

**SISTEM MONITORING
AKTIVITAS KARYAWAN LAPANGAN DENGAN METODE *LOCK*
GPS BERBASIS *CLOUD* PADA PTP. NUSANTARA II**

SKRIPSI

**HERU IRFANSYAH
NIM.0702163065**



**JURUSAN SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGRI SUMATERA UTARA
MEDAN**

2021 M/1442 H

**SISTEM MONITORING
AKTIVITAS KARYAWAN LAPANGAN DENGAN METODE *LOCK*
GPS BERBASIS *CLOUD* PADA PTP. NUSANTARA II**

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana

**HERU IRFANSYAH
NIM.0702163065**



**JURUSAN SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGRI SUMATERA UTARA
MEDAN**

2021 M/1442 H

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lampiran :-

Kepada Yth:

Dekan Fakultas Sains dan Teknologi

UIN Sumatera Utara

Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi sudara:

Nama Lengkap : Heru Irfansyah
Nomor Induk Mahasiswa : 0702163065
Program Studi : Sistem Informasi
Judul Skripsi : Sistem Monitoring Aktivitas Karyawan Lapangan
Dengan Metode Lock GPS Berbasis Cloud Pada PTP.
Nusantara II

Dengan ini kami menilai skripsi tersebut dapat disetujui untuk dapat segera dimunagasyahkan. Atas perhatiannya kami ucapkan terimakasih.

Medan, Agustus 2021

Komisi Pembimbing

Pembimbing I



(SUENDRI, M.Kom)

NIP. 19871208 201503 1 033

Pembimbing II



(Ali Ikhwan, M.Kom)

NIP. 1100000109

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama Lengkap : Heru Irfansyah
Nomor Induk Mahasiswa : 0702163065
Program Studi : Sistem Informasi
Juduk Skripsi : Sistem Monitoring Aktivitas Karyawan Lapangan
Dengan Metode Lock GPS Berbasis Cloud Pada PTP.
Nusantara II

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dari sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 16 Agustus 2021



(Heru Irfansyah)
NIM. 0702163065



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARAMEDAN
FAKULTAS SAINS DAN TEKNOLOGI

Jl. IAIN No. 1 Medan, Kode Pos 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: www.saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id


PENGESAHAN SKRIPSI

Nomor : B.202/ST.V.2/PP.01.1/11/2021

Judul : Sistem Monitoring Aktivitas Karyawan Lapangan Dengan Metode Lock GPS Berbasis Cloud Pada PTP. Nusantara II
Nama : Heru Irfansyah
Nomor Induk Mahasiswa : 0702163065
Program Studi : Sistem Informasi
Fakultas : Sains dan Teknologi
Telah dipertahankan dihadapan Dewan Penguji Skripsi Jurusan Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**
Pada Hari/Tanggal : Selasa / 7 September 2021
Tempat : Aplikasi Zoom

Tim Ujian Munaqasyah

Ketua


(Samsudin, ST, M.Kom)
NIP. 197612272011011002

Penguji I


(Priase, ST, M.Kom)

NIP. 1100000122

Penguji III


(Suendri, M.Kom)

NIP. 19871208 201503 1 033

Penguji II


(Aninda Muliani Harahap, M.Kom)

NIP. 198611292019032009

Penguji IV


(Ali Ikhwan, M.Kom)

NIP. 1100000109

Mengesahkan

Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan




(Dr. Mhd. Syahnan, MA)

NIP. 196609051991031002

MOTTO

“ Sistem pendidikan yang bijaksana setidaknya akan mengajarkan kita betapa sedikitnya yang belum diketahui oleh manusia, seberapa banyak yang masih harus ia pelajari ”

(SIR JOHN LUBBOCK)

PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah dan mengharap rahmat Allah SWT, karya ilmiah ini saya persembahkan untuk:

1. Kepada kedua orang tua saya , Ayah Sofyan dan Ibu Junira atas ketulusan hati dalam mendidik, mengasuh, membimbing penulis dengan kasih dan doa agar penulis dapat menyelesaikannya kegiatan akademik di UIN Sumatera Utara Medan.
2. Kepada Kakak dan adik-adik kandung saya, terima kasih atas do'a dan semangat dukungannya selama ini.
3. Skripsi ini adalah hadiah spesial untuk orang-orang yang saya sayangi. Terima kasih atas bantuan, kebaikan, perhatian, dan wawasan Anda. Terima kasih telah mengajari saya bagaimana menjalani hidup yang jujur dan bahagia.
4. Almamater tercinta UIN Sumatera Utara Medan yang saya banggakan

SISTEM MONITORING

AKTIVITAS KARYAWAN LAPANGAN DENGAN METODE LOCK GPS BERBASIS CLOUD PADA PTP. NUSANTARA II

ABSTRAK

Metode *LockGPS* dan penyimpanan dengan *cloud* dapat diterapkan pada aplikasi monitoring aktivitas karyawan lapangan untuk menjadi solusi dalam meminimalisir kecurangan-kecurangan yang ada pada setiap pekerjaan karyawan di lapangan. *LOCKGPS* dapat mengunci lokasi karyawan sehingga karyawan tidak dapat lagi melakukan ke tidak jujuran pada saat absensi. Untuk mempermudah pemantauan aktivitas karyawan dari jarak jauh ini akan di sediakan menu *upload* aktivitas karyawan berupa foto yang nantinya akan di lengkapi tanggal dan waktu secara otomatis. Pada pendataan hasil panen atau laporan bukti hasil panen disediakan menu untuk mengupload foto proses penimbangan hasil panen dan hasil penimbangannya agar tidak ada lagi ke tidak jujuran dalam pendataan hasil panen. Data untuk penelitian ini dikumpulkan melalui observasi, wawancara, dan studi pustaka yang relevan seperti jurnal dan buku. Ada dua jenis data yang dikumpulkan: primer dan sekunder. Dengan menggunakan UML, langkah-langkah yang dilakukan untuk merancang database dan menggambarkan alur kerja dari aplikasi yang akan dibangun. Pada tahap ini penulis membuat antarmuka yang sesuai dengan fungsi yang dibutuhkan aplikasi yang akan dibangun. RAD (Rapid Application Development) merupakan metode pengembangan sistem yang digunakan dalam penelitian ini. RAD adalah versi adaptasi cepat dari model air terjun yang menggunakan pendekatan konstruksi komponen.

Kata kunci : Lock GPS, Cloud, interface, RAD

MONITORING SYSTEM
FIELD EMPLOYEE ACTIVITIES WITH CLOUD-
BASED GPS LOCK METHOD ON PTP. NUSANTARA II

ABSTRACT

The GPS Lock method and cloud storage can be applied to the field employee activity monitoring application to be a solution in minimizing the frauds that exist in every employee's job in the field. LOCK GPS can lock the location of employees so that employees can no longer commit dishonesty during absenteeism. To make it easier to monitor employee activities remotely, a menu of employee activity uploads will be provided in the form of photos which will later be completed with the date and time automatically. In harvesting data collection or reports of harvest evidence, a menu is provided to upload photos of the weighing process for harvests and weighing results so that there is no more dishonesty in harvesting data collection. The data for this study were collected through observation, interviews, and relevant literature studies such as journals and books. There are two types of data collected: primary and secondary. By using UML, the steps taken to design the database and describe the workflow of the application to be built. At this stage the author creates an interface that is in accordance with the functions required for the application to be built. RAD (Rapid Application Development) is a system development method used in this research. RAD is a fast-adapted version of the waterfall model that uses a component construction approach.

Keywords :Lock GPS, Cloud, interface, RAD

KATA PENGANTAR

الرَّحْمٰنُ الرَّحِیْمُ

Alhamdulillah rabbil'alamin, Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini guna memenuhi salah satu syarat untuk memperoleh gelar Strata-1 (S-1) pada Program Studi Sistem Informasi. Program Studi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara dengan mengangkat judul skripsi "Sistem Monitoring Aktivitas Lapangan Karyawan Dengan Metode *Lock GPS* Berbasis *Cloud* Pada PTP.Nusantara II"

Selama penyusunan skripsi ini, penulis menyadari bahwa banyak orang yang memiliki dampak signifikan dalam penulisan skripsi ini. Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini:

1. Bapak Prof. Dr. Syahrin Harahap, MA selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd. Syahnan, MA selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Samsudin, S.T, M.Kom selaku Ketua Program Studi Sistem Informasi, Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
4. Bapak Suendri M.Kom selaku Sekertaris Prodi Sistem Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara dan Selaku Dosen Pembimbing I yang telah berkenan memberikan tambahan ilmu dan solusi dalam penulisan skripsi ini.
5. Bapak Ali Ikhwan, M.Kom Selaku Dosen Pembimbing II yang telah bersedia membimbing dan memberikan banyak ilmu dalam penulisan skripsi ini

6. Raissa Amanda Putri, S.M.Kom.,M.TI selaku dosen pendamping yang telah memberikan ilmu dalam penyusunan skripsi ini.
7. Kedua orang tua, ayahanda Sofyan dan ibunda Junira Gorat yang telah memberikan dukungan baik moral maupun materi serta doa yang tiada henti kepada penulis
8. Seluruh pejuang Angkatan 16 khususnya Sistem Informasi-2 yang telah memberikan dukungan dan motivasi dalam penyelesaian skripsi kepada penulis.

Penulis mohon maaf atas kesalahan yang telah dilakukan. Semoga skripsi ini bermanfaat untuk kedepannya dan mendorong penelitian lebih lanjut.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Medan , 26, Agustus ,2021
Penyusun,



Heru Irfansyah

0702163065

DAFTAR ISI

MOTTO	i
PERSEMBAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR TABEL	vx
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	6
2.1 Sistem	6
2.2 Monitoring.....	6
2.3 <i>GlobalPositioningSystem (GPS)</i>	7
2.3.1 Bagian-Bagian <i>GPS</i>	8
2.3.2 Prinsip Kerja <i>GPS</i>	9
2.3.3 Fungsi Dan Kegunaan <i>GPS</i>	10
2.3.4 Cara Kerja <i>GPS</i> pada <i>Smartphone</i>	11
2.3.5 Kelebihan dan Kekurangan <i>GPS</i> pada <i>Handphone</i>	11
2.4 <i>Lock GPS</i>	11
2.5 <i>Cloud</i>	12
2.5.1 Sejarah <i>CloudComputing</i>	12
2.5.2 Sejarah Perkembangan <i>CloudComputing</i>	13
2.5.3 Pengertian <i>CloudComputing</i>	17
2.5.4 Layanan Yang ditawarkan <i>CloudComputing</i>	19
2.5.5 Penyebaran Komputasi Awan	20

2.5.6 Fungsi dan Manfaat <i>CloudComputing</i>	20
2.5.7 Kelebihan dan Kekurangan <i>CloudComputing</i>	21
2.5.8 Sistem Kerja <i>CloudComputing</i>	22
2.6 <i>Android</i>	23
2.6.1 Sejarah <i>Android</i>	23
2.6.2 Pengertian <i>Android</i>	23
2.6.3 Fitur-fitur <i>Android</i>	24
2.6.4 Arsitektur <i>Android</i>	24
2.6.5 Versi-versi <i>Android</i>	26
2.7 <i>Android Studio</i>	32
2.8 <i>AndroidSDK(SoftwareDevelopmentKit)</i>	33
2.9 Bahasa Pemrograman Kotlin	34
2.10 Google Map API	35
2.11 UML (<i>UnifiedModellingLanguage</i>).....	35
2.11.1 <i>UseCase</i> Diagram.....	36
2.11.2 <i>Class</i> Diagram	37
2.11.3 <i>Activity</i> Diagram	38
2.11.4 <i>Sequence</i> Diagram.....	39
2.12 Penelitian Terkait	40
2.13 Pembuatan Sistem Yang Dirancang	43
BAB III METODOLOGI PENELITIAN	44
3.1 Tempat dan Waktu Penelitian	44
3.2 Kebutuhan Sistem	46
3.2.1 Perangkat Keras.....	46
3.2.2 Perangkat Lunak.....	46
3.3 Cara Kerja Sistem.....	47
3.3.1 Metode Pengumpulan Data	47
3.3.2 Jenis Data	48
3.3.3 Metode Pengembangan Sistem	49
3.4 Kerangka Berfikir.....	53
BAB IV HASIL DAN PEMBAHASAN	55

4.1 Analisis kebutuhan Sistem	55
4.1.1 Profil Kantor PT.Perkebunan Nusantara II	55
4.1.2 Visi dan Misi PT.Perkebunan Nusantara II.....	56
4.1.3 Tujuan Perusahaan	56
4.1.4 Sasaran Perusahaan	56
4.1.5 Struktur Organisasi PTP.Nusantara II Perk Bandar Klippa ...	57
4.1.6 <i>Job Description</i> Dari Struktur Organisasi	58
4.1.7 Analisis Sistem Berjalan	63
4.1.8 Analisis Sistem Usulan	64
4.2 Penerapan Metode <i>Lock GPS</i> Berbasis <i>Cloud</i>	69
4.3 Penerapan <i>Cloud</i> Pada Plikasi	73
4.3.1 Langkah-langkah Pembuatan VPS dengan ECS Alibaba	74
4.3.2 Penerapan Aplikasi Pada <i>Server Alibaba Cloud</i>	81
4.4 <i>Workshop design</i>	83
4.4.1 <i>Design Model</i>	83
4.4.1.1 <i>Use Case Diagram</i>	83
4.4.1.2 <i>Activity Diagram</i>	84
4.4.1.3 <i>Sequential Diagram</i>	94
4.4.1.4 <i>Class Diagram</i>	101
4.4.2 <i>Design Database</i>	101
4.4.3 <i>Design Interface</i>	103
4.5 Implementasi dan Pengujian Sistem	119
4.5.1 Implementasi Antarmuka Karyawan	119
4.5.2 Implementasi Antarmuka Mandor	127
4.5.3 Pengujian Sistem	136
BAB VKESIMPULAN DAN SARAN	159
5.1 Kesimpulan	159
5.2 Saran	159
DAFTAR PUSTAKA	161
LAMPIRAN	165

DAFTAR TABEL

Tabel 2.1 Simbol <i>UseCase</i> Diagram	36
Tabel 2.2 Simbol <i>Class</i> Diagram	37
Tabel 2.3 Simbol <i>Activity</i> Diagram	38
Tabel 2.4 Simbol <i>Sequence</i> Diagram	39
Tabel 2.5 Penelitian Terdahulu	40
Tabel 3.1 Waktu Penelitian	45
Tabel 4.1 <i>Job Description</i>	58
Tabel 4.2 <i>Database</i> Absensi	101
Tabel 4.3 <i>Database</i> Aktivitas	102
Tabel 4.4 <i>Database</i> Hasil Panen	102
Tabel 4.5 <i>Database</i> Karyawan	102
Tabel 4.6 <i>Database</i> Admin	103
Tabel 4.7 <i>Database</i> Wilayah	103
Tabel 4.8 Macem-Macem <i>Widget Android</i>	104
Tabel 4.9 Data Diri Validator	136
Tabel 4.10 Pengujian Sistem Pada Karyawan	136
Table 4.11 Pengujian Sistem Pada Mandor	146

DAFTAR GAMBAR

Gambar 2.1 Prinsip Kerja <i>GPS</i>	9
Gambar 2.2 Perkembangan <i>CloudComputing</i>	13
Gambar 2.3 Pengertian <i>CloudComputing</i>	17
Gambar 2.4 <i>Android</i> Versi 1.1	27
Gambar 2.5 <i>Android</i> Versi 1.5 (<i>Cupcake</i>)	27
Gambar 2.6 <i>Android</i> Versi 1.6 (<i>Donut</i>)	28
Gambar 2.7 <i>Android</i> Versi 2.0/2.1 (<i>Eclair</i>)	28
Gambar 2.8 <i>Android</i> Versi 2.2 (<i>Froyo</i>)	29
Gambar 2.9 <i>Android</i> Versi 2.3 (<i>Gingerbread</i>)	29
Gambar 2.10 <i>Android</i> Versi 3.0/3.1 (<i>Honeycomb</i>)	30
Gambar 2.11 <i>Android</i> Versi 4.0 (<i>Ice CreamSandwich</i>)	30
Gambar 2.12 <i>Android</i> Versi 4.1 (<i>Jelly Bean</i>)	30
Gambar 2.13 <i>Android</i> Versi 4.4 (<i>Kit Kat</i>)	31
Gambar 2.14 <i>Android</i> Versi 5.0.2 (<i>Lolipop</i>)	31
Gambar 2.15 <i>Android</i> Versi 6.0 (<i>Marshmallow</i>)	32
Gambar 2.16 <i>Android</i> Studio	32
Gambar 2.17 <i>Android</i> SDK	33
Gambar 3.1 Peta Lokasi PTP.Nusantara II Perk. Bandar Klippa	44
Gambar 3.2 Siklus Pengembangan Sistem RAD	49
Gambar 3.3 Kerangka Berfikir	53
Gambar 4.1 Struktur Organisasi PTP.Nusantara II Perk Bandar Klippa	57
Gambar 4.2 <i>Flowmap</i> Sistem Berjalan	63
Gambar 4.3 <i>Flowmap</i> Sistem Usulan	66
Gambar 4.4 Penerapan Metode <i>Lock GPS</i>	70
Gambar 4.5 Menjalankan Fungsi Peta Pada Apk Karyawan	71
Gambar 4.6 Proses Kinerja Pada Maps <i>View</i>	71
Gambar 4.7 Menjalankan Fungsi Peta Pada Apk Mandor	72
Gambar 4.8 Pembuatan Marker di Maps Aplikasi	72
Gambar 4.9 Pencaraian ECS (<i>Elastic Compute Service</i>).....	74

Gambar 4.10	Tampilan <i>Creat ECS Instance</i>	75
Gambar 4.11	Konfigurasi ECS Bagian 1	75
Gambar 4.12	Konfigurasi ECS Bagian 2	76
Gambar 4.13	Konfigurasi ECS Bagian 3	77
Gambar 4.14	Konfigurasi Network ECS Bagian 1	78
Gambar 4.15	Konfigurasi Network ECS Bagian 2	78
Gambar 4.16	Sistem Konfigurasi ECS	79
Gambar 4.17	Tampilan <i>Instance</i>	80
Gambar 4.18	Scripting Koneksi <i>Server Alibaba Cloud</i>	81
Gambar 4.19	CpanelYangBerisi <i>Backend</i>	82
Gambar 4.20	Tampilan Folder <i>Scripting</i> Aplikasi	82
Gambar 4.21	<i>Use Case Diagram</i>	83
Gambar 4.22	<i>ActivityDiagram Login</i>	84
Gambar 4.23	<i>Activity Diagram</i> Absen Karyawan	85
Gambar 4.24	<i>Activity Diagram</i> Aktivitas Karyawan	86
Gambar 4.25	<i>Activity Diagram</i> Hasil Panen Karyawan	87
Gambar 4.26	<i>Activity Diagram</i> Laporan Absen (Mandor)	88
Gambar 4.27	<i>Activity Diagram</i> Laporan Aktivitas Karyawan (Mandor)	89
Gambar 4.28	<i>Activity Diagram</i> Data Panen Karyawan (Mandor)	90
Gambar 4.29	<i>Activity Diagram</i> Master Karyawan (Mandor)	91
Gambar 4.30	<i>Activity Diagram</i> Laporan Hasil Panen (Mandor)	92
Gambar 4.31	<i>ActivityDiagram</i> Admin.....	93
Gambar 4.32	<i>Sequential Login</i> (Karyawan dan Mandor)	94
Gambar 4.33	<i>Sequential</i> Absen Karyawan	95
Gambar 4.34	<i>Sequential</i> Aktivitas Karyawan.....	96
Gambar 4.35	<i>Sequential</i> Hasil Panen Karyawan	96
Gambar 4.36	<i>Sequential</i> Master Absen Karyawan (Mandor).....	97
Gambar 4.37	<i>Sequential</i> Master Aktivitas (Mandor).....	98
Gambar 4.38	<i>Sequential</i> Master Panen (Mandor).....	99
Gambar 4.39	<i>Sequential</i> Master Karyawan (Mandor).....	99
Gambar 4.40	<i>Sequential</i> Laporan (Mandor)	100

Gambar 4.41 <i>Calss Diagram</i>	101
Gambar 4.42 <i>Login</i> (Karyawan)	104
Gambar 4.43 Absensi (Karyawan)	105
Gambar 4.44 Tampilan Tambah Aktivitas (Karyawan)	106
Gambar 4.45 Tampilan Tambah Panen (Karyawan)	107
Gambar 4.46 Tampilan <i>Login</i> (Mandor)	107
Gambar 4.47 Tampilan Tambah Karyawan (Mandor)	108
Gambar 4.48 Tampilan Ubah dan Hapus Data Karyawan (Mandor)	109
Gambar 4.49 Tampilan Cetak Laporan (Mandor)	110
Gambar 4.50 <i>Splash Screen</i> (Karyawan)	110
Gambar 4.51 Tampilan Aktivitas (Karyawan)	111
Gambar 4.52 Tampilan Ket Aktivitas (Karyawan)	111
Gambar 4.53 Tampilan Panen (Karyawan)	112
Gambar 4.54 Tampilan Ket Hasil Panen (Karyawan)	112
Gambar 4.55 Tampilan Menu <i>Logout</i> (Karyawan)	113
Gambar 4.56 Tampilan Menu (Mandor)	113
Gambar 4.57 Tampilan Data Absen (Mandor)	114
Gambar 4.58 Tampilan Konfirmasi Absen (Mandor)	115
Gambar 4.59 Tampilan Data Aktivitas (Mandor)	115
Gambar 4.60 Tampilan Detail Aktivitas	116
Gambar 4.61 Tampilan Data Panen (Mandor)	116
Gambar 4.62 Tampilan Detail Panen (Mandor)	117
Gambar 4.63 Data Karyawan (Mandor)	117
Gambar 4.64 Tampilan Laporan (Mandor)	118
Gambar 4.65 Tampilan Laporan Panen PDF	118
Gambar 4.66 Implementasi Tampilan <i>Login</i> Karyawan	119
Gambar 4.67 Impelementasi Tampilan Menu Absen Karyawan	120
Gambar 4.68 Impelementasi Tampilan Menu Aktivitas Karyawan	120
Gambar 4.69 Impelementasi Tampilan <i>Upload</i> Aktivitas Karyawan	121
Gambar 4.70 Impelementasi Tampilan Detail Aktivitas Karyawan	122
Gambar 4.71 Impelementasi Tampilan Menu Panen Karyawan	123

Gambar 4.72 Impelementasi Tampilan Upload Hasil Panen Karyawan	123
Gambar 4.73 Impelementasi Tampilan Detail Hasil Panen Karyawan	124
Gambar 4.74 Impelementasi Tampilan <i>Logout</i> Karyawan	125
Gambar 4.75 Impelementasi Tampilan Login (Mandor)	125
Gambar 4.76 Impelementasi Tampilan Menu Utama (Mandor)	126
Gambar 4.77 Impelementasi Tampilan Menu Master Absen (Mandor)	127
Gambar 4.78 Impelementasi Tampilan Detail Absen (Mandor)	127
Gambar 4.79 Impelementasi Tampilan Menu Master Aktivitas (Mandor)	128
Gambar 4.80 Impelementasi Tampilan Detail Aktivitas (Mandor)	129
Gambar 4.81 Impelementasi Tampilan Menu Panen (Mandor)	130
Gambar 4.82 Impelementasi Tampilan Detail Panen (Mandor)	130
Gambar 4.83 Impelementasi Tampilan Menu Master Karyawan (Mandor)	131
Gambar 4.84 Impelementasi Tampilan Uдах dan Hapus Data Karyawan	132
Gambar 4.85 Impelementasi Tampilan Tambah Karyawan (Mandor)	133
Gambar 4.86 Impelementasi Tampilan Menu Master Laporan (Mandor)	134
Gambar 4.87 Impelementasi Tampilan Cetak Laporan Absen (Mandor)	135
Gambar 4.88 Impelementasi Tampilan Laporan Panen	135
Gambar 4.89 Halaman <i>Login</i>	136
Gambar 4.90 Tampilan Informasi <i>User</i> dan <i>Pass</i> Salah	137
Gambar 4.91 Menu Utama	137
Gambar 4.92 Informasi Telah Absen	138
Gambar 4.93 Tampilan Informasi Abse	138
Gambar 4.94 Tampilan Informasi Ajukan Ijin	139
Gambar 4.95 Tampilan Data Aktivitas	139
Gambar 4.96 Tampilan Form Tambah Aktivitas	140
Gambar 4.97 Tampilan Informasi Simpan Aktivitas	140
Gambar 4.98 Tampilan Detail Aktivitas	141
Gambar 4.99 Tampilan Informasi Hapus Data Aktivitas	141
Gambar 4.100 Tampilan Informasi Berhasil Hapus Data	142
Gambar 4.101 Tampilan Data Hasil Panen	142
Gambar 4.102 Tampilan Form Hasil Panen	143

Gambar 4.103 Tampilan Informasi Simpan Data	143
Gambar 4.104 Tampilan Detail Hasil Panen	144
Gambar 4.105 Tampilan Hapus Data Hasil Panen	144
Gambar 4.106 Tampilan Informasi Hapus Data	145
Gambar 4.107 Tampilan <i>Logout</i>	145
Gambar 4.108 Tampilan Informasi <i>Logout</i>	146
Gambar 4.109 Tampilan Halaman <i>Login</i>	146
Gambar 4.110 Informasi <i>User</i> dan <i>Pass</i> Salah	147
Gambar 4.111 Tampilan Menu Utama	147
Gambar 4.112 Tampilan Data Absen Karyawan	148
Gambar 4.113 Data Infromasi Absen Karyawan	148
Gambar 4.114 Tampilan Infromasi Konfrimasi Kahadiran	149
Gambar 4.115 Tampilan Infromasi Konfirmasi Tidak Hadir	149
Gambar 4.116 Tampilan Konfirmasi Ijin	150
Gambar 4.117 Tampilan Infromasi Konfirmasi Ijin	150
Gambar 4.118 Tampilan Data Aktivitas Karyawan	151
Gambar 4.119 Tampilan Detail Infromasi Aktivitas	151
Gambar 4.120 Tampilan Data Hasil Panen	152
Gambar 4.121 Tampilan Detail Hasil Panen	152
Gambar 4.122 Tampilan Data Karyawan	153
Gambar 4.123 Tampilan <i>Form</i> Tambah Data Karyawan	153
Gambar 4.124 Informasi Data Berhasil Disimpan	154
Gambar 4.125 Tampilan Detail Data Karyawan	154
Gambar 4.126 Informasi Data Berhasil Diubah	155
Gambar 4.127 Informasi Data Berhasil Dihapus	155
Gambar 4.128 Tampilan Data Nama Laporan Karyawan	156
Gambar 4.129 Tampilan Form cetak Laporan	156
Gambar 4.130 Infromasi Download Data Laporan Panen	157
Gambar 4.131 Tampilan laporan Panen Berbentuk PDF	157
Gambar 4.132 Tampilan Infromasi <i>Logout</i>	158

BAB I

PENDAHULUAN

1.1 Latar Belakang

Memasuki era globalisasi, persaingan di dunia kerja semakin ketat. Perkembangan teknologi juga terus mengalami kemajuan. Pemanfaatan teknologi oleh manusia digunakan untuk membantu menyelesaikan pekerjaan secara lebih efektif dan efisien. Dengan adanya sistem informasi semua pekerjaan dapat diselesaikan dengan lebih terstruktur. Dengan pesatnya perkembangan teknologi navigasi, saat ini menghasilkan sistem navigasi yang sangat canggih, yang dipergunakan untuk memilih posisi suatu objek pada bagian atas bumi yang disebut GPS. Seiring dengan perkembangan teknologi yang semakin maju, teknik pengelolaan penyimpanan informasi juga semakin maju. Salah satu teknik yang dapat digunakan untuk menyimpan informasi yang besar dan aman adalah *cloud storage* dalam bentuk digital.

PTPN 2 adalah salah satu instansi pengelola kebun kelapa sawit. Luasnya lahan kebun kelapa sawit yang harus dikelola oleh PTPN 2 membuat instansi ini membutuhkan banyak karyawan. Maka dari itu, PTPN 2 sangat membutuhkan sebuah aplikasi untuk dapat membantu dalam proses absensi karyawan, monitoring aktivitas lapangan, dan pendataan hasil panen perkebunan. Saat ini, PTPN 2 sudah difasilitasi dengan aplikasi absensi berbasis *android*, akan tetapi aplikasinya masih tergolong sederhana dan penggunaannya masih kurang efektif yaitu, karyawan hanya perlu menginput nama, tanggal, waktu dan keterangan hadirnya. Dengan kerja aplikasi yang demikian dapat ditemukan kelemahan yaitu, mandor lapangan kesulitan untuk mengetahui apakah karyawan yang mengisi absensi benar-benar sudah berada di lapangan, karena tidak adanya fitur cek lokasi. Pada aktivitas karyawan mandor tidak dapat mengetahui keadaan kegiatan karyawan di lapangan. Selain itu dalam hal melaporkan hasil panen, karyawan masih menggunakan cara manual. Pada pelaporan ini mandor mengalami kesulitan untuk mengetahui kebenaran dari data hasil panen yang dilaporkan oleh karyawan dan catatan hasil panen bisa saja rusak atau hilang.

GPS lock adalah suatu kondisi dimana *GPS* terhubung dengan sinyal satelit dan mendapatkan sinyal berupa informasi. Jika *GPS* dapat menentukan lokasi pengguna, maka *GPS* juga dapat menghitung kecepatan, arah, jalur, dan jarak ke tujuan. Oleh karena itu, pembuatan sistem *GPS Lock* di suatu perusahaan bisa bekerjasama langsung menggunakan satelit sehingga mampu menerima koordinat lokasi karyawan secara akurat ketika melakukan absensi di lapangan juga pada luar tempat kerja. Sistem ini juga menggunakan berbasis *Cloud*, istilah *Cloud* mengacu pada penyimpanan yang terdiri dari kumpulan *server* yang disimpan dengan rapi dan dikelola oleh layanan *Cloud*. *Cloud* ini dapat memberikan kapasitas penyimpanan yang lebih besar tanpa harus memindahkan *information*. Keterbatasan penyimpanan *information* sering kita jumpai saat akan menyimpan *document* yang akan kita pindahkan ke data *setweb* seperti yang bisa kita lakukan sekarang dengan menyimpan *information* di email, *googledrive* atau *dropbox*. Penggunaan *Cloud* untuk sistem ini adalah untuk menyimpan *information checking* karyawan berupa foto dan *information* panen yang membutuhkan penyimpanan yang besar untuk setiap jam kerja karyawan.

Metode *Lock GPS* dan penyimpanan dengan *cloud* dapat diterapkan pada aplikasi *monitoring* aktivitas karyawan lapangan untuk menjadi solusi dalam mengatasi kelemahan-kelemahan dari aplikasi yang sebelumnya. Metode *Lock GPS* dapat digunakan untuk mengunci lokasi karyawan pada saat melakukan absensi sehingga mandor dengan mudah mengetahui bahwa karyawannya benar-benar sudah berada di lapangan. Untuk mempermudah pemantauan aktivitas karyawan dari jarak jauh ini akan disediakan menu *upload* aktivitas karyawan berupa foto yang nantinya akan dilengkapi tanggal dan waktu secara otomatis. Pada pendataan hasil panen atau laporan bukti hasil panen akan di sediakan menu untuk *upload* foto proses hasil penimbangannya, sehingga mandor dapat mengetahui kebenaran dari data hasil panen yang dilaporkan nantinya. Untuk penyimpanan data hasil panen akan diterapkan media penyimpanan *cloud* atau penyimpanan digital sehingga menjadi lebih praktis di bandingkan dengan media berupa buku atau catatan. Dengan adanya aplikasi ini, mandor akan lebih mudah dalam memantau aktivitas karyawannya di lapangan. Sesuai latar belakang

persoalan yang sudah diuraikan, penulis tertarik buat mengambil judul “*SISTEM MONITORING AKTIVITAS KARYAWAN LAPANGAN DENGAN METODE LOCK GPS BERBASIS CLOUD PADA PTP. NUSANTARA II*”.

1.2 Rumusan Masalah

Dari latar belakang diatas maka penulis akan melakukan penelitian dengan rumusan masalah :

1. Bagaimanakah membangun aplikasi absensi menggunakan sistem *Lock GPS* berbasis *android*?
2. Bagaimanakah membangun aplikasi *monitoring* aktivitas karyawan yang dapat melaporkan setiap kegiatan karyawan berbasis *android* ?
3. Bagaimanakah menerapkan metode *Cloud* dalam menyimpan laporan kegiatan karyawan lapangan dan laporan hasil panen karyawan ?

1.3 Batasan Masalah

Dalam pembuatan aplikasi ini penulis mendapatkan beberapa batasan masalah yaitu :

1. Teknik *client*(Karyawan) –*server* (Atasan) akan di terapkan pada aplikasi.
2. Aplikasi *monitoring* hanya digunakan pada PTP.Nusantara II Perk Bandar Klippa kebunsawit batang kuis.
3. Pengambil data hasil panen kelapa sawit bagian Afdeling IV di bulan Februari, Maret dan April.
4. Mengambil data karyawan lapangan pada bagian Afdeling IV.
5. Aplikasi absensi akan dibangun menggunakan metode *Lock GPS* yang menampilkan titik koordinat karyawan.
6. Pembuatan aplikasi *monitoring* ini menggunakan *Cloud* untuk penyimpanan data absen, data aktivitas karyawan dan data hasil panen berbasis *Android*.

7. Aplikasi pada menu aktivitas hanya menampilkan aktivitas kegiatan karyawan lapangan seperti pemberian pupuk, pemeliharaan pohon sawit, penurunan sawit dan pengumpulan sawit berupa foto dan keterangan saja dan juga terdapat tanggal dan waktu secara otomatis.
8. Aplikasi pada menu hasil panen hanya menampilkan hasil panen karyawan lapangan berupa foto, keterangan dan kuantitas timbangan saja serta tanggal dan waktu secara otomatis.
9. Aplikasi juga dapat menampilkan laporan hasil panen bulanan karyawan berbentuk pdf.

1.4 Tujuan Penelitian

1. Membangun aplikasi absensi menggunakan GPS Lock untuk mengetahui lokasi karyawan secara akurat menggunakan Android.
2. Membangun aplikasi *monitoring* aktivitas karyawan yang dapat melaporkan setiap kegiatan karyawan yang berada di lapangan menggunakan *android*.
3. Mengubah penyimpanan data hasil laporan karyawan ke dalam *database* digital berbasis *cloud*.

1.5 Manfaat

Penulis juga mendapatkan manfaat dari penelitian ini antara lain::

- a. Bagi Mahasiswa
 1. Menambah wawasan atau pengetahuan penulis pada bidang teknologi *GPS Lock*.
 2. Mengetahui penyimpanan data yang besar menggunakan *Cloud*.
 3. Melatih berpikir secara terstruktur dan sistematis
- b. Bagi Universitas Islam Negeri Sumatera Utara
 1. Menjadikan bahan referensi penelitian mahasiswa lain yang lebih mendalami tentang penyimpanan berbasis *Cloud*
 2. Meningkatkan kualitas lulusan terbaik melalui karya ilmiah yang dihasilkan oleh mahasiswa UIN Sumatera Utara.

3. Mengetahui kemampuan mahasiswa dalam bidang IT yang dimiliki setiap mahasiswa
- c. Bagi Perusahaan PTP. Nusantara II
1. Dapat mengisi absensi menggunakan *GPS* yang dapat mengetahui titik koordinat karyawan itu berada.
 2. Dapat melaporkan data hasil panen tanpa harus bertatap muka atau dengan via *telephone* dan dapat menyimpan data menggunakan *Cloud* tanpa harus khawatir akan kehilangan laporan hasil panen .
 3. Dapat melihat kegiatan terkini tanpa harus bertatap muka ke tempat lokasi karyawan bekerja, mereka cukup mengirim foto aktivitas mereka.
 4. Atasan bisa mengawasi karyawan tanpa harus khawatir karyawan sudah hadir atau belum.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Seperti yang dinyatakan sebelumnya, ada banyak orang yang mendefinisikan sistem saat ini. Beberapa peningkatan sistem yang telah diterapkan selama penelitian ini adalah yang disediakan oleh Sutabri. Sutabri memberikan panduan sistem sebagai sekelompok elemen yang saling berhubungan yang bekerja sama untuk mencapai tujuan bersama. Dengan kata lain, sistem didefinisikan sebagai kumpulan elemen, komponen, atau variabel yang terorganisir, berinteraksi, stabil, dan terintegrasi. Pada tingkat yang paling dasar, sistem adalah kumpulan entitas (*hardware, brainware, software*) yang berinteraksi, berkolaborasi, dan bekerja sama untuk mencapai tujuan tertentu. (Herliana & Rasyid, 2016)

2.2 Monitoring

Monitoring, juga dikenal sebagai pengawasan, adalah proses yang dilakukan untuk memastikan bahwa pekerjaan yang telah dilakukan diselesaikan atau ditingkatkan sedemikian rupa sehingga memenuhi tenggat waktu. (Riyanto, 2019) *Monitoring* adalah proses pengumpulan data dan penentuan tujuan program. Awasi perubahan yang berpusat pada proses dan garis waktu. Pemantauan menyediakan data mentah untuk menjawab pertanyaan evaluasi melalui data dan memberikan tambah. Evaluasi adalah waktu untuk memahami peristiwa, pertanyaan yang harus dijawab, rekomendasi yang akan dibuat, saran perbaikan. Namun, tanpa pemantauan, evaluasi tidak akan ada, tidak akan ada data untuk dikerjakan, dan akan mengalami kekurangan sumber daya. Oleh karena itu, pemantauan dan evaluasi harus dilakukan dengan hati-hati.

Menurut (Mardiani, 2013) Tujuan *monitoring* adalah:

1. Menilai apakah kegiatan yang dilakukan sudah sesuai dengan rencana.
2. Identifikasi masalah yang muncul sehingga dapat segera diatasi.
3. Menilai apakah pola kerja dan manajemen yang digunakan sudah sesuai untuk

mencapai tujuan kegiatan.

4. Mengetahui hubungan antara kegiatan dengan tujuan untuk memperoleh suatu ukuran kemajuan.
5. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan.

Sistem *monitoring* adalah sistem yang dirancang untuk memberikan peringatan ketika suatu program menjalankan fungsinya. Umpan balik untuk memberikan informasi tentang keadaan sistem saat ini. Pemeliharaan sistem juga dapat disebut sebagai kumpulan prosedur dan program untuk menghitung kemampuan sistem informasi untuk membaca dan mengirimkan data berdasarkan informasi yang disediakan. Selain itu, sistem pemeliharaan adalah kumpulan fitur informasi yang memberikan informasi tentang apa yang terjadi dengan sistem yang sedang dirawat. (Salamun, 2017)

2.3 Global Positioning System (GPS)

Global Positioning System (GPS) adalah sistem yang menentukan sinyal lokasi permukaan bumi dengan sinyal satelit bantuan. *GPS* adalah sebuah sistem yang dapat membantu kita dalam menentukan dimana kita berada pada waktu tertentu. Selain itu, untuk mengambil data yang dikumpulkan oleh *GPS*, diperlukan perangkat yang mampu membaca data tersebut. Karena perangkat ini dapat memberikan lokasi atau koordinat tempat *GPS* digunakan, maka perangkat ini biasa disebut sebagai penerima. Akibatnya, memiliki *GPS* merupakan manfaat yang signifikan bagi SIG. *ArcView GIS* adalah salah satu alat dalam *Sistem Informasi Geografis (SIG)* atau *Geographics Information System (GIS)* yang memiliki pendekatan langsung untuk analisis data. (Amri et al., 2019)

GPS terdiri dari konstelasi 24 satelit yang beroperasi. Untuk memastikan cakupan yang berkelanjutan dari seluruh dunia, 4 satelit ditempatkan di setiap 6 bidang orbit. Setiap satelit *GPS* akan memancarkan sinyal radio gelombang mikro terus menerus yang terdiri dari dua pembawa, dua kode dan menu navigasi. Operator dan kode umumnya digunakan untuk menentukan jarak dari penerima pengguna ke satelit *GPS*. Antena L-band dan antena S-band adalah bagian dari

segmen luar angkasa yang berkomunikasi dengan segmen kontrol yang terletak di Amerika Serikat di Colorado Springs, Colorado. Tugasnya adalah mengunduh dan mengunggah informasi data pelacakan satelit *GPS*.

Terlepas dari kenyataan bahwa tujuan utama *GPS* adalah menjadi sistem navigasi, *GPS* telah merevolusi cara kita mensurvei daratan dan menavigasi. Namun, karena pentingnya aplikasi *GPS* untuk masyarakat umum, *GPS* telah menjadi komoditas global, dan pengembangan lanjutan aplikasi berbasis *GPS* terbatas pada beberapa individu. (Rahman et al., 2018)

2.3.1 Bagian–Bagian *GPS*

GPS terdiri dari 3 segmen yaitu bagian ruang, bagian kontrol/pengontrol, dan bagian pengguna. (Maftukhin, 2018)

1. Ruang utama terdiri dari 24 satelit. Setiap satelit *GPS* memancarkan sinyal yang terdiri dari beberapa komponen, termasuk dua sinyal sinusoidal yang dikenal sebagai frekuensi pembawa, dua kode digital, dan data navigasi. Sinyal telah ditambahkan sebagai modul untuk navigasi dan kode pembayaran. Jarak antara penerima pengguna dan satelit *GPS* ditentukan oleh pembawa sinyal. Pesan navigasi mencakup lokasi serta informasi lain seperti koordinat (lokasi).
2. Bagian Kontrol terdiri dari jaringan stasiun pelacakan global, dengan pusat kendali utama di Colorado Springs. Tanggung jawab utama departemen kontrol adalah memantau satelit *GPS* untuk menentukan dan memprediksi lokasi satelit, integritas sistem, kondisi jam atom satelit, data atmosfer, dan faktor lainnya. Data ini kemudian dikemas dan ditransmisikan ke satelit *GPS*, dan sebaliknya.
3. Bagian pengguna mencakup pengguna militer dan sipil. Saat penerimaan *GPS* dihubungkan ke antena *GPS*, pengguna dapat menerima sinyal *GPS* yang dapat digunakan untuk menentukan posisinya di mana pun dia berada. *GPS* saat ini tersedia secara gratis untuk semua pengguna di seluruh dunia.

2.3.2 Prinsip Kerja GPS



Gambar 2.1 Prinsip Kerja GPS(Adikara, 2013)

Setidaknya tiga hingga empat satelit dapat menjangkau setiap area di permukaan bumi. Setiap GPS yang dilengkapi dengan teknologi mutakhir dapat menerima hingga 12 saluran satelit secara bersamaan. GPS menerima sinyal satelit dengan lebih mudah saat langit cerah dan bebas penghalang. Semakin banyak satelit yang diterima GPS, semakin akurat hasilnya. Penerima GPS melakukan triangulasi posisi yang akan ditentukan dengan mengukur lamanya waktu sinyal dikirim dari satelit dan mengalikannya dengan kecepatan cahaya, yaitu 3×10^8 meter per detik, untuk menentukan seberapa jauh penerima. GPS berasal dari masing-masing satelit, dan penerima GPS dapat menghitung posisi tetap suatu titik, yaitu garis lintang dan garis bujur bumi, menggunakan sinyal yang dikirim oleh satelit setidaknya tiga kali. Penggunaan sinyal satelit keempat memungkinkan penerima GPS untuk menghitung posisi ketinggian titik dalam kaitannya dengan permukaan laut rata-rata, yang ideal untuk navigasi. (Azzahra, 2016)

2.3.3 Fungsi dan Kegunaan GPS

Fungsi dan Kegunaan *GPS* bukan tanpa kelebihan, ada banyak sekali keuntungan yang bisa didapatkan dari sistem navigasi *GPS* bagi masyarakat di seluruh dunia, khususnya bagi pemerintah Amerika Serikat. Beberapa fungsi dan aplikasi *GPS* dapat dibagi menjadi lima kategori, yaitu: (Avelina, 2018)

1. *GPS* untuk Militer

Teknologi *GPS* dapat digunakan untuk membantu sistem pertahanan militer. Selanjutnya dapat memantau pergerakan musuh saat war dan memandu arah bom agar lebih tepat sasaran.

2. *GPS* untuk Navigasi

Sistem *GPS* sangat berguna dalam kebutuhan berkendara, dengan dipasangnya *GPSTracker* pada kendaraan, perjalanan akan lebih nyaman karena arah dan tujuan jalan dapat diketahui setelah *GPS* mengirimkan posisi kendaraan kita yang diterjemahkan ke dalam peta digital.

3. *GPS* untuk Sistem Informasi Geografis

GPS juga sering digunakan dalam sistem informasi geografis, seperti membuat peta, mengukur jarak perbatasan, atau berfungsi sebagai referensi untuk mengukur suatu wilayah.

4. *GPS* untuk Sistem Pelacakan Kendaraan

Fungsi ini hampir identik dengan navigasi, kecuali bahwa navigasi menggunakan perangkat penerima sinyal *GPS* dan menampilkan koordinatnya dalam satu perangkat, sedangkan sistem pelacakan memerlukan perangkat yang menampilkan dan menerima sinyal dari beberapa lokasi. Misalnya, kita dapat menentukan lokasi kendaraan yang hilang dengan memeriksa koordinat yang dihasilkan oleh alat yang dipasang di kendaraan, yang dapat dilihat melalui media *smartphone* atau alat khusus lainnya.

5. *GPS* untuk Pemantau Gempa

Teknologi *GPS* saat ini sedang ditingkatkan untuk menghasilkan tingkat presisi dan akurasi yang sangat tinggi, memungkinkan *GPS* digunakan untuk memantau pergerakan tanah di Bumi. Ahli geologi dapat menggunakan ini untuk memperkirakan kemungkinan gempa bumi di daerah tertentu.

2.3.4 Cara Kerja GPS Pada Smartphone

Cara kerja *GPS* di smartphone hampir identik dengan cara kerja smartphone dengan provider telekomunikasi. Sebuah ponsel mampu berkomunikasi dengan BTS di kedua arah. BTS (Base Transceiver Station) merupakan elemen terluar yang bersentuhan langsung dengan semua telepon seluler yang terhubung dengan operator. Saat berpindah dari satu lokasi ke lokasi lain, ponsel secara tidak sadar akan mengambil sinyal dari BTS yang berbeda tetapi masih dalam satu operator telekomunikasi yang sama, jangkauan BTS tidak terlalu jauh sehingga operator telekomunikasi memiliki sinyal yang baik di semua tempat, operator telekomunikasi harus memiliki besar jumlah menara BTS. Sehingga operator telekomunikasi dapat mengetahui secara tepat lokasi menara BTS yang sinyalnya ditangkap oleh ponsel tersebut. (Tafa et al., 2018)

2.3.5 Kelebihan dan Kekurangan GPS Pada Handphone

Ketika *GPS* di ponsel dibandingkan dengan *GPS* di kendaraan, ada beberapa kelebihan dan kekurangan masing-masing. Manfaat memiliki *GPS* di ponsel Anda adalah dapat digunakan sebagai alat komunikasi dan cukup kecil untuk dibawa kemana-mana. Kekurangannya adalah akurasinya masih kurang karena jarak miss mencapai 5 sampai 10 meter, sedangkan *GPS* yang terdapat pada kendaraan yang bergerak hanya 1 sampai 3 meter. (Ramadi, 2011)

2.4 Lock GPS

Lock GPS mengacu pada kondisi di mana penerima *GPS* terhubung ke sinyal satelit dan menerima sinyal informasi dari satelit. Untuk menghitung posisi 2D (lintang dan bujur) dan jalur pergerakan, penerima *GPS* harus mengunci sinyal dari setidaknya tiga satelit. Jika penerima *GPS* menerima empat atau lebih satelit, ia dapat menghitung posisi tiga dimensi (lintang, bujur, dan ketinggian). Jika *GPS* dapat menentukan posisi pengguna, maka *GPS* dapat menghitung data lain seperti kecepatan, arah, jalur, tujuan perjalanan, jarak ke tujuan, matahari terbit dan terbenam, dan sebagainya. (Makhfuddin & Prabowo, 2015)

2.5 *Cloud*

2.5.1 *Sejarah Cloud Computing*

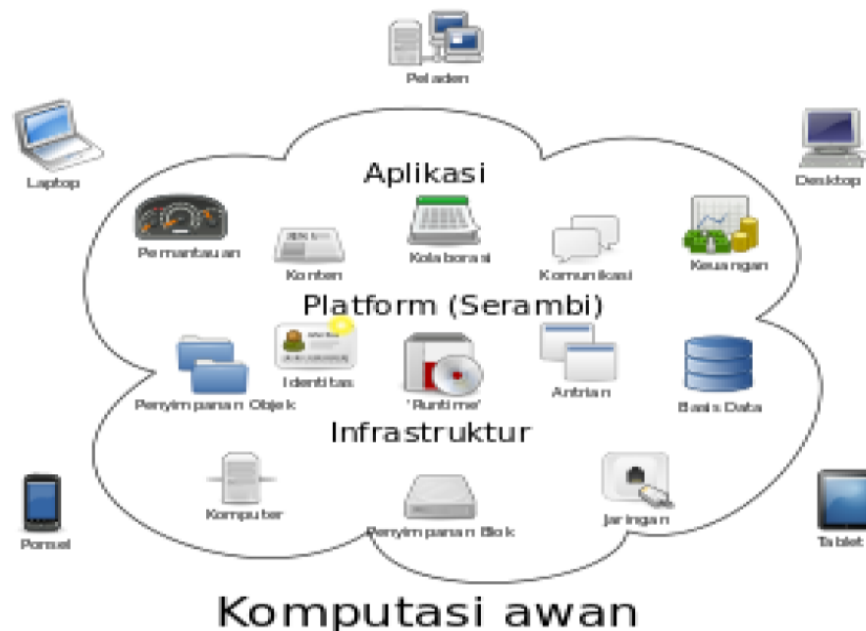
Evolusi komputasi peringatan, juga dikenal sebagai komputasi awan, dimulai pada tahun 1960. Fasilitas utama yang memungkinkan terciptanya teknologi *Cloud Computing* adalah pembagian waktu dan multitasking, yang merupakan fitur umum dalam sistem operasi saat ini. Awalnya, revolusi komputer, seperti IBM, menjadikan komputasi awan sebagai pengembang paling signifikan dalam teknologi ini. *Mainframe* IBM adalah sistem komersial yang sepenuhnya dapat diakses oleh publik, meskipun hanya dapat memproses beberapa perintah sederhana di sisi pengguna. *Tymeshare*, sebagai penyedia layanan, mengembangkan sistem penyewaan kamar mainframe yang dapat diakses melalui saluran telepon pada tahun 1970-an. Perusahaan ini menawarkan layanan komputer tingkat tinggi kepada perusahaan besar pada saat itu.

Tahun 1980-an dan 1990-an melihat kelahiran perusahaan yang memproduksi mainframe besar dan komputer skala kecil. Karena penyewa dari perusahaan terhubung ke pusat data melalui jalur ISDN, maka perangkat keras yang mereka produksi kemudian disewakan kepada perusahaan untuk kebutuhan pusat data, dengan biaya bulanan atau tahunan, dan kemampuan komputer yang mereka miliki juga diteruskan ke pelanggan. Data pelanggan dicadangkan dan dapat diakses jika terjadi kegagalan sistem. Selama waktu ini, sistem SLA, atau perjanjian tingkat layanan, dikembangkan sebagai hasil dari pengembangan sistem penyewaan sumber daya. Dekade antara 1980 dan 1990 juga menjadi saksi lahirnya *Application Service Provider* (ASP), yang pada saat itu juga dikenal sebagai layanan pihak ketiga yang berfungsi dalam hal pendelegasian.

Data, pengelolaan data, dan software hosting berupa layanan rental. ASP memungkinkan usaha kecil untuk memanfaatkan teknologi mutakhir dengan biaya rendah. Parameter ASP meliputi harga, layanan pelanggan, dan RAS (*Reliability, Availability, dan Serviceability*). Dekade 2000-an melihat lonjakan penggunaan ASP. Di mana perusahaan besar, seperti AT&T dan Oracle, memasukkan layanan ASP ke dalam operasi mereka. Namun, layanan ini hanya tersedia untuk pengguna

perusahaan dan tidak menjangkau pengguna akhir sebagai konsumen. Hal ini disebabkan sulitnya ASP dan tingginya biaya instalasi dan pemeliharaan. Dibutuhkan sejumlah besar sumber daya manusia untuk memberikan bantuan kepada pengguna akhir, serta untuk memelihara sistem ini. Akibatnya, banyak perusahaan ASP gagal, dan hanya sedikit yang bertahan. Dan seiring berkembangnya sistem komputer, muncullah dua konsep dasar yang nantinya akan melahirkan sebuah teknologi baru yang dikenal dengan *Cloud Computing*. Web Hosting dan Aplikasi Web Asli adalah konsepnya. (Husain et al., 2018)

2.5.2 Sejarah Perkembangan *Cloud Computing*



Gambar 2.2 Perkembangan *Cloud Computing* (Sulaiman & Widarma, 2017)

Istilah "I" berasal dari kata "telepon". Sirkuit data (termasuk yang membawa lalu lintas Internet) diprogram antar tujuan hingga tahun 1990-an. Kemudian, perusahaan telepon jarak jauh mulai menyediakan layanan *Virtual Private Network* (VPN) untuk komunikasi data. Karena perusahaan telepon dapat mengarahkan lalu lintas untuk menyeimbangkan penggunaan sesuai keinginan mereka, mereka dapat menyediakan layanan berbasis VPN dengan *bandwidth* terjamin sebagai sirkuit tetap dengan biaya lebih rendah. Akibatnya,

jaringan mereka secara keseluruhan dapat digunakan lebih efisien. Sebagai hasil dari pengaturan ini, dimungkinkan untuk menentukan jalan mana yang harus diambil dengan cepat dan tepat. Simbol awan digunakan untuk mewakili tanggung jawab penyedia (penyedia layanan), dan *Cloud Computing* memperluasnya untuk melindungi server dan infrastruktur jaringan.

Ide di balik komputasi awan dimulai pada tahun 1960-an, ketika John McCarthy, seorang ahli komputasi MIT dan pelopor kecerdasan buatan, menyatakan keyakinannya bahwa "suatu hari komputasi akan menjadi infrastruktur publik seperti listrik dan telepon." Namun baru pada tahun 1995 Larry Ellison, pendiri Oracle, mengusulkan konsep "*Network Computing*" sebagai kampanye untuk menantang dominasi Microsoft, yang pada saat itu mendominasi komputasi desktop dengan Windows 95. Larry Ellison mengusulkan agar pengguna tidak memerlukan sejumlah besar perangkat lunak, termasuk sistem operasi dan berbagai program lain, dijejalkan ke PC desktop mereka.

PC Desktop dapat diganti dengan terminal yang terhubung langsung ke server dan menyediakan lingkungan yang ramah pengguna dengan berbagai persyaratan perangkat lunak. Munculnya beberapa produsen, seperti Sun Microsystems dan Novell Netware, yang menawarkan klien *Network Computing* sebagai pengganti desktop menghangatkan konsep "*Network Computing*." Namun pada akhirnya gaung *Network Computing* hilang dengan sendirinya, karena kualitas jaringan komputer pada saat itu masih kurang memadai, sehingga akses NC (*Network Computing*) menjadi sangat lambat, dan masyarakat akhirnya kembali ke memilih kenyamanan dari PC desktop, bersama dengan harga yang lebih rendah.

Sebagai akibat dari ketidakpraktisan program berbasis web, terobosan baru, *Cloud Computing*, telah dikembangkan. Aplikasi *Cloud Computing* tidak bergantung pada sistem operasi yang digunakan oleh pengguna (jadi boleh saja menggunakan Linux, Mac OS, MS Windows, atau bahkan PDA atau ponsel). Yang penting, pengguna dapat terhubung ke Internet dan menavigasi ke alamat atau situs tertentu untuk menjalankan program yang ia butuhkan. Aplikasi Google adalah contoh yang paling umum, dan termasuk penyelenggara (pengelola data

hubungan, jadwal, atau kalender, dan email) serta aplikasi bisnis (pengolah kata, pengolah angka, dan program presentasi). Aplikasi ini tidak hanya gratis, tetapi juga terus diperbarui oleh pengembang. Pengguna tidak diharuskan membayar apa pun kecuali mereka memerlukan fitur tambahan. (Anggeriana, 2011).

Akhir 1990-an melihat pengenalan konsep ASP (*Application Service Provider*). Akses ke aplikasi menjadi lebih cepat seiring dengan peningkatan kualitas jaringan komputer. Sejumlah pemilik data *center* melihat ini sebagai peluang untuk menawarkan fasilitas mereka sebagai tempat meng-host aplikasi yang dapat diakses pelanggan melalui jaringan komputer. Akibatnya, pelanggan tidak perlu berinvestasi dalam peralatan pusat data. Hanya saja ASP ini masih “*private*”, artinya layanan ini hanya dikustomisasi untuk satu pelanggan, sedangkan aplikasi yang disediakan saat itu kebanyakan *client-server*. Karena hadirnya berbagai teknik baru dalam pengembangan perangkat lunak di awal abad kedua puluh satu, khususnya di bidang pemrograman berbasis web, serta peningkatan kapasitas jaringan internet, situs internet tidak lagi hanya berisi informasi statis. . Namun, sudah mulai mengarah ke aplikasi bisnis yang lebih kompleks. Dan, seperti yang telah disebutkan sebelumnya, *popularitas Cloud Computing* tumbuh ketika, pada awal 2000-an, Marc Benioff, mantan VP Oracle, meluncurkan layanan aplikasi CRM dalam bentuk *Software as a Service*, Salesforce.com, yang mendapat sambutan luar biasa. dalam dunia Teknologi Informasi. Dengan misinya yang terkenal "*The End of Software*", Benioff dapat dikatakan telah berhasil mewujudkan visi bosnya di Oracle, Larry Ellison, tentang *Network Computing* menjadi kenyataan satu dekade kemudian.

Kemudian *Cloud Computing* menyapu dunia teknologi informasi seperti bola salju. Mulai tahun 2005, ada inisiatif yang dipimpin oleh nama-nama besar seperti Amazon.com meluncurkan Amazon EC2 (*Elastic Compute Cloud*), Google dengan Google App Engine-nya, dan IBM meluncurkan Blue Cloud Initiative, antara lain. Semua inisiatif ini masih dalam pengerjaan, dan bentuk *Cloud Computing* masih mencari bentuk terbaik, baik secara praktis maupun akademis. Bahkan dari sisi akademis, jurnal yang membahas topik ini baru muncul dalam tiga tahun terakhir. Akhirnya, seperti yang bisa kita lihat, semua nama

besar terlibat dalam pertempuran untuk kontrol cloud ini. Bahkan pabrikan Dell, pernah mencoba mematenkan istilah "*Cloud Computing*", tetapi ditolak oleh otoritas paten Amerika. Meski di luar konflik mendung mendung, namun tidak demikian di tanah air tercinta Indonesia ini. Masih ada beberapa pemain yang mencoba memasuki area ini, bahkan tidak cukup untuk mengisi satu jari tangan. PT Telkom yang saat ini menawarkan setidaknya dua layanan aplikasi berbasis *Software as a Service* merupakan salah satu yang cukup serius untuk bermain di bidang ini. Salah satunya disediakan oleh anak perusahaannya, "Sigma Cipta Caraka", yang menyediakan layanan aplikasi core banking untuk bank kecil dan menengah. Kemudian, bekerja sama dengan IBM Indonesia dan mitra bisnisnya, PT Codephile, Telkom menawarkan layanan e-Office on Demand untuk kebutuhan kolaborasi/korespondensi perusahaan atau organisasi.

Respon tenang dunia teknologi informasi dalam negeri terhadap *Cloud Computing* dapat disebabkan oleh beberapa faktor, antara lain:

1. Penetrasi infrastruktur internet bisa dibilang masih terbatas.
2. Tingkat kedewasaan pengguna internet yang masih menjadikan media internet utamanya sebagai media hiburan atau sosialisasi..
3. Diperlukan investasi yang tinggi untuk menyediakan layanan cloud ini, karena harus merupakan kombinasi infrastruktur jaringan, hardware dan software sekaligus..

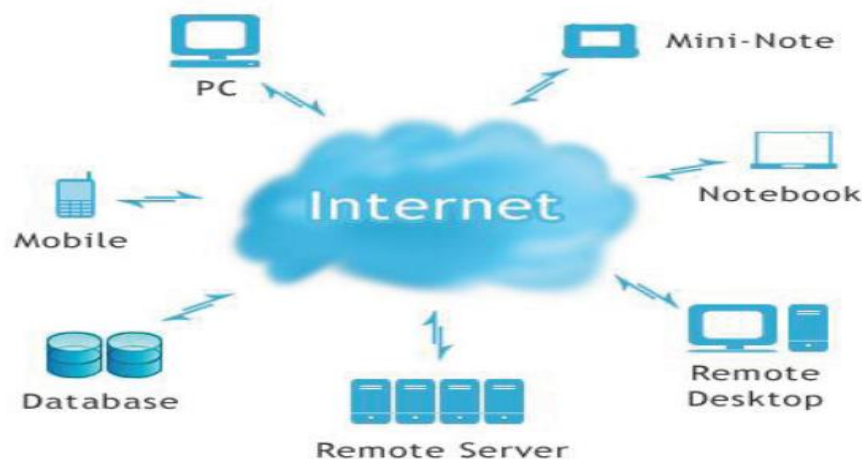
Namun, sebagai penduduk terbesar kelima di dunia, dan dengan demikian pasar terbesar kelima di dunia, para pelaku teknologi informasi dalam negeri harus segera mempersiapkan diri dengan mulai mengembangkan layanan siap pakai. menjadi berkabut Akibatnya, ketika gelombang besar *Cloud Computing* tiba, itu akan menguntungkan lebih dari sekadar pemain asing besar. Tentu saja peran pemerintah sebagai fasilitator dan regulator sangat dibutuhkan dalam situasi ini. Paradigma atau pandangan *Cloud Computing* ini masih terus berkembang dan menjadi sumber perdebatan di kalangan akademisi, vendor teknologi informasi, instansi pemerintah, dan pihak berkepentingan lainnya. Untuk memberikan landasan bersama bagi publik, pemerintah AS telah mengeluarkan beberapa rekomendasi standar tentang berbagai aspek *Cloud Computing* melalui *National*

Institute of Science and Technology (NIST), yang merupakan bagian dari Departemen Perdagangan AS. (Anggeriana, 2011)

Beberapa contoh historis menunjukkan bagaimana konsep pengembangan kerangka kerja komputasi online berkembang sebagai berikut: (Anggeriana, 2011)

1. Portal internet yang memiliki berbagai fasilitas layanan publik mulai dari surat elektronik (e-mail), forum diskusi hingga penyimpanan dokumen dengan media penyimpanan yang sangat luas (bahkan ada yang menyediakan ruang penyimpanan tidak terbatas) - hingga mekanisme berbagi dokumen, blog layanan dll. Semuanya disediakan di satu tempat.
2. Perangkat Lunak sebagai Layanan atau layanan SaaS dari vendor teknologi informasi terkemuka - mulai dari layanan pemindaian virus online hingga layanan pemindaian spam, dll
3. Layanan *SpeedyWiki* ini secara sederhana dapat disebut sebagai dasar-dasar *Cloud Computing* dalam arti fasilitas *SpeedyWiki* ini dapat diakses dan digunakan secara bersamaan untuk berkolaborasi dalam menyusun dokumentasi yang sangat kompleks.
4. Aplikasi *Point of Sale* atau POS di kasir supermarket dengan metode Terminal *Service* juga bisa dikategorikan sebagai dasar *Cloud Computing*.

2.5.3 Pengeretian *CloudComputing*



Gambar 2.3 *Cloud Computing* (Ashari & Setiawan, 2014)

Komputasi awan adalah hibrida dari teknologi komputer dan pengembangan berbasis internet. Karena awan sering digambarkan dalam diagram jaringan komputer, itu adalah metafora untuk internet. *Cloud* di *CloudComputing*, seperti *cloud* dalam diagram jaringan komputer, adalah abstraksi dari infrastruktur kompleks yang disembunyikannya. Ini adalah metode komputasi di mana kemampuan terkait teknologi informasi disajikan sebagai layanan, memungkinkan pengguna untuk mengaksesnya melalui Internet ("Cloud") tanpa mengetahui apa yang ada di dalamnya, menjadi ahli dengannya, atau memiliki kendali atas infrastruktur teknologi yang membantunya. Menurut makalah IEEE *Internet Computing* yang diterbitkan pada tahun 2008, "*CloudComputing* adalah suatu paradigma dimana informasi disimpan secara permanen di server di internet dan disimpan sementara di komputer pengguna (klien) termasuk desktop, komputer tablet, komputer notebook, komputer dinding, genggam, sensor, monitor, dan sebagainya."

Komputasi awan adalah konsep luas yang mencakup SaaS, Web 2.0, dan tren teknologi terkini lainnya yang terkenal, semuanya memiliki tema yang sama yaitu mengandalkan Internet untuk memenuhi kebutuhan komputasi pengguna. Google Apps, misalnya, menawarkan akses online ke aplikasi bisnis umum melalui browser web, dengan perangkat lunak dan data disimpan di *server*. Komputasi awan adalah tren teknologi terkini, dan iCloud adalah salah satu contoh perkembangan teknologi komputasi awan jenis ini. *Cloudcomputing* merupakan istilah baru dalam dunia teknologi informasi.

Cloud adalah metode komputasi yang mengandalkan teknologi internet. Awan, seperti yang didefinisikan dalam bahasa Indonesia, adalah simbol internet, tempat penyimpanan sumber daya, perangkat lunak, dan informasi bersama. Disediakan untuk komputer sesuai permintaan dan perangkat lain, seperti jaringan listrik. *Cloudcomputing* merupakan perubahan paradigma yang terjadi pada awal tahun 1980-an, mengikuti transisi dari mainframe ke *clientserver*. Karena *cloud* memusatkan semua data, maka akan dapat memberikan kemudahan dalam proses layanan jika diterapkan di instansi pemerintah. Konsep implementasi *cloud* dapat ditelusuri kembali ke pola teknologi dan sistem

informasi di perbankan. Namun, tidak semua layanan internet masuk dalam kategori *cloudcomputing*.(Fajrin, 2012).

2.5.4 Layanan yang ditawarkan *cloudcomputing*

Dengan menyediakan komponen-komponen yang diperlukan seperti *server*, *hardware*, dan jaringan. Pengguna komputasi awan dapat menginstal aplikasi mereka pada infrastruktur. Pengguna komputasi awan juga dapat memilih cara menggunakan layanan komputasi awan yang disediakan oleh vendor berdasarkan kebutuhan spesifik mereka.

1) Infrastructure as a service

Layanan ini disediakan dengan menyediakan komponen-komponen yang diperlukan berupa *server*, perangkat keras, dan jaringan dengan harga yang ditetapkan. Pengguna komputasi awan dapat menginstal aplikasi mereka pada infrastruktur. Contoh: hosting untuk aplikasi web Sebagai contoh vendor penyedia layanan, pertimbangkan semua penyedia hosting.

2) Platform as a service

Layanan yang menyediakan perangkat lunak sistem dan perangkat lunak pendukung yang diperlukan untuk membangun aplikasi yang akan diinstal pada *server* berdasarkan kebutuhan organisasi/instansi. Organisasi/lembaga tersebut kemudian membangun dan menggunakan aplikasi yang dibutuhkan pada platform ini. Analogi yang dapat digunakan untuk menggambarkan layanan ini adalah menyewakan rumah dengan segala isinya sehingga pengguna dapat langsung menggunakan rumah tersebut. Pengembangan aplikasi jarak jauh adalah contoh bagaimana hal ini dapat dilakukan. *BizNet*, *Microsoft*, dan *LintasArta* adalah beberapa contoh vendor.

3) Software as a service

Disediakan dengan membuat perangkat lunak dan aplikasi tersedia untuk pelanggan melalui internet. Panel *front-end* digunakan oleh penyedia layanan komputasi awan untuk berinteraksi dengan pengguna dan pelanggan. Email dan dokumen online adalah dua contoh layanan sederhana. Contoh layanan (agak

sulit): SAP di internet Google, Amazon, SAP, dan vendor lainnya adalah contohnya.(Fajrin, 2012)

2.5.5 Penyebaran komputasi awan

Menurut (Snaith et al., 2011),ada beberapa penyebaran komputasi awan sebagai berikut:

a. *Private cloud* (Swasta awan)

Infrastruktur *cloud* hanya digunakan oleh organisasi. Itu dapat dikelola di tempat atau di luar situs oleh organisasi atau pihak ketiga.

b. *Community cloud* (Masyarakat awan)

Infrastruktur *cloud* digunakan bersama oleh beberapa organisasi dan melayani komunitas tertentu dengan masalah serupa (mis., misi, persyaratan keamanan, kebijakan, dan pertimbangan kepatuhan). Itu dapat dikelola di tempat atau di luar situs oleh organisasi atau pihak ketiga.

c. *Public cloud*.

Infrastruktur *cloud* terbuka untuk umum atau kelompok industri besar, dan dimiliki oleh perusahaan yang menyediakan layanan *cloud*.

d. *Hybrid cloud* (Hybrid awan)

Infrastruktur *cloud* terdiri dari dua atau lebih *cloud* (swasta, komunitas, atau publik) yang masih merupakan entitas yang berbeda tetapi dihubungkan oleh standar kepemilikan atau teknologi yang memungkinkan portabilitas data dan aplikasi.

2.5.6 Fungsi dan Manfaat *CloudComputing*

Cloud Compuitng memanfaatkan server komputer yang sangat cepat dan cepat, yang saat ini disediakan oleh banyak penyedia yang menyediakan *server media*, memungkinkan universitas menghemat uang dan ruang untuk instalasi *server*.*Dropbox*, *GoogleDrive*, dan *iCloud* adalah contoh aplikasi berbasis *cloudcomputing*. Teknologi informasi, seperti tren pembelajaran jarak jauh, pembelajaran elektronik (*e-learning*), perpustakaan elektronik (*e-library*), dan

multimedia, memainkan peran penting dalam menyediakan fasilitas belajar mengajar yang lebih efisien di bidang pendidikan.(Mutia, 2016)

2.5.7 Kelebihan dan Kekurangan *Cloud Computing*

Menurut (Mutia, 2016), ada beberapa Kelebihan atau Keuntungan *Cloud Computing* antara lain:

1. Manfaat bagi spesialis keuangan adalah meminimalkan biaya usaha yayasan publik sehingga organisasi dapat membidik tambahan pada sudut pandang kegunaan.
2. Untuk perancang aplikasi, administrasi PaaS (*Stage as a help*) memungkinkan pengembangan dan eksekusi aplikasi yang cepat dengan cara ini meningkatkan kegunaan..
3. Bagi para ahli yang menggeluti bisnis TI, ini berarti peluncuran sektor bisnis baru untuk industri administrasi kemajuan inovasi data.
4. Bagi organisasi di bidang kerangka kerja, ini merupakan peluang yang luar biasa karena perluasan penggunaan administrasi PaaS akan membangun penggunaan transmisi data web.
5. Penggabungan aplikasi dengan gadget yang berbeda.

Resiko atau kekurangan yang harus dihadapi pengguna *Cloud Computing* ini antara lain:

1. Tingkat administrasi adalah kemungkinan terjadinya konflik pelaksanaan bantuan dari pemasok. Ketidakteraturan pemasok cloud ini mencakup keamanan informasi dan pemulihan informasi.
2. Kerahasiaan, yang mengandung arti adanya bahaya bahwa informasi klien akan diperoleh oleh orang lain dengan alasan bahwa memfasilitasi dilakukan bersama-sama.
3. Konsistensi, yang menyinggung bahaya penyimpangan dari tingkat konsistensi pemasok dengan pedoman yang diterapkan oleh klien.
4. Kepemilikan informasi menyinggung bahaya kehilangan kepemilikan informasi setelah informasi disimpan di *cloud*.

5. Portabilitas informasi, yang menyinggung kemungkinan membagi informasi di antara administrasi cloud dan bagaimana memulihkan informasi jika suatu hari klien mengakhiri administrasi komputasi terdistribusi.

2.5.8 Sistem Kerja *Cloud Computing*

Kerangka kerja komputasi terdistribusi dipisahkan menjadi dua bagian: ujung depan dan ujung belakang. Mereka terkait satu sama lain melalui sebuah organisasi, biasanya web. Ujung depan adalah sisi klien PC (klien), atau (pelanggan), dan ujung belakang adalah bagian "*cloud*" dari kerangka kerja. Ujung depan menggabungkan PC pelanggan (atau PC organisasi) dan aplikasi yang diperlukan untuk mencapai kerangka kerja komputasi terdistribusi. Tidak semua kerangka kerja komputasi terdistribusi memiliki UI yang serupa. Administrasi, misalnya, program email elektronik mengeksplorasi browser Internet yang ada seperti Web Pioneer atau Firefox. Kerangka kerja yang berbeda memiliki aplikasi luar biasa yang memberikan akses jaringan kepada pelanggan.

Di bagian belakang kerangka kerja adalah berbagai PC, *server*, dan kerangka penyimpanan informasi yang menjadikan "*cloud*" administrasi pendaftaran. Biasanya setiap aplikasi akan memiliki dedicated server sendiri. *Server* fokus menangani kerangka kerja, memeriksa lalu lintas, dan permintaan pelanggan untuk menjamin semuanya berjalan sesuai rencana. Kerangka kerja ini menyimpan banyak pedoman yang dianggap sebagai konvensi dan menggunakan jenis pemrograman yang tidak biasa yang disebut *middleware*. *Middleware* organisasi memungkinkan PC untuk berbicara satu sama lain. Umumnya, server tidak berjalan pada batas penuh. Itu menyiratkan ada daya penanganan yang hasilnya tidak digunakan. Kemudian, pada saat itu, Anda akan membutuhkan jalan. Prosedur ini disebut virtualisasi *server*. Dengan memperkuat hasil setiap *server*, virtualisasi *server* mengurangi kebutuhan mesin untuk berjalan. (Mutia, 2016)

2.6 *Android*

2.6.1 *Sejarah Android*

Pada Oktober 2003, empat profesional TI, Andi Rubin, Rich Miner, Scratch Singes, dan Chris White, mendirikan *Android.Inc* di California, AS. Visi *Android* untuk memahami ponsel yang lebih sensitif dan pemiliknya kemudian didasarkan pada dunia virtual raksasa Google. Google membeli *Android* pada Agustus 2005 saat itu. Sistem operasi *Android* didasarkan pada platform Linux, yang bersifat *opensource*. *Android* juga *opensource*. Dengan nama besar Google dan konsep *opensource* pada sistem operasi *Android*, tidak butuh waktu lama bagi *Android* untuk bersaing dan menyelamatkan sistem operasi Serbaguna lainnya seperti Symbian, Windows Portable, Blackberry, dan iOS. Saat ini, siapa yang tidak mengenal *Android*, yang telah menjadi pemimpin dalam kerangka kerja untuk ponsel. (Lengkong et al., 2015)

Android merupakan *frameworkmobile* berbasis Linux yang menggabungkan framework, middleware, dan aplikasi. *Android* menyediakan platform terbuka di mana pengembang dapat membangun aplikasi mereka. Awalnya, Google Inc. membeli *Android Inc.*, sebuah startup yang membuat pemrograman untuk ponsel atau ponsel. Open Handset Union, konsorsium dari 34 komunikasi media, pemrograman, dan organisasi perangkat, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Versatile, dan Nvidia, dibentuk pada saat itu untuk mengembangkan *Android*. Pada hari debut *Android*, 5 November 2007, *Android* dan Open Handset Coalition menyatakan bahwa mereka mendukung kemajuan open source di ponsel. Google, di sisi lain, merilis kode *Android* di bawah izin Apache, izin produk, dan ponsel panggung terbuka. (Andi, 2015)

2.6.2 *Pengertian Android*

Android merupakan framework ponsel yang menggabungkan framework, middleware, dan aplikasi. *Android* menyediakan platform gratis dan terbuka bagi pengembang untuk membuat aplikasi mereka sendiri. Awalnya, Google Inc. membeli *Android Inc.*, sebuah startup yang membuat pemrograman untuk ponsel

atau ponsel. *Open Handset Partnership*, sebuah konsorsium dari 34 perusahaan peralatan penyiaran, pemrograman, dan komunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Versatile, dan Nvidia, dibentuk pada saat itu untuk mengembangkan *Android*. (Sanitasari et al., 2017).

2.6.3 Fitur-Fitur *Android*

Android adalah kerangka kerja serbaguna *opensource* dan dapat dibuat untuk dikerjakan ulang sehingga tidak ada pengaturan yang berbeda sehubungan dengan produk dan peralatan. Meskipun demikian, secara keseluruhan *Android* sendiri menjunjung tinggi unsur-unsur yang menyertainya: (Zikria, 2016).

1. *Storage*: Mendukung *SQL Lite*. *SQL Lite* adalah sebuah *database relational lite* (versi ringan) yang digunakan untuk penyimpanan data.
2. *Konektivitas*: Mendukung koneksi GSM/*EDGE*, IDEN, CDMA, EVDO, UMTS, Bluetooth, Wifi, LTE, dan Wimax.
3. *Messaging*: Mendukung SMS dan MMS.
4. *Web Browser* yang digunakan adalah browser berbasis *opensource* Webkit, dengan *engine java script Chrome V8*.
5. *Media Support*: Dukungan media meliputi *file* media bertipe: H.263, H.264 (dalam bentuk 3GP atau MP4 *container*), MPEG-4 SP, AMR, AMR-WB (3GP *container*), AAC, HE-AAC (dalam bentuk MP4 atau 3GP *container*), MP3, MIDI, Ogg, Vorbis, WAV, JPEG, PNG, *GIF*, dan BMP.
6. Dukungan *hardware*: Sensor *accelometer*, *camera*, kompas digital, *sensor proximity*, dan *GPS (Global Positioning System)*.
7. *MultiTouch*: Mendukung layar dengan dukungan *multitouch*.
8. *Multitasking*: Kemampuan untuk melaksanakan tugas secara bersamaan.
9. *Flash*: Mendukung animasi *Flash*.
10. *Tethering*: mendukung berbagi koneksi internet.

2.6.4 Arsitektur *Android*

Menurut (Nazruddin, 2012) Arsitektur *Android* terdiri dari bagian-bagian seperti berikut :

1. *Applications* dan *Widgets*

Ini adalah lapisan di mana kami hanya mengelola aplikasi, di mana kami biasanya mengunduh aplikasi dan kemudian memperkenalkan dan menjalankannya. Aplikasi pusat pada lapisan termasuk pelanggan email, program SMS, jadwal, peta, program, kontak, dan daftar terus. Bahasa pemrograman Java digunakan untuk mengkompilasi semua aplikasi.

2. *Applications Frameworks Android* adalah "*OpenDevelopment Platform*"

Artinya, *Android* menawarkan desainer atau memungkinkan para insinyur untuk membuat aplikasi yang hebat dan imajinatif. Desainer diizinkan untuk mengakses peralatan, mengakses aset data, menjalankan administrasi yayasan, menetapkan peringatan, dan menambahkan peringatan status, dll. Desainer memiliki akses penuh ke API struktur seperti halnya aplikasi klasifikasi pusat. Rekayasa aplikasi ini direncanakan agar kita dapat dengan mudah menggunakan kembali bagian-bagian yang telah dimanfaatkan (reuse). Jadi dapat kita simpulkan bahwa *Application Structures* adalah lapisan tempat pembuat aplikasi membuat/membuat aplikasi yang akan berjalan pada kerangka kerja *Android*, karena pada lapisan inilah aplikasi dapat direncanakan dan dibuat, misalnya *content-supplier* seperti SMS dan panggilan. Bagian-bagian yang diingat untuk Sistem Aplikasi adalah sebagai berikut:

- a. *Views*
- b. *ContentProvider*
- c. *ResourceManager*
- d. *NotificationManager*
- e. *ActivityManager*

3. *Libraries* adalah lapisan di mana *highlightAndroid* ditemukan, umumnya produsen aplikasi mengakses perpustakaan untuk menjalankan aplikasi mereka. Berjalan di atas bagian, lapisan ini menggabungkan pustaka C/C++ pusat seperti *Libc* dan *SSL*, seperti:

- a. *libraries* media untuk pemutaran media audio dan video.
- b. *libraries* untuk manajemen tampilan.

- c. *librariesGraphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
- d. *libraries SQLite* untuk dukungan *database*.
- e. *libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
- f. *librariesLiveWebcore* mencakup modern *web browser* dengan *engine* embeded *webview*.
- g. *libraries 3D* yang mencakup implementasi OpenGL ES 1.o API's

4. *Android RunTime*

Ini adalah lapisan yang membuat aplikasi *Android* berjalan yang dalam prosesnya menggunakan eksekusi Linux.*DalvikVirtualMachine* (DVM) adalah mesin yang membentuk premis dari struktur aplikasi *Android*. Di *AndroidRunTime* dipartisi menjadi dua bagian, lebih spesifiknya:

a. *Core Libraries*

Aplikasi *Android* dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa *java/c* yang ditangani oleh *Core Libraries*.

b. *Dalvik Virtual Machine*

Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi - fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux *kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*

Ini adalah lapisan tempat pusat kerangka kerja *Android* ditemukan. Berisi catatan kerangka kerja yang mengawasi kerangka kerja penanganan, memori, *aset*, *driver*, dan kerangka kerja *Android* lainnya. Bagian Linux yang digunakan oleh *Android* adalah Linux bit discharge 2.6.

2.6.5 **Versi-Versi *Android***

Menurut (Kusniyati & Pangondian Sitanggang, 2016), Sejak rilis awalnya pada tahun 2007, *Android* telah berkembang dengan sangat pesat, sebagaimana dibuktikan oleh berbagai versi sistem operasi *Android*.

1. *Android* versi 1.1



Gambar 2.4 *Android* Versi 1.1 (Adikara, 2013)

Android pertama kali diluncurkan pada tahun 2007, tetapi kerangka kerja ini mulai dikirimkan dan disebarluaskan ke perangkat yang berbeda pada Walk 9 sebelumnya, 2009. Adaptasi *Android* 1.1 adalah *Android* yang mendasarinya di mana bentuk ini hanya memberikan sentuhan pada beberapa aplikasi seperti kerangka kerja UI yang unggul, serta beberapa aplikasi yang berbeda.

2. *Android* versi 1.5 (*Cupcake*)



Gambar 2.5 *Android* Versi 1.5 (*Cupcake*)(Adikara, 2013)

Pada Mei 2009, *Android* mengalami perubahan rendisi lain. *Android* versi 1.1 kemudian ditingkatkan ke *Android* 1.5, yang juga dikenal sebagai *AndroidCupcake*.

3. *Android* versi 1.6 (*Donut*)



Donut
Android 1.6

E.

Gambar 2.6 *Android* Versi 1.6 (*Donut*)(Adikara, 2013)

Sejak rilis awal *AndroidCupcake* pada September 2009, *Donut* (*Donut* 1.6) telah dikirimkan dalam waktu kurang dari 4 bulan.

4. *Android* versi 2.0/2.1 (*Eclair*)



Eclair
Android 2.0/2.1

Gambar 2.7 *Android* Versi 2.0/2.1 (*Eclair*)(Adikara, 2013)

Pada tahun yang sama, *Android* kembali merilis versi terbaru dari sistem operasi, *Android 2.0/2.1 Eclair*. Google merilis *AndroidEclair* tiga bulan setelah rilis awal.

5. *Android* versi 2.2 (*Froyo: Frozen Yoghurt*)



E.

Gambar 2.8 *Android* Versi 2.2 (*Froyo: Frozen Yoghurt*)(Adikara, 2013)

Google menghabiskan waktu 5 bulan untuk mengonversi dari *AndroidEclair* versi sebelumnya ke versi *Froyo, Frozen, Yoghurt*. *Android* versi 2.2, dijuluki *AndroidFroyo*, dirilis pada 20 Mei 2010.

6. *Android* versi 2.3 (*Gingerbread*)



Gambar 2.9 *Android* Versi 2.3 (*Gingerbread*)(Adikara, 2013)

7 bulan kemudian, *Android* membuat terobosan lain dengan merilis kembali *Android* versi 2.3, juga dikenal sebagai *AndroidGingerbread*.

7. *Android* versi 3.0/3.1 (*Honeycomb*)



Gambar 2.10 *Android* Versi 3.0/3.1 (*Honeycomb*)(Adikara, 2013)

Android 3.0/3.1, juga dikenal sebagai *AndroidHoneycomb*, dirilis pada Mei 2011. *AndroidHoneycomb* adalah versi sistem operasi *Android* yang dirancang khusus untuk digunakan dengan tablet berbasis *Android*.

8. *Android* versi 4.0 (ICS: *Ice Cream Sandwich*)



Gambar 2.11 *Android* Versi 4.0 (ICS: *Ice Cream Sandwich*)(Adikara, 2013)

Android ICS atau *Ice Cream Sandwich* juga dirilis pada tahun yang sama dengan *Honeycomb*, yaitu pada bulan Oktober 2011.

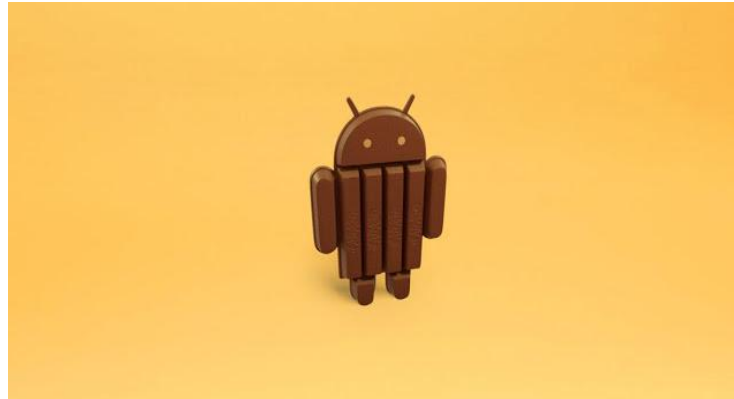
9. *Android* versi 4.1 (*Jelly Bean*)



Gambar 2.12 *Android* Versi 4.1 (*Jelly Bean*)(Adikara, 2013)

Android Jelly Bean adalah versi terbaru dari *Android* yang tersedia saat ini. Google Nexus 7 adalah salah satu perangkat yang menjalankan sistem operasi *JellyBean*, dan dikembangkan oleh ASUS, sebuah perusahaan Taiwan yang berbagi kota kelahiran Acer.

10. *Android* versi 4.4 (*Kit Kat*)



Gambar 2.13 *Android* Versi 4.4 (*Kit Kat*)(Adikara, 2013)

Kehadiran *AndroidKitKat* ini merupakan peluncuran produk OS baru yang diluncurkan pada 4 September 2013. Sebelumnya memang ramai dibicarakan bahwa *Android* akan meluncurkan OS baru bernama *AndroidKeyLimePie*, namun setelah dianalisa ternyata menemukan bahwa itu tidak cocok dengan ejaan orang biasa, sehingga namanya diubah menjadi *Android OS KitKat*. yang kebanyakan orang kenal.

11. *Android* versi 5.0.2 (*Lollipop*)



Gambar 2.14 *Android* Versi 5.0.2 (*Lollipop*)(Adikara, 2013)

Android Lollipop adalah sistem operasi *Android* yang saat ini membentuk pola baru di industri telepon seluler; Hal ini tidak lepas dari keunikan dan

keunggulan OS tersebut. Banyak orang yang menantikan kedatangan *Android* versi ini karena diharapkan sistem operasi *Lollipop* akan lebih baik dari versi sebelumnya.

12. *Android* versi 6.0 (*Marshmallow*)



Gambar 2.15 *Android* Versi 6.0 (*Marshmallow*) (Adikara, 2013)

Android 6.0 Marshmallow adalah versi sistem operasi seluler yang dikembangkan oleh Google. *Android M* pertama kali diumumkan pada Mei 2015 di Google I/O dan dirilis secara resmi pada Oktober 2015. *Android Marshmallow* memperkenalkan model izin aplikasi yang didesain ulang; sekarang hanya ada delapan kategori izin, dan aplikasi tidak lagi diberikan semua hak akses secara default.

2.7 *Android Studio*



Gambar 2.16 *Android Studio* (Andi, 2015)

Google memperkenalkan *Android Studio*, sebuah IDE untuk pengembangan *Android*, pada acara Google I/O 2013. *Android Studio* adalah pengembangan *Eclipse IDE* yang didasarkan pada Java IDE IntelliJ IDEA yang populer. *Android Studio* adalah IDE resmi untuk mengembangkan aplikasi *Android*. Jika dibandingkan dengan *Eclipse IDE*, *Android Studio* sebagai pengembangan

Eclipse memiliki banyak fitur baru. Tidak seperti *Eclipse*, yang menggunakan Ant sebagai lingkungan fabrikasinya, *Android Studio* menggunakan *Gradle*.

Fitur-fitur lainnya adalah sebagai berikut :

1. Menggunakan *Gradle-based build system* yang fleksibel.
2. Bisa mem-build *multiple APK* .
3. *Templates support* untuk *Google Services* dan berbagai macam tipe perangkat.
4. *Layout editor* yang lebih bagus.
5. *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*.

Karena *Android Studio* menggunakan *Gradle*, maka lebih unggul dari *Eclipse* dalam hal konstruksi. Selanjutnya, tidak seperti *Eclipse*, kita tidak perlu lagi mengutak-atik dependensi paket di *Android Studio*. Fitur lain yang membedakan *Android Studio* dari *Eclipse* adalah dukungan format manajer xml yang unggul secara visual. (NUR YATI, 2018)

2.8 *Android SDK (Software Development Kit)*



Gambar 2.17 *Android SDK* (NUR YATI, 2018)

Android-SDK adalah kit pengembangan untuk programmer yang ingin membuat aplikasi berbasis Google *Android*. *Android SDK* dilengkapi dengan banyak alat pengembangan. Debugger, pustaka, emulator handset, dokumentasi, contoh kode, dan tutorial adalah bagian dari *Android SDK*. *Android* saat ini

mendukung arsitektur x86 di Linux (distribusi Linux modern apa pun), Mac OS X 10.4.8 atau lebih baru, dan Windows XP atau Vista. JDK, Apache Ant, dan Python 2.2 atau yang lebih baru diperlukan. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih tinggi dengan plugin *AndroidDevelopmentTools* (ADT), yang memungkinkan pengembang menggunakan editor teks untuk mengedit file Java dan XML serta alat baris perintah untuk membuat, membangun, men-debug, dan mengontrol perangkat *Android*. (NUR YATI, 2018)

2.9 Bahasa Pemrograman Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM). Kotlin adalah bahasa pemrograman Android praktis yang menggabungkan bahasa berorientasi objek (OO) dan fungsional. Kotlin juga merupakan bahasa pemrograman yang dapat dioperasikan, memungkinkannya untuk digunakan bersama dengan bahasa pemrograman Java dalam satu proyek. Bahasa pemrograman ini juga dapat digunakan untuk membuat aplikasi untuk area kerja, web, dan backend. JetBrains, perusahaan di belakang IntelliJ IDEA, menciptakan Kotlin terlebih dahulu. JetBrains merilis Kotlin *opensource* setelah banyak pengembangan, dan sekarang berkembang lebih jauh. Kotlin didukung penuh oleh Google untuk pengembang aplikasi *Android*. (Febriandirza, 2020)

Menurut (Sibarani et al., 2018) Berikut ini adalah beberapa keuntungan yang mungkin didapat jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi pada tahap JVM:

1. Dapat mengatasi *NullPointerException* yang umumnya terdapat pada Java.
2. Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.
3. Mudah dipelajari.
4. Dukungan IDE untuk mempermudah dalam pemrograman.

2.10 *GoogleMapsAPI*

Google Maps adalah layanan gratis populer yang disediakan oleh Google. *GoogleMaps* adalah peta dunia yang dapat kita gunakan untuk melihat lokasi tertentu. *GoogleMaps*, seperti organisasi individual lainnya, adalah peta yang dapat dilihat menggunakan program. Kita dapat menggunakan *GoogleMapsAPI* untuk menambahkan fitur Google Maps yang telah kita buat di web atau di blog kita, baik yang berbayar maupun yang gratis. *GoogleMapsAPI* adalah perpustakaan berbasis *JavaScript*. (Ariyanti et al., 2015)

Google menyediakan empat jenis opsi model peta di Google Maps API, antara lain:

- a. *ROADMAP*, menampilkan peta biasa 2 dimensi
- b. *SATELLITE*, menampilkan foto satelit
- c. *TERRAIN*, menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
- d. *HYBRID*, menunjukkan foto satelit yang di atasnya tergambar jalan dan nama kota pada tampilan *ROADMAP*

2.11 UML




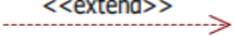

Unified Modeling Language (UML) adalah standar industri untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. UML adalah standar untuk membuat model sistem. (Dharwiyanti & Wahono, 2003). *Unified Modeling Language* (UML) adalah bahasa berbasis grafik atau gambar yang digunakan untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem pengembangan perangkat lunak OOP (*Object-Oriented*). UML juga menyediakan standar untuk membuat cetak biru sistem, yang mencakup konsep proses bisnis, kelas penulisan dalam bahasa pemrograman tertentu, skema informasi premis, dan komponen perangkat lunak sistem. (Suendri, 2018)

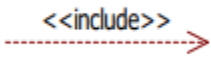
Adapun beberapa diagram yang digunakan pada UML yaitu sebagai berikut:

2.11.1 Use Case Diagram

Diagram *use case* adalah deskripsi fungsionalitas yang diharapkan dari sistem. Penekanannya adalah pada "apa" yang dilakukan sistem daripada "bagaimana" melakukannya. *Use case* adalah representasi dari interaksi antara aktor dan sistem. Kasus penggunaan adalah tugas khusus, seperti masuk ke sistem atau membuat daftar belanja. Entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan tugas tertentu disebut sebagai entertainer. Grafik *use case* bisa sangat berguna saat mengkompilasi persyaratan sistem, mengkomunikasikan desain dengan klien, dan merancang kasus uji untuk semua fitur sistem. (Dharwiyanti & Wahono, 2003)

Tabel 2.1 Simbol *Use Case* Diagram (Hendini, 2016)



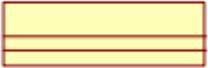


NO	Simbol	Keterangan
1		<i>Entertainer</i> mendefinisikan serangkaian peran yang diambil pengguna saat berinteraksi dengan use case atau eksposisi orang.
2		<i>Use case</i> Contoh kasus Sebuah deskripsi dari sistem urutan tindakan yang menghasilkan hasil yang terukur untuk aktor.
3		Association/Association Menyatukan satu objek dengan objek lainnya.
4		Augmentasi/Extend Hubungan kasus penggunaan target yang memperluas perilaku kasus penggunaan sumber pada titik tertentu dijelaskan.
5		Speculation/Generalization Hubungan generalisasi dan spesialisasi (khusus umum) antara dua kasus penggunaan di mana satu fungsi lebih umum daripada yang lain.
6		Menggunakan/ <i>Include</i>



	Tunjukkan kasus penggunaan sumber secara eksplisit.
---	---

2.11.2 *Classdiagram*

Diagram kelas adalah spesifikasi yang, ketika dipakai, menghasilkan objek, dan merupakan pusat desain dan pengembangan berorientasi objek. Diagram kelas juga menggambarkan status sistem (atribut/properti). Kelas adalah format untuk membentuk objek. *Classdiagram* adalah chart yang terdiri dari sekumpulan objek yang memiliki atribut dan metode. Diagram kelas menggambarkan struktur dan deskripsi kelas, paket, dan objek, serta hubungan mereka. (Ali Ikhwan, 2018)

Tabel 2.2 Simbol *Classdiagram* (Hendini, 2016)




NO	Simbol	Keterangan
1		Spekulasi Hubungan di mana objek anak (anak-anak) berbagi perilaku dan struktur informasi objek yang lebih tinggi dalam hierarki daripada objek induk (leluhur).
2		<i>Nary Association</i> berusaha menghindari asosiasi dengan lebih dari dua objek.
3		<i>Class</i> Kumpulan objek dengan atribut dan operasi yang sama.
4		<i>Collaboration</i> Kolaborasi Sebuah deskripsi dari sistem urutan tindakan yang menghasilkan hasil yang terukur untuk aktor.
5		<i>Realization</i> Operasi yang benar-benar dilakukan oleh sebuah objek.




6		<i>Dependency</i> Sebuah hubungan di mana perubahan dalam elemen independen berdampak pada elemen yang bergantung pada elemen dependen.
7		<i>Association</i> Apa yang menghubungkan antara objek satu dengan objek lainnya

2.11.3 Activity Diagram

Activity Diagram adalah deskripsi aliran aktivitas yang berbeda dalam sistem yang dirancang, termasuk bagaimana setiap aliran dimulai, keputusan yang mungkin dibuat, dan bagaimana mereka berakhir. Eksposisi paralel yang mungkin terjadi dalam beberapa eksekusi juga dapat digambarkan menggunakan diagram aktivitas. *Activity Diagram* adalah jenis diagram keadaan di mana sebagian besar keadaan adalah tindakan dan sebagian besar transisi dipicu oleh penyelesaian keadaan sebelumnya (di dalam pemrosesan). Akibatnya, garis besar aktivitas tidak secara tepat menggambarkan perilaku dalam sistem (atau interaksi antar subsistem), tetapi lebih menggambarkan proses dan jalur aktivitas dari tingkat atas secara umum. (Dharwiyanti & Wahono, 2003)

Tabel 2.3 Simbol *Activity Diagram* (Hendini, 2016)


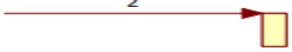
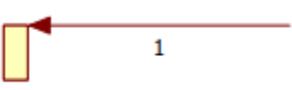
NO	Simbol	Keterangan
1		Status Awal/ <i>Initial</i> Keadaan awal aktivitas sistem diwakili oleh diagram aktivitas.
2		Aktivitas/ <i>Activity</i> Aktivitas sistem, yang biasanya didahului oleh kata kerja.
3		Percabangan / <i>Decision</i> Asosiasi percabangan di mana beberapa kegiatan digabungkan menjadi satu

4		Penggabungan/ <i>JoinAsosiasi</i> penggabungan di mana ada lebih dari satu aktivitas
5		Status Akhir/ <i>Final</i> Status akhir sistem.
6		<i>Swimlane</i> Pisahkan organisasi bisnis yang membawahi kegiatan yang berlangsung.

2.11.4 Sequence Diagram

Sequence diagram adalah representasi berbasis waktu dari interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya). Diagram urutan memiliki dimensi vertikal (waktu) dan dimensi horizontal (objek terkait). Garis besar pengelompokan biasanya digunakan untuk menggambarkan skenario atau serangkaian langkah yang diambil dalam menanggapi suatu peristiwa untuk menghasilkan hasil tertentu. Dimulai dengan apa yang mengawali kegiatan, apa eksposisi dan perubahan yang terjadi di interior, dan apa hasil yang diperoleh. (Dharwiyanti & Wahono, 2003)

Tabel 2.4 Simbol *Sequence Diagram* (Hendini, 2016)

NO	Simbol	Keterangan
1		<i>LifeLine</i> Objek entitas adalah antarmuka yang berkomunikasi satu sama lain.
2		<i>Message</i> Spesifikasi komunikasi antar objek yang berisi informasi tentang aktivitas yang berlangsung
3		<i>Message</i> Spesifikasi komunikasi antar objek yang berisi informasi tentang aktivitas yang berlangsung

2.12 Penelitian Terkait

Penelitian sebelumnya merupakan penelitian terdahulu yang dijadikan acuan oleh penulis ketika melakukan penelitian baru. Tabel di bawah ini merupakan penelitian terdahulu yang relevan dengan judul penelitian penulis.

Tabel 2.5 Penelitian Terdahulu

NO	Judul	Metode	Kelebihan	Kelemahan
1	APLIKASI ABSENSI MENGGUNAKAN METODE <i>LOCK GPS</i> DENGAN <i>ANDROID</i> di PT. PLN (Persero) APP MALANG BASECAMP MOJOKERTO, Thn.2015, (Makhfuddin & Prabowo, 2015)	<i>Lock GPS</i>	<ul style="list-style-type: none"> • Memudahkan karyawan melakukan absensi tanpa harus datang langsung ke kantor • Karyawan yang ingin absen harus benar-benar berada di area unit kerja. 	<ul style="list-style-type: none"> • Aplikasi memiliki tampilan yang masih sederhana yang menggunakan web. • Keamanannya kurang terjamin • Aplikasi tersebut hanya menampilkan koordinatnya saja • Fitur-fiturnya masih sangat terbatas dan minimalis
2	SISTEM MONITORING KARYAWAN DENGAN METODE <i>LBS (LOCATION BASED</i>	<i>LBS (LOCATION BASED SERVICE)</i>	<ul style="list-style-type: none"> • Memiliki tampilan yang cukup mudah dimengerti dan menarik oleh <i>staff</i>. • Aplikasi 	<ul style="list-style-type: none"> • Aplikasi tersebut tidak ada waktu dan tanggal pada hari apa dia melakukan izin keluar, dan <i>alpha</i> • Aplikasi

	<i>SERVICE</i>) BERBASIS <i>ANDROID</i> ,Thn.2 019,(Hayati, 2019)		tersebutmemb erikan data berupa jabatan pegawai yang berada di luar kantor berdasarkan status izinnya	tersebuthanya memunculkan waktu lama izin keluar saja yang di tentukan perusahaan
3	APLIKASI ABSENSI TENAGA <i>MARKETING</i> MENGGUNAK AN METODE <i>LOCATIONBASE</i> <i>DSERVICE</i> (<i>LBS</i>) BERBASIS <i>ANDROID</i> ,Thn.2 014, (Hutagaol et al., 2014)	<i>LOCAT</i> <i>IONBA</i> <i>SEDSE</i> <i>RVICE</i> (<i>LBS</i>)	<i>Monitoring</i> pegaw ai yang biasa memantau pegawai marketing yang ditugaskan di luar area kantor, sangat baik dan jelas dengan aplikasinya	<ul style="list-style-type: none"> • laporan point of sale dan komunikasi masih dalam bentuk chat kepada pimpinan perusahaan. • Absensi di <i>android</i> hanya menampilkan jam berapa karyawan masuk ke kantor
4	SISTEM <i>MONITORING</i> MOBIL RENTAL MENGGUNAK AN <i>GPSTRACKER</i> ,T hn.2018,	<i>GPSTR</i> <i>ACKER</i>	<ul style="list-style-type: none"> • Sistem ini sudah menggunakan komponen Arduino uno sebagai papan sirkuit dan juga modul 	<ul style="list-style-type: none"> • Transfer data dari alat arduino ke <i>server</i> masih menggunakan <i>internet</i> atau GPRS

	(Mahendra et al., 2018)		<p><i>GPS</i> untuk menangkap koordinat dari satelit</p> <ul style="list-style-type: none"> • Sistem juga sudah menampilkan <i>marker</i> lokasi keberadaan mobil pada map <i>monitoring</i> sehingga <i>user</i> mengetahui keberadaan mobil 	
5	IMPLEMENTASI DAN MONITORING LAYANAN CLOUD STORAGE SEAFILE DENGAN NAGIOS DI FAKULTAS ILMU TERAPAN, Thn.2017, (Reinaldi et al., 2017)	LAYANAN CLOUD STORAGE DENGAN SEAFILE	<ul style="list-style-type: none"> • Kemanana <i>Cloudstorage</i> Seafile sudah dilengkapi enkripsi <i>password</i> pada setiap masing-masing <i>library</i> meskipun <i>library</i> tersebut sudah di-share • <i>cloudstorage</i> Seafile file-file yang di-<i>upload</i> jadi lebih terstruktur 	<ul style="list-style-type: none"> • Hanya berbasis <i>Website</i> • Fitur-fiturnya masih sangat terbatas atau belum lengkap

2.13 Pembuatan Sistem Yang Dirancang

Pada perancangan sistem ini menggunakan metode *Lock GPS* dan penyimpanan dengan *cloud* dapat diterapkan pada aplikasi *monitoring* aktivitas karyawan lapangan untuk menjadi solusi dalam meminimalisir ke tidak jujur kegiatan yang ada pada setiap pekerjaan karyawan di lapangan. Pada aplikasi karyawan akan dilengkapi dengan menu absen, aktivitas dan panen, pada menu absen dilengkapi dengan *gps* untuk mengetahui titik lokasi karyawan, selanjutnya pada menu aktivitas dilengkapi dengan upload foto aktivitas karyawan dan keterangan kegiatan aktivitas karyawan, pada menu hasil panen dilengkapi dengan upload foto timbangan agar tidak adanya kecurangan dalam penimbangan setelah itu menulis keterangan wilayah panen lalu mengisi kuantitas (kg) panen. Berikutnya pada aplikasi mandor dilengkapi menu master absen karyawan, master aktivitas, master panen, master karyawan dan laporan. Pada menu master absen mandor bisa mengetahui lokasi/titik koordinat karyawan melakukan absen dan dilengkapi dengan tanggal dan waktu yang otomatis, selanjutnya pada menu master aktivitas mandor bisa melihat aktivitas/kegiatan karyawan yang terdapat foto, keterangan, waktu dan tanggal secara otomatis, pada menu master panen mandor bisa melihat foto timbangan karyawan, keterangan, kuantitas/berat, tanggal dan waktu secara otomatis. Pada master karyawan mandor bisa menambah karyawan, mengubah data karyawan dan menghapus data karyawan, pada bagian menu laporan mandor bisa mencetak hasil laporan panen bulanan karyawan yang langsung bisa dilihat dalam bentuk pdf.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian



Gambar 3.1 Peta Lokasi Kantor PTP. Nusantara II Perk. Bandar Klippa

Dalam pelaksanaan penulis mengambil tempat penelitian di Desa Batang Kuis, Sumatera Utara. Peneliti melakukan observasi langsung ke kantor PTP. Nusantara II Perk. Bandar Klippa Batang Kuis. Alamat Jl. Batang Kuis, Bakaran Batu, Kec. Batang Kuis, Kabupaten Deli Serdang, Sumatera Utara 20372. Pada tanggal 14 September 2020 waktu observasi pada saat jam istirahat, Selain itu penelitian juga mewawancarai mandor panen bagaimana memonitoring karyawan di kebun dan melihat sistematis dalam pendataan hasil panen tersebut. Penelitian ini juga di butuhkan jadwal dan tahapan penelitian untuk mencapai batas waktu dalam pembuatan sistem. Tahapan penelitian direncanakan mulai pada bulan Desember 2021 – bulan April 2021 yaitu sebagai berikut:

Tabel 3.1 Waktu Penelitian

Tahapan	2021																			
	Desember				Januari				Februari				Maret				April			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengajuan Proposal Skripsi	■	■																		
Pengumpulan Data			■	■	■	■														
Penyelesaian Proposal Skripsi			■	■	■	■	■	■	■											
Seminar Proposal Skripsi									■											
Analisis data									■	■	■	■	■							
Tahap Penyusunan Skripsi									■	■	■	■	■							
Perancangan Sistem													■	■						
Desain <i>Interface</i>														■	■					
Pembuatan <i>Coding</i>															■	■	■	■	■	■
Uji Coba																			■	■

Tahapan penelitian yang dilakukan yaitu sebagai berikut:

1. Pengajuan Proposal Skripsi
Tahap mengajukan penemuan masalah terhadap sebuah studi kasus beserta solusi yang akan diberikan penulis untuk mengatasi masalah tersebut.
2. Pengumpulan Data
Pengajuan proposal skripsi diterima, maka selanjutnya penulis melakukan tahap pengumpulan data yaitu observasi ke tempat objek penelitian dan melakukan wawancara, serta melakukan studi pustaka terkait penelitian. Data yang didapat sangat dibutuhkan untuk kelanjutan penyelesaian penelitian.
3. Penyelesaian Proposal Skripsi
Tahap ini penulis menyelesaikan isi proposal skripsi dan juga melalui proses bimbingan oleh pembimbing I dan pembimbing II yang telah ditugaskan.
4. Seminar Proposal

Tahap seminar proposal skripsi dilakukan untuk menguji kesesuaian pada penelitian yang telah diajukan dengan presentasi proposal skripsi.

5. Analisis Data

Setelah pengumpulan data maka penulis akan melakukan analisis data untuk kesesuaian pengembangan sistem dalam proses pembuatannya.

6. Tahap Penyusunan Skripsi

Proses penyusunan skripsi dilakukan dengan dibina oleh dosen pembimbing yang telah ditentukan prodi.

7. Perancangan Sistem

Pada Pada tahap ini penulis mulai merancang database dan menggambarkan alur kerja dari aplikasi yang akan dibangun menggunakan UML.

8. Desain *Interface*

Penulis melakukan desain *interface* untuk menghasilkan *interface* yang sesuai dengan fungsi yang dibutuhkan pada aplikasi yang akan dibangun.

9. Pembuatan *Coding*

Penulis melakukan pembuatan *coding* sistem

10. Uji Coba

Pengujian sistem dilakukan untuk mengetahui apakah sistem berjalan dengan baik atau tidak.

3.2 Kebutuhan Sistem

3.2.1. Perangkat Keras

Berikut merupakan Spesifikasi perangkat keras yang dibutuhkan untuk membangun aplikasi:

1. *Processor* Intel ® *Core* ™ i7-3610QM
2. NVIDIA ® *GeForce* ® GT 640M
3. RAM 4GB DDR3 *Memory*
4. HDD 500 GB

3.2.2. Perangkat Lunak

Berikut merupakan daftar *software* yang dibutuhkan untuk membangun aplikasi:

1. Sistem Operasi, *Windows 10*
2. *Android Studio 4.0*
3. *Balsamiq Mockups Version 3.5.17*
4. *Visual Paradigm Version 16.0*
5. *Android Versi 10*
6. *Microsoft Office 2010*
7. *Microsoft Visio 2010*

3.3 Cara Kerja

Metode kualitatif digunakan dalam penelitian ini. Apa saja yang termasuk dalam metode kualitatif seperti observasi, wawancara, dan studi pustaka. Hal ini dilakukan untuk mendapatkan bahan-bahan yang akan membantu penelitian. Alasan menggunakan metode kualitatif dalam penelitian ini adalah untuk mencapai tujuan penelitian.

3.3.1 Metode Pengumpulan Data

Dalam penelitian ini, informasi dikumpulkan melalui observasi, wawancara, dan tinjauan literatur yang relevan seperti jurnal dan buku. Berikut penjelasan dari sumber informasi tersebut:

1. Observasi

Observasi adalah upaya memperoleh informasi atau information dengan cara mengamati secara langsung objek yang menjadi bahan dalam penelitian ini. Untuk memperoleh informasi yang lebih jelas dan memperoleh information yang dibutuhkan oleh penulis. Penulis melakukan pengamatan secara langsung dengan mengunjungi kebun sawit PTP.Nusantara II *Perk* Bandar Klippa yang berlokasi di Jl. Besar Batang Kuis. Penulis mengamati kegiatan yang berlangsung pada objek yang menjadi bahan penelitian. Penulis juga mengamati bagaimana aktivitas karyawan yang berada di kebun sawit dan melihat pendataan hasil panen dari PTP.Nusantara II Bandar Klippa ini.

2. Wawancara

Wawancara adalah kegiatan tanya jawab yang ditunjukkan kepada

seseorang yang ahli dalam suatu bidang atau disebut sebagai nara sumber dengan tujuan untuk memperoleh informasi yang berguna untuk mendukung pengembangan penelitian. Dalam hal ini, penulis melakukan metode wawancara untuk mencari informasi sebanyak-banyaknya baik itu mengenai profil perusahaan, operasional perusahaan, keadaan perusahaan saat ini, dan informasi yang lain. Penulis selaku pewawancara melakukan wawancara dengan seorang narasumber yaitu mandor di kebun sawit yang bertugas di kebun PTP.Nusantara 2 Bandar Klippa Batang Kuis.

3. Studi Pustaka

Studi Pustaka adalah metode pengumpulan data dengan membaca buku dan jurnal sesuai dengan data yang dibutuhkan. Dalam hal ini, penulis mencari data dan informasi dengan mempelajari beberapa bahan referensi jurnal dan internet yang terkait atau relevan dengan hal yang menjadi objek penelitian, yaitu seperti permasalahan yang terjadi pada perusahaan PTP.Nusantara dan juga metode penerapan absensi menggunakan *Lock GPS*.

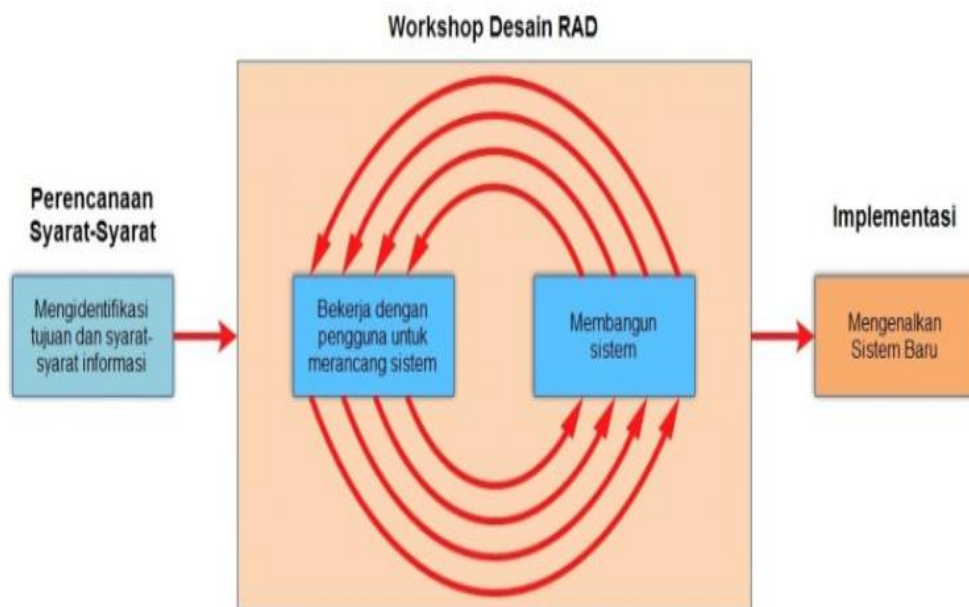
3.3.2 Jenis Data

Information yang diperoleh dan dikumpulkan oleh penulis dibagi menjadi dua jenis information, yaitu:

- a. Data Primer: yaitu data yang dikumpulkan melalui perorang ataupun melalui instansi pada lokasi penelitian dengan cara melakukan wawancara, observasi, atau observasi langsung. Dalam penelitian ini, penulis melakukan observasi dan wawancara kepada bapak menejer utama PTP. Nusantara II *Perk* Bandar Klippa. Adapun data yang didapat berupa kekurangan-kekurangan sistem aplikasi absensi karyawan yang pernah dibangun sebelumnya sepertinya tidak berjalan dengan baik. Karyawan sering melakukan absensi di rumah atau tidak dilokasi mereka bekerja. Pada sistem pendataan dan laporan hasil panen sering terjadi ke tidak jujur saat penimbangan dan juga buku laporan hasil panen sering terjadi kehilangan.
- b. Data Sekunder: yaitu data yang dikumpulkan melalui penelitian terdahulu ataupun jurnal yang terkait dengan judul penelitian yang menjadi landasan

ataupun acuan dalam membuat beberapa pertanyaan saat melakukan observasi. Adapun penelitian terdahulu yang dimaksud seperti yang di buat oleh Makhfuddin, dkk, dengan judul “ Aplikasi Absensi Menggunakan Metode *Lock GPS Dengan Android* di PT. PLN (Persero)”, dan juga jurnal-jurnal ataupun penelitian terdahulu lainnya.

3.3.3 Metode Pengembangan Sistem



Gambar 3.2 Siklus Pengembangan Sistem RAD (Kendall, 2010)

Dalam penelitian ini, metode pengembangan sistem yang digunakan adalah RAD (*Rapid Application Development*), sebuah model proses perangkat lunak yang menekankan pada siklus hidup pengembangan yang singkat. RAD adalah adaptasi cepat dari model air terjun yang menggunakan pendekatan konstruksi komponen. Untuk mempercepat pengembangan sistem/aplikasi, RAD menggabungkan berbagai teknik terstruktur dengan teknik prototyping dan teknik pengembangan aplikasi bersama. Konsep RAD mengacu pada model yang menekankan pengembangan aplikasi dalam waktu yang relatif singkat. (Puteri & Effendi, 2018)

Tiga tahap RAD (Rapid Application Development) yaitu:

1. *Requirements Planning* (Perancangan Persyaratan)

Tahap ini merupakan tahap dimana pengguna dan analisis bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem, dan pada tahap ini lebih berorientasi pada pemecahan masalah bisnis. Pada tahap ini penulis melakukan hal-hal sebagai berikut:

- a. Pengumpulan data informasi yang akan digunakan untuk tahap berikutnya, data tersebut ialah berupa lokasi-lokasi panen kelapa sawit di PTP. Nusantara II yang berada di kota Batang Kuis. Pengumpulan data ini dilakukan dengan observasi ketempat dan wawancara. Adapun data yang didapatkan penulis disini adalah data karyawan dan data hasil panen karyawan.
- b. Identifikasi Sistem yang dilakukan pada tahap ini dilakukan untuk mengembangkan sistem yang sudah ada. Identifikasi sistem dalam penelitian ini adalah identifikasi sistem saat ini/lama dan identifikasi sistem yang diusulkan. Sedangkan untuk sistem lama/saat ini yang masih sederhana, dimana pegawai melakukan absensi hanya memasukkan tanggal kehadiran dan informasi absensi. Hal ini kadang memiliki beberapa kendala, seperti karyawan bisa saja melakukan absensi di mana saja tidak harus di lokasi kerja. Kemudian pada aktivitas lapangan karyawan sering kedatangan atasan bersantai pada saat jam kerja. Kemudian pada pendataan hasil panen karyawan masih sangat manual yaitu menggunakan buku di mana karyawan memberi tahu atasan berapa hasil panen mereka. Hal ini juga memiliki kendala yaitu ke tidak jujuran hasil penimbangan yang di buat oleh karyawan. Kemudian sistem usulan yang ingin diangkat adalah untuk memangkas birokrasi sistem tersebut dengan membuat sistem absensi yang menggunakan *GPS* yang mengetahui titik koordinat karyawan berada yang akurat, pada bagian aktivitas lapangan dengan membuat sistem yang bisa mengupload aktivitas karyawan yang berbentuk foto, pada pendataan hasil panen akan di buat

bukti seperti upload foto penimbangan hasil panen agar tidak adanya ke tidak jujur dalam penimbangan.

2. *Design Workshop*

Tahap ini merupakan tahap perancangan dan penyempurnaan dengan menggunakan kelompok pendukung keputusan sistem untuk membantu pengguna menyepakati sistem yang dibangun. Biasanya dilakukan dengan menunjukkan tampilan visual desain menjadi bagian sebagai berikut:

a. Desain Proses

Pada tahap perancangan expositions ini, penulis mengidentifikasi aktor-aktor yang terlibat dalam sistem kegiatan pegawai lapangan dimana information diperoleh dari tahap sebelumnya. Menggunakan *Unified Model Language* dengan diagram model yang digunakan yaitu:

1. Membuat *Use Case*

Dalam graph ini penulis menggambarkan hubungan antara sistem yang diusulkan dan aktor tentang apa yang dapat dilakukan aktor sebagai pengguna dalam aliran sistem. Dalam *Use Case* ini terdapat 2 aktor yang menjadi *end user* yaitu mandor panen sawit dan karyawan panen sawit. Mandro dapat melakukan registrasi kesistem, melakukan *login*, pengecekan absensi karyawan, melihat aktivitas dan melihat hasil panen karyawan. Sedangkan pada karyawan, dapat melakukan registrasi ke sistem, melakukan *login*, melakukan absensi, melakukan *upload* aktivitas karyawan, malakukan *upload* hasil panen karyawan.

2. Membuat *Classdiagram*

Setalah mengetahui aliran sistem selanjutnya dilakukan pembuatan diagram yang untuk mengetahui *class-class* yang nantinya akan dipakai dalam implementasi kode secara OOP (*Object Oriented Programming*). Adapun *classdiagram* pada sistem ini terdapat 6 *class* yaitu *class admin*, *class karyawan*, *class absensi*, *class aktivitas harian*, *class wilayah* dan hasil panen.

3. Membuat *Activity Diagram*

Dalam pembuatan diagram ini berisikan aliran sistem usulan atau proses dalam sistem. Terdapat 3 proses (3 *Activity Diagram*) yang terjadi dalam sistem ini yaitu pada sisi mandor dan pada posisi karyawan. Aktivitas tersebut adalah, proses absensi, proses aktifitas, proses hasil panen.

4. Membuat *Sequence diagram*

Pada tahap pembuatan ini, penulis menggambarkan interaksi objek yang tersusun dalam urutan waktu dan hubungan timbal balik dengan sistem. Ada 2 succession outline yaitu grouping chart untuk aktor mandor/director dan aktor pegawai.

b. Desain *Database*

1. Menentukan potensial objek

Pada tahap ini penulis membuat daftar potensial objek dengan cara menemukan objek yang penting berlandaskan *use case*. Dari diagram tersebut terdapat 6 *object* yaitu, *admin*, karyawan, absensi, aktifitas harian, wilayah dan hasil panen.

2. Membuat rancangan *Database*

Pada tahap ini penulis membuat rancangan *database* dengan membuat dari nama *database* yang di beri nama *db_ptpn admin*, *db_ptpn karyawan*, *db_ptpn absensi*, *db_ptpn wilayah*, *db_ptpn aktifitas_harian*, *db_ptpn hasil_panen*.

c. Desain *Interface*

1. Rancangan Struktur Menu

Pada tahap ini penulis mulai merancang struktur menu untuk dijadikan landasan pembuatan *Interface*. Adapun struktur menu pada sisi mandor adalah, register, *login*, menu laporan absensi karyawan, menu laporan aktifitas karyawan, menu laporan panen karyawan dan menu hasil laporan hasil bulanan karyawan. Sedangkan pada sisi karyawan yaitu register, *login*, menu absensi, menu *upload* aktifitas karyawan, menu hasil panen karyawan.

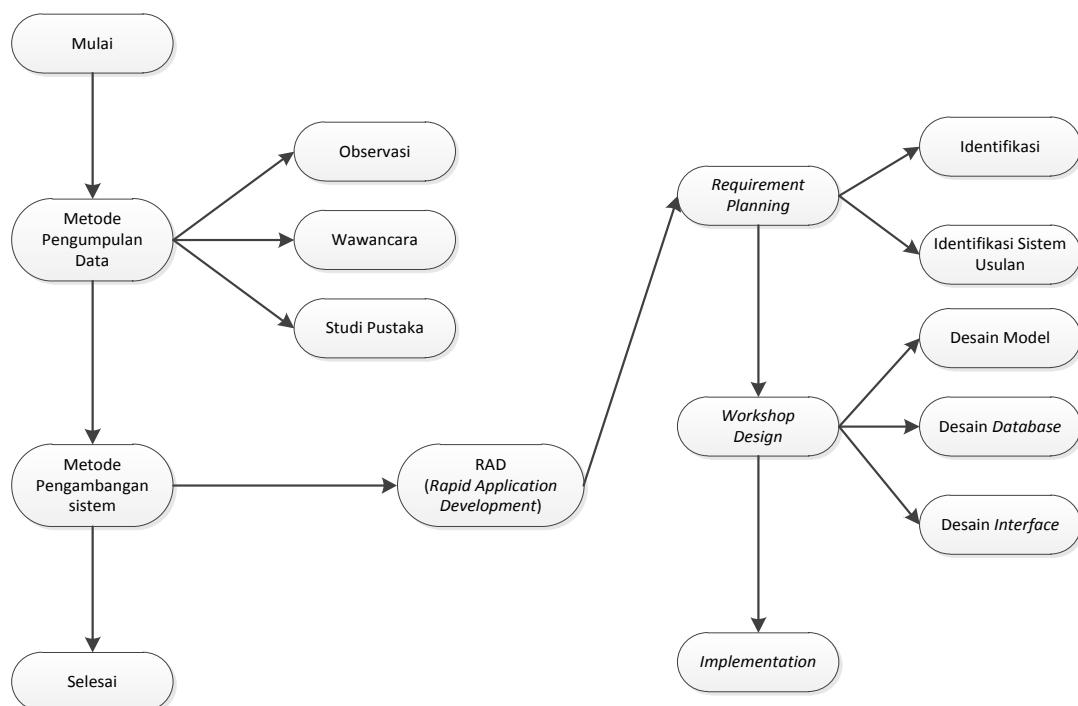
2. Rancangan *Interface*

Pada tahap ini penulis merancang antarmuka sistem yang akan dibuat untuk menggambarkan tampilan sistem. Dalam hal tersebut penulis menggunakan *Microsoft Visio 2010* untuk melakukan desain antarmuka sistem yang dapat dilihat pada bab berikutnya.

3. Implementasi

1. Saat sistem baru dibangun, sistem baru atau sebagian diuji dan diperkenalkan ke organisasi.
2. Sistem yang lama tidak perlu dijalankan secara beriringan.
3. Melakukan pengkodean sistem berdasarkan tahapan *Requirements Planning* dan *Workshop Desain*.

3.4 Kerangka Berfikir



Gambar 3.3 Kerangka Berfikir

Pada penyelesaian dalam penelitian ini memiliki beberapa tahapan proses, langkah pertama dimulai dari metode pengumpulan data. Metode ini merupakan bagaimana penulis memperoleh data dalam penelitian ini yang dimulai dari tahap observasi pada tempat penelitian, dalam hal ini peneliti mendatangi Kantor PTP. Nusantara II *Perk.* Bandar Klippa. Dalam menunjang hasil yang optimal peneliti juga melakukan wawancara kepada menejer, mandor di kantor PTP. Nusantara II dan melakukan studi pustaka terkait penelitian yang di angkat dengan cara membaca dan mengumpulkan refrensi dari berbagai karya ilmiah dan buku-buku terkait penelitian.

Setelah itu dilakukan tahapan metode pengembangan sistem, dimana dalam penelitian ini penulis menggunakan metode pengembangan sistem RAD, sebuah metode pengembangan sistem yang menekankan waktu pembuatan yang efisien dalam pengembangan sistem. RAD sendiri memiliki tiga tahapan yang harus dilakukan, yaitu *Requirements Planning*, *Workshop Design*, dan *Implementation*.

Dalam *Requirements Planning* yaitu suatu kegiatan atau tahapan dimana penulis mengidentifikasi kebutuhan pada sistem dilakukan dengan cara melakukan analisis data apa saja yang di perlukan, hal ini dilakukan setelah penulis melakukan pengumpulan data dan memiliki ide usulan pembuatan sistem dengan melakukan implementasi *Lock GPS* dan penyimpanan *cloud* yang berbentuk digital pada sistem yang akan di kembangkan. Selanjutnya tahap *Workshop Design* dimana pada tahap ini penulis merancang aliran sistem atau model sistem, menampilkan data yang diperoleh dalam bentuk perancangan *database*, melakukan perancangan struktur menu utama, dan antarmuka sistem. Kemudian tahap akhir yaitu *Implemetation*, disinilah implementasi *Lock GPS* diterpkan, penyimpanan *cloud* berbasis digital diterapkan dan *coding* sistem dilakukan, setelah ketiga hal ini dilakukan barulah *unit testing* dilakukan kepada pengguna.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan Sistem

Kebutuhan sistem merupakan sekumpulan beberapa hal yang diperlukan dalam membangun suatu sistem dalam suatu penelitian. Kebutuhan sistem yang dibutuhkan untuk menyelesaikan penelitian ini terdiri dari profil perusahaan tempat penelitian dilakukan, analisis sistem yang sedang berjalan dan analisis sistem yang diusulkan. Sebelumnya pada penelitian ini penulis melakukan penelitian pada PTP.Nusantara II Perk Bandar Klipaa dan mandor lapangan guna mendapatkan information yang dibutuhkan dalam pembuatan sistem atau aplikasi. Berikut ini adalah penjelasan dari beberapa hal tersebut.

4.1.1 Profil Kantor PT.PerkebunanNusantara II

Perkebunan Nusantara II merupakan hasil penggabungan PTP II dan PTP IX yang didirikan berdasarkan akte Notaris Harun Kamil, SH ditetapkan di Jakarta pada tanggal 11 Maret 1996 dengan Akte No. 35 dan diperbarui dengan akte Notaris N.M. Dipo Nusantara Pua Upa, SH No. 33 tanggal 13 Agustus 2008 dan beberapa kali mengalami perubahan dengan diperbaharui akte Notaris Nanda Fauz Iwan, SH, M.Kn No. 19 tanggal 17 Maret 2016. Total areal yang dikelola PTPN II 109.409,61 Ha dan total areal ditanami 52.118,06 Ha. Produk perkebunan yang dihasilkan yaitu Minyak Sawit (CPO), Inti Sawit (Palm Kernel), RSS, Gula, Tetes, dan Tembakau.

Wilayah kerja PTPN II terletak di 6 (enam) Kabupaten/Kota di Propinsi Sumatera Utara, yaitu Kabupaten Langkat, Kabupaten Deli Serdang, Kabupaten Serdang Bedagai, Kabupaten Padang Lawas, Kotamadya Medan, Kotamadya Binjai dan Provinsi Papua terletak 1 (satu) Kabupaten yaitu Keerom. Pada saat ini PTPN II memiliki karyawan sebanyak 6252, terdiri dari 440 karyawan pimpinan dan 5776 karyawan pelaksana. (<https://ptpn2.com/>, diakses pada pukul 12:32 wib pada tanggal 18 Maret 2021).

4.1.2 Visi dan Misi PT. Perkebunan Nusantara II

Visi dan misi suatu perusahaan atau instansi sangat penting karena menjadi landasan untuk mencapai tujuan perusahaan. Visi dan misi PT. Perkebunan Nusantara II adalah sebagai berikut: Visi perusahaan perkebunan adalah menjadi perusahaan multibisnis yang berdaya saing tinggi. Sementara itu, misi perusahaan adalah memaksimalkan semua sumber daya dan peluang bisnis yang ada agar dapat memberikan kontribusi yang sebesar-besarnya dengan tetap menjaga kelangsungan hidup jangka panjang dan nilai tambah..

4.1.3 Tujuan Perusahaan

Tujuan perusahaan adalah sebagai berikut:

1. Meningkatkan kinerja operasional organisasi dan manajemen, serta memanfaatkan peluang bisnis secara ideal, agar menjadi perusahaan perkebunan berkelanjutan yang berdaya saing, sejahtera, dan menguntungkan, dengan kemampuan memberikan kontribusi bagi pembangunan daerah dan nasional serta kesejahteraan karyawan.
2. Melaksanakan pembinaan dan pengembangan agribisnis di bidang perkebunan sesuai dengan prinsip ekonomi perusahaan yang sehat, kuat, dan berkembang.
3. Meningkatkan posisi portofolio bisnis dengan meningkatkan seluruh aspek sumber daya PTP Nusantara II dari dalam.
4. Menjaga dan meningkatkan kontribusi devisa sektor perkebunan dengan meningkatkan produksi sekaligus mendukung upaya peningkatan ekspor nonmigas serta melestarikan sumber daya alam, lingkungan, air, dan tanah.

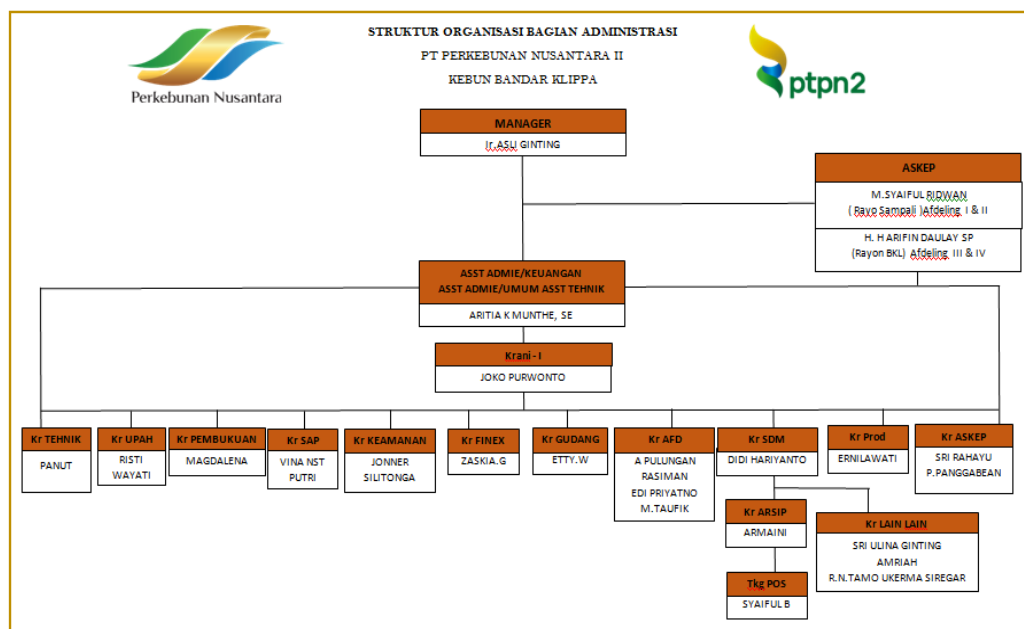
4.1.4 Sasaran Perusahaan

Mempertahankan dan memperluas kontribusi di bidang perkebunan dengan meningkatkan produksi dan mendukung upaya peningkatan ekspor nonmigas. Meningkatkan kesempatan kerja dalam rangka meningkatkan kesejahteraan masyarakat secara umum. Memelihara sumber daya alam dan lingkungan, serta kesuburan air dan tanah. Strategi Perusahaan Dalam rangka meningkatkan kinerja perusahaan dan mengantisipasi masa globalisasi tahun 2000 dan ketidakpastian

ekonomi di tahun-tahun mendatang, perusahaan telah menetapkan beberapa strategi, antara lain sebagai berikut:

- Meningkatkan produksi dan produktivitas dengan mengembangkan dua budidaya unggulan yaitu kelapa sawit dan tebu.
- Meningkatkan kualitas produksi dengan potensi pasar, serta memantau harga produksi dengan keunggulan kompetitif yang dapat menghasilkan keuntungan yang lebih tinggi.
- Meningkatkan kesadaran kesejahteraan karyawan dalam rangka meningkatkan semangat kerja dan produktivitas dalam bekerja.
- Meningkatkan pengetahuan dan keterampilan teknis sumber daya manusia melalui pelatihan dan pendidikan.

4.1.5 Struktur Organisasi PTP. Nusantara II Perk Bandar Klippa



Gambar 4.1 Struktur Organisasi PT. Perkebunan Nusantara II Perk Bandar Klippa

Gambar diatas adalah struktur organisasi PT. Perkebunan Nusantara II Perk Bandar Klippa. Struktur organisasi adalah susunan dan hubungan yang terjalin antara setiap bagian dan kedudukan dalam suatu organisasi atau perusahaan dalam rangka melaksanakan kegiatan operasional untuk mencapai

suatu tujuan. Struktur organisasi harus secara jelas menggambarkan pemisahan kegiatan dan bagaimana hubungan antara kegiatan dan fungsi dibatasi. Struktur organisasi yang baik juga harus menjelaskan siapa melapor kepada siapa.

4.1.6 *Job Description* dari struktur organisasi

Struktur organisasi adalah susunan dan hubungan yang terjalin antara setiap bagian dan kedudukan dalam suatu organisasi atau perusahaan dalam rangka melaksanakan kegiatan operasional guna mencapai tujuan. Struktur organisasi secara jelas menggambarkan pemisahan kegiatan dan bagaimana hubungan antara kegiatan dan fungsi dibatasi; dalam struktur organisasi yang baik harus menjelaskan siapa melapor kepada siapa.

Berikut ini bagan perkebunan nusantara II (PERSERO) cabang kebun Bandar Klippa:

Tabel 4.1 *Job Description*

No	Nama	Jabatan	Uraian Tugas
1	Ir. Asli <i>Ginting</i>	Manager	<ul style="list-style-type: none"> a. Melaksanakan tugas dan kebijakan perusahaan yang telah digariskan b. Melaksanakan perencanaan, pengorganisasian, pengendalian, dan pengawasan unit usahatani dalam rangka mendukung kelancaran tugas pokok secara efektif guna mencapai produktivitas. c. Memberikan informasi yang akurat dan terkini untuk membantu manajemen dalam mengambil keputusan. d. Membantu direksi dalam mencapai tujuan perusahaan e. Mematuhi peraturan/peraturan perusahaan (sistem operasional dan prosedur standar) f. Menyampaikan rencana kerja dan anggaran perkebunan (RKAP). g. Koordinasi tugas bawahan yaitu Dinas Perkebunan dan pembantu (tebu, kelapa sawit,

			<p>admin/keuangan, rancang, BA.PM).</p> <p>h. Mengawasi operasional harian taman</p> <p>i. Menyediakan bahan dan alat pabrik sesuai dengan kapasitas dan persyaratan kualitas yang ideal</p> <p>j. Menjaga kebutuhan aset perusahaan berupa gangguan keamanan yang mungkin terjadi dari luar maupun dari dalam perusahaan. Melaksanakan tugas khusus yang didelegasikan oleh direksi.</p>
2	-M. Syaiful Ridwan -H.H Arifin Daulay SP	Askep	<p>a. Membantu Manajer Perkebunan dalam menyusun rencana kerja dan biaya taman (pabrik).</p> <p>b. Buat jaringan afiliasi.</p> <p>c. Mengawasi pelaksanaan rencana kerja dan rencana anggaran/biaya.</p> <p>d. Mengkoordinasikan perekrutan dan penempatan karyawan di afdeling.</p> <p>e. Mengawasi pendistribusian kebutuhan material di afdeling.</p> <p>f. Melakukan pemeriksaan administrasi dan fisik pada pekerjaan lapangan.</p> <p>g. Mengawasi pelaksanaan tugas Asisten di bidang afdeling.</p>
3	Arita K Munthe, SE	Asst Admin/Keuangan, Asst Admin/Umum Asst Tehnik	<p>Asst Umum</p> <p>a. Mengawasi dan memverifikasi perekrutan pekerja sesuai dengan standar yang ditetapkan.</p> <p>b. Menjaga hubungan positif dengan masyarakat sekitar lokasi perusahaan.</p> <p>c. Mengkoordinasikan kegiatan yang bertujuan untuk meningkatkan kesejahteraan karyawan.</p> <p>d. Menyediakan manajer pertanian dengan informasi tentang produktivitas kerja.</p>

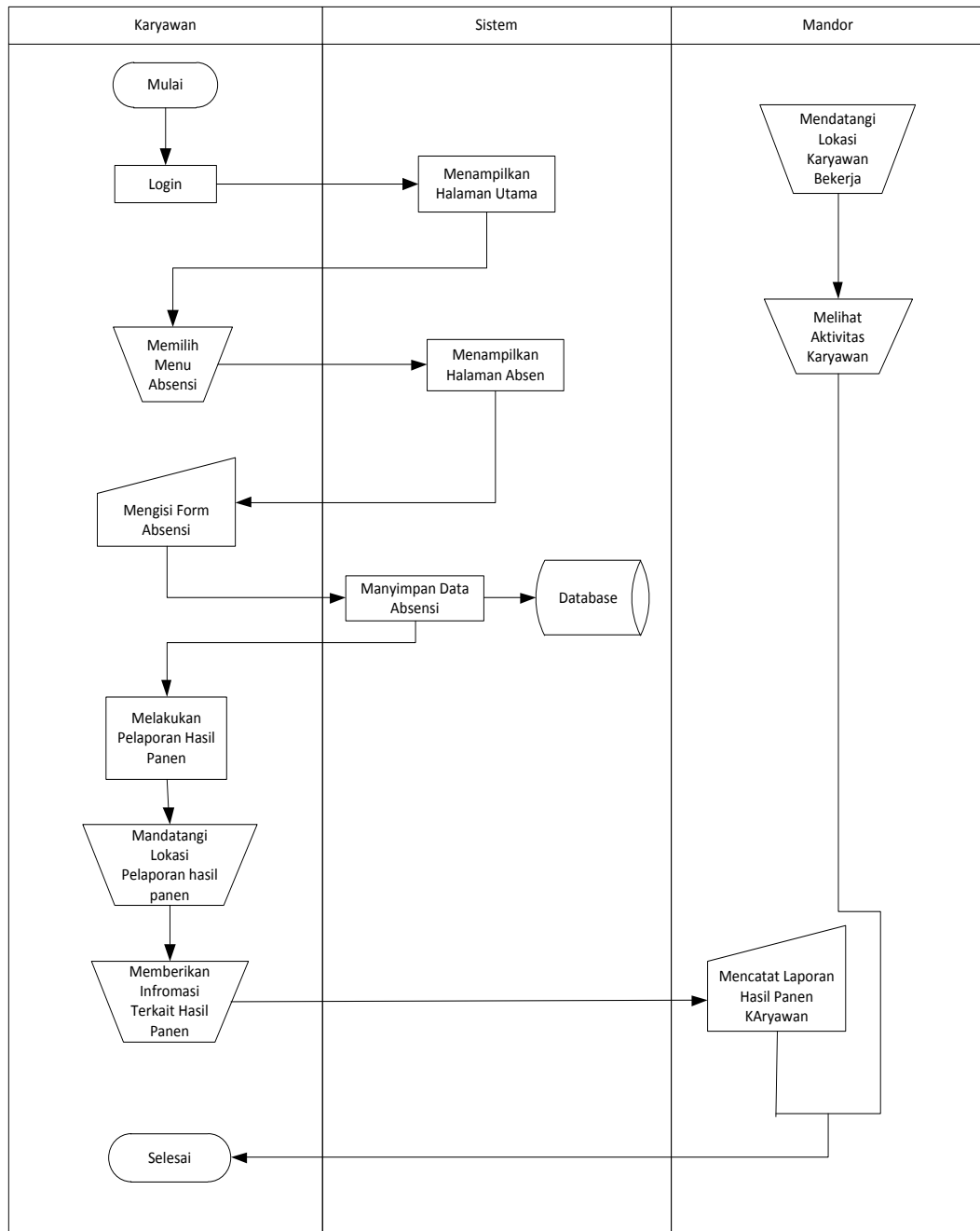
			<p><i>Asst Admin</i></p> <ul style="list-style-type: none"> a. Mengkoordinasikan dan merencanakan kegiatan administrasi. b. Melaksanakan dan mengawasi seluruh kegiatan operasional di bidang keuangan dan administrasi, tenaga kerja, fasilitas, dan perlengkapan kerja. c. Mengkoordinasikan penyusunan rencana kerja anggaran perusahaan perkebunan/afdeling berdasarkan norma kerja yang telah ditetapkan. d. Menyusun laporan neraca saldo dan laporan manajemen secara bulanan sesuai dengan jadwal yang telah ditentukan. e. Mengevaluasi biaya/harga pokok barang secara bulanan dan mengawasi pembukuan biaya sesuai dengan pembukuan dalam sistem administrasi. f. Bekerja sama dengan pejabat publik untuk mendorong dan mengarahkan karyawan di lingkungan kantor perkebunan untuk meningkatkan kesejahteraan, agama, olahraga, lingkungan, gotong royong, koperasi, dan keselamatan karyawan.
4	Joko Purwanto	Krani - I	<ul style="list-style-type: none"> a. Kembangkan atau buat rencana anggaran bulanan b. Menyusun atau menyusun rencana kerja harian, serta daftar laporan kerja harian dan gaji karyawan. c. Lihat buku mandor dan pindahkan hari kerja karyawan ke buku asisten. d. Membuat laporan mingguan dan bulanan. e. Bertanggung jawab kepada asisten.

5	Panut	Kr Thenik	<p>a. Bertanggung jawab atas semua tugas primer dan sekunder yang berkaitan dengan pengelolaan Bengkel Teknik / Bengkel Reparasi, serta kebersihan lingkungan.</p> <p>b. Mengawasi pelaksanaan tugas pekerjaan Engineering Pabrik.</p> <p>c. Mengawasi penyelesaian pekerjaan dalam jangka waktu yang ditentukan.</p> <p>d. Memberikan arahan dan dorongan untuk menumbuhkan lingkungan kerja yang positif.</p> <p>e. Mengelola pencapaian pelayanan kerja karyawan Bengkel Teknik Pabrik seefektif mungkin.</p>
6	Risti Wayati	Kr Upah	-
7	Magdalena	Kr Pembukuan	<i>Input information</i> , dan input semua biaya operasional terkait PKS
8	-Vina Nst -Putri	Kr SAP	-
9	Jonner Silitonga	Kr Keamanan	<p>a. Melapor kepada Manajer Perkebunan dan bertanggung jawab atas keamanan dan ketertiban di Taman Sei Kencana.</p> <p>b. Kerjasama dengan aparat keamanan setempat seperti polisi, pemerintah desa, dan Koramil</p> <p>c. Melindungi aset perusahaan dari segala jenis gangguan, baik eksternal maupun internal.</p> <p>d. Mewakili perusahaan dalam interaksi dengan polisi atau aparat keamanan lainnya</p> <p>e. Mengawasi keamanan inventaris dan informasi perusahaan</p>
10	Zaskia .G	Kr Finex	-
11	Etty.W	Kr Gudang	<p>a. Bertanggung jawab atas penyimpanan gudang melalui laporan harian, bulanan, dan tahunan.</p> <p>b. Memberikan masukan kepada atasan dalam hal analisis biaya divisi.</p>

12	-A pulungan -Rasiman -Edi Priyanto -M.Taufik	Kr AFD	<p>a. Mengatur dan mengawasi pelaksanaan pekerjaan/kegiatan sesuai dengan peraturan perusahaan.</p> <p>b. Memberikan arahan, bimbingan, dan pengawasan teknis terhadap seluruh kegiatan yang dilaksanakan di Afdeling.</p> <p>c. Melakukan observasi dan inspeksi lapangan secara terus menerus.</p> <p>d. Menyelenggarakan administrasi dan pembukuan untuk semua kegiatan Afdeling.</p> <p>e. Melakukan perawatan secara efektif dan efisien sesuai dengan standar yang ditentukan.</p>
13	Didi Harianto	Kr SDM	<p>a. Menyusun rencana implementasi untuk bidang yang dikelolanya.</p> <p>b. Melakukan pengendalian dan pengawasan terhadap kawasan yang dikelolanya.</p> <p>c. Buat rencana jangka panjang dan jangka pendek untuk pendidikan keselamatan dan kesejahteraan tempat kerja dan layanan keselamatan.</p> <p>d. Membuat kebijakan untuk inisiatif pengembangan sumber daya manusia (pendidikan dan pelatihan).</p>
14	Ernilawati	Kr Prod	-
15	-Sri Rahayu -P.Panggabean	Kr Askep	<p>a. Membantu kepala Askep dalam pencatatan informasi pegawai lapangan.</p> <p>b. Mengkoordinir personel lapangan</p>
16	Armaini	Kr Arsip	Bertugas menyimpan <i>information</i> pegawai dan memanen <i>information</i> .
17	Syaiful. B	Tkg POS	Mengantar surat dan mengirimkan surat ke divisi yang bersangkutan
18	-Sri Ulina <i>Ginting</i> -Amriah -R.N.Tamo Ukerma Siregar	Kr Lain-lain	-

4.1.7 Analisis Sistem Berjalan

Pada analisi sistem yang berjalan ini menjelaskan bagaimana alur dari sistem berjalan Absensi, Aktivitas karyawan dan hasil panen karyawan. Monitoring aktivitas karyawan lapangan yang dilakukan secara kesepakatan, adapun alurnya dijelaskan pada diagram berikut :

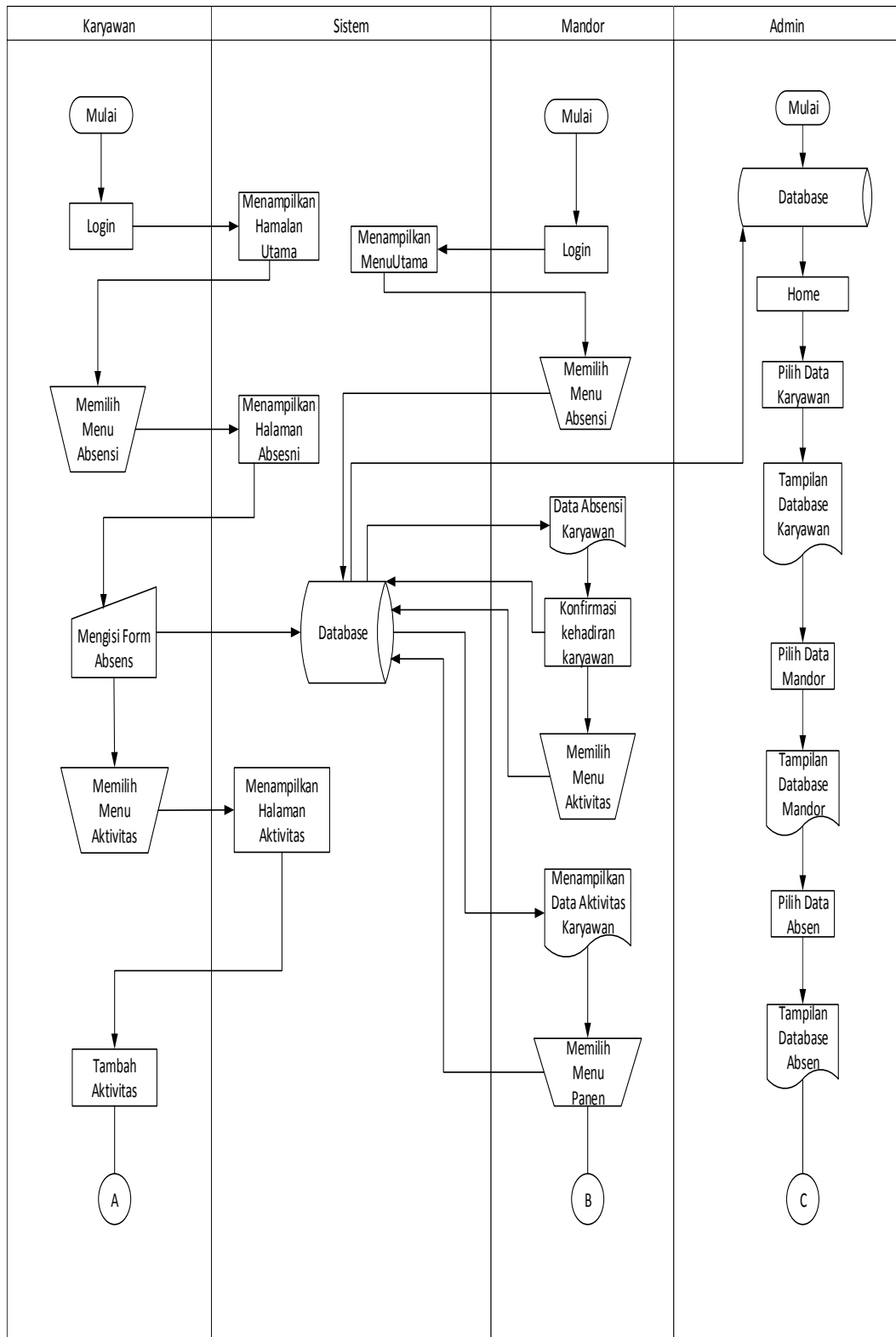


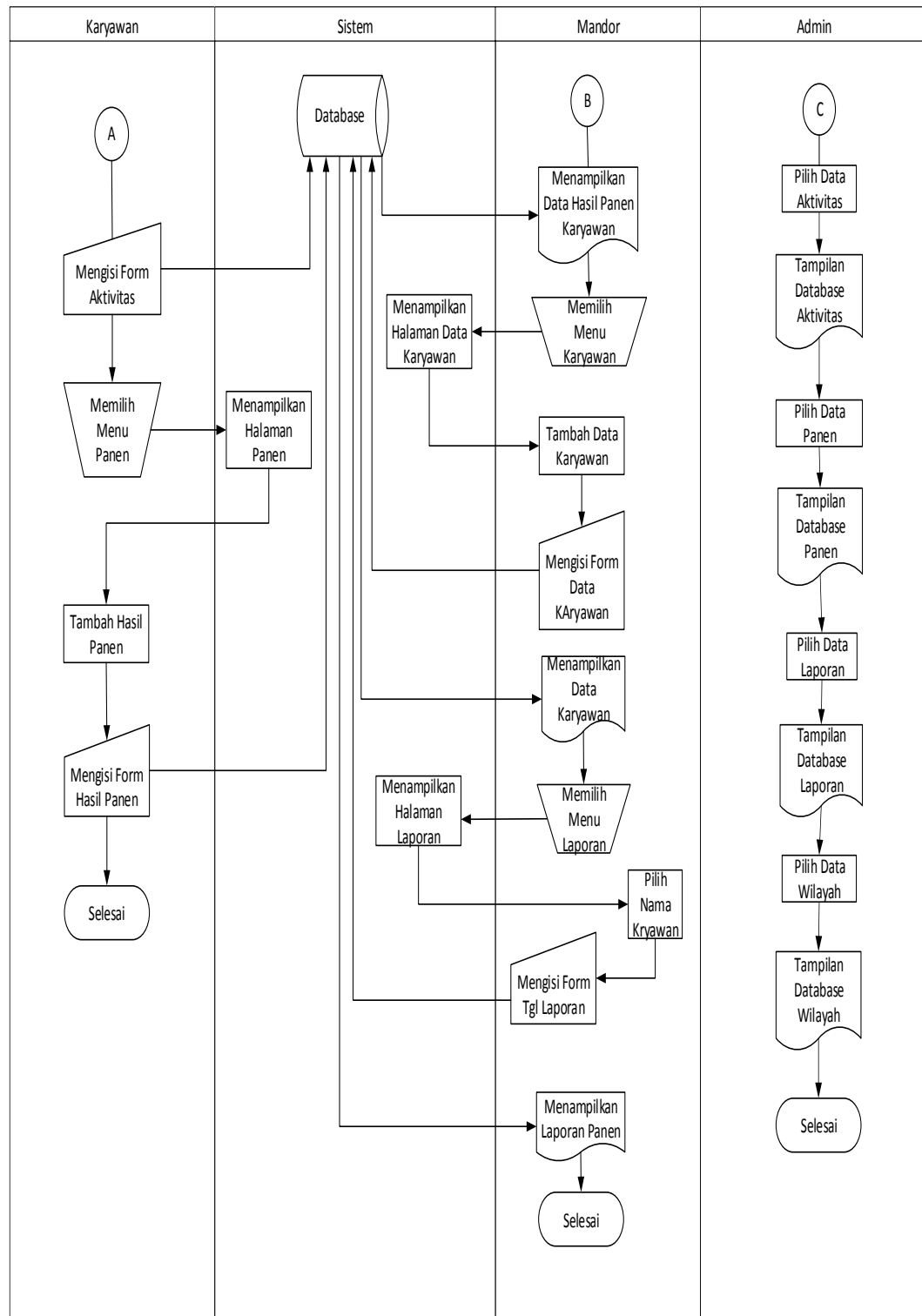
Gambar 4.2 Flowmap Sistem Berjalan

Flowmap sistem berjalan diatas di mulai dengan karyawan melakukan absensi yang sudah menggunakan aplikasi dari perusahaan PTP. Nusantara II PerKBandar Klippa, yang di mulai dengan melakukan *login* yaitu dengan mengisi *username* dan *password* setelah itu sistem akan menampilkan halaman absensi utama, kemudian pegawai memilih menu absensi. Setelah itu sistem akan menampilkan *form* absen, selanjutnya karyawan mengisi *form* absensi tersebut yang berisi jenis ijin (hadir, sakit dan cuti), lalu mengisi keterangan dan tanggal setelah itu menyimpan data tersebut *kedatabase*. Selanjutnya karyawanakan melakukan pelaporan hasil panen kepada mandor yang tidak menggunakan sistem yaitu dengan mendatangi secara langsung tempat pelaporan hasil panen, setelah itu karyawan memberikan informasi panen terhadap mandor lalu mandor akan mencatat hasil panen karyawan dengan menggunakan buku atau catatan. Selanjutnya untuk pengecekan aktivitas karyawan lapangan masih sangat manual yaitu mandor mendatangi secara langsung atau berkeliling setiap karyawan untuk mengawasi aktivitas karyawan bekerja seperti pemberian pupuk, pemeliharaan pohon sawit, penurunan sawit dan pengumpulan sawit, pada saat jam kerja agar tidak ada yang namanya tidak melakukan pekerjaan pada saat jam kerja.

4.1.8 Analisis Sistem Usulan

Sistem yang diusulkan dari sistem yang sudah berjalan pada penelitian ini adalah mengembangkan aplikasi yang dapat berjalan pada sistem operasi *Android* sebagai karyawandan mandor untukmemanfaatkan layananpenyimpanan digital yang berbentuk *Cloud* dan menggunakan sistem yang di sebut dengan *GPS* (*Global Positioning System*) untuk mengetahui posisi karyawan pada saat absen. Usulan sistem ini akan dibuat dua aplikasi yaitu untuk pegawai dan mandor, alasan kenapa harus menggunakan dua aplikasi agar aplikasi tidak *crash* karena saat meminta information antara mandor dengan pegawai dan menghindari pelanggaran information penting. Berikut adalah penjelasan dan arsitektur sistem yang akan diusulkan:





Gambar 4.3 Flowmap Sistem Usulan

Pada Sistem usulan disisi Karyawan, adapun penjelasannya adalah sebagai berikut:

- 1) karyawan memulai dengan membuka aplikasi.
- 2) Lalu melakukan *login* yang sudah di daftarkan oleh mandor.
- 3) Sistem akan menampilkan halaman utama.
- 4) Setelah itu pegawai memilih menu absensi maka sistem akan menampilkan halaman absensi, Setelah itu karyawan mengisi *form* absensi lalu memilih tombol absen atau ajukan izin akan tersimpan ke *database*.
- 5) Pada menu aktivitas karyawan memilih menu aktivitas selanjutnya sistem akan menampilkan halaman aktivitas, Lalu karyawan memilih tambah aktivitas lalu karyawan mengisi *form* aktivitas lalu karyawan menekan tombol simpan lalu akan tersimpan ke *database*. Kegiatan pada aktivitas karyawan seperti pemberian pupuk, pemeliharaan pohon sawit, penurunan sawit dan pengumpulan sawit.
- 6) Pada menu panen karuawan memilih menu panen selanjutnya sistem akan menampilkan halman panen, Lalu karyawan memilih tambah hasil panen lalu karuawan mengisi *form* hasil panrn lalu karyawan menekan tombol simpan lalu akan tersimpan ke *database*. Pada hasil panen ini karyawan wajib memfoto pada saat penimbangan sawit agar tidak terjadi kesalahan pada saat pelaporan hasil panimbangan
- 7) Selesai

Selanjutnya pada sistem usulan disisi mandor. Dimana penjelasannya sebagai berikut:

- 1) Mandor memulai dengan membuka aplikasi tersebut
- 2) Setelah itu melakukan *login* lalu sistem akan menampilkan menu utama
- 3) Slanjutnya mandor mamilih menu absensi lalu lalu sistem akan meminta data absensi ke *database* lalu *database* akan mengeluarkan data absensi karyawan ,Setelah itu mandor akan memilih nama karyawan tersebut lalu mandor akan mengecek posisi karyawan tersebut apakah sudah di posisi atau tidak lalu mandor akan mengkonfirmasi kehadiran lalu akan tersimpan ke *database*.

- 4) Pada pengecekan aktivitas karyawan mandor memilih menu aktivitas lalu sistem akan meminta data aktivitas ke *database* lalu *database* akan mengirim data aktivitas karyawan tersebut lalu mandor akan mengecek data aktivitas karyawan tersebut seperti pemberian pupuk, pemeliharaan pohon sawit, penurunan sawit dan pengumpulan sawit.
- 5) Selanjutnya pada pengecekan panen karyawan mandor memilih menu panen lalu sistem akan meminta data panen ke *database* lalu *database* akan mengirim data panen karyawan tersebut lalu mandor akan mengecek data panen karyawan tersebut dan melihat foto hasil timbangan panen karyawan.
- 6) Selanjutnya pada menu pegawai mandor dapat melihat informasi pegawai, menambah pegawai, merubah informasi pegawai dan menghapus informasi pegawai dengan memilih menu pegawai maka sistem akan menampilkan halaman informasi pegawai.
- 7) Pada menu laporan mandor bisa melihat data hasil panen karyawan bulanan dengan cara mandor memilih menu laporan lalu sistem akan menampilkan halaman laporan lalu memilih nama karyawan yang mau dicek data panennya lalu akan disuruh mengisi *form* tanggal panen dan tanggal akhir panen lalu akan di kirim ke *database* lalu *database* akan menampilkan data tersebut lalu sistem akan langsung mengkonvert data tersebut ke pdf.
- 8) Selesai

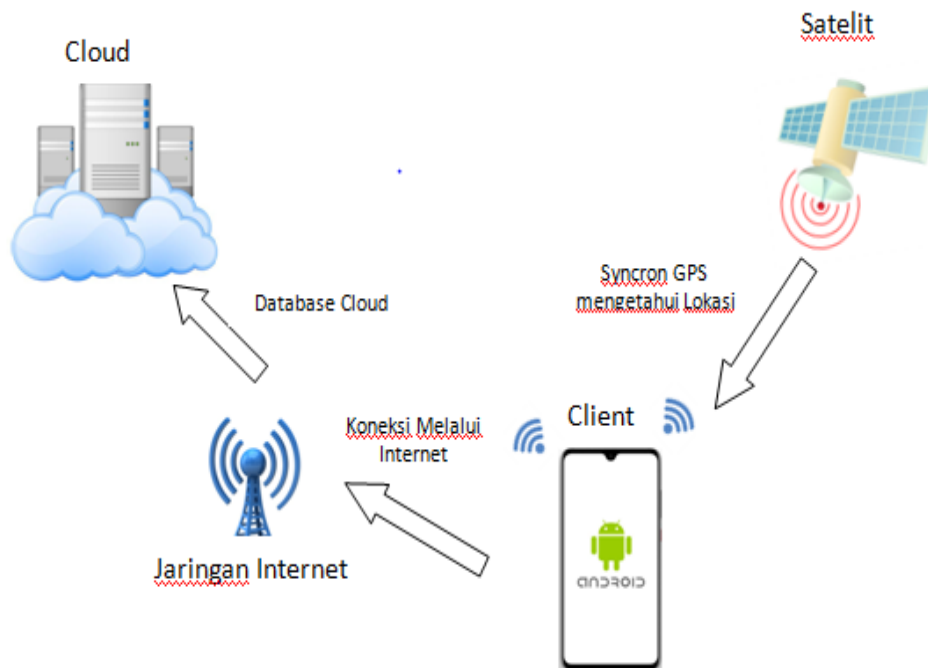
Selanjutnya pada sistem usulan disisi admin. Dimana penjelasannya sebagai berikut:

- 1) *Admin* memulai membuka *database*.
- 2) *Database* menampilkan halaman utama atau *home*.
- 3) Setelah itu *admin* memilih table karyawan, lalu menampilkan data karyawan dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* karyawan.

- 4) Setelah itu *admin* memilih table mandor, lalu menampilkan data mandor dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* mandor.
- 5) Setelah itu *admin* memilih table absen, lalu menampilkan data absen dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* absen.
- 6) Setelah itu *admin* memilih table aktivitas, lalu menampilkan data aktivitas dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* aktivitas.
- 7) Setelah itu *admin* memilih table panen, lalu menampilkan data panen dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* panen.
- 8) Setelah itu *admin* memilih table laporan, lalu menampilkan data laporan dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* laporan.
- 9) Setelah itu *admin* memilih table wilayah, lalu menampilkan data wilayah dipenyimpanan *database cloud*, *administrator* juga dapat mengedit dan menghapus *information* wilayah.
- 10) Selesai

4.2 Penerapan Metode *Lock GPS* Berbasis *Cloud*

Penulis dalam penelitian ini merancang dan membangun sebuah aplikasi berbasis android untuk melakukan absensi karyawan menggunakan metode kunci GPS, tujuan dibuatnya sistem ini adalah agar karyawan lapangan dapat melakukan absensi dari lokasi kerjanya di PTP..Perkebunan kelapa sawit Nusantara II Perk Bandar Klippa. Sistem ini menggunakan *distributedstorage* berupa *computerized*, jaringan web dan *handphoneAndroid* sebagai test system program. Sedangkan *cloud* adalah kapasitasnya, HP adalah media tempat aplikasi akan diinstal, dan jaringan web adalah media yang menghubungkan HP sebagai pelanggan dengan penyimpanan cloud melalui koneksi jaringan, dan menggunakan fitur GPS untuk melacak keberadaan karyawan. .



Gambar 4.4 Penerapan Metode *Lock GPS*

Metode yang digunakan pada gambar di atas adalah kunci GPS. Ini digunakan dengan penguncian GPS untuk menentukan rentang suatu wilayah dengan menentukan koordinat melalui Google API. Titik koordinat yang ditentukan berada di lokasi panen karyawan. Sistem akan dapat menentukan apakah lokasi karyawan pernah bekerja di pemanenan kelapa sawit karyawan; jika karyawan tersebut berada di luar lokasi panen yang telah ditentukan mandor, maka karyawan tersebut dipastikan tidak hadir. Dengan cara ini, karyawan harus benar-benar hadir di lokasi kerjanya yang ditentukan oleh mandor, sehingga kehadirannya dikonfirmasi oleh mandor.

Cloud disini sebagai *capacity* atau tempat penyimpanan data *setadvanced*, dimana *information* absensi, *information* pegawai dan *information* lokasi kerja. Agar karyawan dapat terhubung ke *cloud*, Sebagai koneksi, saya menggunakan jaringan web. Karyawan sekarang terhubung ke jaringan lokal dan di tempat kerja., karyawan dapat melakukan absensi jika karyawan tepat di lokasi tersebut, karyawan akan dikonfirmasi oleh mandor.

Dalam pembuatan sistem monitoring aktivitas karyawan lapangan dengan metode *lock gps* diimplementasikan menggunakan bahasa pemrograman *android*.

```

// bundle data
var mapViewBundle: Bundle? = null
if (savedInstanceState != null)
    mapViewBundle = savedInstanceState.getBundle(MAP_VIEW_BUNDLE_KEY)

// Generate map view
mapView.onCreate(mapViewBundle)
mapView.getMapAsync(this)

```

Gambar 4.5Menjalankan Fungsi Peta Pada Apk Karyawan

Gambar script diatas berfungsi untuk menjalankan fungsi peta secara paralel berguna untuk mendapatkan posisi *latitude* dan *longitude* pengguna pada saat melakukan absensipada aplikasi pegawai untuk mengetahui posisi pegawai pada saat absen.

```

// Implement OnMapReadyCallback
override fun onMapReady(googleMap: GoogleMap?) {
    // agar maps nya tetap tampil
    mapView.onResume()

    // insialisai maps
    mMap = googleMap

    // Tanyakan lagi ask permission
    askPermissionLocation()

    val permissiOne = ActivityCompat.checkSelfPermission(requireActivity(), Manifest.permission.ACCESS_FINE_LOCATION)
    val permissiTwo = ActivityCompat.checkSelfPermission(requireActivity(), Manifest.permission.ACCESS_COARSE_LOCATION)
    val pckgMgr = PackageManager.PERMISSION_GRANTED

    // Tanyakan permission
    if (permissiOne != pckgMgr && permissiTwo != pckgMgr ) {
        return
    }

    // Lokasi saya saat ini
    /*mMap?.isMyLocationEnabled = true*/
    mMap?.setOnCameraIdleListener(this)
}

```

Gambar 4.6Proses Kinerja Pada Maps View

Gambar script diatas disebut *overiding* method dari kinerja peta secara paralel guna mendapatkan posisi *latitude* dan *longitude*, yang mana hasil posisi *latitude* dan *longitude* diproses pada map view yang akan memberikan tampilan map pada aplikasi.

```

:
// generate map view
mapView.onCreate(mapViewBundle)
mapView.getMapAsync(this)

```

Gambar 4.7 Menjalankan Fungsi Peta Pada Apk Mandor

Gambar script diatas berfungsi untuk menjalankan fungsi peta secara paralel berguna untuk mendapatkan posisi latitude dan longitude pengguna pada saat mandor melihat karyawan yang sudah melakukan absensi.

```

override fun onMapReady(googleMap: GoogleMap?) {

    mapView.onResume()
    mMap = googleMap

    // Tanyakan lagi ask permission
    askPermissionLocation()

    val permissiOne = ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
    val permissiTwo = ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION)
    val pkgMgr = PackageManager.PERMISSION_GRANTED

    // Tanyakan permission
    if (permissiOne != pkgMgr && permissiTwo != pkgMgr) {
        return
    }

    // set marker berdasarkan lat dan lng
    val markerOptions = MarkerOptions()
    markerOptions.position(LatLng(dataAbsensi?.lat?.toDouble() ?: 0.0, dataAbsensi?.lng?.toDouble() ?: 0.0))
    markerOptions.title(dataAbsensi?.nama ?: "")
    markerOptions.snippet("Absen di wilayah ${dataAbsensi?.kodeWilayah}")
    mMap?.addMarker(markerOptions)

    // Move camera ke posisi ini
    mMap?.moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng(dataAbsensi?.lat?.toDouble() ?: 0.0, dataAbsensi?.lng?.toDouble() ?: 0.0), DEFAULT_ZOOM))
}

```

Gambar 4.8 Pembuatan Marker di Maps Aplikasi

Gambar di atas adalah *overidingmethod* yang di mana akan membuat marker di maps pada aplikasi mandor pada saat melakukan pengecekan absensi karyawan.

4.3 Penerapan *Cloud* Pada Aplikasi

Implementasi *cloudcomputing* adalah deskripsi fisik tentang bagaimana dan di mana aplikasi dan infrastruktur cloud computing akan dibangun untuk pengguna. Karena perusahaan atau organisasi besar lebih memilih untuk membangun model komputasi terdistribusi untuk kebutuhan mereka sendiri, seperti *server*, pemrograman, dan informasi yang dikelola secara pribadi di *serverfarm*, model implementasi untuk sistem ini adalah *cloud* pribadi. Pengembangan model *privatecloud* ini membutuhkan dana yang besar, serta pengalihan risiko dan beban operasional perusahaan. Perusahaan atau organisasi dapat menetapkan kebijakan mereka sendiri untuk mekanisme, keamanan, dan akses ke sumber daya komputasi awan.

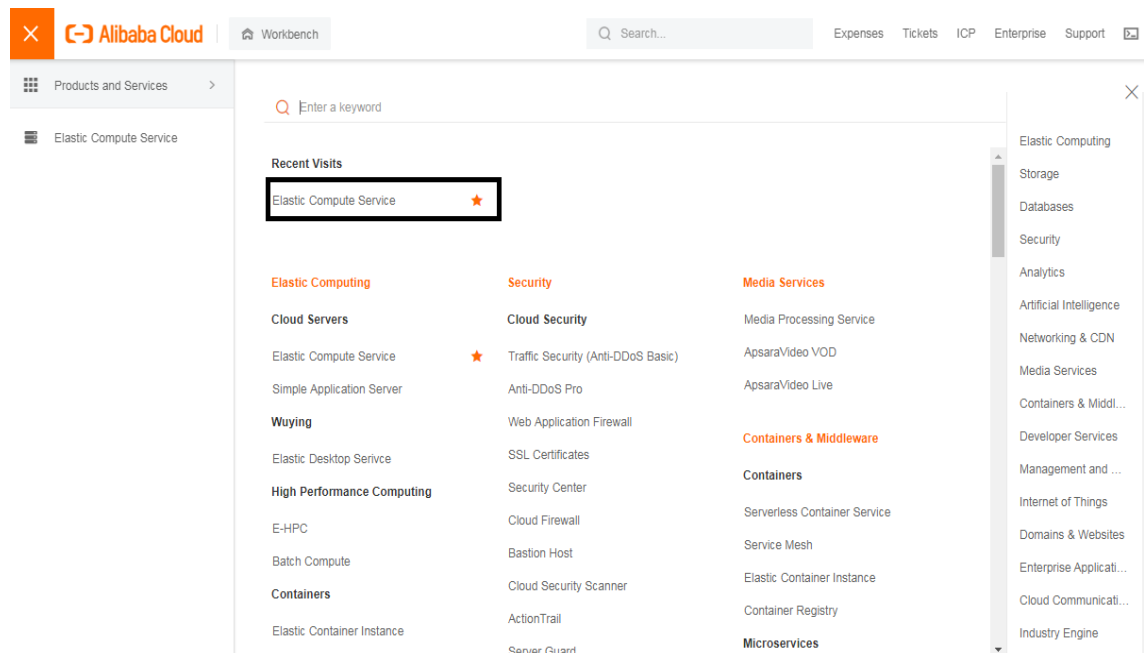
Penyedia *cloud* telah memberikan jaminan informasi agar data tidak mudah rusak atau rusak, tahapan teknologi, dan jaminan ISO dalam penerapan teknologi komputasi terdistribusi. Tentu saja, komputasi awan akan membuat data dan informasi lebih aman daripada metode tradisional yang saat ini digunakan oleh sebagian besar orang. Cloud computing juga menghilangkan kebutuhan media penyimpanan seperti *harddrive* eksternal karena *server* sudah memiliki media penyimpanan terpusat. Karena semua perangkat keras atau produk fisik memiliki masa pakai, dan setelah masa pakai itu, kerusakan, fungsi yang tidak ideal, dan kesalahan sering terjadi.

Pada aplikasi monitoring aktivitas karyawan lapangan menggunakan penyimpanan Alibab *cloud* yaitu sebuah layanan virtual private server, pada aplikasi ini yang dimana nantinya akan memakai VPS (*Virtual Private Server*) atau yang di sebut ECS (*Elastic Compute Service*) pada alibaba *cloud*, sebelum masuk konfigurasi membuat VPS dengan ECS pada Alibaba cloud akan di jelaskan terlebih dahulu tentang VPS. VPS ialah sebuah komputer virtual yang bertindak sebagai server untuk data dan aplikasi, website, dan hosting. Alibaba

Cloud menyebutnya ECS, fungsi VPS yaitu sebagai hosting/server untuk website memang sedikit lebih kompleks untuk mengaturnya, tidak semudah web hosting biasa yang kita tinggal pakai. Tetapi untuk performa, VPS yang terkonfigurasi baik, performanya jauh lebih bagus daripada shared web hosting. Walaupun mirip, setiap provider VPS mempunyai istilah dan fitur-fitur yang berbeda. Artikel ini akan membahas tentang bagaimana cara memilih ECS yang tepat dan sesuai dengan kebutuhan di Alibaba Cloud.

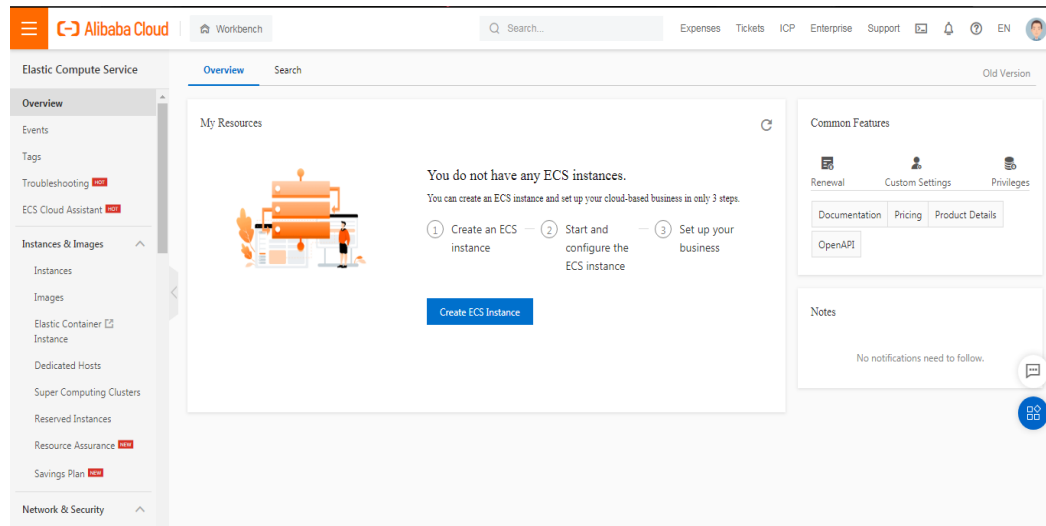
4.3.1 Langkah-langkah Pembuat VPS dengan *Elastic Compute Service (ECS)* Alibaba Cloud

Untuk membuat ECS silahkan masuk ke Console dari Alibaba Cloud. Geser *pointer* mouse pada bar sebelah kiri dan kita akan menemukan produk yang dijadikan favorit. Jika tidak menemukan *Elastic Compute Service* silahkan ke menu *Products* dan pilih *Elastic Compute Service*. Jika ingin membuat ECS sebagai produk favorit klik bintang yang ada di sebelah kanannya.



Gambar 4.9 Pencarian ECS (*Elastic Compute Services*)

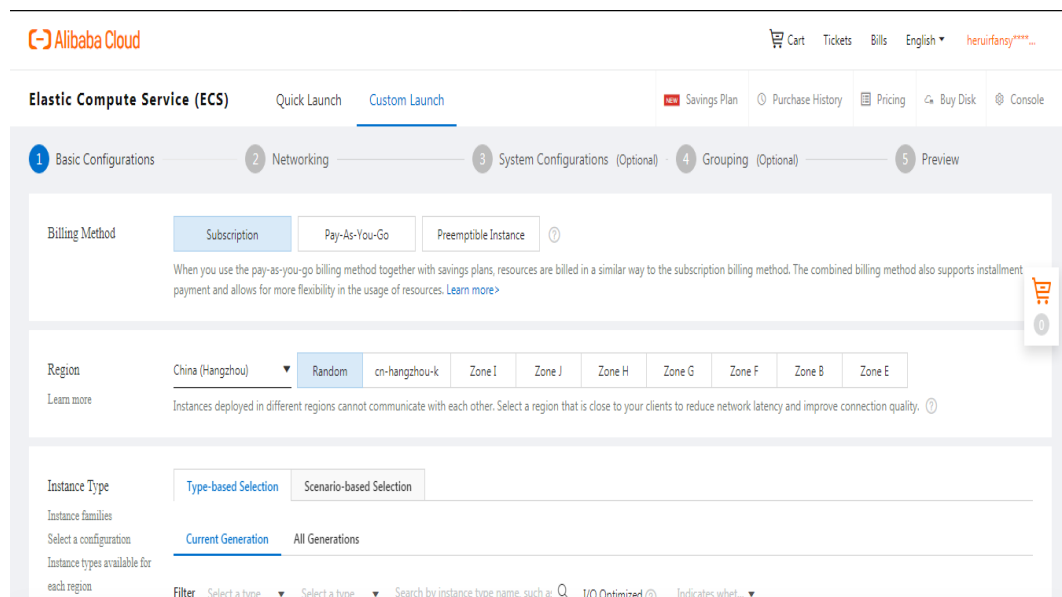
Setelah itu akan langsung masuk ke tampilan creat ECS Instance, setelah itu pilih *Creat ECS Instances* setelah itu akan menampilkan dashboard registrasi ESC (**Elastic Compute Service**).



Gambar 4.10 Tampilan *Creat ECS Instance*

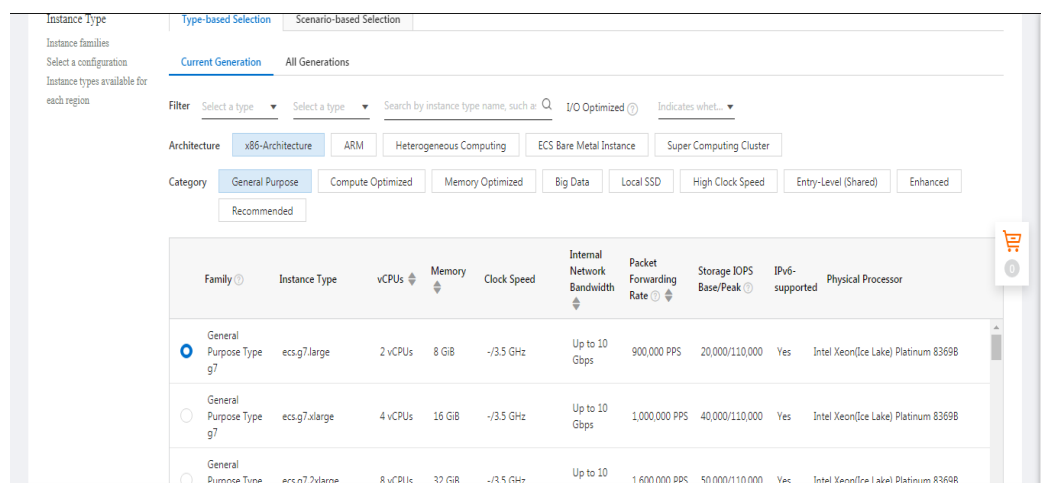
1. Basic Configuration ECS

Pada bagian selanjutnya ialah konfigurasi dasar untuk *Instance* yang akan di buat.



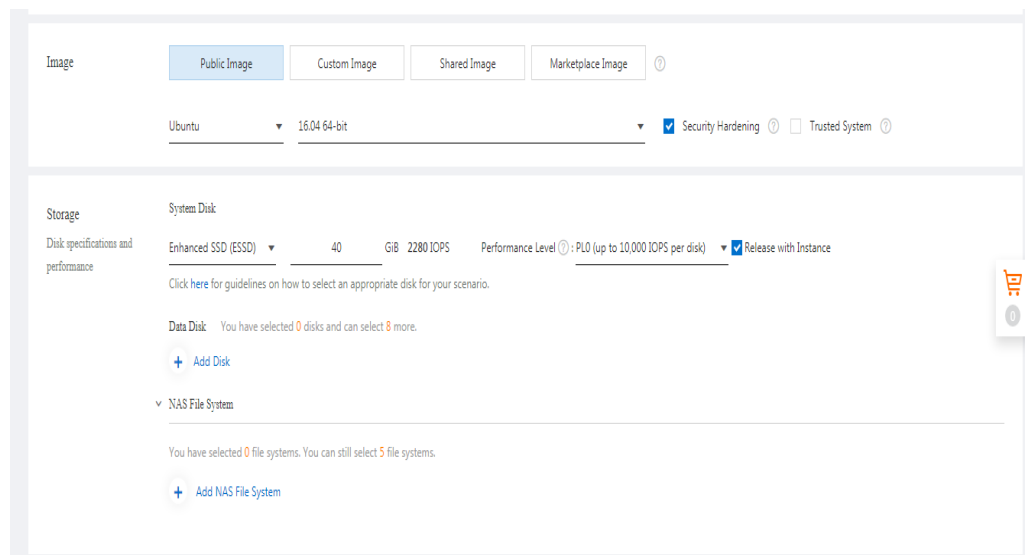
Gambar 4.11 Konfigurasi ECS Bagian 1

Pada bagian *Billing Method*, ini merupakan cara penagihan yang akan dilakukan oleh Alibaba Cloud. Terdapat 2 pilih yang *Subscription* dan *Pay-As-You-Go*. *Subscription* atau yang artinya berlangganan adalah kondisi dimana siklus tagihan akan dilakukan bulanan atau tahunan. Sedangkan untuk *Pay-As-You-Go* adalah metode penagihan yang dihitung setiap jam. Artinya tiap jam saldo pada kartu kredit/paypal milik kita akan terpotong. *Region* adalah memilih lokasi server dari *instance* yang kita buat ini. Saya sarankan untuk memilih lokasi server yang berada paling dekat dengan target pengguna. Jika digunakan untuk web dan penggunanya orang Indonesia maka pilih Jakarta karena lebih dekat maka secara ilmu jaringan dasar maka akan lebih cepat untuk diakses.



Gambar 4.12 Konfigurasi ECS Bagian 2

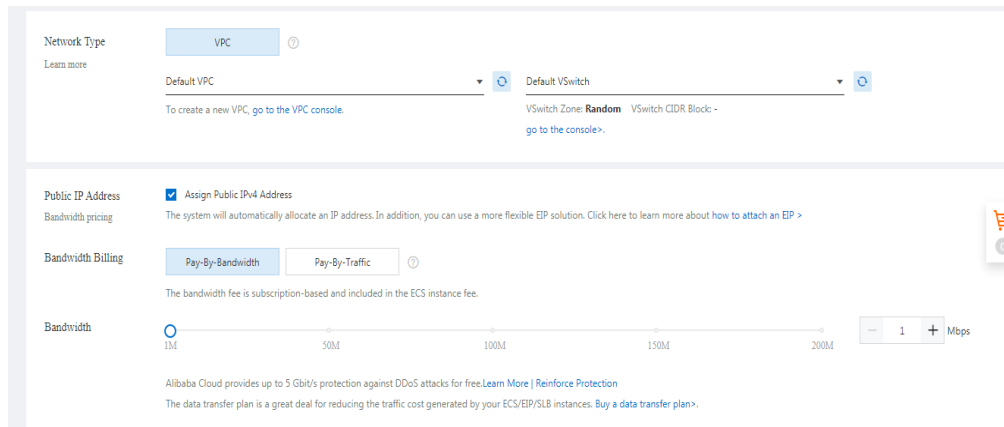
Untuk tipe *instance* adalah spesifikasi dari *instance* yang akan kita gunakan. Terdapat 3 arsitektur yaitu tipe *x86-Architecture* yang memiliki spesifikasi seperti VPS pada umumnya, tipe *Heterogeneous Computing* yang berfokus pada GPU, dan tipe Bare Metal yang digunakan layaknya kita memiliki server fisik. Pada setiap tipe arsitektur masih memiliki beberapa kategori yang digunakan untuk mengkategorikan tipe instance yang tersedia. Dengan cara ini pengguna bisa lebih mudah menentukan *instance* mana yang cocok. Sebenarnya pada bagian atas tersedia juga alat *filter* untuk ini. Jadi silahkan pilih mau *filter* yang bagaimana.



Gambar 4.13 Konfigurasi ECS Bagian 3

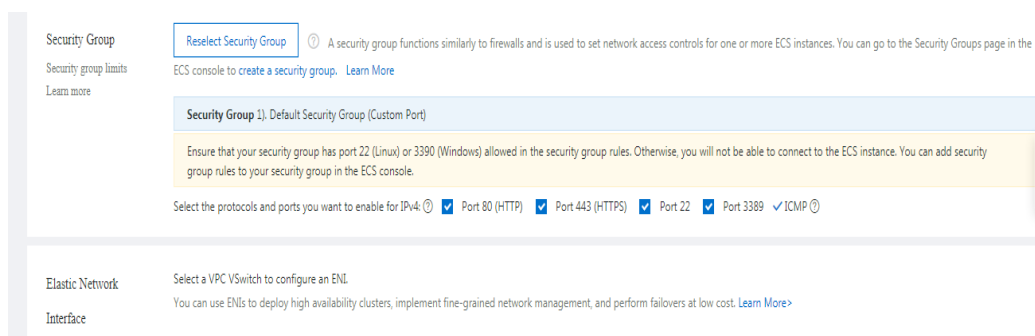
Pada *Image* penggunaannya adalah memilih Sistem Operasi yang akan digunakan pada *instance* ini. Pada pembahasan kali ini kita langsung saja menggunakan **Public Image** yang didalamnya terdapat beberapa *image* yang telah disediakan oleh pihak Alibaba Cloud. Bagian **Storage** merupakan bagian yang mengatur penyimpanan pada *instance*. Terdapat 2 jenis penyimpanan yaitu *System Disk* yang bersifat wajib karena digunakan untuk menyimpan sistem dan *Data Disk* yang bersifat opsional karena digunakan untuk menyimpan data tambahan. Bahasa sederhananya adalah perbedaan partisi antara sistem dengan data. Masing-masing *disk* memiliki ketentuan kapasitas yaitu untuk *System Disk* minimal 40GB dan maksimal 500GB. Sedangkan untuk *Data Disk* kapasitas minimalnya adalah 20GB dan maksimal 32.768GB. Jika *Data Disk* dengan ukuran itu dianggap masih kurang tenang saja karena kita bisa menambahkan hingga 16 buah *Data Disk*. Secara bawaan pada bagian *Storage* akan menggunakan Ultra Cloud Disk yang memiliki IOPS sebesar 2.120 pada *System Disk* dan 5.000 pada *Data Disk*. Jika ingin meningkatkan bisa memilih untuk menggunakan SSD yang memiliki IOPS 3.000 pada *System Disk* dan 25.000 pada *Data Disk*. Jika pada bagian ini telah selesai kita bisa klik **Next** untuk melanjutkan ke bagian *Networking*.

2. Networking Configuration ECS



Gambar 4.14 Konfigurasi Network ECS Bagian 1

Di bagian atas terdapat *Network Type* yang akan mengatur VPC dan VSwitch, biarkan saja sebagaimana mestinya. Lalu untuk Network Billing Method adalah teknik pembayaran dari penggunaan jaringan. Maka akan dikenakan biaya jika *instance* ini memiliki IP Publik, namun karena saat ini kita membutuhkan jadi kita ceklis *AssignPublicIP*. Ketika diceklis maka akan muncul pengaturan besar kecepatan dari jaringan kita. Gunakan secukupnya saja walaupun memperbesar kecepatan tidak dikenakan biaya. Saya menyebut bahwa kita akan dikenakan biaya jika menggunakan IP Publik, pengaturan biayanya adalah berdasarkan besarnya lalu lintas. Harga setiap GB data adalah USD 0,090 jadi masih bisa dibilang murah.



Gambar 4.15 Konfigurasi Network ECS Bagian 2

Pada bagian **Security Group** kita hanya cukup memastikan bahwa *port* yang digunakan terbuka. Jika *instance* ini nantinya akan dijadikan sebagai web server maka pastikan *port* HHTP atau HTTPS terbuka (ceklis). Jika sudah klik *next* untuk ke menu *System Configurations*.

3. System Configurations ECS

The screenshot displays the 'System Configurations' section of the AWS Management Console for ECS. It includes the following fields and options:

- Logon Credentials:** Radio buttons for 'Key Pair', 'Password' (selected), and 'Set Later'. A note states: 'SSH key pairs provide higher security than custom passwords and can prevent brute-force attacks. We recommend that you use SSH key pairs to create instances.'
- Logon Username:** Text input field containing 'root'.
- Logon Password:** Text input field with a note: 'You can set a password after the instance is created.' Below it, a detailed password policy: 'The name must be 8 to 31 characters in length and contain at least three types of the following characters: uppercase letter, lowercase letters, digits, and special characters. Special characters include the following ones: []~!@#%&*'_{-+~}|:;<>.,/;. The logon password for a Windows instance cannot start with a forward slash (/).'
- Confirm Password:** Text input field with a note: 'Remember your password. If you forget your password, log on to the ECS console to reset your password.'
- Instance Name:** Text input field containing 'launch-advisor-20210602'. A link 'Learn how to customize sequential instance names.' is provided. A note states: 'The name must be 8 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), colons (:), and hyphens (-). It must start with a letter.'
- Description:** Text area with a placeholder 'Enter a description'. A note states: 'The description must be 2 to 256 characters in length and cannot start with http:// or https://.'
- Host:** Text input field with a placeholder 'Enter a hostname'. A link 'Learn how to customize sequential hostnames.' is provided. A note states: 'In other operating systems such as Linux: The hostname must be 2 to 64 characters in length. It can contain multiple segments separated by periods (.). Each segment can contain letters, digits, and hyphens (-), but consecutive periods (.) or hyphens (-) are not allowed. The hostname cannot start or end with a period (.) or a hyphen (-).'
- Sequential Suffix:** A checkbox labeled 'Add Sequential Suffix to Instance Name and Hostname'.

Gamabr 4.16 Sistem Konfigurasi ECS

Di *System Configuration* kita akan melakukan konfigurasi sistem operasi yang mengarah kepada kredensial. Terdapat 3 pilihan yaitu *Key Pair*, *Password*, dan *Set Later*. Untuk saat ini kita gunakan *password* agar seperti VPS pada umumnya. Kemudian masukkan *secretword* yang akan digunakan untuk masuk ke VPS nantinya. Untuk *instance name*, *description*, dan *host* silahkan sesuaikan saja karena ini hanyalah sebagai penanda untuk mempermudah kita mengelola jika memiliki banyak *instance* nantinya. Jika sudah klik *Next* untuk melanjutkan ke *Grouping*.

4. Grouping & Preview ECS

Grouping berfungsi untuk mengelompokkan *instance*. Ini digunakan ketika kita memiliki banyak *instance*. Di bagian *preview* hanya menampilkan seluruh konfigurasi kita dari awal, jadi sekedar memastikan saja bahwa *instance* telah sesuai dengan keinginan. Sekedar informasi bahwa *Auto Release Schedule* adalah jadwal penghapusan *instance* secara otomatis. Jadi jika di set maka pada tanggal tersebut *instance* akan otomatis terhapus. Jika sudah pasti maka silahkan klik *Create Instance* lalu klik *Console* untuk melakukan pembuatan *instance* dan tunggu sesaat. Jika sudah aktif di halaman *Console* akan muncul sebuah *instance* dengan status *running*. Ini bertanda bahwa VPS kita telah siap digunakan dengan IP Publik yang tersedia seperti gambar di bawah ini:

The screenshot displays the AWS Management Console interface for Elastic Compute Service (ECS) instances. The page title is 'Elastic Compute Service / Instances'. A 'Create Instance' button is visible in the top right corner. Below the title, there is a search bar with the placeholder text 'Select an instance attribute or enter a keyword' and a 'Tags' button. An 'Advanced Search' link and a settings icon are also present. The main content area shows a table of instances with the following columns: Instance ID/Name, Tag, Monitoring, Zone, IP Address, Status, Network Type, Specifications, Billing Method, and Actions. One instance is listed with the following details:

Instance ID/Name	Tag	Monitoring	Zone	IP Address	Status	Network Type	Specifications	Billing Method	Actions
i-8ps851olzko08z6b68e3 heru-irfansyah			Kuala Lumpur Zone A	47.254.254.112(Public) 172.23.39.198(Private)	Running	VPC	1 vCPU 1 GiB (I/O Optimized) ecs.t5-ic1m1.small 1Mbps	Subscription Expires August 6, 2022, 22:59(Time Zone: UTC + 7)	Manage Connect Upgrade/Downgrade Renew More

Below the table, there is a row of action buttons: Start, Stop, Restart, Reset Password, Renew, Switch to Subscription, Release, and More. At the bottom right, there is a pagination control showing 'Total: 1 item(s), Per Page: 20 item(s)' and a page number '1'.

Gambar 4.17 Tampilan *Instance*

4.3.2 Penerrapan Aplikasi Pada Server Alibaba Cloud

Setelah melakukan pembuatan VPS dengan *Elastic Compute Service* (ECS) Alibaba *Cloud*. Selanjutnya melakukan konfigurasi pada terminal ECS ya itu dengan mensetting atau meninstal Linux, Apache, Mysql dan PHP pada ECS untuk mengkoneksikan *database* aplikasi ke alibaba *cloud*.

```

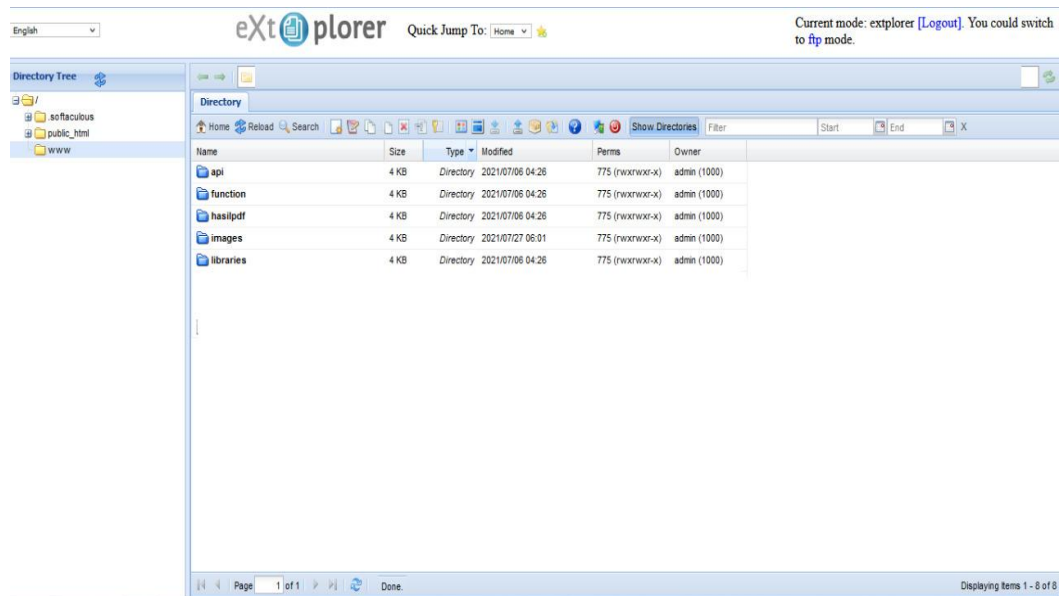
<?php
...
$hostname = 'localhost';
$username = 'root';
$password = '';
$database = 'db_ptpn';

$connect = mysqli_connect($hostname, $username, $password, $database);
if (mysqli_connect_errno()){
    set_response("failed", mysqli_connect_error());
    die();
} else {
    //echo "Success connect to mysql";
}

```

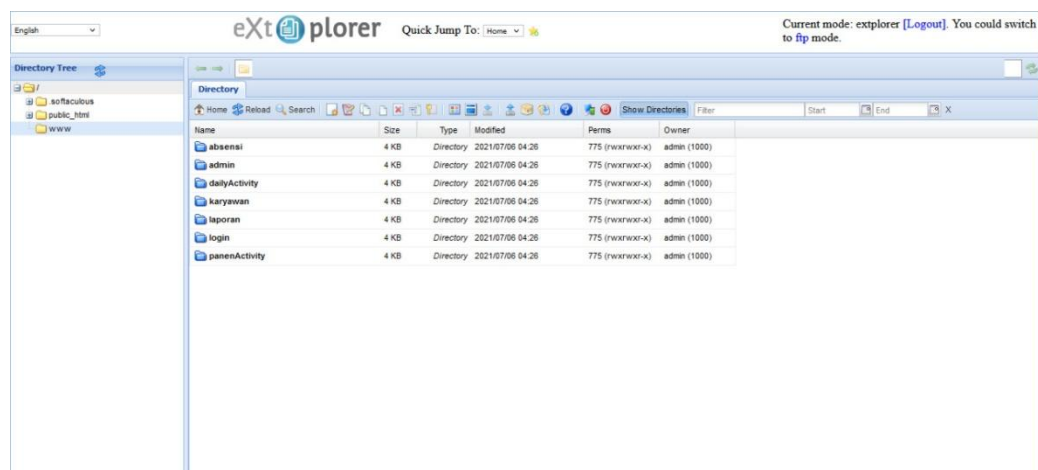
Gambar 4.18 Scripting Koneksi *Server Alibab Cloud*

Pada gambar di atas adalah skrip PHP yang terhubung ke koneksi server yang sudah dibuat di alibab cloud untuk terhubung ke database mysql yang dibuat di cpanel. Setelah konfigurasi selesai bahwasannya cpanel yang di mana *app backend* akan ditaruh pada *Cpanel*, *CPanel* adalah *controlboard* yang digunakan untuk melakukan *setting* pada layanan *webfacilitating* yang meliputi manajemen *document*, *informationbase*, *space*, *security*, *programming* dan konfigurasi lainnya, Sedangkan *backend* ialah bagian pada website atau aplikasi mobile untuk mengatur logika aplikasi dan mengolah data-datanya.



Gambar 4.19 Cpanel yang Berisi Backend

Pada gambar diatas ialah cpanel yang dimana didalamnya terdapat backend yang ada pada *cloud*. Agar aplikasi bisa dijalankan secara online, apabila tidak ada backend maka aplikasi tersebut tidak akan bisa dijalankan atau server terputus. Pada absen karyawan, admin, login, aktivitas karyawan, hasil panen karyawan, data karyawan dan laporan karyawan sudah masuk ke database alibaba cloud yang dimana otomatis data tersebut sudah online.



Gambar 4.20 Tampilan Folder *Scripting* Aplikasi

Pada gambar di atas ialah Folder Scripting aplikasi yang sudah terkoneksi pada alibaba *cloud* isi folder tersebut terdapat absensi, admin, aktivitas, hasil panen, karyawan dan laporan.

4.4 Workshop Design

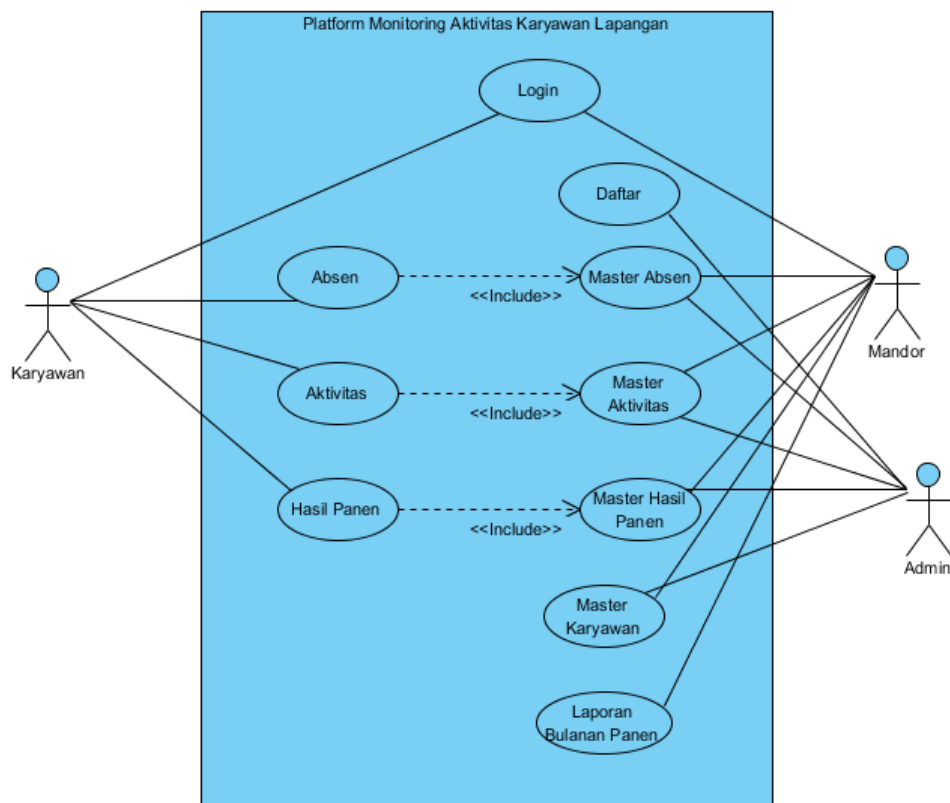
Bagian *Design Workshop* ini membahas analisis yang lebih detail dari usulan jalur sistem yang telah dibahas pada bagian sebelumnya. Bagian ini akan membahas tiga hal: analisis sistem menggunakan diagram UML, analisis basis data, dan analisis antarmuka.

4.4.1 Design Model

Pada perancangan UML penulis hanya membuat 4 diagram yaitu *use case*, *activity diagram*, *sequence diagram*, dan *classdiagram*.

4.4.1.1 Use Case Diagram

Singkatnya, use case adalah sketsa yang menggambarkan siapa yang menggunakan sistem dan apa yang dapat mereka lakukan.



Gambar 4.21 Use Case Diagram

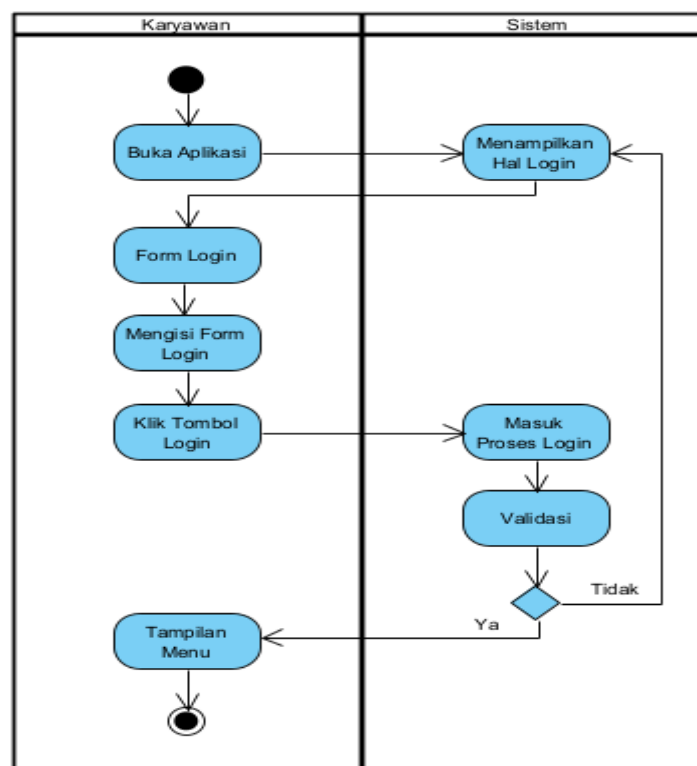
Pada gambar di atas dapat dilihat bahwa yang berperan sebagai aktor ialah karyawan, mandor dan *admin*, dari diagram terlihat kedua aktor dapat melakukan

login. Aktor karyawan dapat melakukan absensi yang menggunakan *GPS*, pengajuan ijin kerja, mengupload aktivitas mereka selama jam kerja dan hasil panen mereka. Sementara mandor dapat melakukan pengkonfirmasiannya karyawan hadir atau pun ijin, melihat aktivitas karyawan dan melihat data timbangan hasil panen karyawan, menambahkan data karyawan baru dan bisa membuat laporan data bulanan hasil panen karyawan yang bisa diconvertkan ke PDF. Sedangkan *admin* mendaftarkan *login* mandor dan dapat melihat data absen, aktivitas karyawan, hasil panen karyawan, dan data karyawan.

4.4.1.2 Activity Diagram

Diagram activity menggambarkan aliran aktivitas yang dapat dilakukan pengguna dan administrator. Kegiatan yang akan dipresentasikan dibagi menjadi dua kategori yaitu untuk klien dan untuk mandor. Beberapa kegiatan yang akan dibahas pada platform monitoring aktivitas pegawai lapangan adalah sebagai berikut.

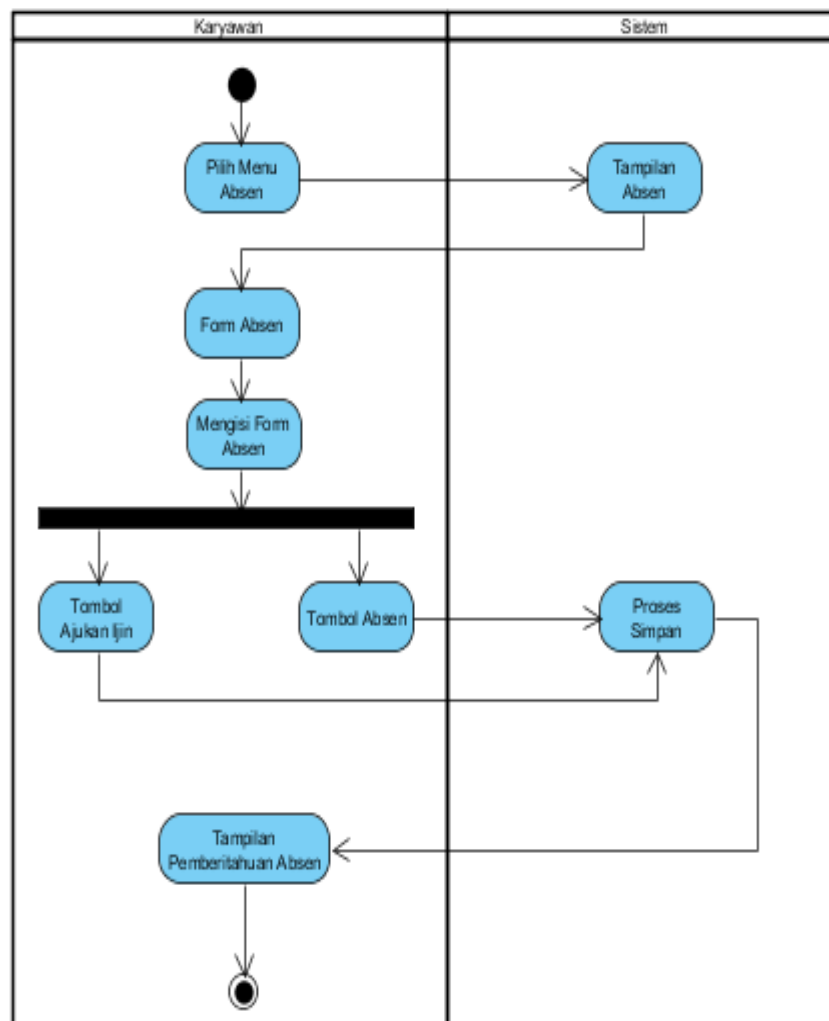
1. Activity Diagram Login



Gambar 4.22 Activity Diagram Login

Proses *login* digambarkan pada gambar di atas; pada saat *login* karyawan akan membuka aplikasi, setelah itu sistem akan menampilkan halaman *login*, setelah itu karyawan akan menampilkan struktur *login*, setelah itu karyawan akan mengisi struktur *login* yang telah didaftarkan oleh mandor, setelah itu karyawan akan menekan *login*, dan sistem akan memproses informasi *login* mereka; jika *username* atau *secretkey* salah maka pegawai akan kembali ke halaman *login*. Jika *username* dan *secretphrase* karyawan sudah benar, dia akan langsung dibawa ke tampilan menu.

2. Activity Diagram Absensi (Karyawan)

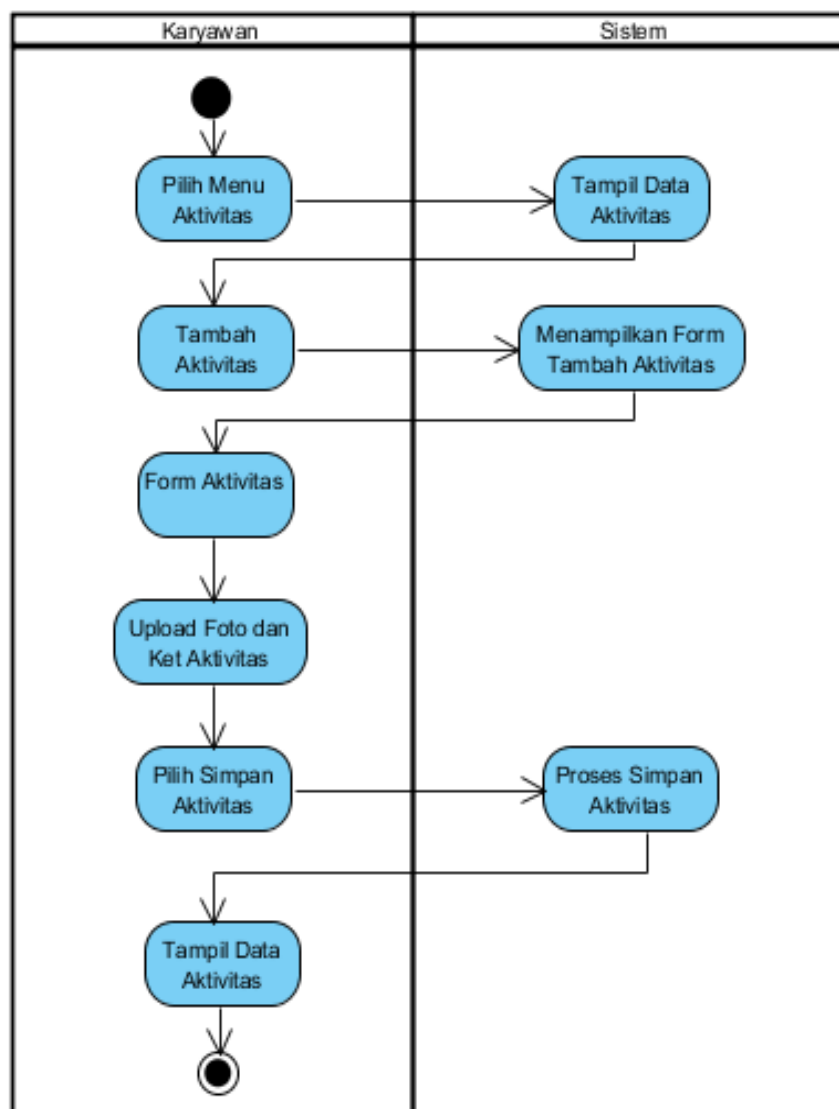


Gambar 4.23 Activity Diagram Absensi Karyawan

Pada gambar di atas adalah compositions melakukan absensi karyawan, saat melakukan absensi karyawan akan memilih menu absensi setelah itu sistem akan

memunculkan *form* absensi lalu karyawan diwajibkan mengidupkan lokasi GPS mereka yang di *android* masing-masing. Setelah itu karyawan diwajibkan mengisi kode wilayah panen yang sudah di tentukan mandor, lalu karyawan menekan tombol absen untuk menyimpan absensi mereka. Apabila karyawan ingin melakukan pengajuan ijin karyawan akan menekan tombol ajukan izin setelah itu sistem akan memproses simpan dan sistem akan memberitahukan bahwasannya sudah melakukan absen atau ajukan izin

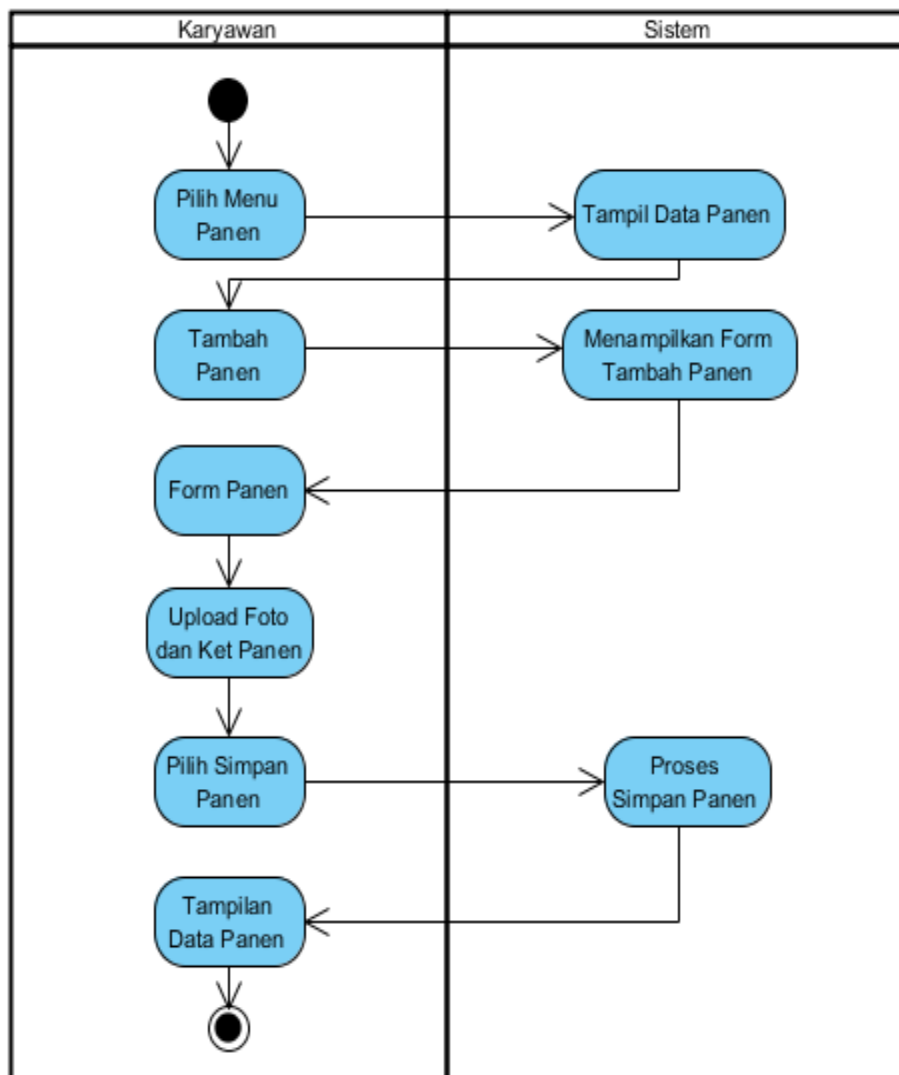
3. Activity Diagram Aktivitas Karyawan



Gambar 4.24 Activity Diagram Aktivitas Karyawan

Pada gambar diatas adalah proses melakukan aktivitas karyawan, pada saat melakukan aktivitas karyawan akan memilih menu aktivitas setelah itu sistem akan memproses halaman aktivitas karyawan, selanjutnya karyawan akan di suruh menambahkan aktivitas mereka dengan menekan tombol tambah aktivitas, setelah itu karyawan di suruh *upload* foto aktivitas mereka lalu membuat keterangan mereka sedang apa selanjutnya apa bila sudah selesai karyawan menekan simpan lalu karyawan akan langsung menuju data tampilan aktivitas mereka.

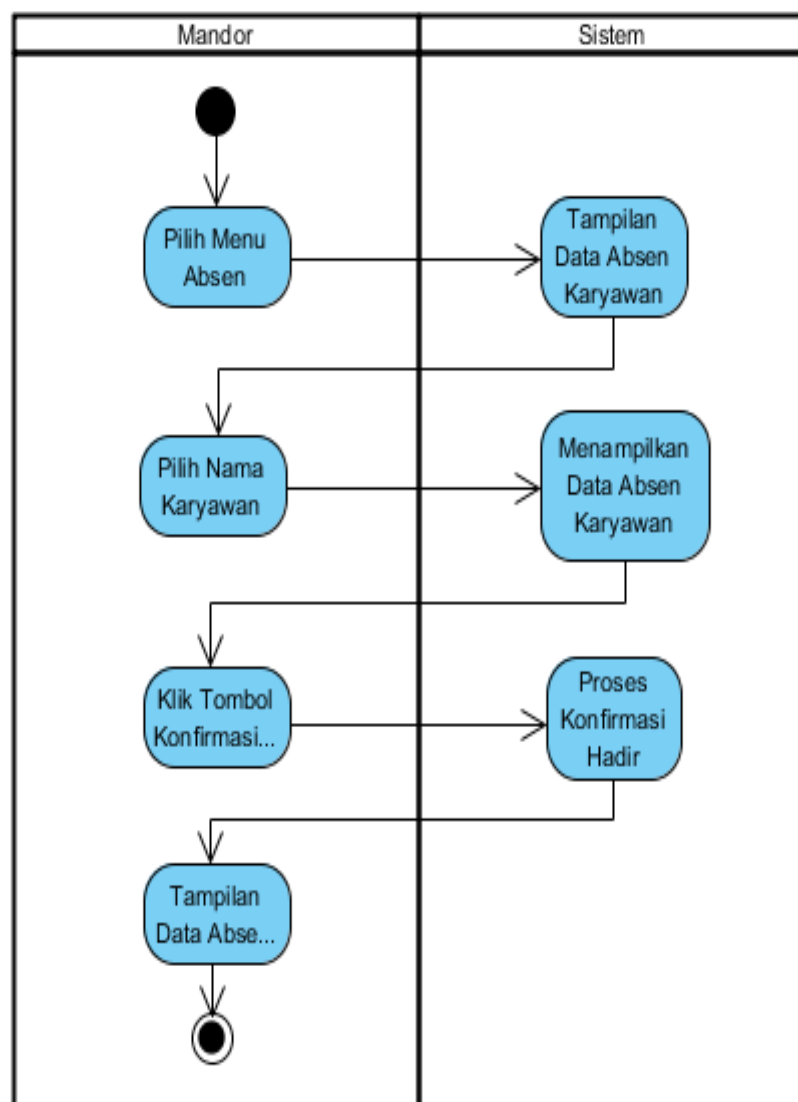
4. Activity Diagram Hasil Panen Karyawan



Gambar 4.25 Activity Diagram Hasil Panen Karyawan

Pada gambar diatas adalah proses melakukan hasil panen karyawan, pada saat melakukan hasil panen karyawan akan memilih menu panen setelah itu sistem akan memproses ke halaman panen karyawan, selanjutnya karyawan di suruh melakukan tambah hasil panen lalu sistem akan menampilkan *form* panen, setelah itu karyawan diwajibkan menguploadfoto timbangan panen mereka, membuat keterangan dan menulis kuantitas (kg) hasil panen mereka lalu menekan tombol simpan.

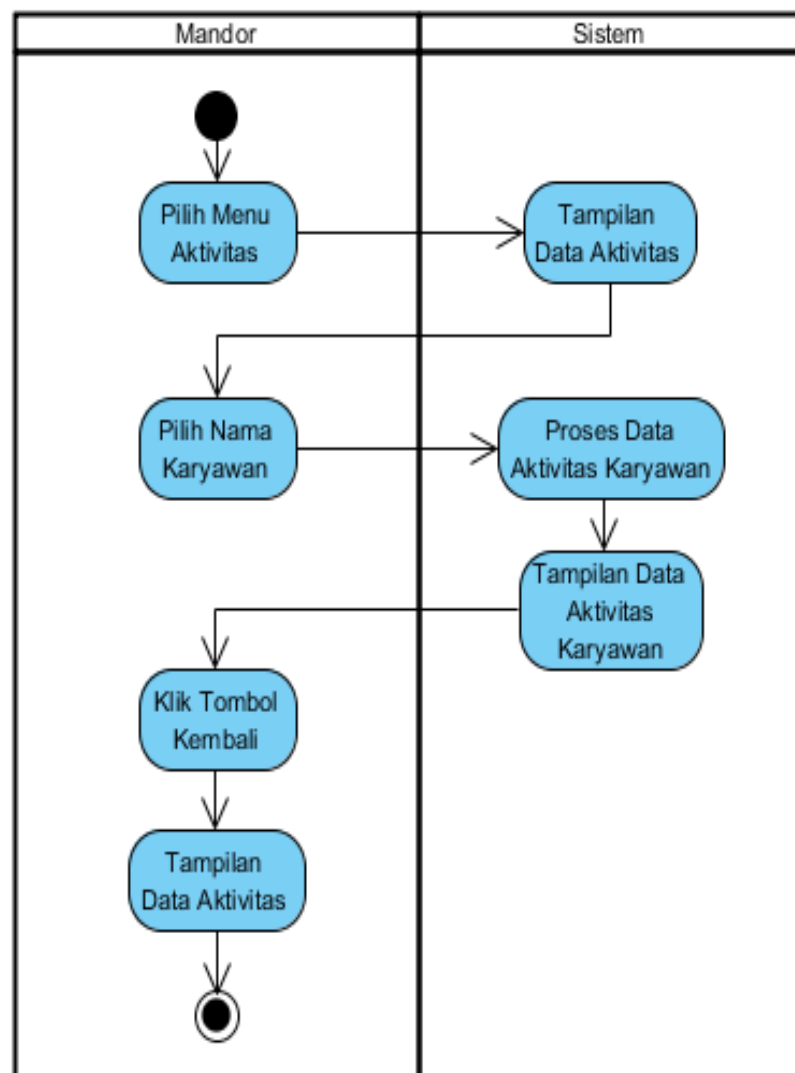
5. Activity Diagram Laporan Absen (Mandor)



Gambar 4.26 Activity Diagram Laporan Absen (Mandor)

Pada gambar diatas adalah proses melakukan pengecekan absen karyawan pada aplikasi mandor, Pada saat melakukan pengecekan absen karyawan, mandor akan memilih menu absen setelah itu sistem akan memproses halaman absen karyawan, lalu sistem akan menampilkan data absen karyawan. Selanjutnya mandor akan memilih nama karyawan setelah itu sistem akan menampilkan data absen karyawan tersebut lalu apabila karyawan benar berada di posisi kerja mandor akan mengkonfirmasi kehadiran karyawan tersebut setelah itu mandor akan kembali lagi ke tampilan absen karyawan.

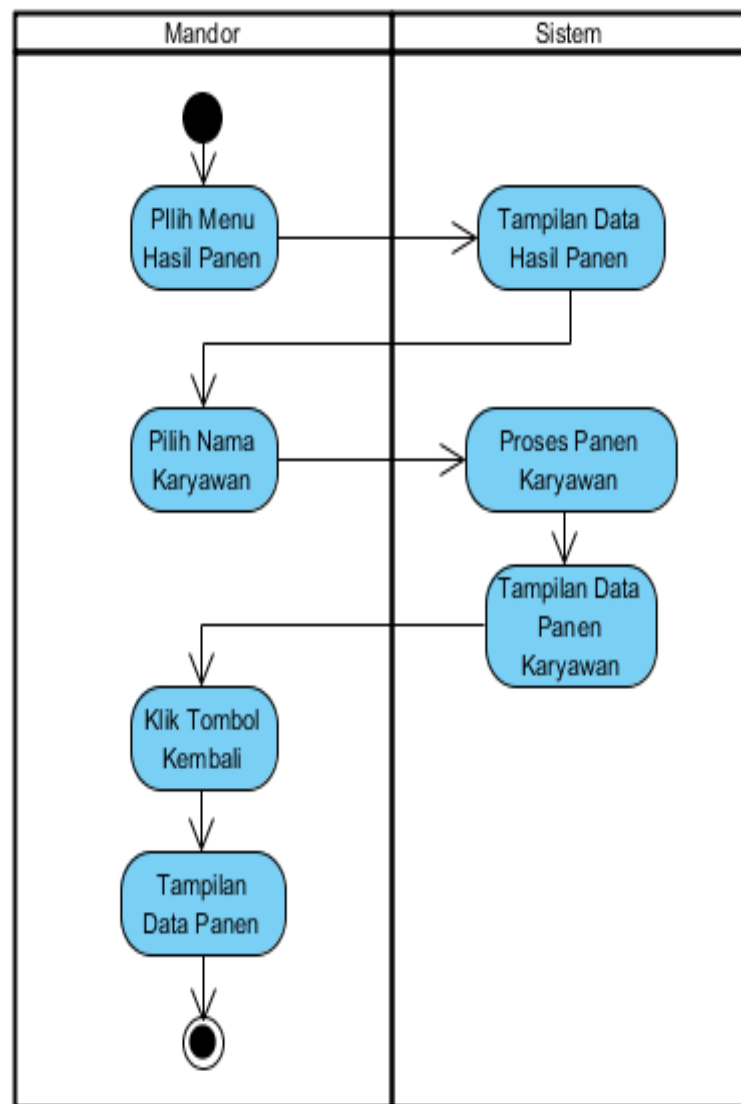
6. Activity Diagram Laporan Aktivitas Karyawan (Mandor)



Gambar 4.27 Activity Diagram Laporan Aktivitas karyawan (Mandor)

Pada gambar diatas merupakan expositions pengecekan aktivitas pegawai pada aplikasi mandor, pada saat pengecekan aktivitas pegawai mandor akan memilih menu master aktivitas setelah itu sistem akan menampilkan halaman information aktivitas pegawai, setelah itu mandor akan memilih nama pegawai maka sistem akan mengolah *information* aktivitas pegawai dan menampilkan *information* aktivitas kemudian setelah itu mandor menekan tombol kembali dan mandor akan kembali ke halaman tampilan information aktivitas.

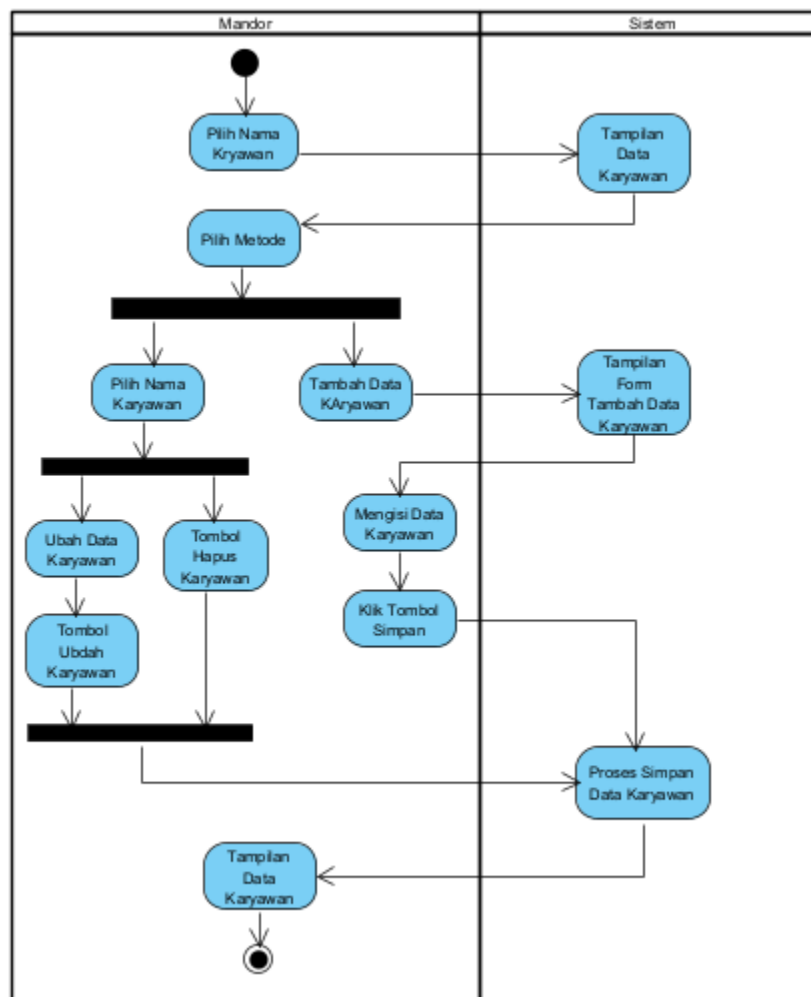
7. Activity Diagram Data Panen Karyawan(Mandor)



Gambar 4.28Activity DiagramData Panen Karyawan(Mandor)

Pada gambar diatas merupakan *expositions* pengecekan panen karyawan pada aplikasi mandor, pada saat melakukan pengecekan panen karyawan mandor akan memilih menu master panen setelah itu sistem akan menampilkan halaman *information* panen karyawan, setelah itu mandor akan memilih nama pegawai maka sistem akan mengolah *information* panen pegawai dan menampilkan *information* panen kemudian setelah itu mandor menekan tombol kembali dan mandor akan kembali ke halaman tampilan *information* induk panen.

8. Activity Diagram Master Karyawan (Mandor)

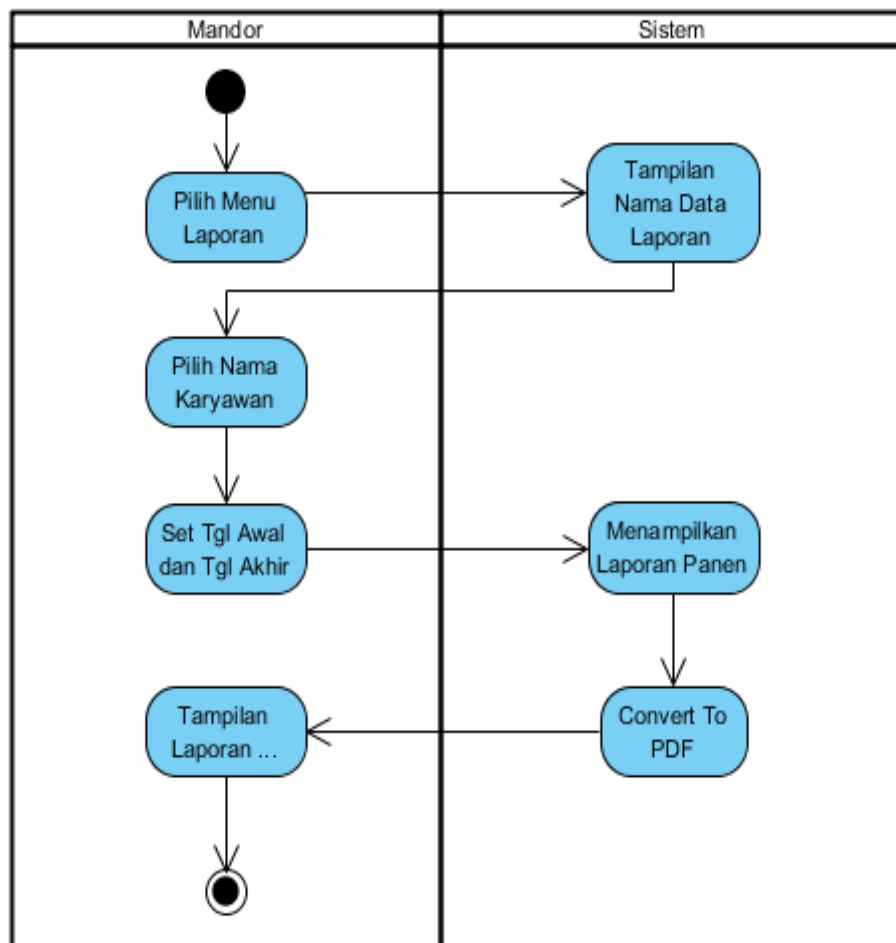


Gambar 4.29 Activity Diagram Master Karyawan (Mandor)

Pada gambar diatas adalah proses melakukan penambahan,ubah dan hapus data karyawan pada aplikasi mandor, pertama-tama mandor memilih menu

karyawan setelah itu sistem akan memunculkan tampilan data karyawan selanjutnya mandor memilih tombol tambah karyawan untuk menambah data karyawan sistem akan menampilkan *form* tampilan data karyawan. Lalu mandor akan mengisi data karyawan tersebut lalu menekan tombol simpan setelah itu karyawan akan kembali ke tampilan data karyawan, untuk bagian hapus dan ubah data karyawan, mandor memilih karyawan tersebut setelah itu mandor tinggal mengubah data karyawan tersebut dengan menekan tombol ubah atau hapus data karyawan, setelah itu karyawan akan kembali ketampilan data karyawan.

9. Activity Diagram Laporan Hasil Panen (Mandor)

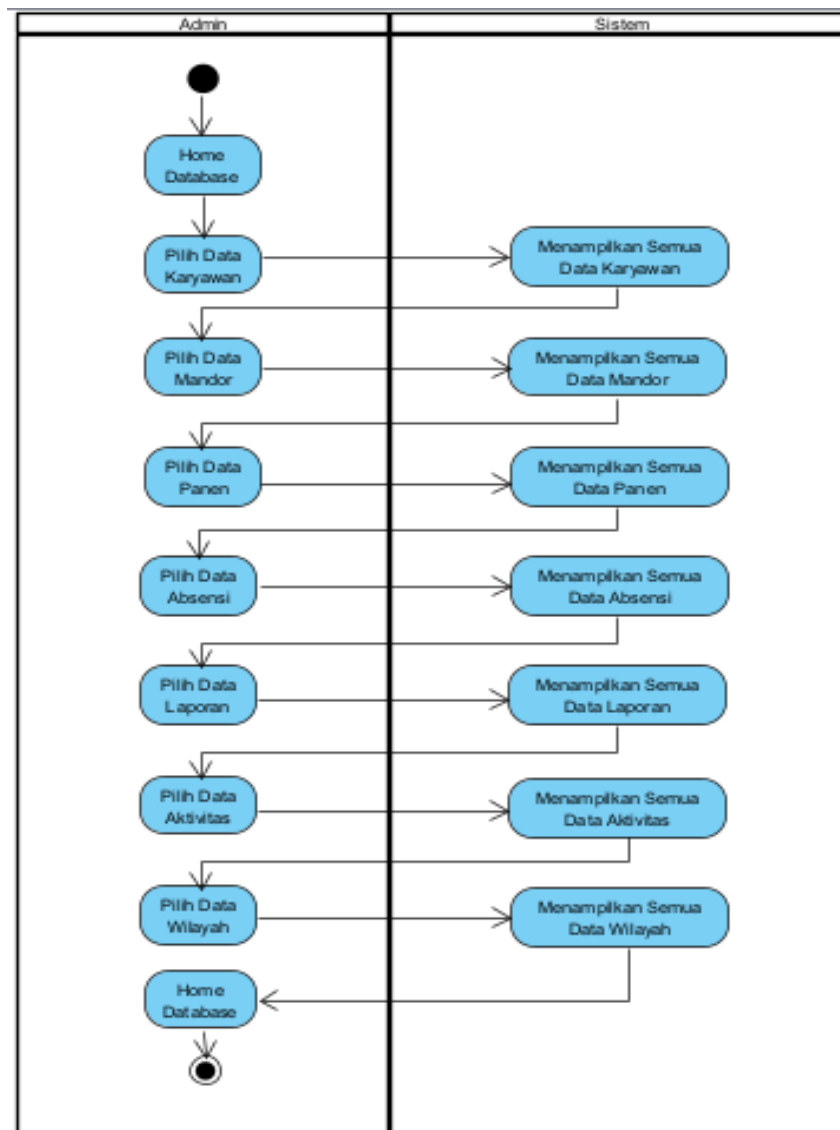


Gambar4.30 Activity Diagram Laporan Hasil Panen (Mandor)

Pada gambar diatas adalah proses melakukan pencetakan laporan bulanan hasil panen karyawan pada aplikasi mandor, pertama-tama mandor memilih menu

laporan setelah itu sistem akan menampilkan nama karyawan setelah itu mandor akan memilih nama karyawan tersebut setelah itu mandor akan memasukkan tanggal awal dan akhir pengecekan laporan panen karyawan setelah itu sistem akan memproses laporan panen lalu sistem akan mengkonvert ke pdf lalu mandor masuk ke tampilan laporan panen pdf.

10. Activity Diagram Admin



Gambar 4.31 Activity Diagram Admin

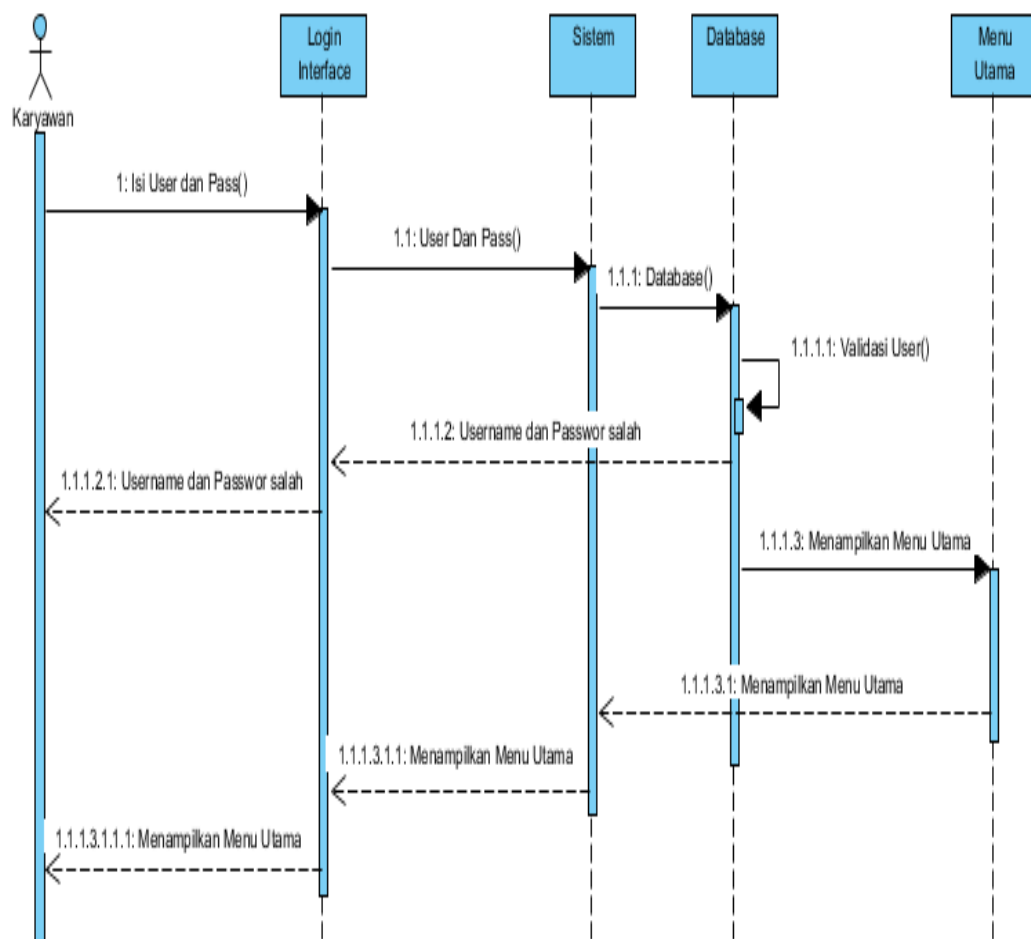
Pada gambar diatas adalah *expositionsadministrator* dimana *administrator* dapat melihat semua *information* yang ada di data set. Pertama *administrator* membuka data set, setelah itu *administrator* masuk ke *informationbasehome* atau

menu data set utama, setelah itu *administrator* memilih data set yang ingin dilihat setelah itu sistem akan menampilkan datanya, *administrator* memilih data set karyawan setelah itu sistem akan menampilkan *information* pegawai.

4.4.1.3 Sequential Diagram

Berikut ialah beberapa diagram *sequential* yang akan digunakan dalam pengembangan aplikasi ini:

1. Sequential Login (Karyawan dan Mandor)

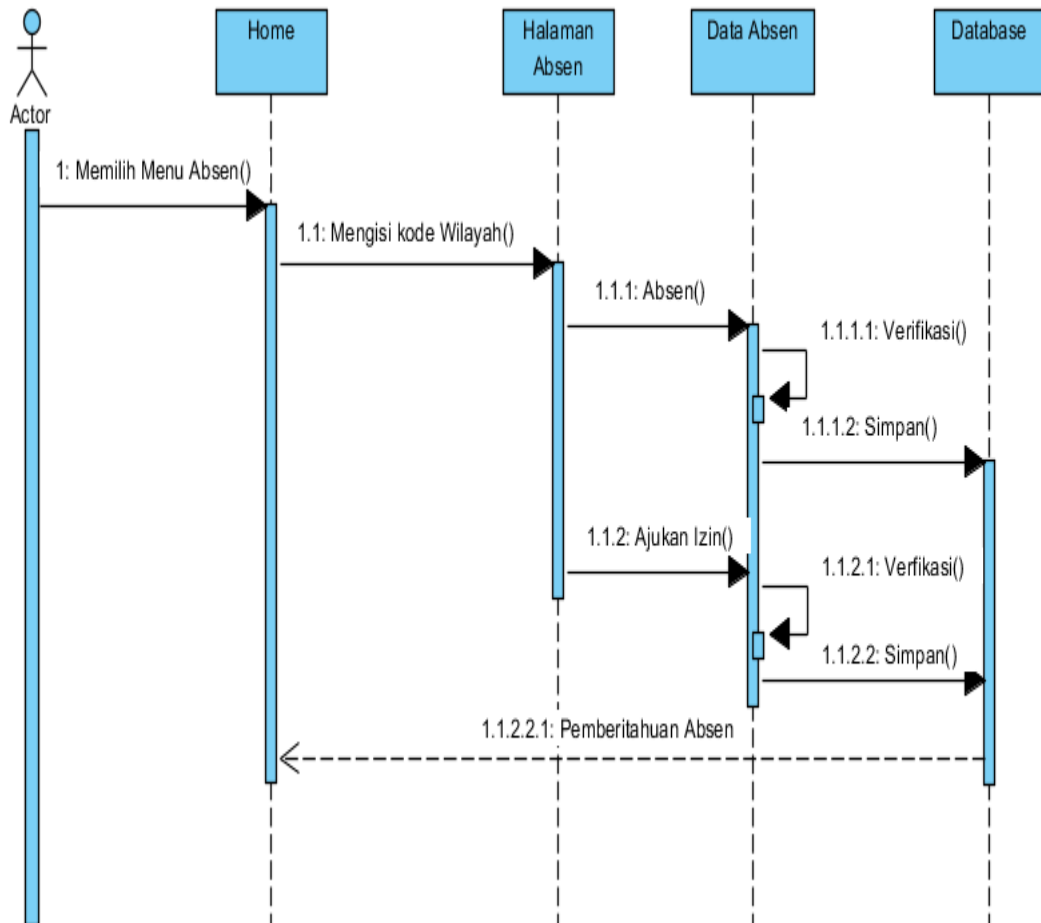


Gambar 4.32 Sequential Login (Karyawan dan Mandor)

Pada gambar di atas merupakan rangkaian antara objek saat karyawan melakukan *login*. Dimulai dengan karyawan membuka *login* lalu mengisi *username* dan *password* setelah itu sistem akan memproses data *login* ke *database* apa bila terdaftar maka akan terverifikasi masuk ke tampilan menu

utama, apabila tidak terferifikasi data didatabase maka akan kembali ke logininterface.

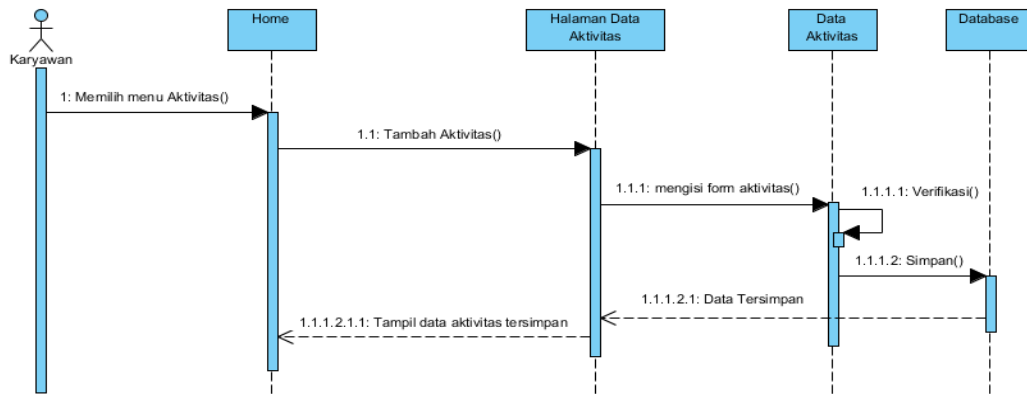
2. Sequential Absen (karyawan)



Gambar 4.33 Sequential Absen (Karyawan)

Pada gambar diatas menjelaskan *sequential* dari absensi karyawan yang dilakukan karyawan, yang dimulai dengan memilih menu absen, lalu akan beralih ke tampilan absenselanjutnya karyawan mengisi kode wilayah mereka berada untuk titik koordinat lokasi karyawan akan otomatis terdeteksi apabila karyawan mengidupkan *GPS* mereka di *android* mereka setelah itu menekan absen atau ajukan ijinlalu *database* akan memverifikasi data tersebut untuk disimpan. Setelah itu sistem akan memberitahu bahwasannya sudah berhasil melakukan absen

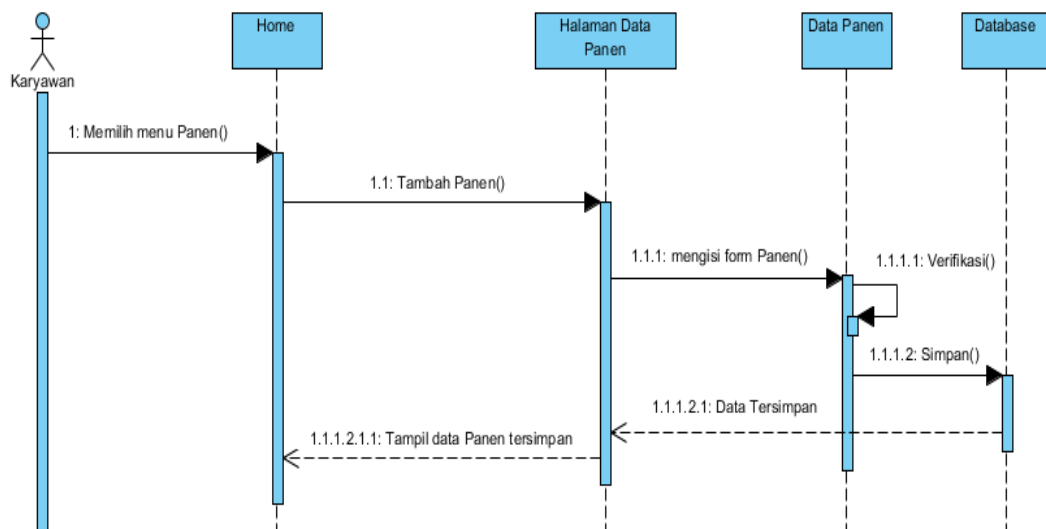
3. Sequential Aktivitas (Karyawan)



Gambar 4.34 *Sequential* Aktivitas (Karyawan)

Diagram diatas merupakan diagram *sequential* dari penginputan data kegiatan aktivitas karyawan yang dilakukan oleh karyawan, yang di mulai dari kegiatan memilih menu aktivitas setelah itu akan beralih ke tampilan aktivitas, setelah itu karyawan di suruh melakukan tambah aktivitas lalu sistem akan beralih ke tambah aktivitas lalu karyawan wajib menguploadfoto aktivitas mereka di lapangan, lalu menulis keterangan mereka sedang melakukan kegiatan apa, setelah itu *database* akan memverifikasi data tersebutkemudian simpan datanya maka sistem akan menampilkan *information* yang telah ditambahkan.

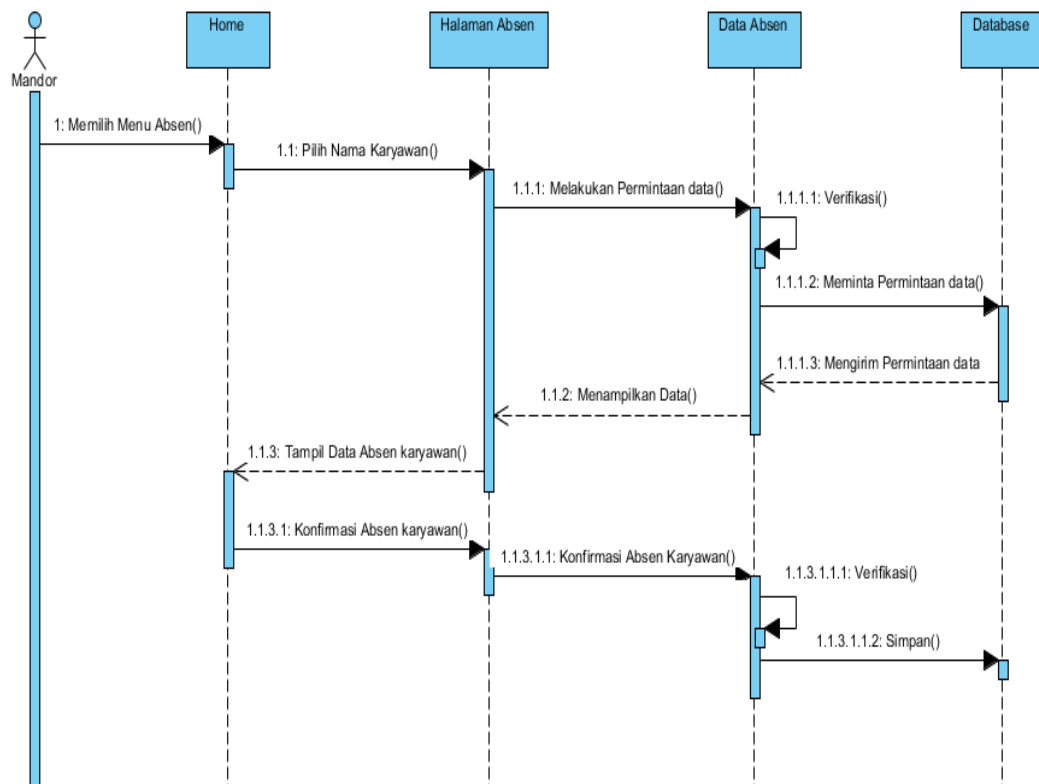
4. *Sequential* Hasil Panen (Karyawan)



Gambar 4.35 *Sequential* Hasil Panen (Karyawan)

Diagram diatas merupakan diagram *sequential* dari penginputan data kegiatan panen karyawan yang dilakukan oleh karyawan, yang di mulai dari kegiatan memilih menu panen setelah itu akan beralih ke tampilan panen, setelah itu karyawan di suruh melakukan tambah aktivitas lalu sistem akan beralih ke tambah panen lalu karyawan wajib menguploadfoto timbangan panen mereka di lapangan, lalu menulis keterangan dan kuantitas timbangan, setelah itu *database* akan memverifikasi data tersebutkemudian simpan datanya maka sistem akan menampilkan *information* yang telah ditambahkan.

5. *Sequential* Master Absen Karyawan (Mandor)

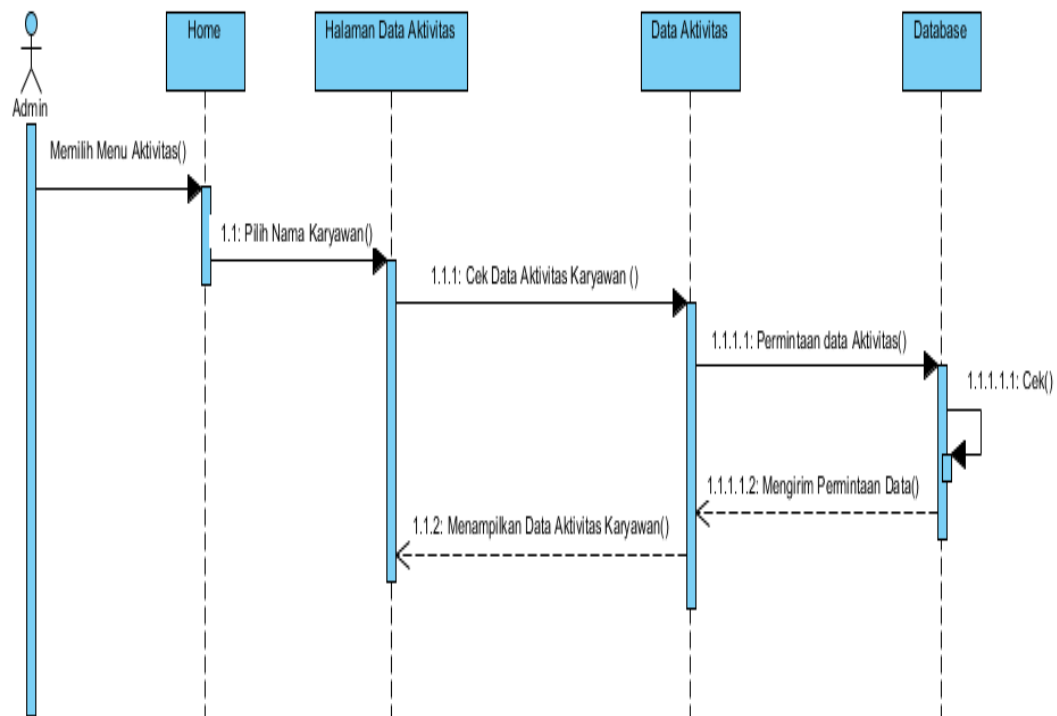


Gambar 4.36 *Sequential* Master Absensi Karyawan (Mandor)

Pada gambar diatas menjelaskan *sequential* dari laporan absensi kehadiran karyawan yang di lakukan oleh mandor, yang di mulai dari kegiatan memilih menu master absensi lalu sistem akan beralih kemaster absensi, setelah itumandor akan masuk ke halaman master absen, lalumandor akan memilih karyawan yang sudah melakukan absensi untuk mengkonfirmasi kehadiran mereka, setelah

karyawan memilih karyawan sistem akan melakukan permintaan data ke *database* lalu sistem memverifikasi data tersebut lalu *database* akan mengirim permintaan data tersebut lalu sistem akan menampilkan absensi karyawan, selanjutnya mandor akan mengecek absensi karyawan, lalu mandor mengkonfirmasi data absen tersebut data tersebut diverifikasi untuk di simpan ke *database*.

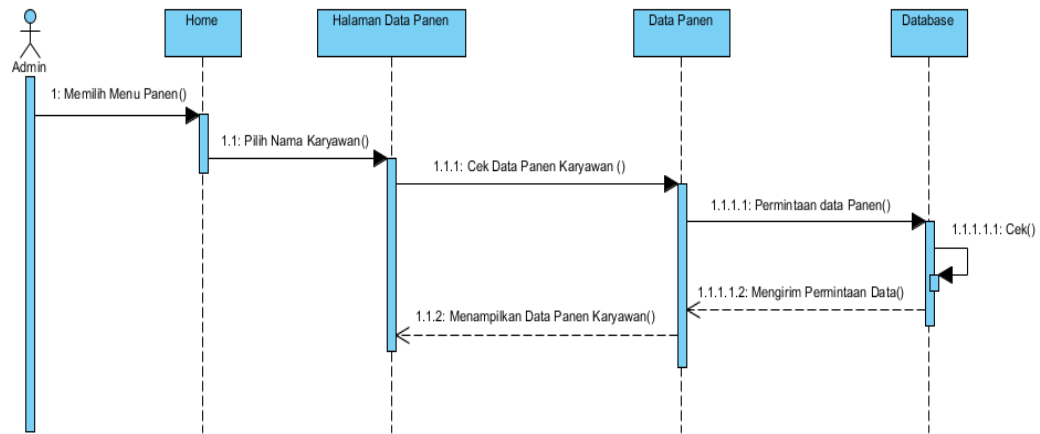
6. *Sequential* Master Aktivitas Karyawan (Mandor)



Gambar 4.37 *Sequential* Master Aktivitas (Mandor)

Pada gambar di atas menjelaskan *Sequential* dari laporan data aktivitas karyawan yang dilakukan oleh mandor, yang di mulai dengan memilih menu master aktivitas lalu beralih ke master aktivitas, lalu mandor memilih nama karyawan yang ingin di lihat kegiatannya di lapangan, setelah itu sistem akan meminta *information* pada data set kemudian data set akan mengecek *information* pegawai kemudian mengirimkan permintaan *information* ke halaman aktivitas, setelah itu data tersebut akan menampilkan data karyawan tersebut ke halaman data aktivitas.

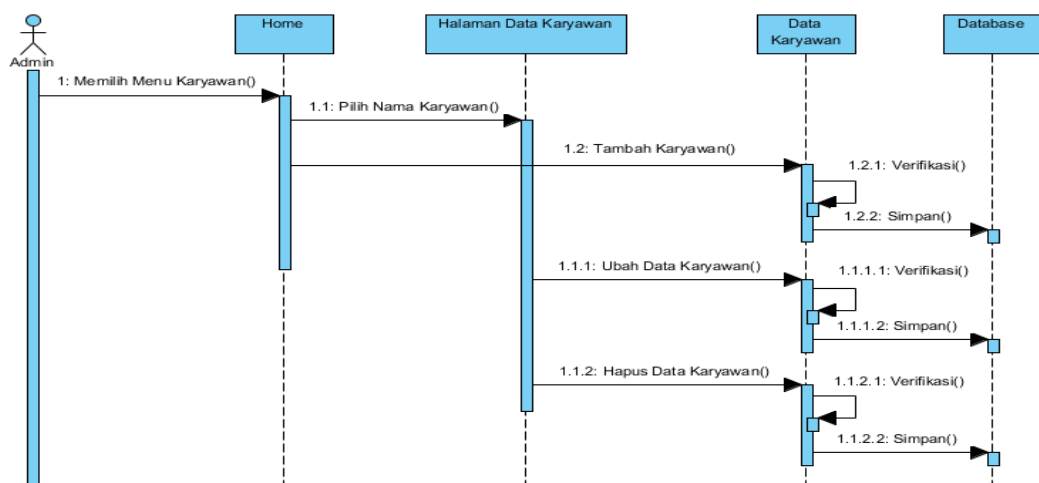
7. *Sequential* Master Panen (Mandor)



Gambar 4.38 *Sequential* Master Panen (Mandor)

Pada gambar di atas menjelaskan *Sequential* dari laporan data panen karyawan yang dilakukan oleh mandor, yang di mulai dengan memilih menu master panen lalu beralih ke master panen, lalu mandor memilih nama karyawan yang ingin di lihat hasil panennya, kemudian sistem akan melakukan permintaan data pada *database* lalu *database* akan mengecek data panen karyawan tersebut lalu mengirim permintaan data ke halaman data panen, setelah itu data tersebut akan menampilkan data panen karyawan tersebut ke halamn data master panen.

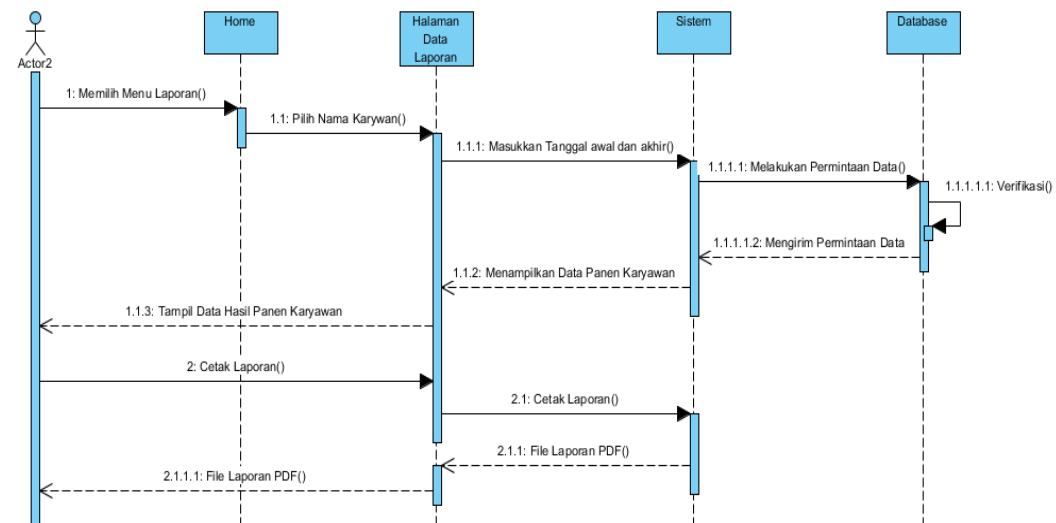
8. *Sequential* Master Karyawan (Mandor)



Gambar 4.39 *Sequential* Master Karyawan (Mandor)

Pada diagram diatas menjelaskan *sequential* dari penambahan,ubah dan hapus data karyawan yang di lakukan oleh mandor, yang di mulai dengan memilih menu master karyawan, lalu beralih ke master karyawan, setelah itu menekan tombol tambah data karyawan, lalu beralih ke tambah data karyawan setelah itu mengisi data karyawan, lalu data tersebut akan di verifikasi ke *database* untuk di simpan, pada bagian ubah dan hapus data karyawan, yg pertama mandor memilih karyawan yang akan diubah atau di hapus datanya lalu akan di verifikasi *database* untuk disimpan.

9. *Sequential* Laporan (Mandor)

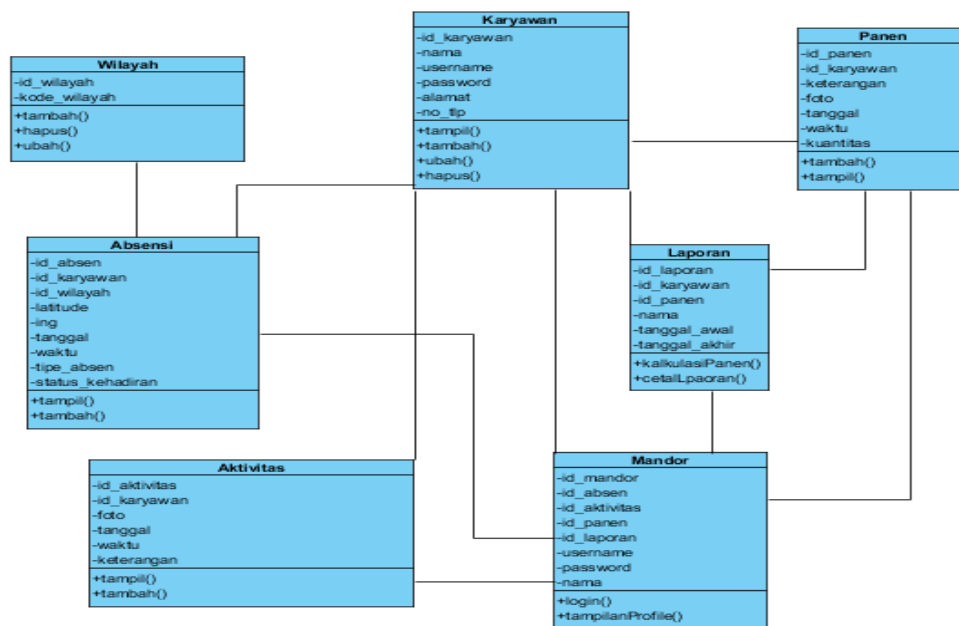


Gambar 4.40 *Sequential* Laporan (Mandor)

Diagram diatas merupakan diagram *Sequential* dari proses laporan data hasil panen karyawan yang bisa di lakukan oleh mandor. Yang dimulai dari menekan tombol laporan setelah itu akan beralih ke laporan, setelah itu memilih nama karyawan yang ini di cetak laporan panenanya, lalu masukkan tanggal awal dan akhir panen, setelah itu sistem melakukan permintaan ke *database* lalu *database* mengirim permintaan data kemudian menampilkan data yang ada, lalu sistem akan memberikan informasi tersebut ke pada mandor, selain itu mandor juga bisa melakukan kegiatan cetak laporan dan sistem akan mengolah data yang ada dalam bentuk file berektensi PDF.

4.4.1.4 Classdiagram

Berikut ini adalah rancangan *classdiagram* untuk sistem yang dibangun :



Gambar 4.41 *Classdiagram*

Diagram diatas menjelaskan tentang suatu hubungan anatar sebuah objek-objek yang akan diimplementasikan kode. Menunjuk dari dua aktor dalam *usecase* yaitu mandor dan karyawan, dapat dilihat bahwa karyawan dapat memiliki lebih dari satu aktivitas dan mandor hanya memiliki satu objek berupa kumpulan data laporan.

4.4.2 Desain Database

Berikut adalah desain *database* yang akan digunakan dalam aplikasi data mining mengetahui Aktivitas Karyawan yang berada di lapangan.

a. Tabel Absensi

Nama table : tb_ptpn_absensi

Primary key : id_absensi

Tabel 4.2 *Database Absensi*

No	Nama Field	Type Data	Ukura n
1	id_absensi	int	11
2	id_karyawan	int	11

3	id_wilayah	int	11
4	latitude	varchar	30
5	Ing	varchar	30
6	tanggal	date	30
7	waktu	varchar	100
8	tipe absen	enum (“absensi”, ”pengajuan”)	
9	status_kehadiran	varchar	30

b. TabelAktivitas

Nama table : tb_ptpn_aktifitas_harian

Primary key : id_aktifitas_harian

Tabel 4.3Database Aktivitas

No	Nama Field	Type Data	Ukuran
1	id_aktifitas_harian	int	11
2	id_karyawan	int	11
3	file_foto	varchar	50
4	tanggal	date	
5	waktu	varchar	20
6	keterangan	varchar	50

c. TabelHasil Panen

Nama table : tb_ptpn_hasil_panen

Primary key : id_hasil_panen

Tabel 4.4Database Hasil Panen

No	Nama Field	Type Data	Ukuran
1	id_hasil_panen	int	11
2	id_karyawan	int	11
3	keterangan	varchar	50
4	file_foto	varchar	50
5	tanggal	date	
6	waktu	varchar	50
7	kuantitas	int	11

d. TabelKaryawan

Nama table : tb_ptpn_karyawan

Primary key : id_karyawan

Tabel 4.5Database Karyawan

No	Nama Field	Type Data	Ukuran
----	------------	-----------	--------

1	id_karyawan	<i>int</i>	11
2	nama	<i>varchar</i>	30
3	<i>username</i>	<i>varchar</i>	30
4	<i>password</i>	<i>varchar</i>	30
5	alamat	<i>varchar</i>	225
6	no_telepon	<i>varchar</i>	50

e. TabelAdmin

Nama table : tb_ptpn_admin

Primary key : id_admin

Tabel 4.6Database Admin

No	Nama Field	Type Data	Ukuran
1	id_admin	<i>int</i>	11
2	<i>username</i>	<i>varchar</i>	30
3	<i>password</i>	<i>varchar</i>	30
4	nama	<i>varchar</i>	40

f. TabelWilayah

Nama table : tb_ptpn_wilayah

Primary key : id_wilayah

Tabel 4.7Database Wilayah

No	Nama Field	Type Data	Ukuran
1	id_wilayah	<i>int</i>	11
2	kode_wilayah	<i>varchar</i>	30

4.4.3 Design Interface

Dalam Subbab ini menjelaskan bagaimana tampilan visual yang akan diterapkan dalam sebuah sistem yang disesuaikan dengan berbagai perangkat sehingga pengguna mudah dan nyaman untuk digunakan website atau mobile., *Desain interface* (antarmuka aplikasi) disusun menggunakan *wirefarm*, secara singkat *Wirefarm* adalah kerangka atau susunan untuk mengatur *thing* di situs web atau halaman seluler sebelum *compositionsdesain* yang sebenarnya dimulai.

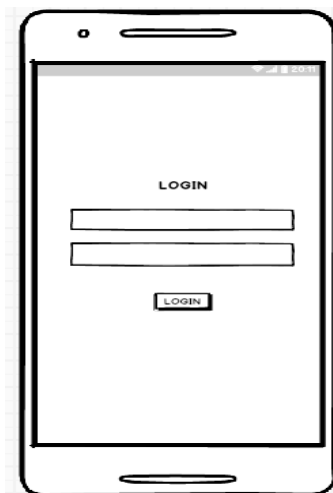
Pada pengembangan aplikasi berbasis *android* ada beberapa komponen penting yang disebut *View* atau *Widget* adapun beberapa komponen tersebut adalah sebagai berikut :

Tabel 4.8 Macam-Macam Widget *Android*

Nama Widget	Fungsi
<i>TextView</i>	Merupakan salah satu <i>View</i> atau <i>widget</i> yang berfungsi untuk menampilkan sebuah tes, baik itu <i>individualizedstructure</i> atau kalimat, paragraf dan lain-lain yang digunakan untuk pengembangan aplikasi.
<i>EditText</i>	Merupakan suatu komponen yang berfungsi untuk memasukkan suatu <i>text</i> yang diperlukan ke dalam sebuah sistem.
<i>ImageView</i>	Merupakan suatu komponen yang menampilkan sebuah gambar kepada para pengguna sistem.
<i>Button</i>	Merupakan suatu komponen tindakan atau aksi, yang dimana <i>user</i> bisa memberikan sebuah perintah kepada sistem sesuai dengan kebutuhannya.
<i>RecyclerView</i>	Merupakan suatu komponen yang menampilkan data dalam bentuk daftar yang dapat diubah sesuai kebutuhan.
<i>FrameLayout</i>	Merupakan suatu komponen yang saling menimpah misalkan memasukkan komponen tombol ke dalam komponen gambar.

1 .Desain *Interface Input*

a) Tampilan *Login*(Karyawan)

**Gambar 4.42** *Login*(Karyawan)

Keterangan :

- a. Tampilan *Login*
- b. Karyawan menginputkan *Username*
- c. Karyawan Menginputkan *Password*
- d. Klik *buttonlogin* untuk masuk kedalam menu utama

b) Tampilan Absensi (Karyawan)



Gambar4.43 Absensi (Karyawan)

Keterangan :

- a. Tampilan absen karyawan
- b. Karyawan diwajibkan menhidupkan titik lokasi di hp mereka untuk mengetahui titik lokasi mereka berada
- c. Karyawan menginputkan kode wilayah mereka

- d. Klik *button* absen untuk melakukan absen klik *button* ajukan ijin untuk melakukan pengajuan ijin tidak masuk

c) Tampilan Tambah Aktivitas

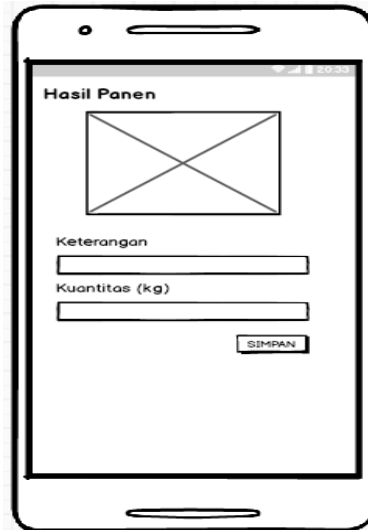


Gambar 4.44 Tampilan Tambah Aktivitas (Karyawan)

Keterangan :

- Tampilan tambah aktivitas karyawan.
- Karyawann mengklik foto untuk menguploadkegiatan aktivitas karyawan.
- Karywan memfoto kegiatan mereka lalu mengkonfirmasi foto mereka.
- Karyawan menginputkan keterangan mereka sedang melakukan kegiatan.
- Klik *button* simpan untuk menyimpan kegiatan aktivitas karyawan ke database

d) Tampilan Tambah Hasil Panen (Karyawan)

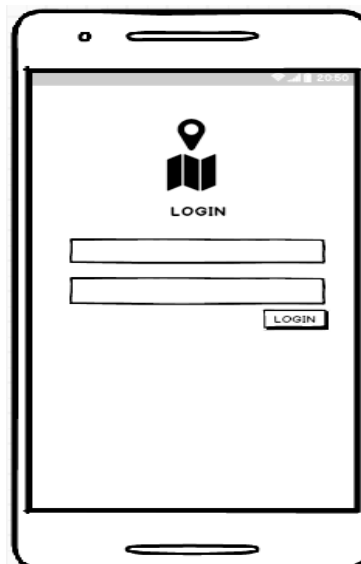


Gambar 4.45 Tampilan Tambah Panen (Karyawan)

Keterangan :

- a. Tampilan tambah hasil panen
- b. Karyawan mengklik foto untuk mengupload hasil timbangan karyawan
- c. Karyawan menginput keterangan panen
- d. Karyawan menginput satuan timbangan mereka berbentuk kg
- e. Klik *button* simpan untuk menyimpan data hasil panen karyawan

e) Tampilan *Login*(Mandor)

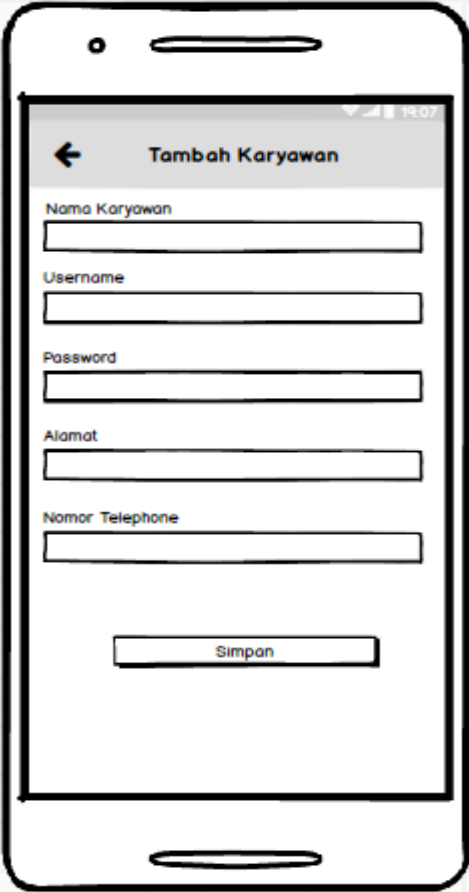


Gambar 4.46 Tampilan *Login*(Mandor)

Keterangan :

- a. Tampilan *Login* mandor
- b. Mandor menginputkan *username*
- c. Mandor menginputksn *password*
- d. Klik *buttonlogin* untuk masuk ke menu utama

f) Tampilan Tambah Karyawan (Mandor)

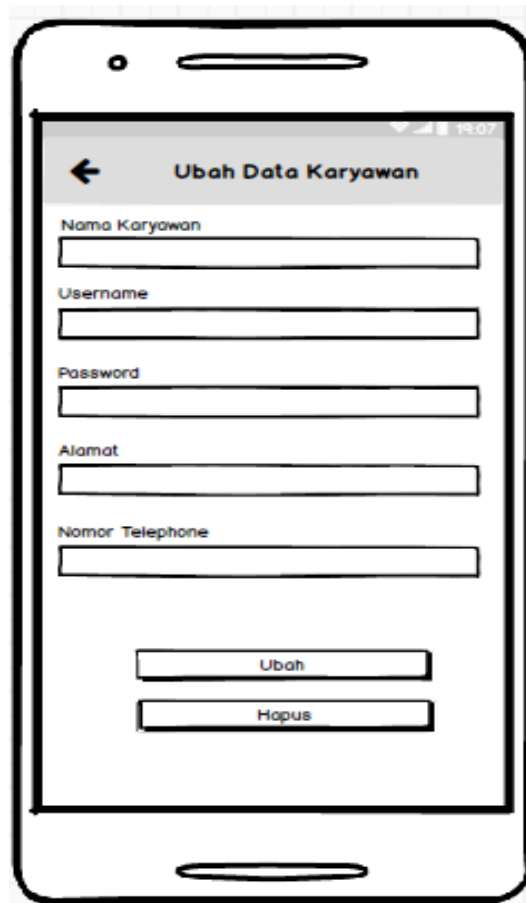


Gambar 4.47 Tampilan Tambah Karyawan (Mandor)

Keterangan:

- a. Tampilan tambah karyawan pada aplikasi mandor
- b. Mandor menginputkan nama karyawan
- c. Mandor menginputkan *username*
- d. Mandor menginputkan *password*
- e. Mandor menginputkan alamat

- f. Mandor menginputkan no tlp
 - g. Klik *button* simpan untuk menyimpan data
- g) Tampilan Ubah dan Hapus Data Karyawan(Mandor)



The image shows a mobile application interface for editing and deleting employee data. The screen is titled "Ubah Data Karyawan" (Edit Employee Data) and features a back arrow on the left. Below the title, there are five input fields for "Nama Karyawan" (Employee Name), "Username", "Password", "Alamat" (Address), and "Nomor Telephone" (Phone Number). At the bottom of the form, there are two buttons: "Ubah" (Edit) and "Hapus" (Delete).

Gambar 4.48 Tampilan Ubah dan Hapus Data Karyawan (Mandor)

Keterangan :

- a. Tampilan Ubah dan hapus data karyawan
- b. Mandor menginput apa yang mau di ubah
- c. Klik *button* ubah untuk mengubah data karyawan dan menyimpannya ke database
- d. Klik *button* hapus untuk menghapus data karyawan dan menghapus data tersebut di *database*

h) Tampilan Cetak Laporan (Mandor)



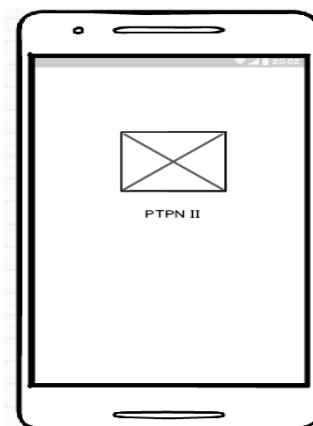
Gambar 4.49 Tampilan Cetak Laporan (Mandor)

Keterangan :

- a. Tampilan cetak laporan panen
- b. Mandor menginput tanggal awal panen karyawan
- c. Mandor menginput tanggal akhir panen karyawan
- d. Klik *button* cetak laporan untuk melihat data laporan panen karyawan sesuai tanggal awal dan akhir

2DesainInterface Output

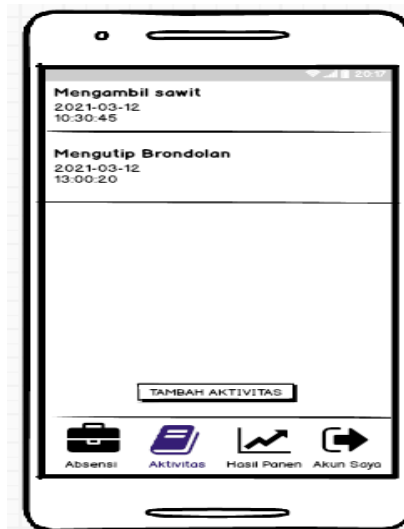
a) *Splash Screen*(Karyawan)



Gambar 4.50*Splash Screen*(Karyawan)

Pada gambar diatas adalah gambar awal pada saat membuka aplikasi akan menampilkan *Splash Screen* atau layar percikan sebelum masuk ke *formlogin* yaitu sebuah gambar untuk membuat lebih menarik.

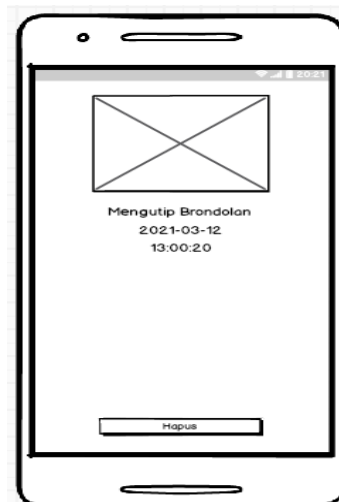
b) Tampilan Aktivitas(Karyawan)



Gambar 4.51 Tampilan Aktivitas (Karyawan)

Pada gambar diatas adalah tampilan data aktivitas karyawan yang di mana karyawan sudah menambahkan kagiatan aktivitas mereka dengan cara melakukan *upload* kegiatan foto dan keterangan kegiatan

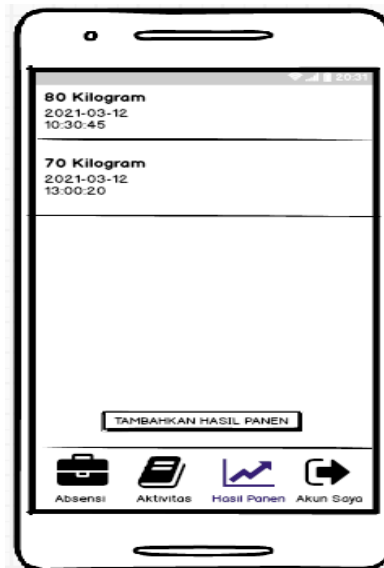
c) Tampilan Ket Aktivitas (Karyawan)



Gambar 4.52 Tampilan Ket Aktivitas (Karyawan)

Pada gambar diatas adalah tampilan *Interfacedetail* aktivitas karyawan yang di lengkapi dengan foto aktivitas karyawan dan keterangan aktivitas karyawan di lengkapi dengan tanggal dan waktu *upload* aktivitas secara otomatis.

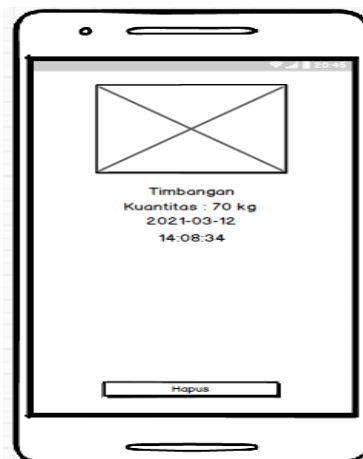
d) Tampilan Panen (Karyawan)



Gambar 4.53 Tampilan Panen (Karyawan)

Pada gambar diatas adalah tampilan data hasil panen karyawan yang sudah melakukan *upload* data akan tampil seperti gambar diatas yang lengkap dengan satuan timbangan tanggal dan waktu data tersebut sudah tersimpan *kedatabase*.

e) Tampilan Ket Hasil Panen (Karyawan)



Gambar 4.54 Tampilan Ket Hasil Panen (Karyawan)

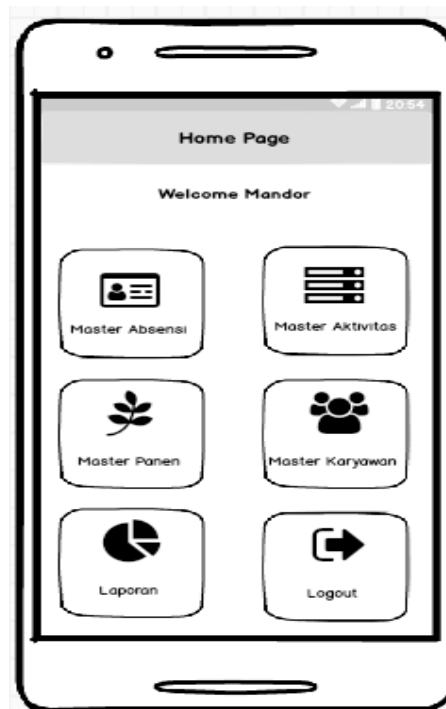
Pada gambar diatas adalah tampilan detail hasil panen karyawan lengkap dengan foto, keterangan dan kuantitas hasil panen karyawan dan dilengkapi tanggal dan waktu secara otomatis.

f) Tampilan Menu *Logout*(Karyawan)

Gambar 4.55 Tampilan Menu *Logout*(Karyawan)

Pada gambar di atas ialah tampilan *logout*, apa bila karyawan selesai melakukan absen,aktivitas, lapor hasil panen maka mereka *logout* aplikasi tersebut.

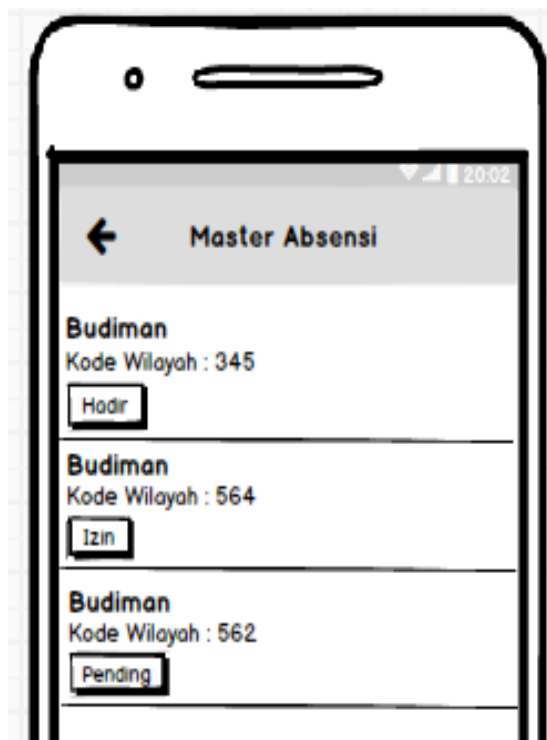
g) Tampilan Menu (Mandor)



Gambar 4.56 Tampilan Menu (Mandor)

Pada gambar diatas ialah gambar tampilan menu pada aplikasi mandor yang terdapat, master absensi ialah pengecekan kehadiran karyawan, pada master aktivitas ialah pengecekan atau memonitoring aktivitas karyawan sedang melakukan apa, pada bagian master panen mandor dapat melihat data timbangan hasil panen mereka agar tidak terjadinya ke tidak jujuran karyawan, pada bagian master karyawan ialah untuk menambah karyawan baru, pada bagian laproan ialah untuk melihat data hasil panen karyawan bulanan dan untuk menu *logout* untuk keluar dari aplikasi tersebut.

h) Tampilan Data Absensi (Mandor)



Gambar 4.57 Tampilan Data Absen (Mandor)

Pada gambar diatas ialah tampilan data absen karyawan yang belum di konfirmasi dan sudah di konfirmasi kehadiran oleh mandor. Apa bila karyawan belum di konfirmasi kehadiran maka akan tampil *text* pending bagi yang sudah di konfirmasi kehadiran maka akan ada *text* pemberitahuan hadir atau izin seperti gambar 4.41 di atas.

i) Tampilan Konfirmasi Absensi (Mandor)



Gambar 4.58 Tampilan Konfirmasi Absen (Mandor)

Pada gambar diatas ialah tampilan konfirmasi absensi kehadiran karyawan apabila karyawan tepat berada di titik lokasi kerja mereka maka mandor akan mengkonfirmasi kehadiran karyawan.

j) Tampilan Data Aktivitas (Mandor)



Gambar 4.59 Tampilan Data Aktivitas (Mandor)

Pada gambar di atas adalah tampilan data aktivitas karyawan pada aplikasi mandor, di data tersebut akan menampilkan data data aktivitas karyawan setiap melakukan pekerjaan.

k) Tampilan Detail Aktivitas (Mandor)



Gambar 4.60 Tampilan Detail Aktivitas

Pada gambar di atas adalah tampilan detail aktivitas karyawan tampilan *admin* lengkap dengan foto dan keterangan aktivitas karyawan dan dilengkapi tanggal dan waktu secara otomatis.

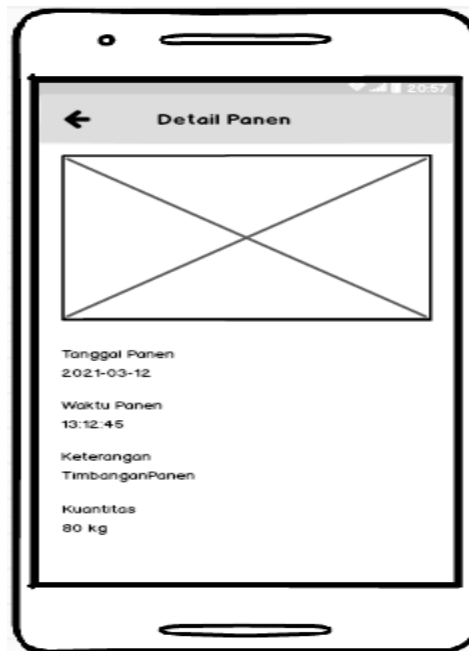
l) Tampilan Data Panen (Mandor)



Gambar 4.61 Tampilan Data Panen (Mandor)

Pada gambar di atas adalah tampilan data hasil panen pada mandor pada aplikasi *admin*, di data tersebut akan menampilkan data hasil panen karyawan setiap melakukan penimbangan panen kelapa sawit.

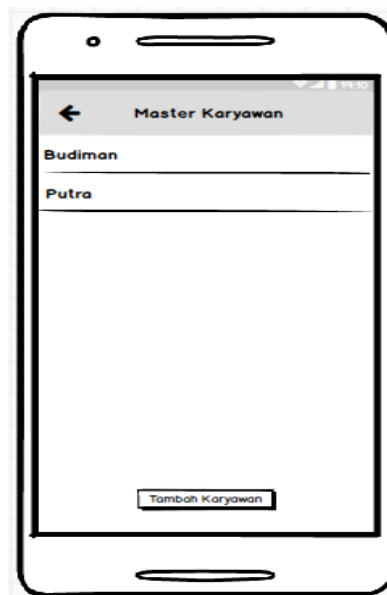
m) Tampilan Detail Panen (Mandor)



Gambar 4.62 Tampilan Detail Panen (Mandor)

Pada gambar diatas adalah tampilan detail hasil panen pada mandor lengkap dengan foto, keterangan dan kuantitas hasil panen karyawan dan di lengkapi tanggal dan waktu secara otomatis.

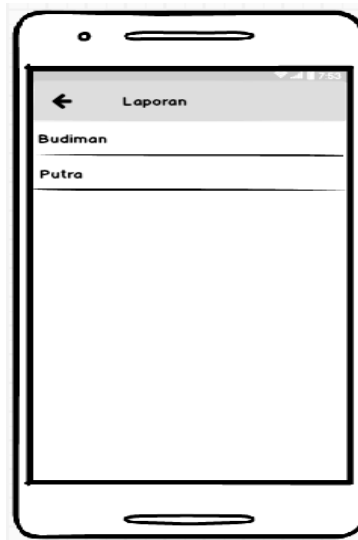
n) Tampilan Data Karyawan (Mandor)



Gambar 4.63 Data Karyawan (Mandor)

Pada gambar diatas adalah tampilan data-data karyawan lapangan yang berada di PTP.Nusnatara II *Perk* Bandar Klippa Bagian Afdeling IV.

o) Tampilan Data Laporan Panen Karyawan (Mandor)



Gambar 4.64 Tampilan Laporan (Mandor)

Pada gambar diatas adalah tampilan laporan yang dimana terdapat nama karyawan yang ingin di cek berapa pendapatan panennya setiap bulan.

p) Tampilan Laporan Panen PDF

Laporan Panen PTPN II

Pada Tanggal // - .s/d. // -

Nama Karyawan :

NO	Tanggal	Kuantitas	Ket

Gambar 4.65 Tampilan Laporan Panen PDF

Pada gambar diatas ialah tampilan laporan hasil panen karyawan tiap bulan yang sudah terbentuk menjadi pdf.

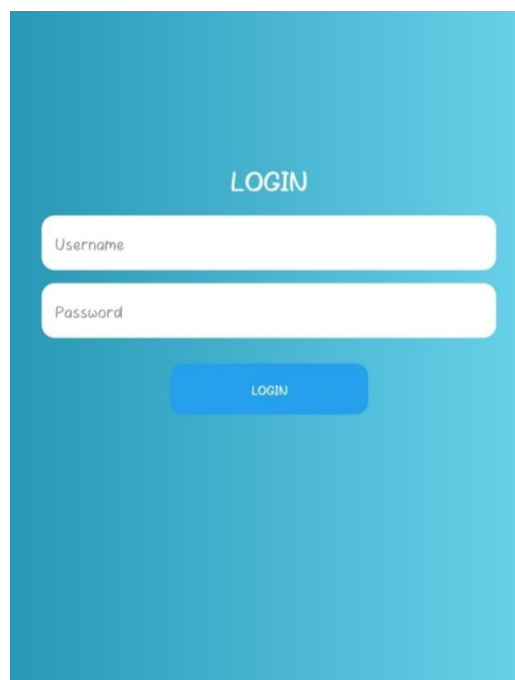
4.5 Implementasi dan Pengujian Sistem

Setelah peneliti melakukan analisis dan perancangan sistem sebelumnya, maka peneliti melakukan tahap implementasi dari perancangan menjadi sebuah sistem *android* menggunakan metode *Lock Gps* yang menggunakan penyimpanan *cloud* yang dapat memonitoring karyawan yang berada dilapangan terutama karyawan bagian panen pada PTP. Nusantara II *Perk* Bandar Klippa.

4.5.1 Implementasi Antarmuka Karyawan

1. Implementasi tampilan *login* karyawan

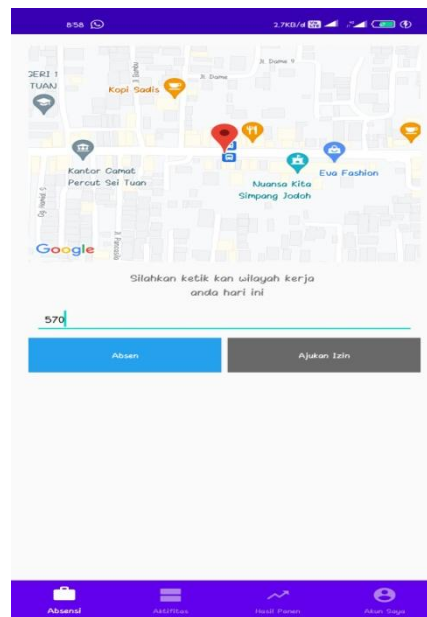
Berikut ini adalah tampilan *Login* tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.66 Implementasi tampilan *login* karyawan

2. Implementasi tampilan menu absen karyawan

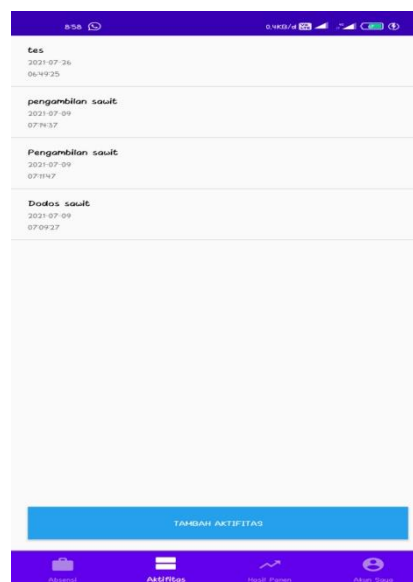
Berikut ini adalah tampilan absensi karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.67 Implementasi tampilan menu absen karyawan

3. Implementasi tampilan menu aktivitas karyawan

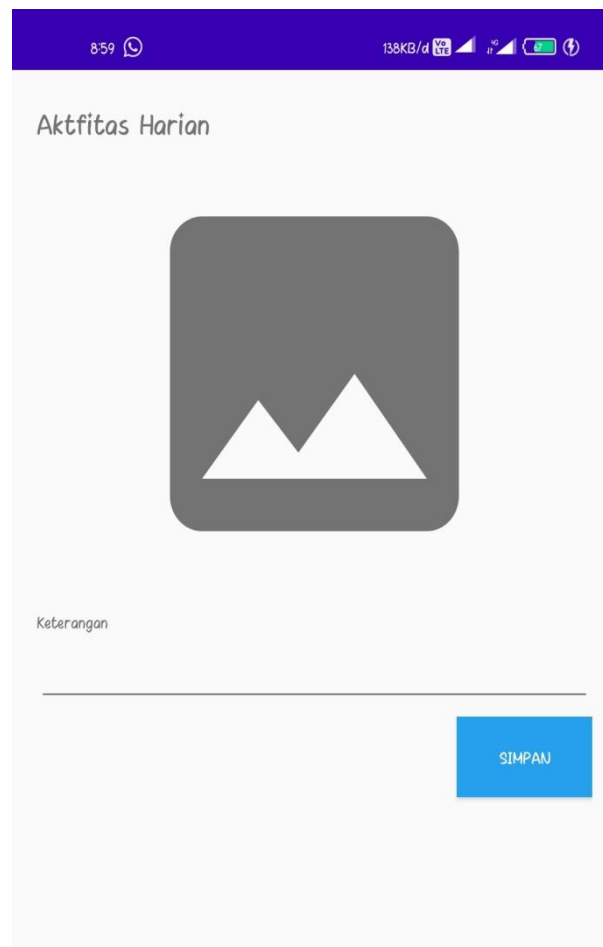
Berikut ini adalah tampilan data aktivitas karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.68 Implementasi tampilan menu aktivitas karyawan

4. Implementasi tampilan *upload* aktivitas karyawan

Berikut ini adalah tampilan *upload* aktivitas karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.69 Implementasi tampilan *upload* aktivitas karyawan

5. Implementasi tampilan detail aktivitas karyawan

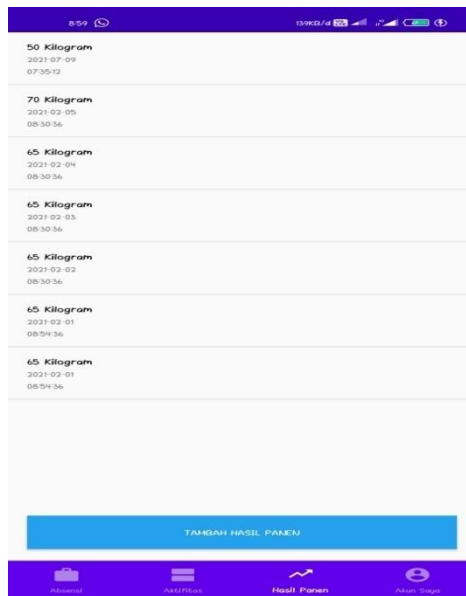
Berikut ini adalah tampilan detail aktivitas karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.70 Implementasi tampilan detail aktivitas karyawan

6. Implementasi tampilan menu panen karyawan

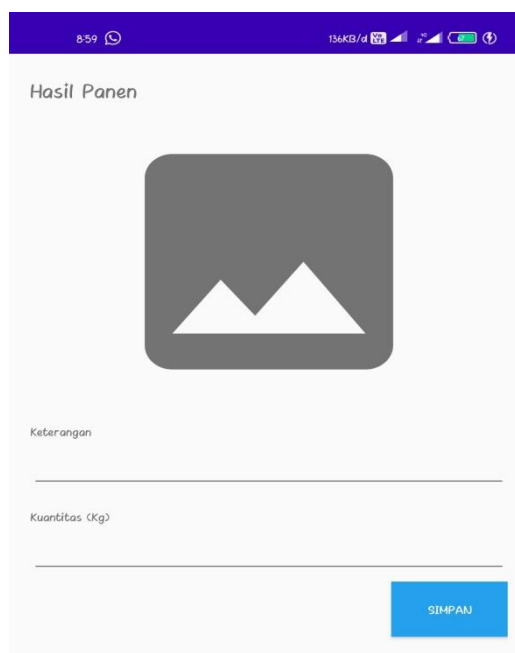
Berikut ini adalah tampilan data hasil panen karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.71 Implementasi tampilan menu panen karyawan

7. Implementasi tampilan *Upload* hasil panen karyawan

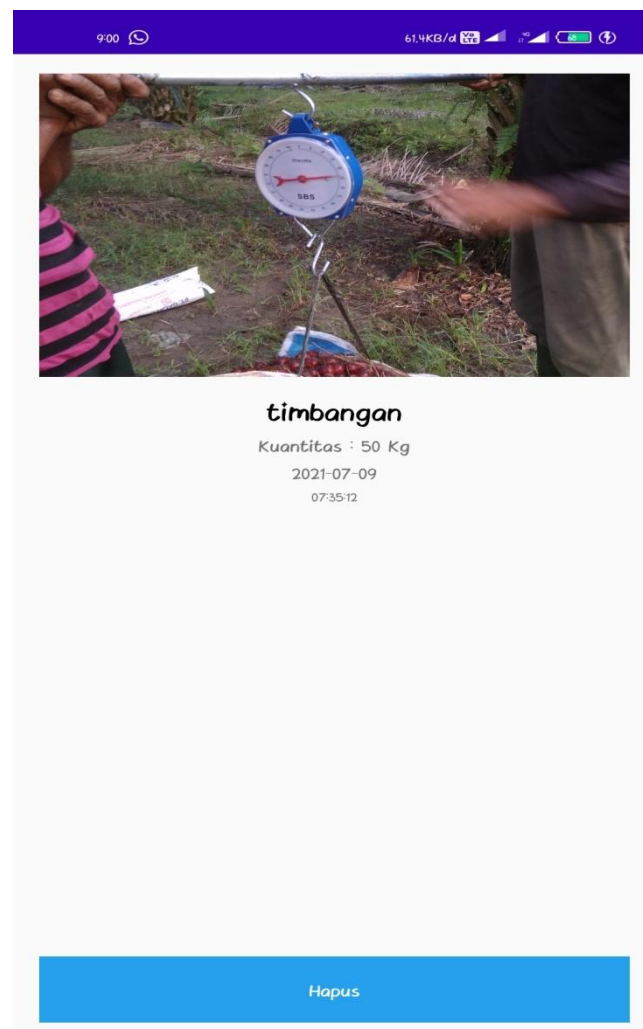
Berikut ini adalah tampilan *upload* hasil panen karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.72 Implementasi tampilan *Upload* hasil panen karyawan

8. Implementasi tampilan detail hasil panen karyawan

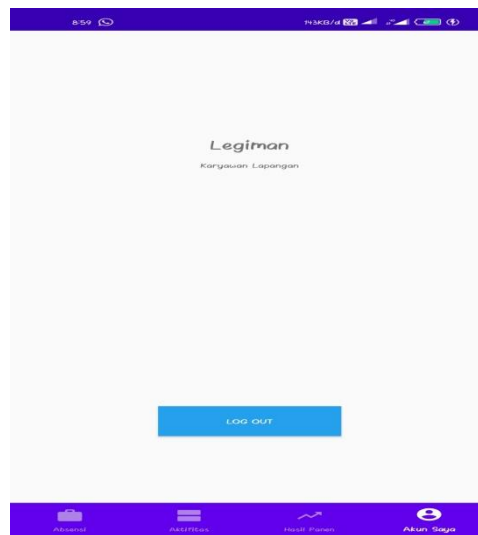
Berikut ini adalah tampilan detail hasil panen karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.73 Implementasi tampilan detail hasil panen karyawan

9. Implementasi tampilan *logout* karyawan

Berikut ini adalah tampilan *logout* karyawan tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:

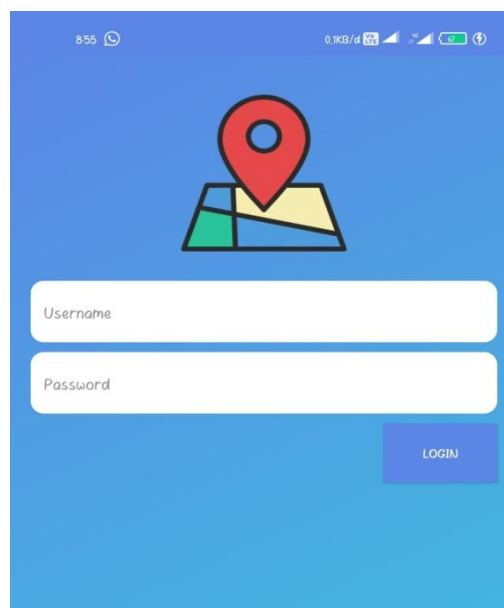


Gambar 4.74Implementasi tampilan *logout* karyawan

4.5.2 Implementasi Antarmuka Mandor

1. Implementasi tampilan *login*(Mandor)

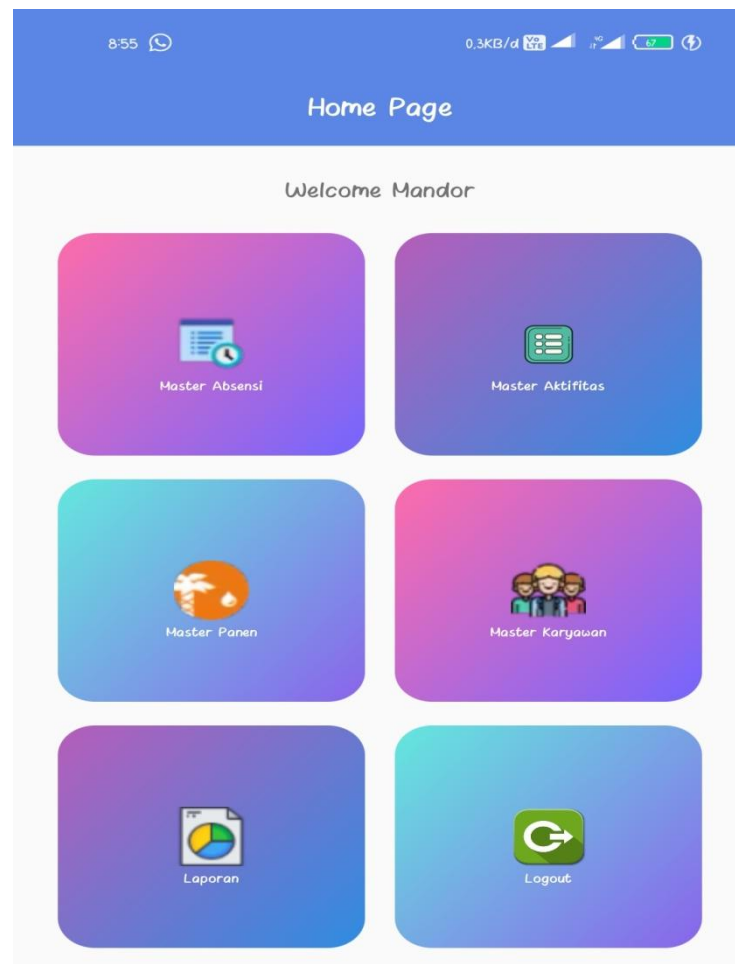
Berikut ini adalah tampilan *login* mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.75Implementasi tampilan *login*(Mandor)

2. Implementasi tampilan menu utama(Mandor)

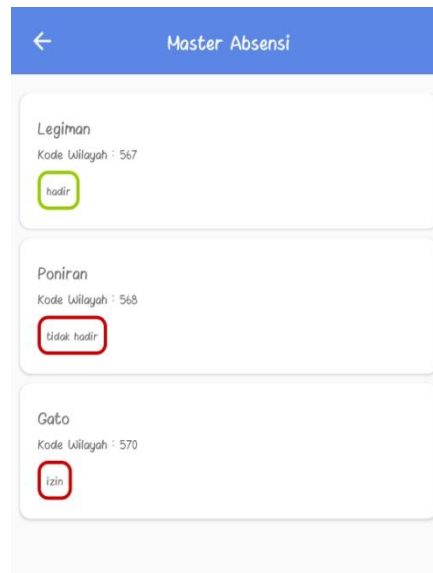
Berikut ini adalah tampilan menu mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.76Implementasi tampilan menu utama(Mandor)

3. Implementasi tampilan menu master absen(Mandor)

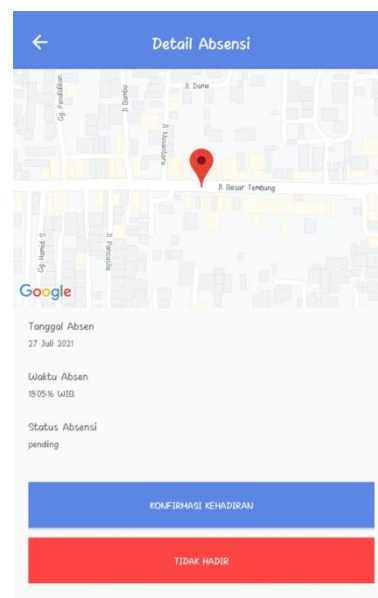
Berikut ini adalah tampilan master absensi mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.77Implementasi tampilan menu master absen(Mandor)

4. Implementasi tampilan detail absen(Mandor)

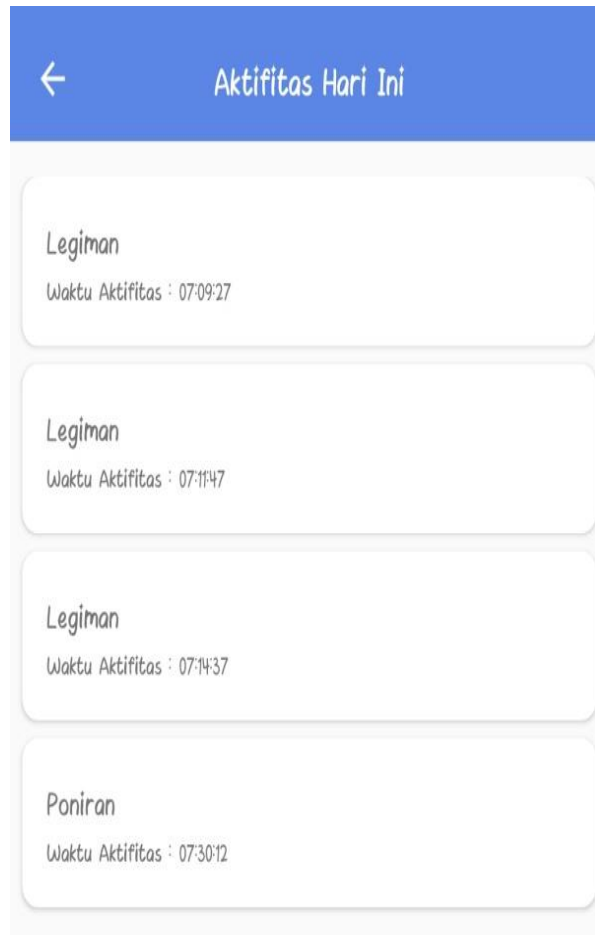
Berikut ini adalah tampilan detail absensi pada mandortentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.78Implementasi tampilan detail absen(Mandor)

5. Implementasi tampilan menu master aktivitas(Mandor)

Berikut ini adalah tampilan data aktivitas karyawan hari ini pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.79Implementasi tampilan menu master aktivitas(Mandor)

6. Implementasi tampilan detail aktivitas(Mandor)

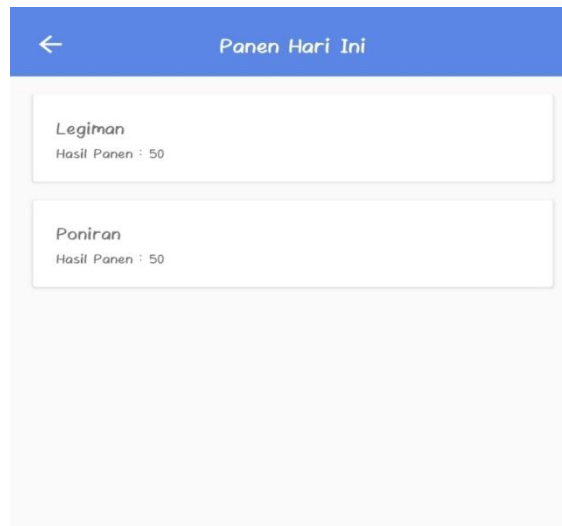
Berikut ini adalah tampilan detail aktivitas karyawan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.80 Implementasi tampilan detail aktivitas(Mandor)

7. Implementasi tampilan menu panen(Mandor)

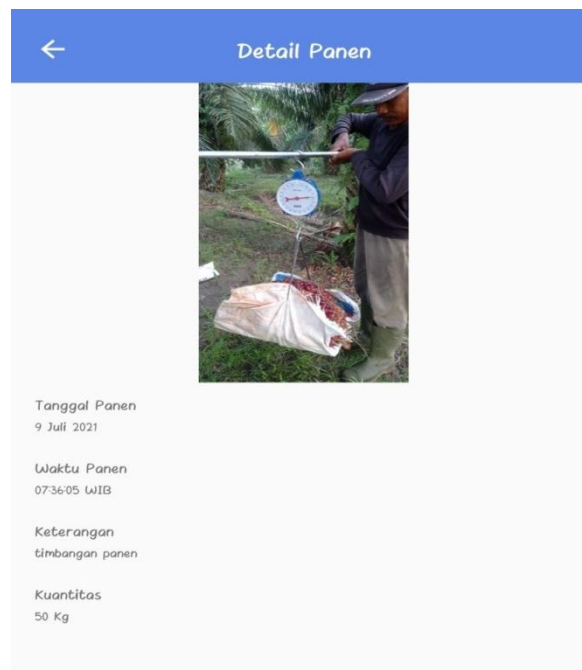
Berikut ini adalah tampilan data panen karyawan hari ini pada mandortentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.81Implementasi tampilan menu panen(Mandor)

8. Implementasi tampilan detail panen(Mandor)

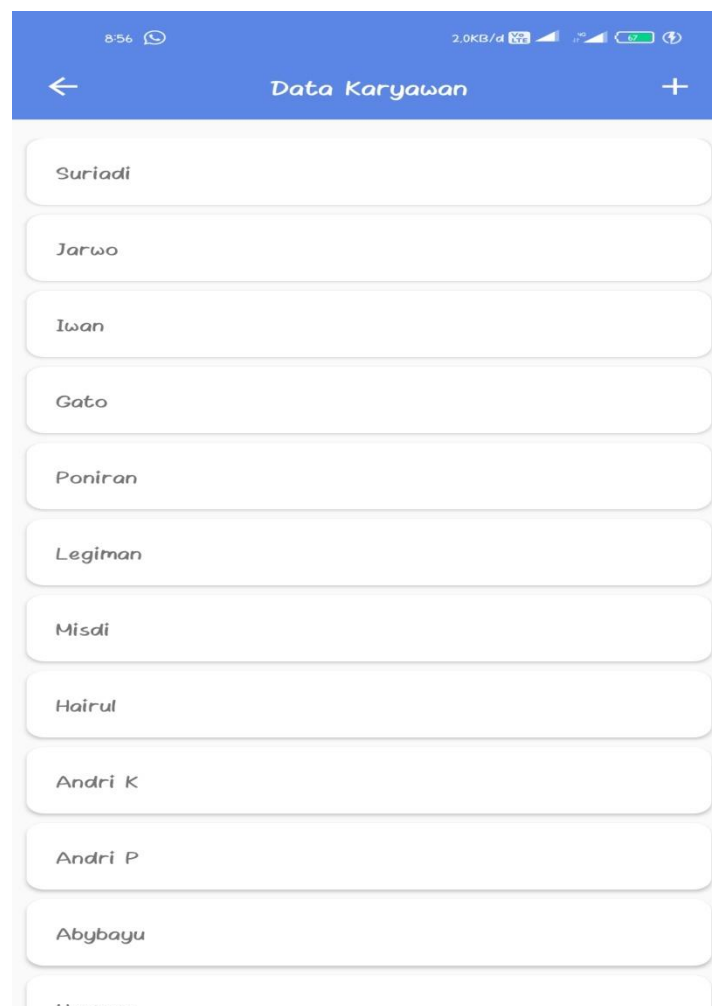
Berikut ini adalah tampilan detail panen karyawan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.82Implementasi tampilan detail panen(Mandor)

9. Implementasi tampilan menu master karyawan(Mandor)

Berikut ini adalah tampilan data karyawan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.83Implementasi tampilan menu master karyawan(Mandor)

10. Implementasi tampilan ubah dan hapus data karyawan(Mandor)

Berikut ini adalah tampilan udah dan hapus data karyawan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:

8:56 0.8KB/d

← Ubah Data Karyawan

Nama Karyawan

Suriadi

Username

Suriadi

Password

suriadi

Alamat

Jln. Batang jambu gg mawar 07

Nomor Telepon

082376187565


Ubah

Hapus

Gambar 4.84 Implementasi tampilan ubah dan hapus data karyawan

11. Implementasi tampilan tambah karyawan(Mandor)

Berikut ini adalah tampilan tambah karyawan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:

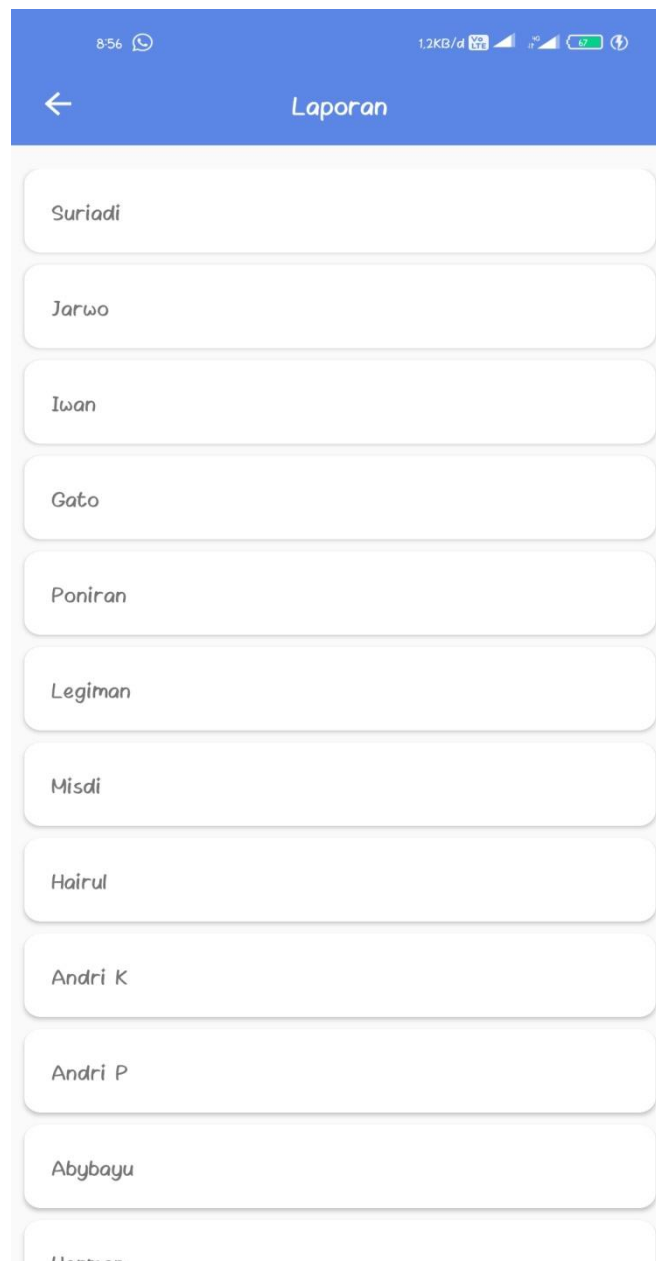


The image shows a mobile application interface for adding a new employee. The screen has a blue header with a back arrow and the title "Tambah Karyawan". Below the header are five white input fields with labels: "Nama Karyawan", "Username", "Password", "Alamat", and "Nomor Telepon". At the bottom is a blue button labeled "Simpan".

Gambar 4.85Implementasi tampilan tambah karyawan(Mandor)

12. Implementasi tampilan menu master laporan(Mandor)

Berikut ini adalah tampilan nama karyawan untuk mencetak laporan pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



Gambar 4.86Implementasi tampilan menu master laporan(Mandor)

13. Implementasi tampilan cetak laporan panen(Mandor)

Berikut ini adalah tampilan cetak laporan mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:

Gambar 4.87Implementasi tampilan cetak laporan absen(Mandor)

14. Implementasi tampilan laporan panen

Berikut ini adalah tampilan laporan panen pada mandor tentang aplikasi pada sisten monitoring aktivitas karyawan lapangan dengan metode *Lock GPS* berbasis *Cloud* pada PTP. Nusantara II:



No	Tanggal	Jumlah	Keterangan
1	2021-01-01	90 Kilogram	Panen 1
2	2021-02-01	90 Kilogram	Panen 1
3	2021-02-01	90 Kilogram	Panen 1
4	2021-02-02	90 Kilogram	Panen 2

Gambar 4.88Implementasi tampilan laporan panen

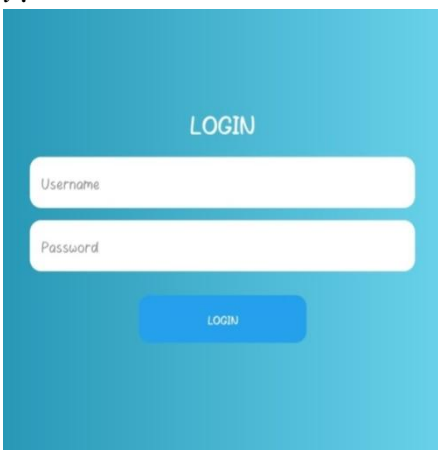
4.5.3 Pengujian Sistem

Pengujian sistem yang akan digunakan menggunakan metode pengujian *blackbox*. Metode ini merupakan metode pengujian yang menitikberatkan pada kebutuhan fungsional sistem. Oleh karena itu pengujian *blackbox* memungkinkan pengembangan perangkat lunak untuk membuat serangkaian kondisi *input* yang akan melatih semua persyaratan fungsional suatu program.

Tabel 4.9 Data Diri Validator

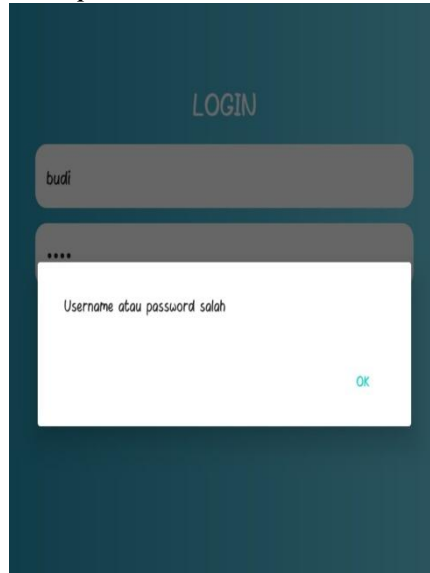
Validator	Nama Validator	Tanda Tangan
Validator 1	Nama :SOFYAN Jabatan : Mandor Lapangan pada PTP. Nusantara II Perk Bandar Klippa	
Validator 2	Nama : Legiman Jabatan : Karyawan Lapangan pada PTP. Nusantara II Perk Bandar Klippa	

Tabel 4.10 Pengujian Sistem Pada Karyawan

No	Rancangan Input/Output	Hasil Yang Diharapkan	Validator I	Validator II
1.	Membuka Sistem	Masuk ke halaman Login	√	√
Tampilan Halaman <i>login</i> :				
				
Gambar 4.89 halaman <i>login</i>				
2.	Klik → Login	Menampilkan	√	√

	(Username dan Password salah)	informasi bahwasan username dan password salah		
--	--------------------------------	--	--	--

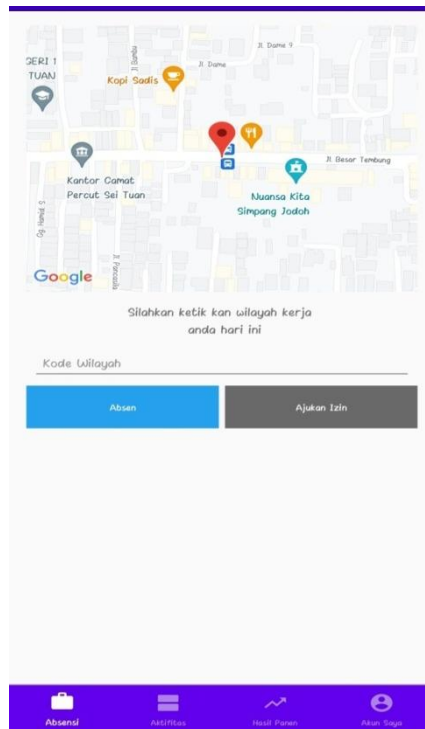
Tampilan Informasi *User* dan *pass* salah:



Gambar 4.90 Tampilan Infromasi *User* dan *pass* salah

3	Klik → Login (Username dan Password Benar)	Masuk ke halaman utama	√	√
---	--	------------------------	---	---

Tampilan menu utama :

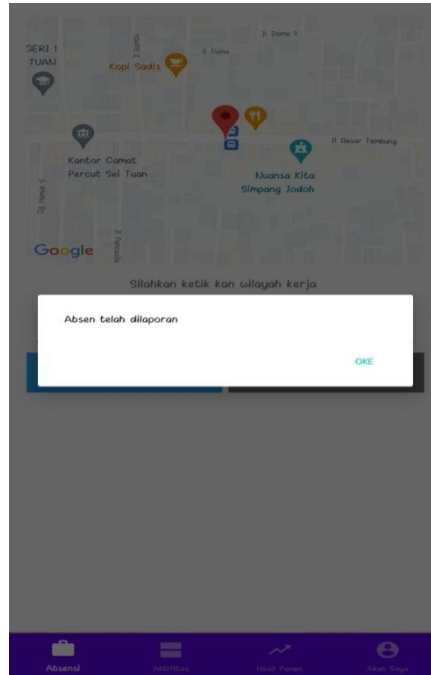


Gambar 4.91 Menu Utama

4	Klik → Absen	Menmpilkan	√	√
---	--------------	------------	---	---

		informasi sudah melakukan absen		
--	--	---------------------------------	--	--

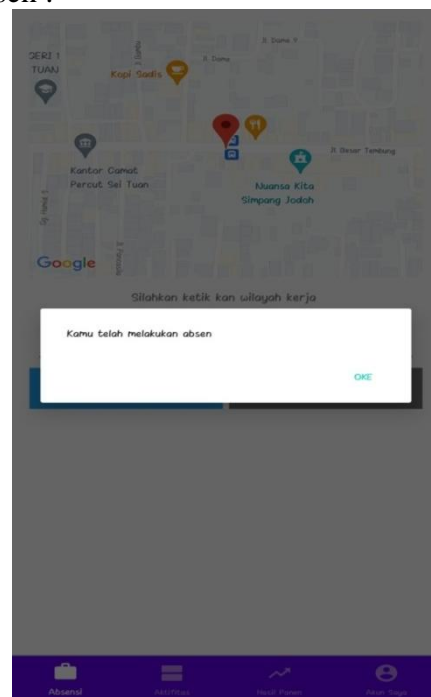
Tampilan Informasi Telah Absen :



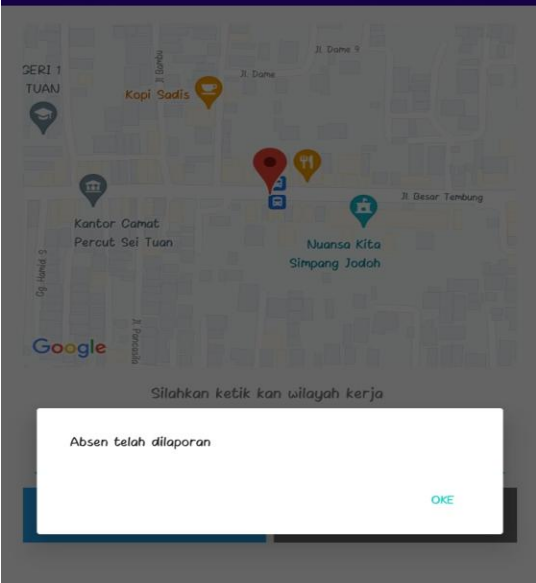
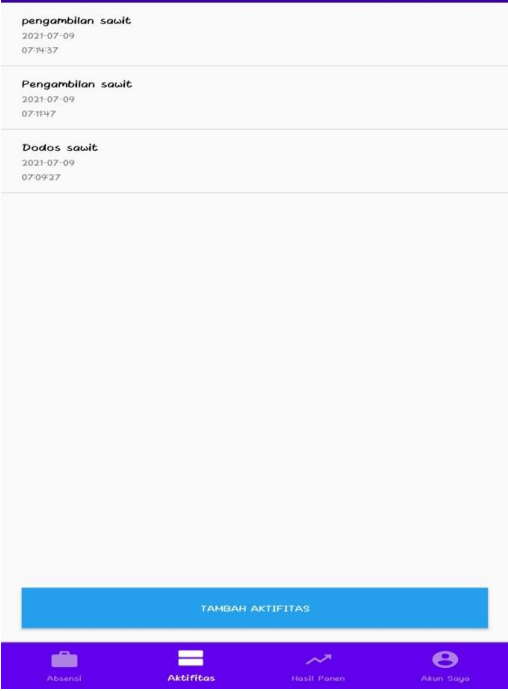
Gambar 4.92 Informasi Telah absen

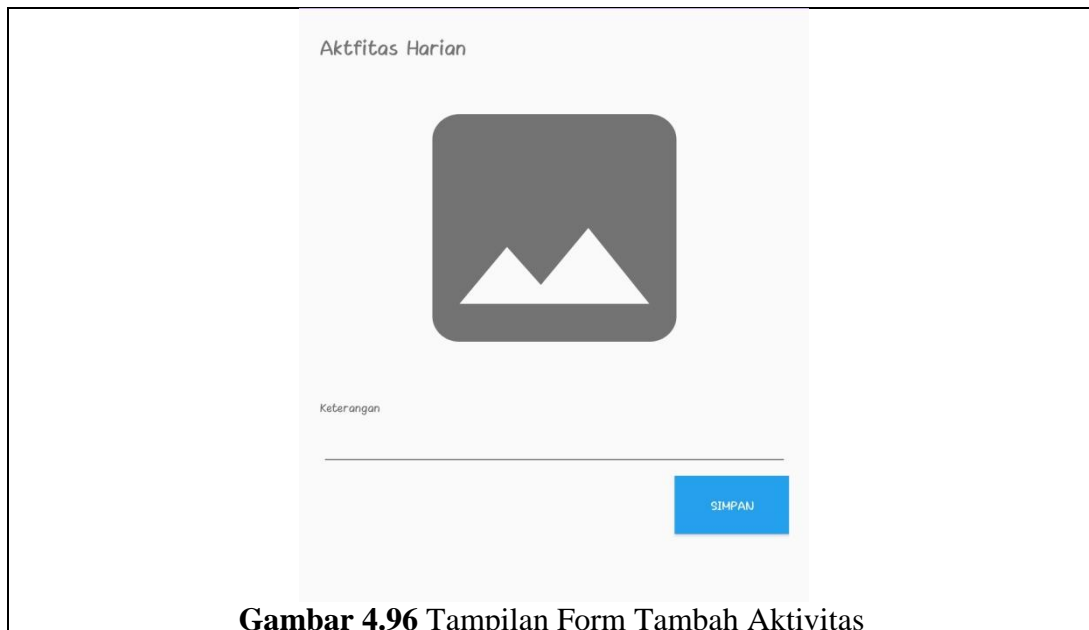
5	Klik → Absen (Melakukan Absen Kedua kalinya)	Menampilkan informasi sudah melakukan absen	√	√
---	--	---	---	---

Tampilan Informasi Absen :



Gambar 4.93 Tampilan Informasi Absen

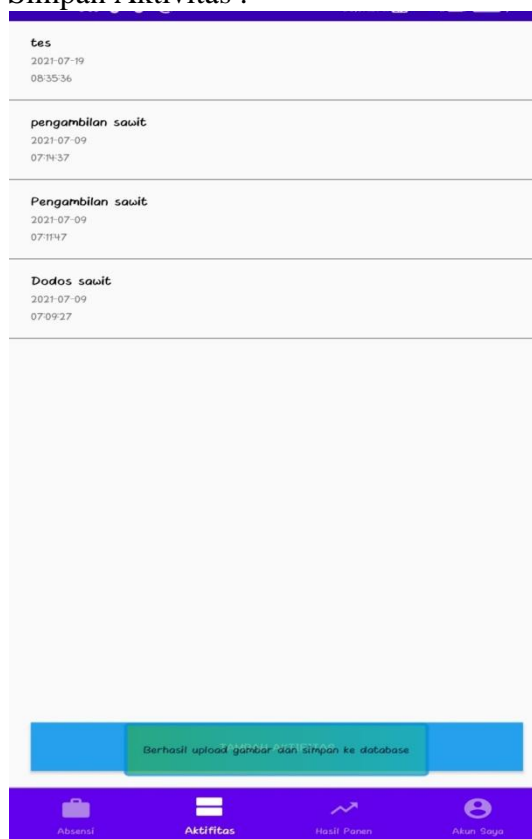
6	Klik → Ajukan Ijin	Menampilkan informasi absen	√	√
<p style="text-align: center;">Tampilan Informasi Ajukan ijin :</p>  <p style="text-align: center;">Gambar 4.94 Tampilan Informasi Ajukan Ijin</p>				
7	Klik → Menu Aktivitas	Menampilkan data aktivitas karyawan	√	√
<p>Tampilan Form Aktivitas :</p>  <p style="text-align: center;">Gambar 4.95 Tampilan Data Aktivitas</p>				
8	Klik → Tambah Aktivitas	Menampilkan form tambah aktivitas	√	√
<p>Tampilan Tambah Aktivitas :</p>				



Gambar 4.96 Tampilan Form Tambah Aktivitas


9	Klik → Simpan Aktivitas	Menampilkan Informasi berhasil upload aktivitas	√	√
---	-------------------------	---	---	---

Tampilan Informasi Simpan Aktivitas :



Gambar 4.97 Tampilan Informasi Simpan Aktivitas

10	Pilih → Data	Menampilkan detail	√	√
----	--------------	--------------------	---	---

	Aktivits	aktivitas		
Tampilan Pilih Data Aktivitas:				
				

Gambar 4.98 Tampilan Detail Aktivitas

11	Klik → Hapus Data Aktivitas	Menampilkan informasi hapus data (Iya/Batal)	√	√
----	-----------------------------	--	---	---

Tampilan Hapus Data Aktivitas:



Gambar 4.99 Tampilan Informasi Hapus Data Aktivitas

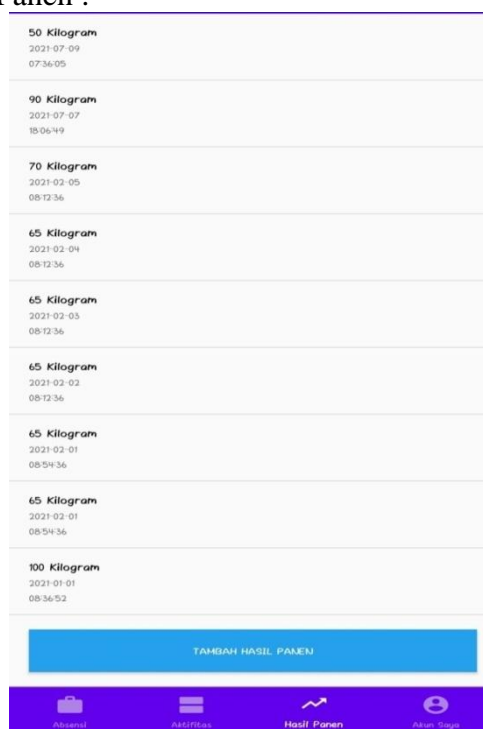
12	Klik → Hapus Data (Iya)	Menampilkan informasi data berhasil	√	√
----	-------------------------	-------------------------------------	---	---



Gambar 4.100 Tampilan Informasi Berhasil Hapus Data

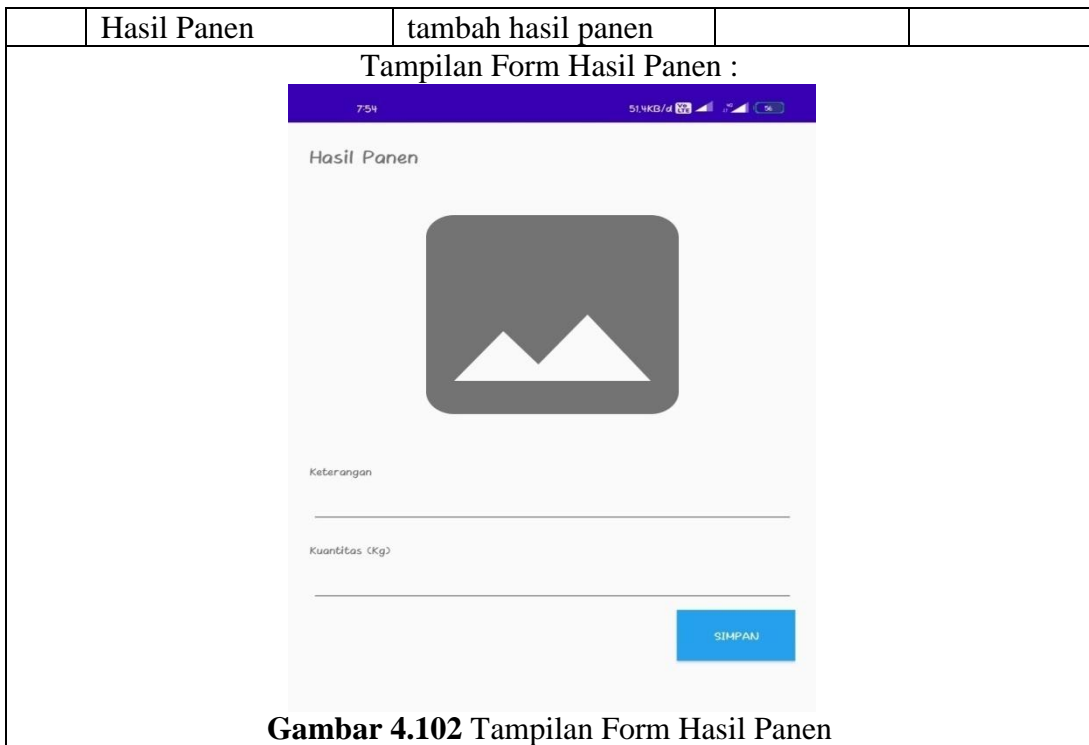
13	Klik → Menu Hasil Panen	Menampilkan data hasil panen	√	√
----	-------------------------	------------------------------	---	---

Tampilan Data Hasil Panen :



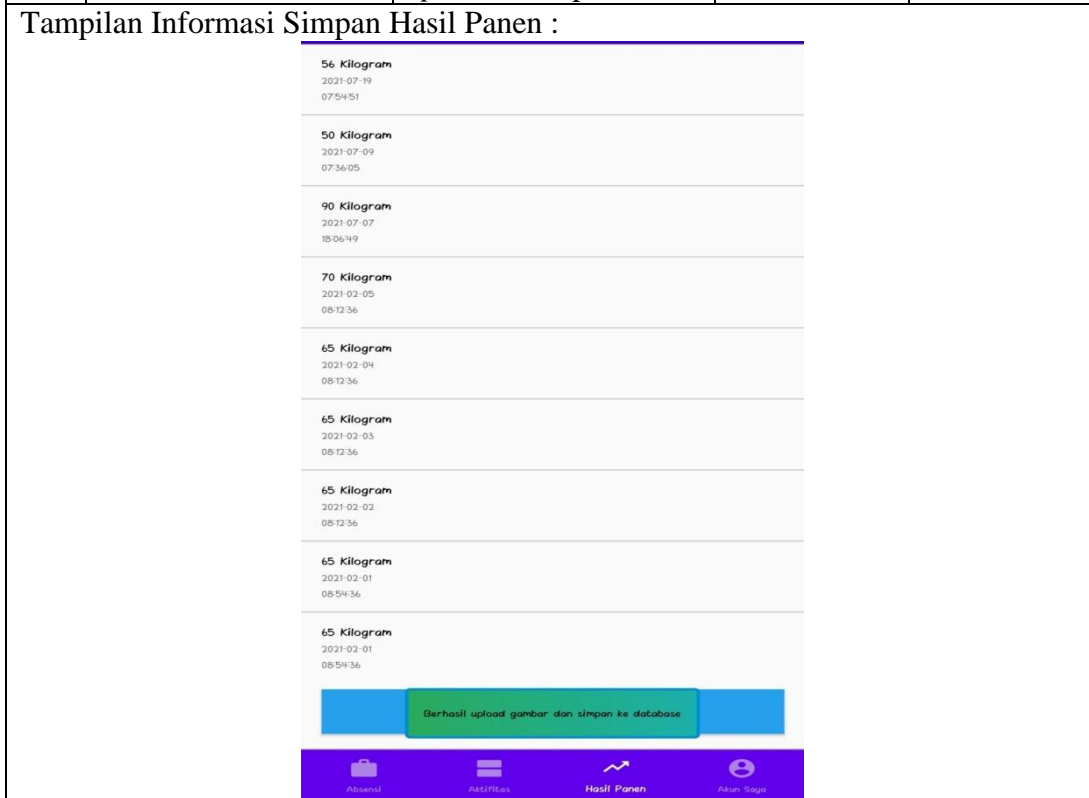
Gambar 4.101 Tampilan Data Hasil Panen

14	Klik → Tambah	Menampilkan form	√	√
----	---------------	------------------	---	---

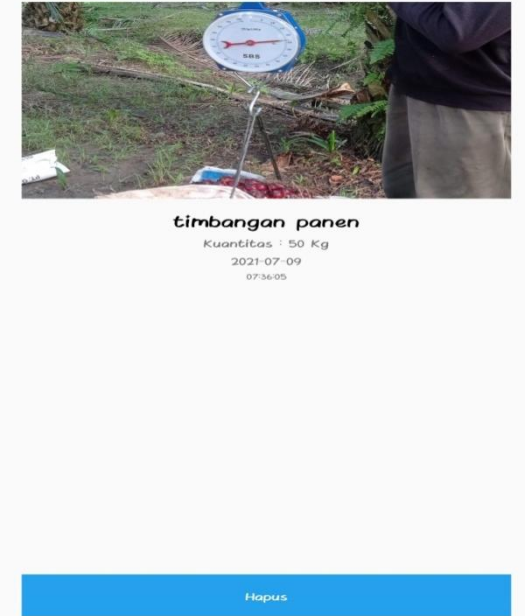



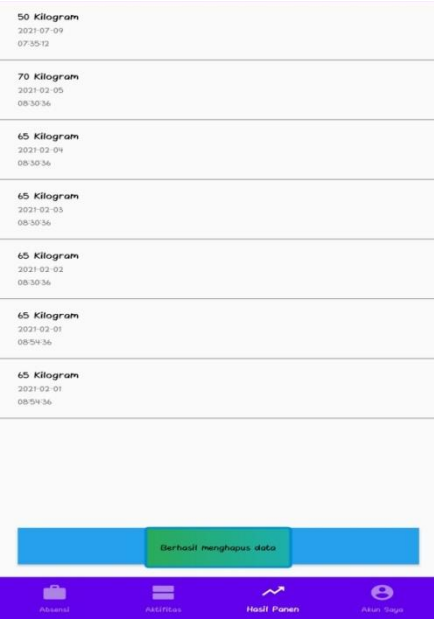
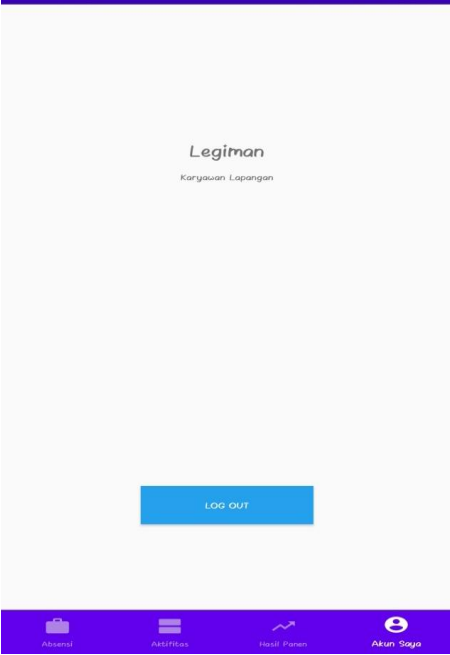
Gambar 4.102 Tampilan Form Hasil Panen

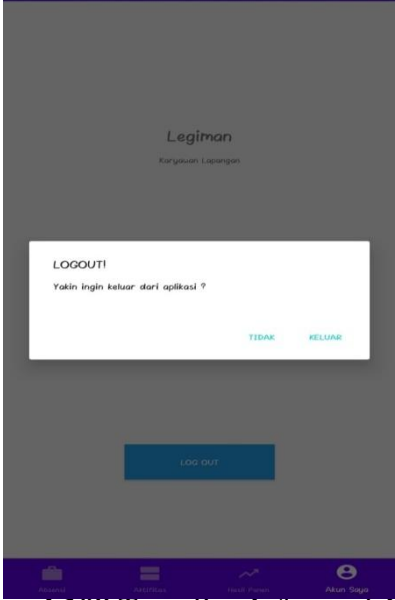
15	Klik \longrightarrow Simpan Hasil Panen	Menampilkan informasi berhasil <i>upload</i> hasil panen	√	√
----	---	--	---	---



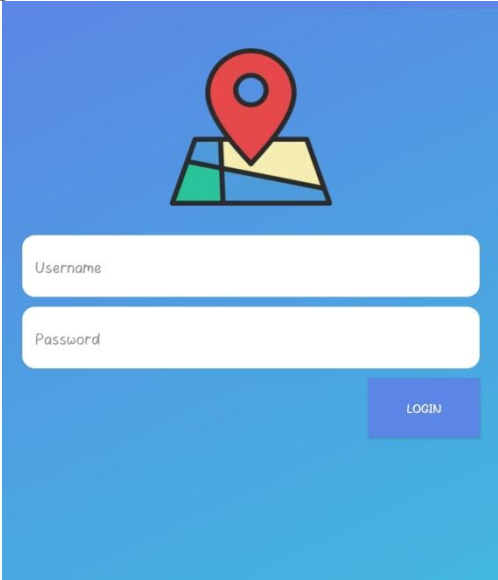
Gambar 4.103 Tampilan Informasi Simpan Data

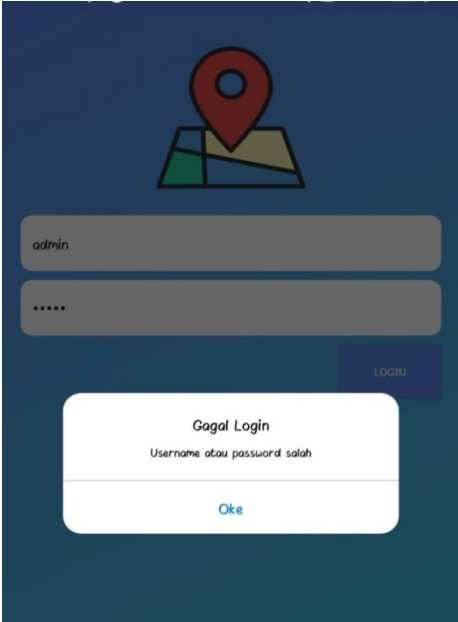
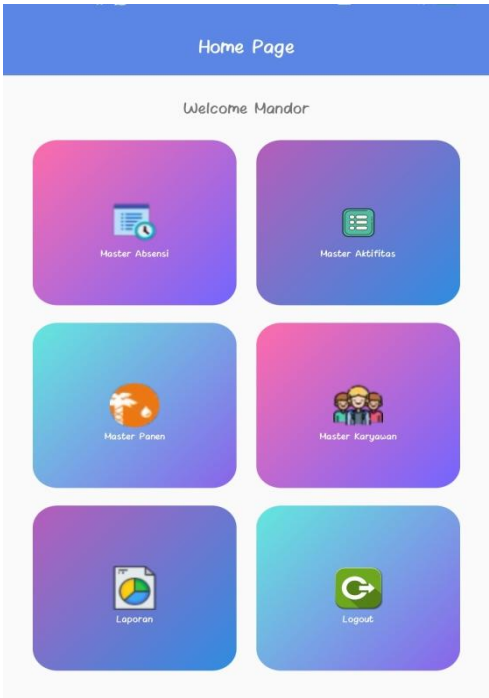
16	Pilih → Data Hasil Panen	Menampilkan detail hasil panen	√	√
<p>Tampilan Detail Hasil Panen:</p>  <p style="text-align: center;">Gambar 4.104 Tampilan Detail Hasil Panen</p>				
17	Klik → Hapus Data Hasil Panen	Menampilkan informasi hapus data (Iya/Batal)	√	√
<p>Tampilan Hapus Data Hasil Panen:</p>  <p style="text-align: center;">Gambar 4.105 Tampilan Hapus Data Hasil Panen</p>				
18	Klik → Hapus	Menampilkan	√	√

	Data (Iya)	informasi data berhasil di hapus		
Tampilan Informasi Hapus Data:				
				
Gambar 4.106 Tampilan informasi Hapus Data				
19	Klik \longrightarrow Menu <i>Logout</i>	Menampilkan form <i>logout</i>	√	√
Tampilan <i>Logout</i> :				
				
Gambar 4.107 Tampilan Logout				
20	Klik \longrightarrow Tombol <i>Logout</i>	Menampilkan Infromasi keluar	√	√

		aplikasi (Tidak / Keluar)		
<p>Tampilan Informasi <i>Logout</i> :</p> 				
<p>Gambar 4.108 Tampilan Informasi <i>Logout</i></p>				

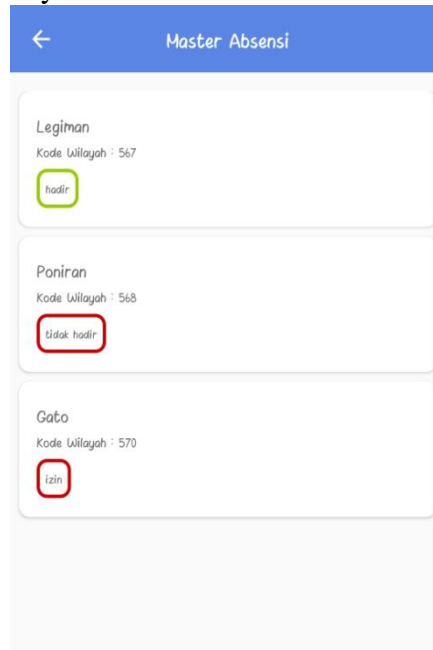
Tabel 4.10 Pengujian Sistem Pada Mandor

No	Rancangan Input/Output	Hasil Yang Diharapkan	Validator I	Validator II
1.	Membuka Sistem	Masuk ke halaman Login	√	√
<p>Tampilan Halaman <i>login</i> :</p> 				
<p>Gambar 4.109 Tampilan Halaman <i>Login</i></p>				
2.	Klik → Login	Menampilkan	√	√

	(Username dan Password salah)	informasi bahwasan username dan password salah		
Tampilan Informasi <i>User</i> dan <i>pass</i> salah :				
				
Gambar 4.110 Informasi User dan Pass Salah				
3	Klik → Login (Username dan Password Benar)	Masuk ke halaman menu utama	√	√
Tampilan Menu Utama:				
				
Gambar 4.111 Tampilan Menu Utama				

4	Klik Absen → Menu	Menampilkan data absen karyawan yang sudah melakukan absen	√	√
---	-------------------	--	---	---

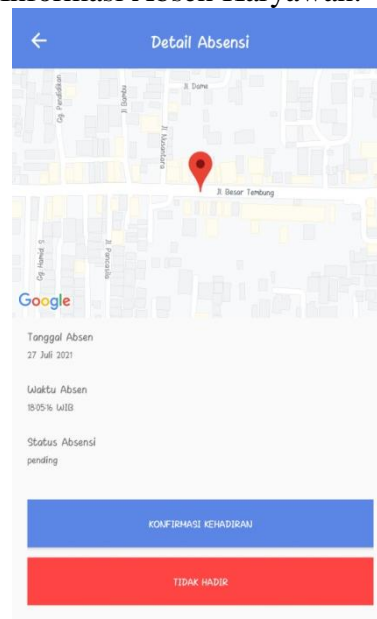
Tampilan Data Absen Karyawan:



Gambar 4.112 Tampilan Data Absen Karyawan

5	Pilih Nama karyawan →	Menampilkan data informasi absen karyawan	√	√
---	-----------------------	---	---	---

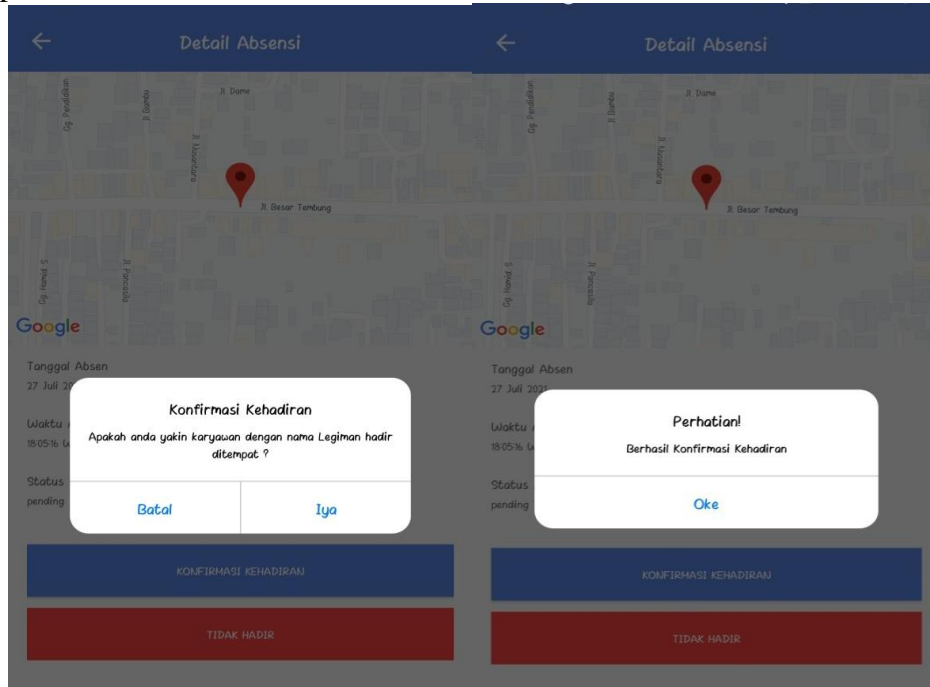
Tampilan Data Informasi Absen Karyawan:



Gambar 4.113 Data Informasi Absen Karyawan

6	Klik → Konfirmasi Absen	Menampilkan informasi konfirmasi kehadiran karyawan	√	√
---	----------------------------	---	---	---

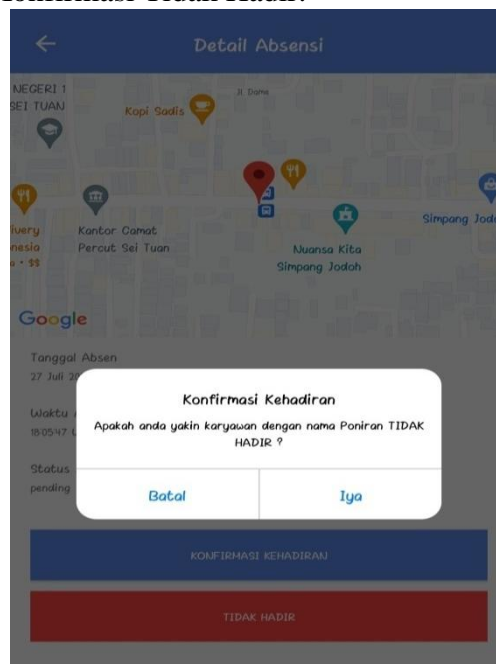
Tampilan Informasi Konfirmasi Kehadiran:



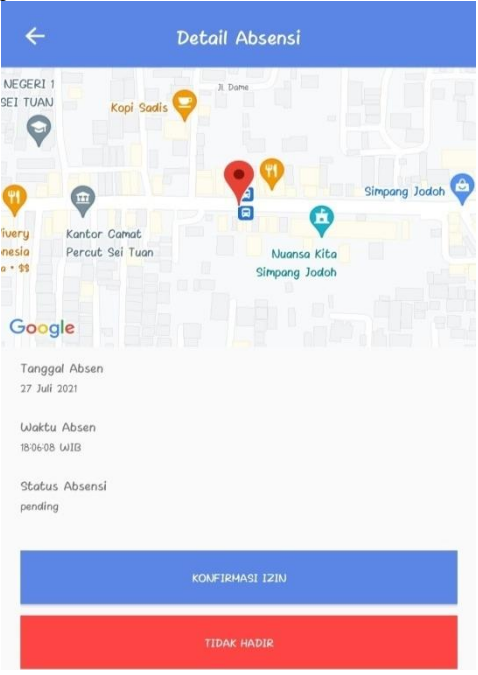
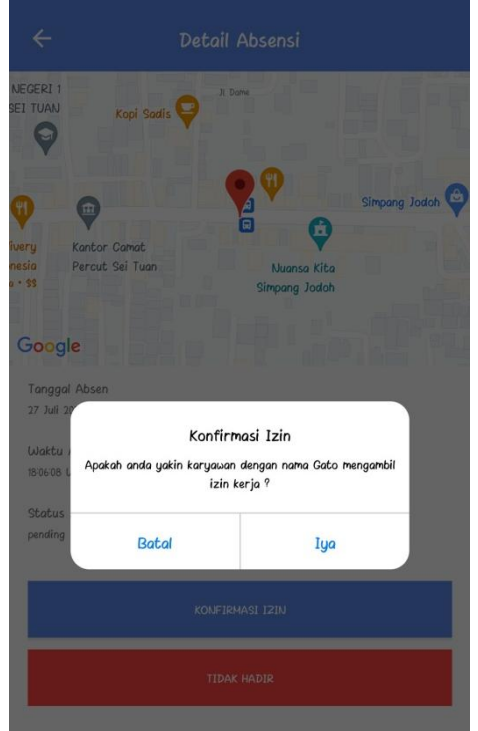
Gambar 4.114 Tampilan Informasi Konfirmasi Kehadiran

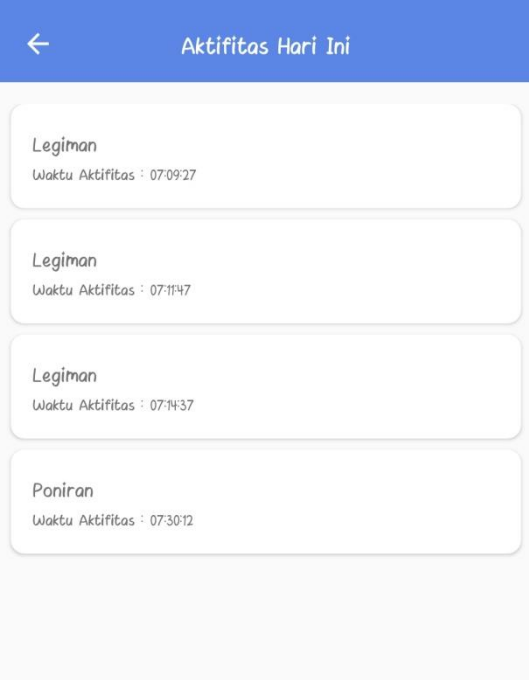
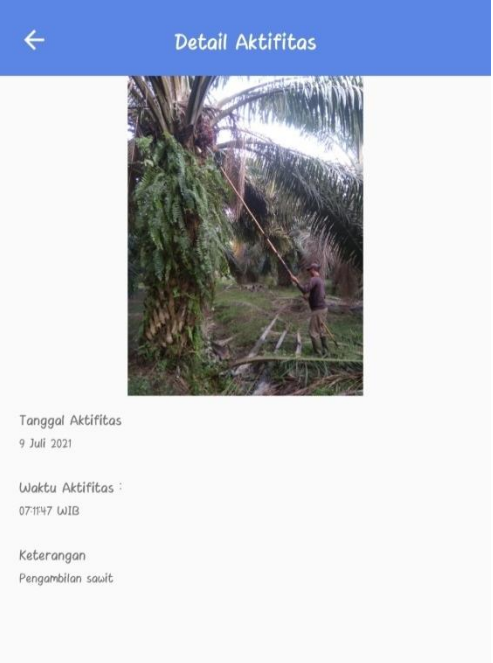
7	Klik → Tidak Hadir	Informasi konfirmasi tidak hadir	√	√
---	--------------------	----------------------------------	---	---

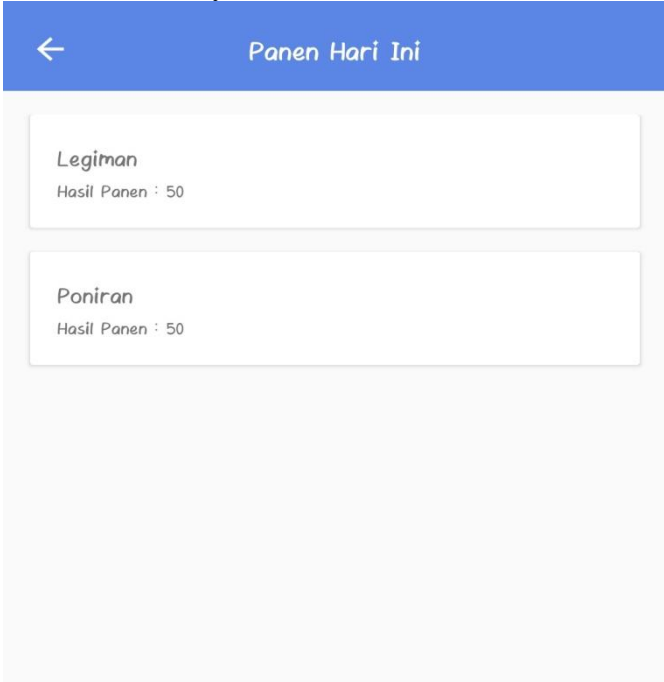
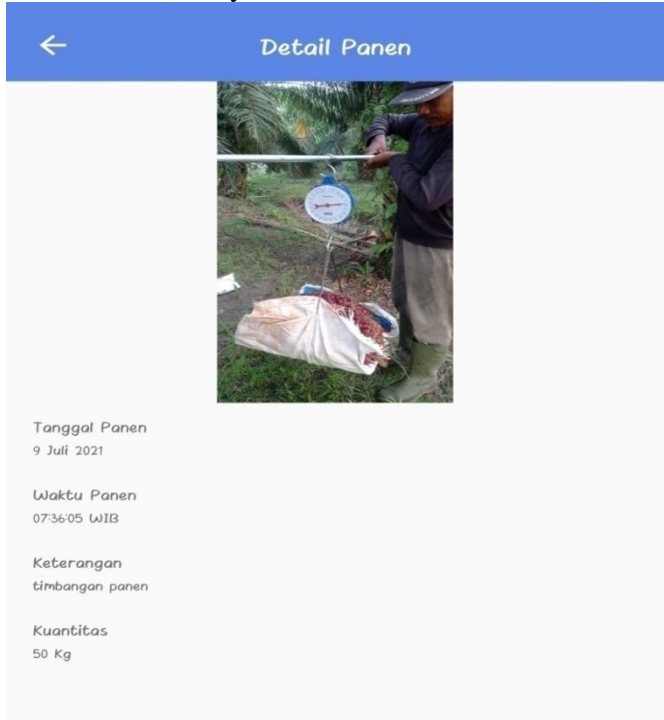
Tampilan Informasi Konfirmasi Tidak Hadir:

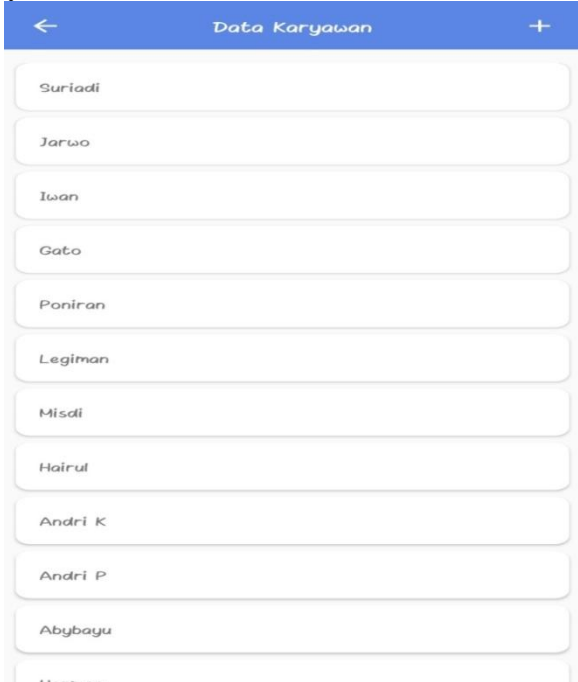




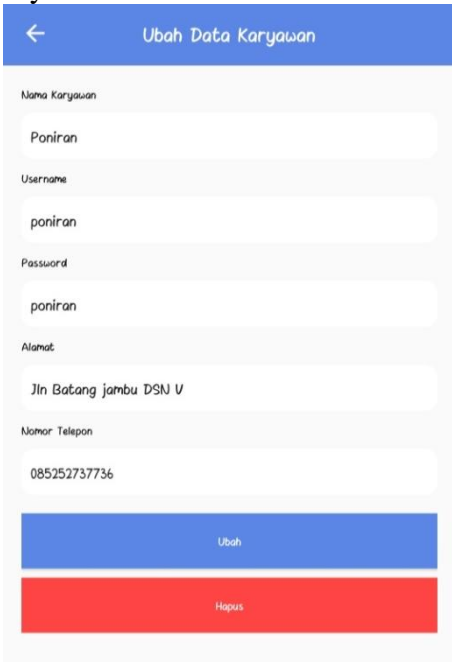
Gambar 4.115 Tampilan Informasi Konfirmasi Tidak Hadir

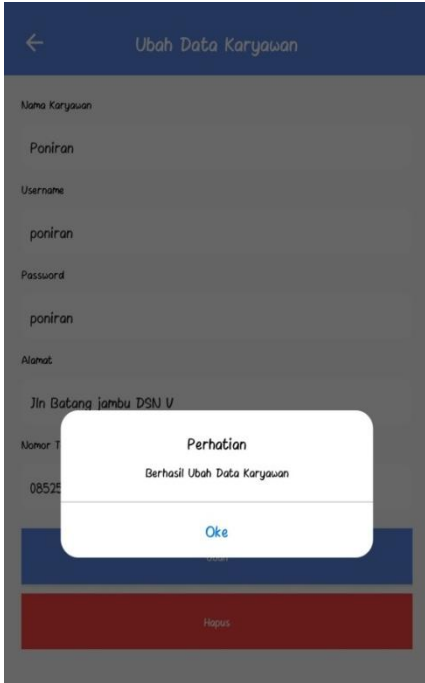
8	Klik → Nama karyawan	Tampilan data ajukan ijin	√	√
<p>Tampilan Konfirmasi Ijin :</p>  <p>Gambar 4.116 Tampilan Konfirmas ijin</p>				
9	klik → Konfirmasi ijin	Tampilan Informasi konfirmasi ajukan ijin	√	√
<p>Tampilan Informasi Konfirmasi ijin :</p>  <p>Gambar 4.117 Tampilan Informasi Konfirmas Ijin</p>				

10	Klik → Menu Aktivitas	Menampilkan data aktivitas karyawan	√	√
<p>Tampilan Data Aktivitas Karyawan:</p>  <p style="text-align: center;">Gambar 4.118 Tampilan Data Aktivitas Karyawan</p>				
11	Pilih → Nama Karyawan	Menampilkan detail informasi aktivitas karyawan	√	√
<p>Tampilan Detail Informasi Aktivitas :</p>  <p style="text-align: center;">Gambar 4.119 Tampilan Detail Informasi Aktivitas</p>				

12	Klik \longrightarrow Menu Hasil Panen	Menampilkan data hasil panen karyawan	√	√
<p>Tampilan Data Hasil Panen Karyawan :</p>  <p style="text-align: center;">Gambar 4.120 Tampilan Data Hasil Panen</p>				
13	Pilih \longrightarrow Nama karyawan	Menampilkan detail hasil panen karyawan	√	√
<p>Tampilan Detail Hasil Panen Karyawan:</p>  <p style="text-align: center;">Gambar 4.121 Tampilan Detail Hasil Panen</p>				

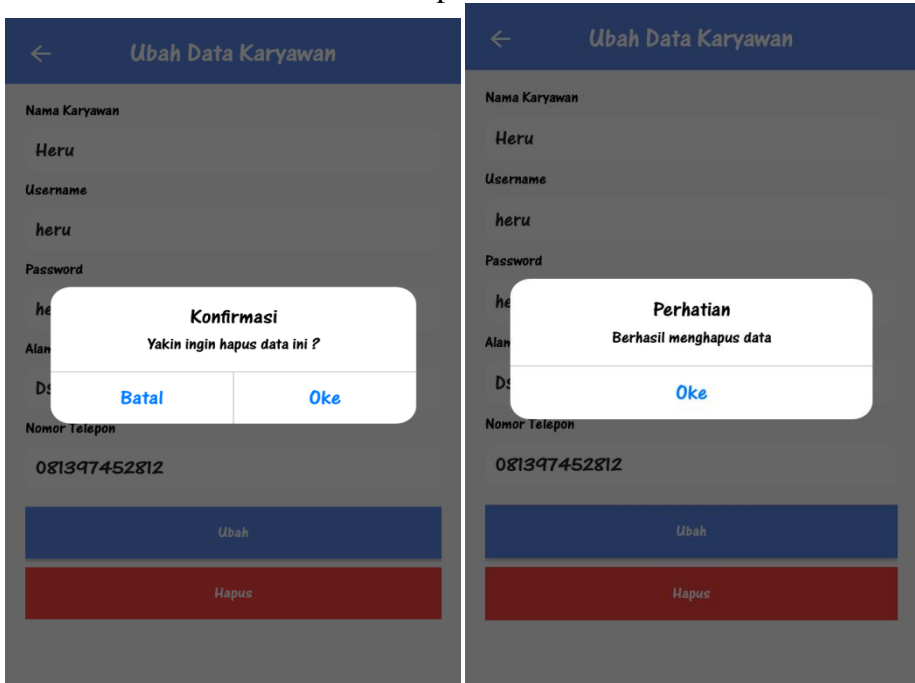
14	Klik \longrightarrow Menu Master karyawan	Menampilkan Data Karyawan	√	√
<p>Tampilan Data Karyawan:</p>  <p style="text-align: center;">Gambar 4.122 Tampilan Data Karyawan</p>				
15	Klik \longrightarrow Tambah Karyawan	Menampilkan form tambah data karyawan	√	√
<p>Tampilan Form Tambah Data Karyawan:</p>  <p style="text-align: center;">Gambar 4.123 Tampilan <i>Form</i> Tambah Data Karyawan</p>				
16	Klik \longrightarrow Simpan	Menampilkan informasi data	√	√

		berhasil di simpan		
<p>Tampilan Informasi Data Berhasil Disimpan:</p>  <p>Gambar 4.124 Informasi Data Berhasil Disimpan</p>				
17	Pilih → Nama karyawan	Menampilkan detail data karyawan	√	√
<p>Tampilan Detail Data Karyawan:</p>  <p>Gambar 4.125 Tampilan Detail Data Karyawan</p>				
18	Klik → Ubah	Menampilkan	√	√

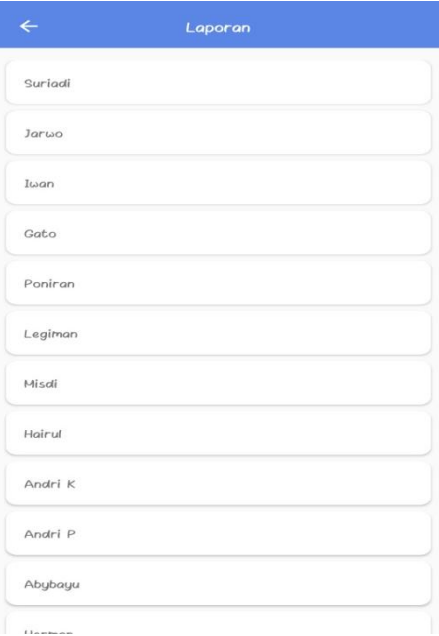

	Data karyawan	informasi data berhasil di ubah		
<p>Tampilan Infromasi Data Berhasil Diubah:</p>  <p>The screenshot shows a mobile application screen titled 'Ubah Data Karyawan'. The form contains fields for Nama Karyawan, Poniran, Username, Password, Alamat, and Nomor T. A white dialog box with the title 'Perhatian' and the message 'Berhasil Ubah Data Karyawan' is overlaid on the form. The dialog box has an 'Oke' button. At the bottom of the form, there are 'Ubah' and 'Hapus' buttons.</p>				

Gambar 4.126 Informasi Data Berhasil Diubah

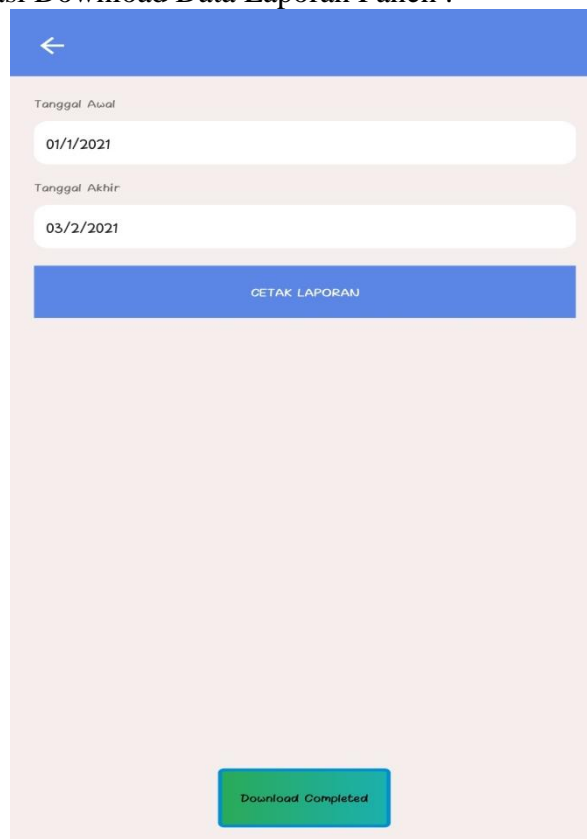
19	klik → Hapus Data Karyawan	Menampilkan informasi data berhasil di hapus	√	√
----	----------------------------	--	---	---

<p>Tampilan Informasi Data Berhasil Dihapus:</p>  <p>The image shows two side-by-side screenshots of the 'Ubah Data Karyawan' form. The left screenshot shows a confirmation dialog box with the title 'Konfirmasi' and the message 'Yakin ingin hapus data ini?'. The dialog box has 'Batal' and 'Oke' buttons. The right screenshot shows a success dialog box with the title 'Perhatian' and the message 'Berhasil menghapus data'. The dialog box has an 'Oke' button. Both screenshots show the form with fields for Nama Karyawan, Username, Password, Alamat, and Nomor Telepon. The form has 'Ubah' and 'Hapus' buttons at the bottom.</p>				
--	--	--	--	--

Gambar 4.127 Informasi Data Berhasil Dihapus

20	Klik → Menu laporan	Menampilkan data nama laporan karyawan	√	√
<p>Tampilan Data Nama laporan Karyawan:</p>  <p style="text-align: center;">Gambar 4.128 Tampilan Data Nama Laporan Karyawan</p>				
21	Pilih → Nama Karyawan	Menampilkan form cetak laporan	√	√
<p>Tampilan Tampilan Form Cetak Laporan:</p>  <p style="text-align: center;">Gambar 4.129 Tampilan Form Cetak Laporan</p>				
22	Klik → Cetak Laporan	Menampilkan Informasi download data laporan panen karyawan	√	√

Tampilan Informasi Download Data Laporan Panen :



Gambar 4.130 Informasi Download Data Laporan Panen

23	Folder PDF Laporan	Menampilkan laporan panen karyawan berbentuk pdf	√	√
----	--------------------	--	---	---

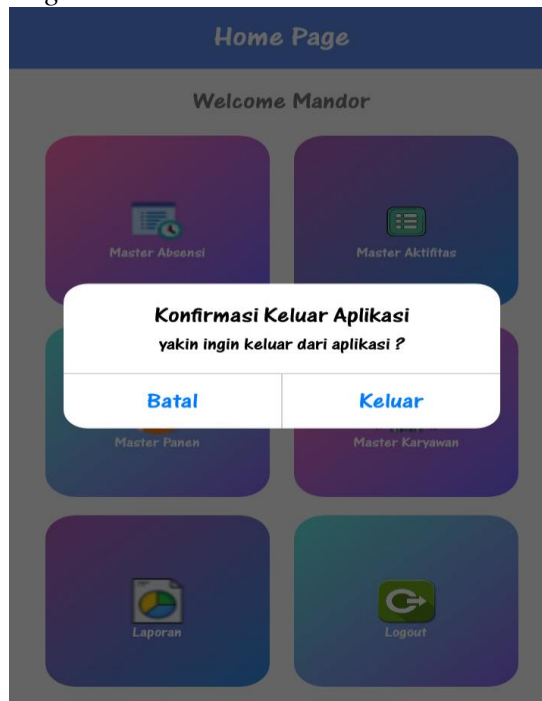
Tampilan Laporan Panen Berbentuk PDF



Gambar 4.131 Tampilan Laporan Panen Berbentuk PDF

24	Klik → Menu <i>Logout</i>	Menampilkan Infomasi keluar aplikasi (Tidak / Keluar)	√	√
----	------------------------------	--	---	---

Tampilan Infomasi *Logout* :



Gambar 4.132 Tampilan Informasi *Logout*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan bab-bab sebelumnya, penulis dapat mencapai kesimpulan sebagai berikut:

1. Metode *Lock GPS* dan penyimpanan dengan *cloud* dapat diterapkan pada aplikasi *monitoring* aktivitas karyawan lapangan pada PTP Nusantara II untuk menjadi solusi dalam meminimalisir ke tidak jujuran yang ada pada setiap pekerjaan karyawan di lapangan. *LOCK GPS* dapat mengunci lokasi karyawan sehingga karyawan tidak dapat lagi melakukan ke tidak jujuran pada saat absensi.
2. Pada sistem ini dibuat dua aplikasi yaitu untuk karyawan dan mandor, agar aplikasi tidak mengalami crash karena saat permintaan data antara mandor dan karyawan serta menghindari pembobolan data penting.
3. Pada sistem ini dibuat dengan pelaporan data hasil panen bulanan karyawan yang terdapat pada aplikasi mandor yang dimana bisa langsung di cetak berbentuk pdf untuk mempermudah pengecekan data hasil panen karyawan dan tidak terjadinya kehilangan data panen karyawan.
4. Dengan menggunakan aplikasi ini, penulis dan pembaca dapat lebih memahami bagaimana mengontrol aktivitas pegawai lapangan dengan menggunakan Android.

5.2 SARAN

Berdasarkan kesimpulan dan tahapan analisis sebelumnya, maka penulis memberikan rekomendasi untuk menggunakan aplikasi sistem monitoring aktivitas pegawai lapangan yaitu :

1. Pada penggunaan aplikasi diwajibkan untuk menghidupkan gps terlebih dahulu untuk mengetahui lokasi titik koordinat karyawan pada saat melakukan absensi.

2. Untuk mengoptimalkan kestabilan sistem karyawan wajib menggunakan *smartphone* minimal versi 7.0 (Nougat) agar tidak terjadinya *crash* dalam menjalankan sistem.
3. Untuk kestabilan *Upload* data dan permintaan laporan data diharapkan koneksi jaringan internet stabil agar tidak mengalami kegagalan dalam *upload* dan permintaan data.

DAFTAR PUSTAKA

- Adikara, F. (2013). Analisis Dan Perancangan Sistem Absensi Berbasis Global Positioning System (GPS) Pada Android 4.x. *Seminar Nasional Teknologi Informasi*.
- Ali Ikhwan, M. K. (2018). *IMPLEMENTASI DATA MINING UNTUK ANALISA POLA PENERIMAAN SECURITY PADA PT.DM SECURITY MENGGUNAKAN METODE ALGORITMA APRIORI*. 21, 1–9.
- Amri, K., Hasdiana, & Rafika Dewi, A. (2019). Rancang Bangun Sistem Informasi Absensi Berbasis GPS Pada Kantor Badan Pusat Statistik Provinsi Sumatera Utara Menggunakan Android. *Snastikom*.
- Andi, J. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System (A-GPS) Dengan Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*.
- Ariyanti, R., Khairil, & Kanedi, I. (2015). Pemanfaatan Google Maps Api Pada Sistem Informasi Geografis Direktori Perguruan Tinggi Di Kota Bengkulu. *Jurnal Media Infotama*.
- Ashari, A., & Setiawan, H. (2014). Cloud Computing: Solusi ICT? *Jurnal Sistem Informasi*.
- Dharwiyanti, S., & Wahono, R. S. (2003). Pengantar Unified Modeling LAnguage (UML). *IlmuKomputer.Com*.
- Fajrin, T. (2012). Analisis Sistem Penyimpanan Data Menggunakan Sistem Cloud Computing Studi Kasus SMK N 2 Karanganyar. *Analisis Sistem Penyimpanan Data Menggunakan Sistem Cloud Computing Studi Kasus SMK N 2 Karanganyar*.
- Febriandirza, A. (2020). Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin. *Pseudocode*, 7(2). <https://doi.org/10.33369/pseudocode.7.2.123-133>
- Hayati, L. N. (2019). SISTEM MONITORING KARYAWAN DENGAN METODE LBS (LOCATION BASED SERVICE) BERBASIS ANDROID. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 2(1). <https://doi.org/10.31598/jurnalresistor.v2i1.347>
- Hendini, A. (2016). PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK. *JURNAL KHATULISTIWA INFORMATIKA*, VOL. IV, NO. 2 DESEMBER 2016 PEMODELAN. *Crop Science*.
- Herliana, A., & Rasyid, P. M. (2016). Sistem Informasi Monitoring Pengembangan Software

Pada Tahap. *Jurnal Informatika*.

- Husain, Zarlis, M., Nasution, Z., Sihotang, H. T., & Wahyuni, S. (2018). FILSAFAT ILMU KOMPUTER DAN CLOUD COMPUTING SECARA ETIMOLOGIS. *Jurnal Mantik Penusa*.
- Hutagaol, N. H., Kurniawan, H., & Chahyadi, F. (2014). Aplikasi Absensi Tenaga Marketing Menggunakan Location Based Service (LBS) Berbasis Android. *Universitas Maritim Raja Ali Haji*.
- Kusniyati, H., & Pangondian Sitanggang, N. S. (2016). APLIKASI EDUKASI BUDAYA TOBA SAMOSIR BERBASIS ANDROID. *JURNAL TEKNIK INFORMATIKA*.
<https://doi.org/10.15408/jti.v9i1.5573>
- Lengkong, H. N., Sinsuw, A. A. E., & Lumenta, A. S. M. (2015). Route Designing on Private Vehicles Using Android-Based Gis Mobile Applications Integrated in Google Maps. *E-Journal of Electrical and Computer Engineering*.
<https://doi.org/10.35793/jtek.4.2.2015.6817>
- Mahendra, D. C., Susyanto, T., & Siswanti, S. (2018). SISTEM MONITORING MOBIL RENTAL MENGGUNAKAN GPS TRACKER. *Jurnal Ilmiah SINUS*.
<https://doi.org/10.30646/sinus.v16i2.357>
- Makhfuddin, R. A., & Prabowo, N. (2015). Aplikasi Absensi Menggunakan Metode Lock GPS dengan Android di PLN APP Malang Basecamp Mojokerto. *Issn*.
- Mardiani, G. T. (2013). SISTEM MONITORING DATA ASET DAN INVENTARIS PT TELKOM CIANJUR BERBASIS WEB. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*. <https://doi.org/10.34010/komputa.v2i1.78>
- Mutia, I. (2016). Pemanfaatan Komputasi Awan (Cloud Computing) Bagi Pembelajaran Mahasiswa Perguruan Tinggi. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*.
<https://doi.org/10.30998/string.v1i1.963>
- Nazruddin, S. (2012). Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. *דגנרע עלון*.
- NUR YATI, M. (2018). APLIKASI BELAJAR MENGAJI BERBASIS ANDROID. *Jurnal Manajemen Informatika*.
- Puteri, M. P., & Effendi, H. (2018). Implementasi Metode RAD Pada Website Service Guide “Tour Waterfall South Sumatera.” *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 7(2),

130–136. <https://doi.org/10.32736/sisfokom.v7i2.570>

- Rahman, A. F., Kharisma, A. P., & Dewi, R. K. (2018). Rancang Bangun Aplikasi Geofence Marketing Cafe Berbasis Android Studi Kasus : Ice Ah ! *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*.
- Reinaldi, N., Susanti, F., & Gunawan, T. (2017). Implementasi Dan Monitoring Layanan Cloud Storage Seafile Implementation and Monitoring Service Cloud Storage Seafile With Nagios in School of Applied Science. *E-Proceeding of Applied Science*, 3(1).
- Riyanto, N. P. (2019). Reminder System dan Monitoring Proyek untuk Penilaian Kinerja Karyawan Berbasis Web. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*.
<https://doi.org/10.32736/sisfokom.v8i2.634>
- Salamun, S. (2017). SISTEM MONITORING NILAI SISWA BERBASIS ANDROID. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*. <https://doi.org/10.36341/rabit.v2i2.221>
- Sanitasari, R. D., Andreswari, D., & Purwandari, E. P. (2017). Sistem Monitoring Tumbuh Kembang Anak Usia 0-5 Tahun Berbasis Android. *Journal Rekursif*.
- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. *9th Industrial Research Workshop and National Seminar (IRONS), Juli*.
- Snaith, B., Hardy, M., & Walker, A. (2011). Emergency ultrasound in the prehospital setting: the impact of environment on examination outcomes. *Emergency Medicine Journal : EMJ*, 28(12), 1063–1065. <https://doi.org/10.1136/emj.2010.096966>
- Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer Dan Informatika*.
- Sulaiman, O. K., & Widarma, A. (2017). Cloud Computing Based Internet of Things (IoT) System in Campus Area Network. *Computer and Systems Architecture*.
- Tafa, I. A., Suryadi, D., & Pontia, T. (2018). Analisis Tingkat Akurasi Global Positioning System Smartphone Dalam Menentukan Titik Lokasi Pada Google Map. *Jurnal Untan*.
- Weldi, Dedi Triyanto, & Uray Ristian. (2020). Aplikasi Sistem Kontrol Parkir Menggunakan Metode *Lock GPS* Berbasis *Internet Of Thing*. *Jurnal Komputer Dan Aplikasi*.

LAMPIRAN I PENGUJIAN BLACKBOX

VALIDATOR SISTEM

20, Agustus, 2021

Validator I

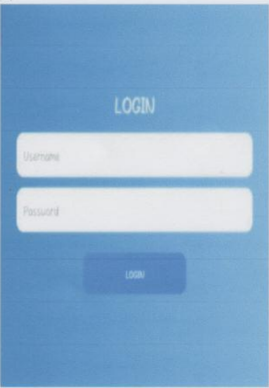
Nama Validator : SOFYAN



Jabatan : Mandor Lapangan PTP. Nusantara II Bandar Klippa

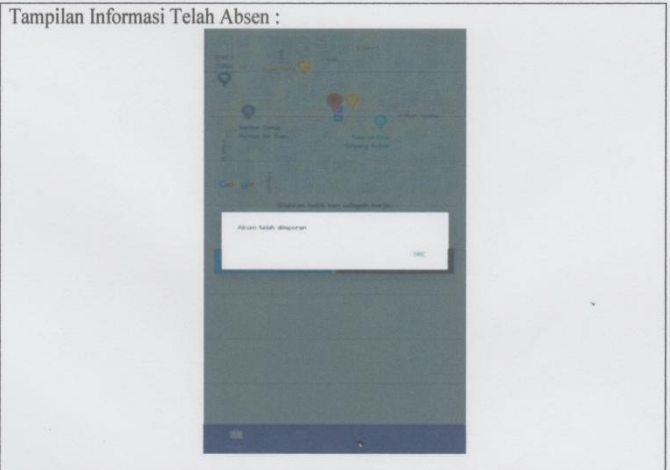
Validator II

Nama Validator : LEGIMAN

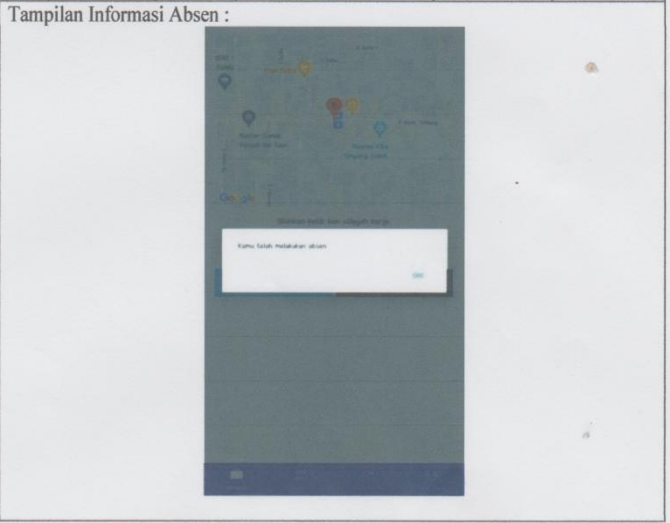
Jabatan : Karyawan Lapangan PTP. Nusantara II Bandar Klippa

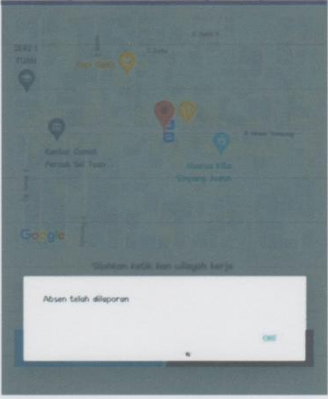

No	Rancangan Input/Output	Hasil Yang Diharapkan	Validator I	Validator II
1.	Membuka Sistem	Masuk ke halaman Login	✓	✓
Tampilan Halaman <i>login</i> :				
				
2.	Klik → Login (Username dan Password salah)	Menampilkan informasi bahwasan username dan password salah	✓	✓


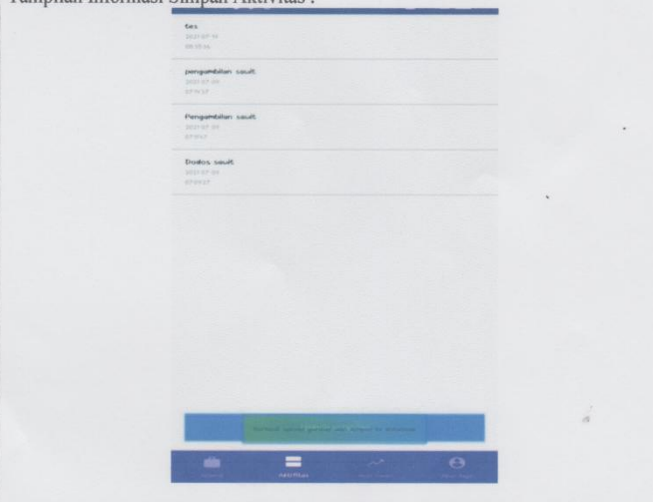
Tampilan Informasi <i>User</i> dan <i>pass</i> salah:				
				
3	Klik → Login (<i>Username</i> dan <i>Password</i> Benar)	Masuk ke halaman utama	✓	✓
Tampilan menu utama :				
				
4	Klik → Absen	Menampilkan informasi sudah melakukan absen	✓	✓


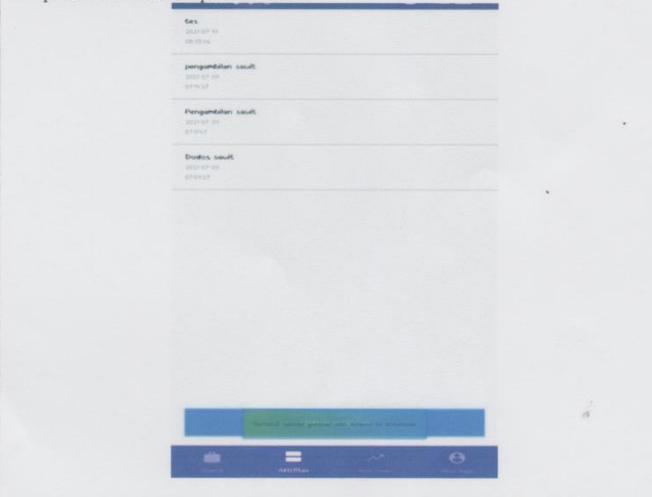




5	Klik → Absen (Melakukan Absen Kedua kalinya)	Menampilkan informasi sudah melakukan absen	✓	✓
---	--	---	---	---






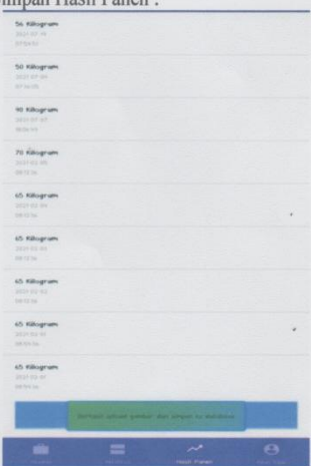
6	Klik → Ajukan Ijin	Menampilkan informasi absen	✓	✓
<p>Tampilan Informasi Ajukan ijin :</p> 				
7	Klik → Menu Aktivitas	Menampilkan data aktivitas karyawan	✓	✓
<p>Tampilan Form Aktivitas :</p> 				

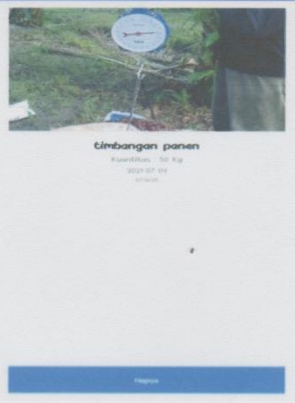
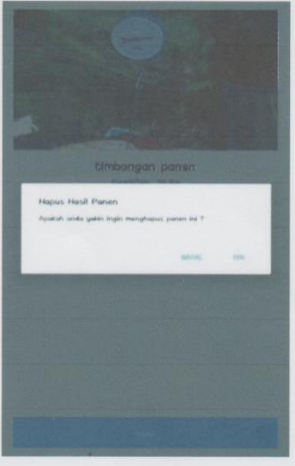
8	Klik → Tambah Aktivitas	Menampilkan form tambah aktivitas	✓	✓
<p>Tampilan Tambah Aktivitas :</p> 				
9	Klik → Simpan Aktivitas	Menampilkan Informasi berhasil upload aktivitas	✓	✓
<p>Tampilan Informasi Simpan Aktivitas :</p> 				

8	Klik → Tambah Aktivitas	Menampilkan form tambah aktivitas	✓	✓
<p>Tampilan Tambah Aktivitas :</p> 				
9	Klik → Simpan Aktivitas	Menampilkan Informasi berhasil upload aktivitas	✓	✓
<p>Tampilan Informasi Simpan Aktivitas :</p> 				

10	Pilih → Data Aktivitas	Menampilkan detail aktivitas	✓	✓
Tampilan Pilih Data Aktivitas:				
				
11	Klik → Hapus Data Aktivitas	Menampilkan informasi hapus data (Iya/Batal)	✓	✓
Tampilan Hapus Data Aktivitas:				
				

12	Klik → Hapus Data (Iya)	Menampilkan informasi data berhasil di hapus	✓	✓
<p>Tampilan Informasi Hapus Data:</p> 				
13	Klik → Menu Hasil Panen	Menampilkan data hasil panen	✓	✓
<p>Tampilan Data Hasil Panen :</p> 				

14	Klik → Tambah Hasil Panen	Menampilkan form tambah hasil panen	✓	✓
<p>Tampilan Form Hasil Panen :</p> 				
15	Klik → Simpan Hasil Panen	Menampilkan informasi berhasil upload hasil panen	✓	✓
<p>Tampilan Informasi Simpan Hasil Panen :</p> 				

16	Pilih → Data Hasil Panen	Menampilkan detail hasil panen	✓	✓
Tampilan Detail Hasil Panen:				
				
17	Klik → Hapus Data Hasil Panen	Menampilkan informasi hapus data (Iya/Batal)	✓	✓
Tampilan Hapus Data Hasil Panen:				
				

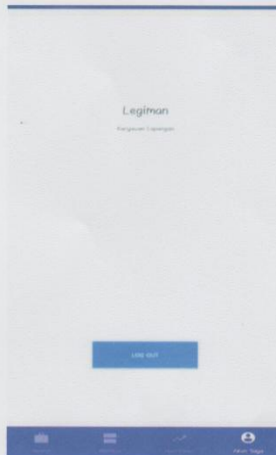
18	Klik → Hapus Data (Iya)	Menampilkan informasi data berhasil di hapus	✓	✓
----	-------------------------	--	---	---


Tampilan Informasi Hapus Data:




19	Klik → Menu Logout	Menampilkan form logout	✓	✓
----	--------------------	-------------------------	---	---

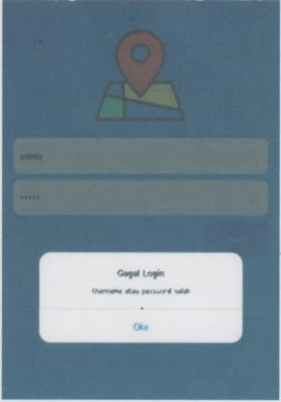

Tampilan Logout :


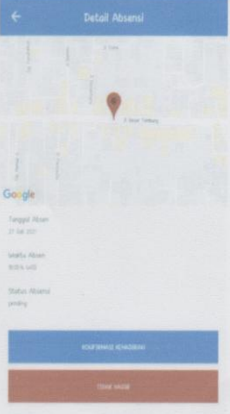


20	Klik → Tombol <i>Logout</i>	Menampilkan Infomasi keluar aplikasi (Tidak / Keluar)	✓	✓
Tampilan Infomasi <i>Logout</i> :				
				

Tabel 4.10 Pengujian Sistem Pada Mandor

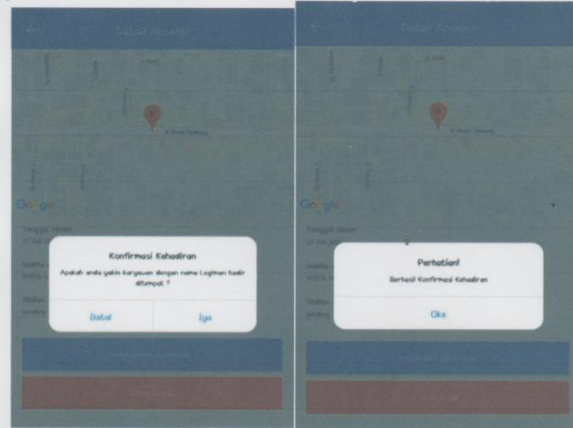
No	Rancangan Input/Output	Hasil Yang Diharapkan	Validator I	Validator II
1.	Membuka Sistem	Masuk ke halaman Login	✓	✓
Tampilan Halaman <i>login</i> :				
				

2.	Klik → Login (<i>Username</i> dan <i>Password</i> salah)	Menampilkan informasi bahwasan username dan password salah	✓	✓
<p>Tampilan Informasi <i>User</i> dan <i>pass</i> salah :</p> 				
3	Klik → Login (<i>Username</i> dan <i>Password</i> Benar)	Masuk ke halaman menu utama	✓	✓
<p>Tampilan Menu Utama:</p> 				

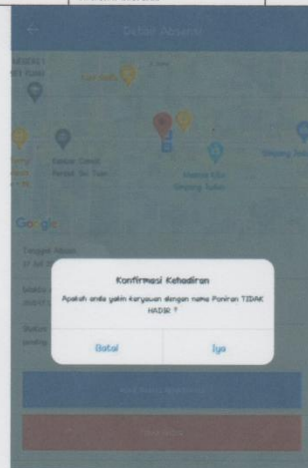
4	Klik → Menu Absen	Menampilkan data absen karyawan yang sudah melakukan absen	✓	✓
<p>Tampilan Data Absen Karyawan:</p> 				
5	Pilih → Nama karyawan	Menampilkan data informasi absen karyawan	✓	✓
<p>Tampilan Data Informasi Informasi Absen Karyawan:</p> 				


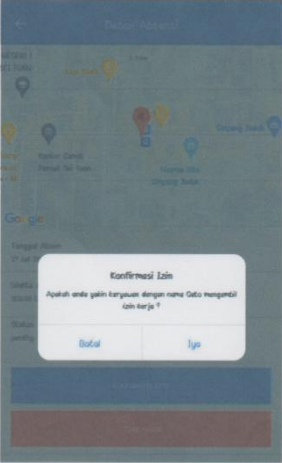
6	Klik → Konfirmasi Absen	Menampilkan informasi konfirmasi kehadiran karyawan	✓	✓
---	----------------------------	---	---	---



Tampilan Informasi Konfirmasi Kehadiran:





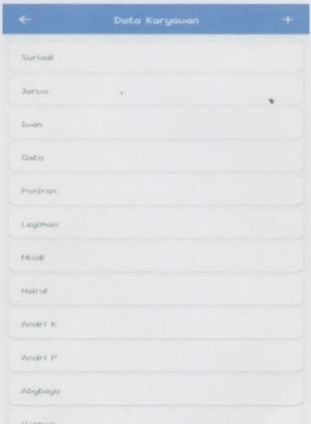
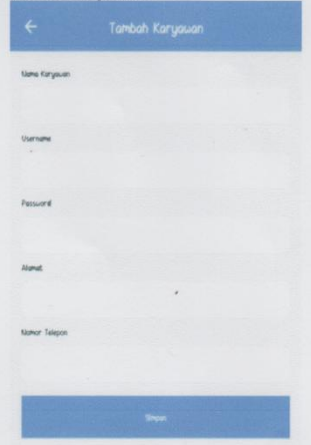
7	Klik → Tidak Hadir	Informasi konfirmasi tidak hadir	✓	✓
---	-----------------------	----------------------------------	---	---

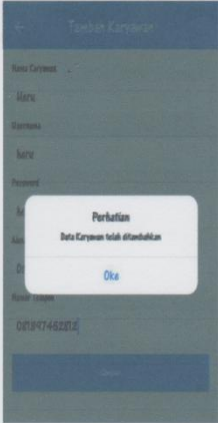



8	Klik → Nama karyawan	Tampilan data ajukan ijin	✓	✓
				
9	klik → Konfirmasi izin	Tampilan Informasi konfirmasi ajukan ijin	✓	✓
				

10	Klik → Menu Aktivitas	Menampilkan data aktivitas karyawan	✓	✓
<p>Tampilan Data Aktivitas Karyawan:</p>  <p>The screenshot shows a mobile application interface titled 'Aktifitas Hari Ini'. It features a blue header with a back arrow and the title. Below the header, there is a list of activity items. Each item consists of a name and a timestamp. The items are: Legiman (07:01:27), Legiman (07:01:47), Legiman (07:01:37), and Puritan (07:01:12).</p>				
11	Pilih → Nama Karyawan	Menampilkan detail informasi aktivitas karyawan	✓	✓
<p>Tampilan Detail Informasi Aktivitas :</p>  <p>The screenshot shows a mobile application interface titled 'Detail Aktivitas'. It features a blue header with a back arrow and the title. Below the header, there is a large image of a tropical landscape with palm trees. Underneath the image, there is text providing details about the activity: 'Tempat Aktivitas: + 00 000', 'Waktu Aktivitas: 07:01:40', and 'Estimasi: Pergerakan saat'.</p>				

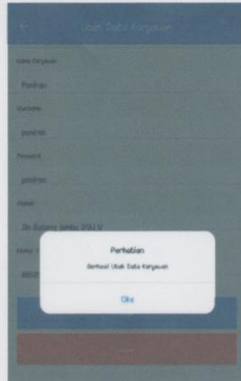
12	Klik → Menu Hasil Panen	Menampilkan data hasil panen karyawan	✓	✓
<p>Tampilan Data Hasil Panen Karyawan :</p> 				
13	Pilih → Nama karyawan	Menampilkan detail hasil panen karyawan	✓	✓
<p>Tampilan Detail Hasil Panen Karyawan:</p> 				

14	Klik → Menu Master karyawan	Menampilkan Data Karyawan	✓	✓
Tampilan Data Karyawan:				
				
15	Klik → Tambah Karyawan	Menampilkan form tambah data karyawan	✓	✓
Tampilan Form Tambah Data Karyawan:				
				

16	Klik → Simpan	Menampilkan informasi data berhasil di simpan	✓	✓
<p>Tampilan Informasi Data Berhasil Disimpan:</p> 				
17	Pilih → Nama karyawan	Menampilkan detail data karyawan	✓	✓
<p>Tampilan Detail Data Karyawan:</p> 				

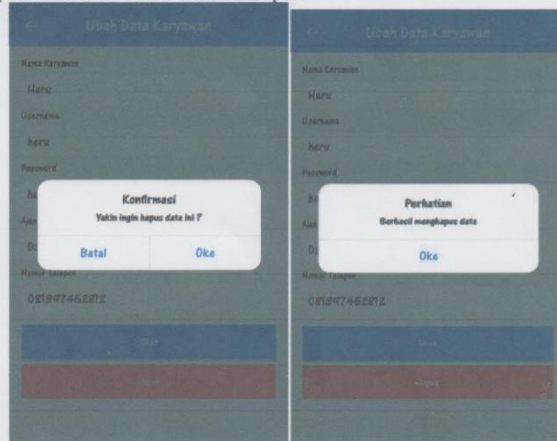
18	Klik → Ubah Data karyawan	Menampilkan informasi data berhasil di ubah	✓	✓
----	---------------------------	---	---	---

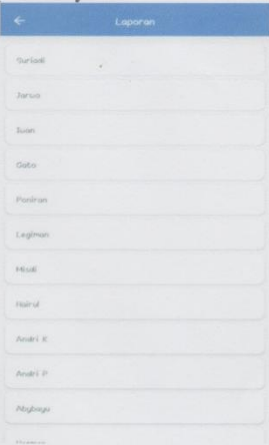
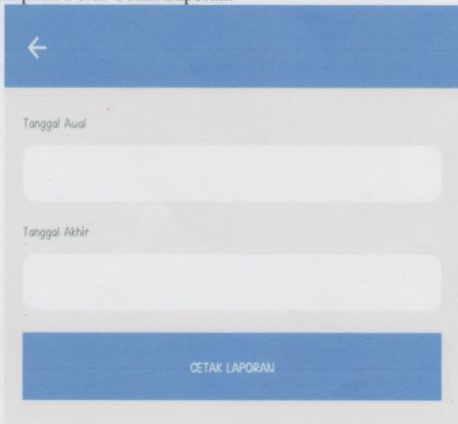
Tampilan Informasi Data Berhasil Diubah:



19	klik → Hapus Data Karyawan	Menampilkan informasi data berhasil di hapus	✓	✓
----	----------------------------	--	---	---

Tampilan Informasi Data Berhasil Dihapus:



20	Klik → Menu laporan	Menampilkan data nama laporan karyawan	✓	✓
Tampilan Data Nama laporan Karyawan:				
				
21	Pilih → Nama Karyawan	Menampilkan form cetak laporan	✓	✓
Tampilan Tampilan Form Cetak Laporan:				
				

22	Klik → Cetak Laporan	Menampilkan Informasi download data laporan panen karyawan	✓	✓
----	----------------------	--	---	---


Tampilan Informasi Download Data Laporan Panen :




23	Folder PDF Lpaoran	Menampilkan laporan panen karyawan berbentuk pdf	✓	✓
----	--------------------	--	---	---

Tampilan Laporan Panen Berbentuk PDF



24	Klik → Menu <i>Logout</i>	Menampilkan Infromasi keluar aplikasi (Tidak / Keluar)	✓	✓
Tampilan Infromasi <i>Logout</i> :				
				

Validator I


SOFYAN

Mandor Lapangan

Validator II


LEGIMAN

Karyawan Lapangan

LAMPIRAN II SURAT IJIN RISET

PT Perkebunan Nusantara II
Tanjung Morawa

Tanjung Morawa, 30 Maret 2021

Nomor : 2.6/X/ 94 /III/2021

Hal : **PENDIDIKAN**
Izin Riset

Kepada Yth. :

Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan
Jl. Willem Iskandar Pasar V Medan estate
20371

Menghunjuk surat Saudara No : B.054/ST.I/ST.V.2/TL.00/01/2021 mengenai hal tersebut diatas, dengan ini disampaikan bahwa pada prinsipnya PT Perkebunan Nusantara II dapat memberikan izin kepada mahasiswa yang namanya tersebut dibawah ini dengan ketentuan tetap mematuhi **Protokol COVID -19** yang berlaku di Perusahaan dan membawa surat keterangan **Hasil Rapid Test Anti Gen**. Adapun nama tersebut adalah :

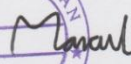
No.	Nama Mahasiswa	NIM	Program Studi
1	Heru Irfansyah	0702163065	Sistem Informasi

Untuk melaksanakan Riset di Kebun Bandar Klippa PTPN II pada tanggal 31 Maret s.d 08 April 2021 dengan judul penelitian "**Sistem Monitoring Aktivitas Lapangan Karyawan Dengan Metode Lock GPS Berbasis Cloud Pada PTP. Nusantara II**". Segala biaya yang berkenaan dengan kegiatan tersebut ditanggung oleh mahasiswa yang bersangkutan.

Kepada mahasiswa yang bersangkutan diharuskan menyampaikan hasil penelitian selama di Kebun Bandar Klippa yang diketahui oleh Manajer . Selanjutnya menyerahkan 1 (satu) exemplar Laporan hasil Penelitian ke Bagian SDM apabila telah selesai.

Demikian disampaikan agar Saudara maklum.

PT Perkebunan Nusantara II
Bagian Sumber Daya Manusia


Drs. M. T. Siagian
Kepala Bagian

Tembusan :
- 2.BKL
- Pertinggal

LAMPIRAN III
SURAT SELESAI RISET

PT Perkebunan Nusantara II
Tanjung Morawa

SURAT KETERANGAN

Nomor : 2.6/S.Ket/ 58 /IV/2021

Tanggal : 09 April 2021


Dengan ini kami sampaikan bahwa Mahasiswa Program Studi Sistem Informasi Universitas Islam Negeri Sumatera Utara Medan yang namanya tersebut di bawah ini :

No.	Nama Mahasiswa	NIM	Program Studi
1.	Heru Irfansyah	0702163065	Sistem Informasi

Benar telah menyelesaikan Riset di Kebun Bandar Klippa PT Perkebunan Nusantara II pada tanggal 31 Maret 2021 s.d 08 April 2021.

Demikian Surat Keterangan ini diperbuat untuk dapat dipergunakan seperlunya.

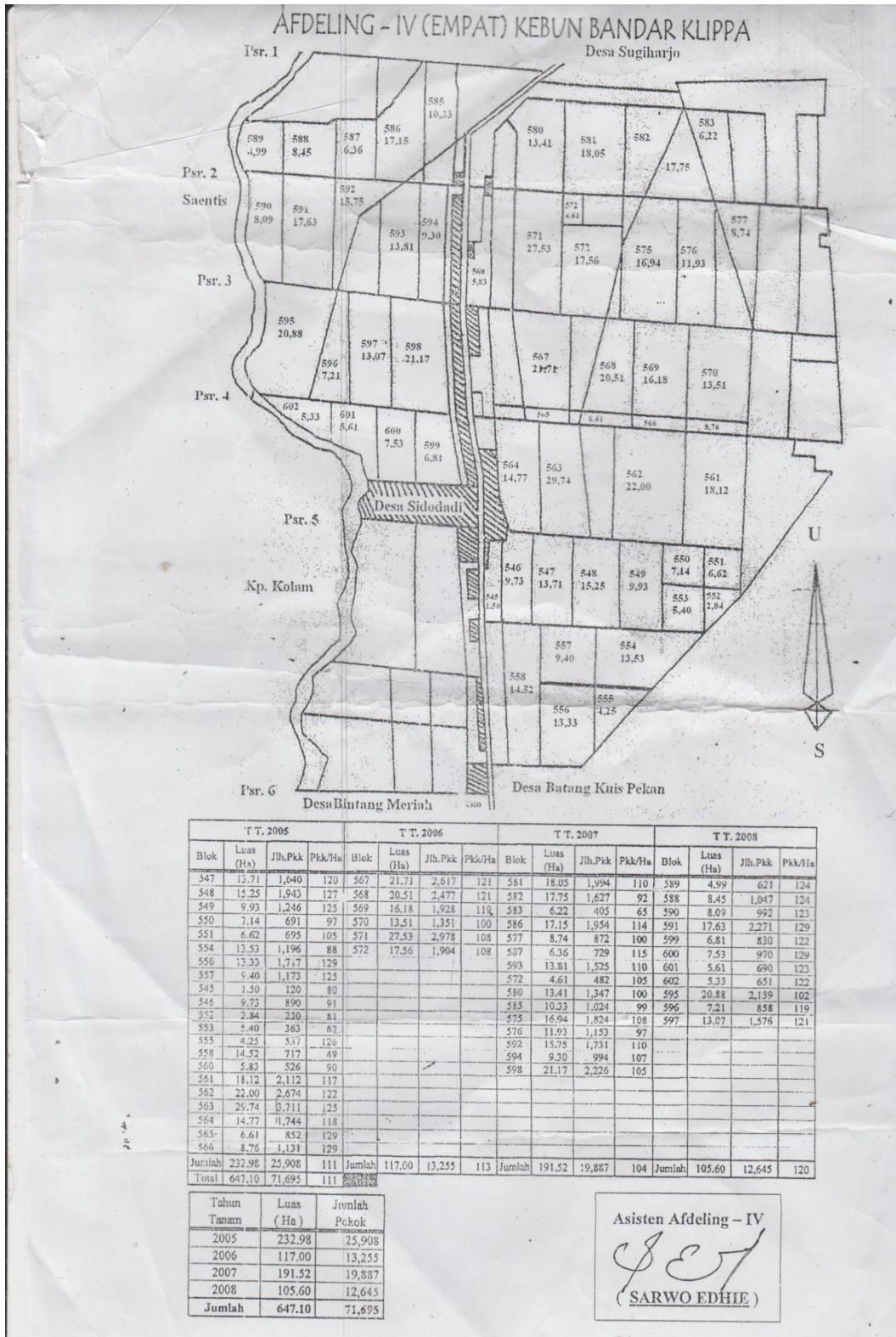
PT Perkebunan Nusantara II
Bagian Sumber Daya Manusia


Drs. M.T. Siagian
Kepala Bagian

Tembusan:

- Pertinggal

LAMPIRAN IV
 PETA KEBUN AFDELING - IV



LAMPIRAN V
DATA HASIL PANEN KARYAWAN

NO	NAMA	Apr-21																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	JARNO	80	50	125	-	100	90	80	80	110	90	110	80	85	-	90	80	100	100	150	80	85	50	80	115	100	110	80	80	
2	SAPHO	80	50	110	100	70	75	110	80	55	110	110	105	85	90	-	60	100	120	110	50	75	50	55	62	120	100	100		
3	IKAN	85	50	105	-	100	100	100	80	120	100	150	100	100	100	85	105	100	100	110	100	-	55	120	100	110	100	100		
4	GATO	75	50	110	110	70	75	110	80	55	110	110	100	80	90	100	60	100	110	80	50	75	50	55	60	100	100	100		
5	PONIRAN	70	50	80	100	85	70	65	65	70	80	80	80	90	70	70	75	80	80	80	70	65	-	70	85	75	75	80		
6	LESMAN	70	50	80	100	85	70	65	65	70	80	80	80	90	70	70	75	80	80	80	-	65	-	70	85	75	75	80		
7	MSDI	70	50	80	-	85	70	65	65	70	80	80	80	90	70	70	75	80	80	80	70	65	-	70	85	75	75	80		
8	HARIL	70	-	80	100	110	100	80	70	110	70	80	11	80	70	75	110	-	140	100	80	90	50	-	90	100	100	100		
9	ANDIK	75	50	80	-	105	85	80	85	125	70	80	85	75	70	80	135	100	135	-	80	90	50	90	130	85	90	100		
10	ABY BAYU	-	50	100	90	55	100	90	80	150	100	80	100	100	80	90	100	100	110	100	100	-	-	60	80	90	100	100		
11	ANDIP	90	50	110	90	115	100	90	80	-	110	80	125	110	80	100	130	100	100	130	100	100	-	115	100	100	-	100		
12	HERMAN	100	-	100	280	140	172	105	125	210	185	230	125	125	110	100	120	120	180	110	105	125	50	140	110	120	90	110		
13	BMO	70	53	90	100	80	136	100	85	125	115	110	150	120	135	-	140	90	120	85	100	100	50	110	80	80	150	85		
14	PANADAN	100	50	80	130	90	100	90	90	-	70	80	100	80	90	80	12	90	90	90	-	100	-	90	100	150	120	90		
15	REZA	80	50	125	100	105	85	80	90	110	100	90	85	90	80	90	80	100	90	105	80	80	50	80	115	100	-	80		
16	SUNARHADI	100	55	120	140	90	105	105	-	100	150	105	110	115	105	110	110	75	150	120	140	140	50	-	110	120	120	120		
17	BANBANG	75	47	110	150	125	105	90	80	90	130	130	105	-	80	100	100	60	130	180	120	100	50	80	100	130	130	100		

NO	NAMA	MARET 2021																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	ABRICO	70	90	50	70	60	80		70	80	90	90	85	90	90	70		100	70		80	90	75	75	100	100	100	100	100	90
2	SARAO	85	100	70	60	70	95		80	90	90	80	85	85	90	90	105	100	75		80	80	75	115	100	90	110	100		
3	IKAN	80	100	70	95	80	80		85	90	100	100	100	105	85	90	105	100	110		80	85	90	85	100	100	110	100	90	
4	GATC	70	100	70	50	60	80		70	80	100	90	85	90	90	95	100	110			80	90	75	75	100	100	100	100	95	
5	PONIRAN	80	90	65	65	70	80		80	65	70	75	90	70	65	70	80	85	70		65	70	65	75	75	80	100	100		
6	LESMAN	60	65	65	65	70	80		60	65	70	75	90	70	65	70	80	85	70		65	70	65	70	75	80	150	150	80	
7	INSO	60	65	65	65	70	80		60	65	70	75	90	70	65	70	80	85	70		65	70	65	70	75	80	150	100	120	
8	HARILL	60	100	65	0	60	80		70	70	80			90	65	70	75	100	100		80	100	80	70	70	115	115	80	80	
9	ANDRIK	60	80	70	70	60	65		105	70	110	90	75	100	80	75	75	100	90		90		80	75	70	90	110	100	90	
10	ARYBAYU	100	80	70	70	80			90	90	85	100	85	85	85	90	90	90	95		115		90	80	90	90	100	100	70	
11	ANDRIP	100	100	70	70	70	80		90	110	100	100	130	100	80	90	120	105	120		110	100	100	100	100	90	110	110	100	
12	HERMAN	80	80	85	70	70	80		60	90	90	100	90	100	90	85	140	130	105		80	90	70	120	134	124	120	110	150	
13	EMO	100	110	60	85	90	70		80	100	100	150	110	90	80	90	100	170			90	70	90	95	90	100	105	110	80	
14	PANAGAN	80	90	0	80	70	80		70	70	70	80	80	90	80		75	100	80		80	90	70	100	70	50	100	115	90	
15	REZA	90	95	75	85	70	85		80	90	90	80	85	90	100	90	105	100	80		100	90	90	80		110	90	100	95	
16	SUMARHADI	80	90	105	85	80	95		100	110	110	130	105	60	105	110	120	150	150		110	110	90	85	85	135	120	80	110	
17	BAMBANG	80	85	60	95		92		100			80	110	95	85	90	85	100	105		80	80	80	120	80	180	100	120	90	

No	Nama	FEBRUARI 2021																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	SURPAKI	75	85	80	50	75	80	85	90	.	.	70	75	80	100	70	75	80	100	75	100	70	70	80	80	80	80	70	90	70	50	60
2	JARWO	70	90	90	90	60	60	70	80	80	80	70	80	80	80	70	70	70	70	85	100	85	100	70	60	.	70	88	90	70	100	
3	IVAN	87	88	90	90	98	85	80	50	75	80	85	90	70	70	75	80	100	70	75	80	100	75	100	70	98	90	98	90	90	90	
4	GATO	88	90	.	90	90	.	85	80	50	75	80	85	90	70	75	80	100	70	75	80	100	75	100	70	90	90	80	80	50	84	90
5	POURANI	80	70	75	80	100	70	75	80	100	75	100	70	80	90	60	67	90	68	.	90	65	67	90	98	78	90	45	45	90	90	
6	LESMAN	90	90	90	90	.	.	90	90	70	75	80	100	70	75	80	100	75	100	70	90	.	.	90	90	90	90	90	90	90	90	90
7	MASDI	70	90	70	75	80	100	70	75	80	100	75	100	70	85	80	50	75	80	85	90	70	75	80	100	70	75	80	100	75	100	
8	HARBU	70	70	75	80	100	70	75	80	100	75	100	70	.	90	70	75	80	100	70	75	80	100	75	100	70	90	.	90	90	.	
9	ANDRIK	80	85	80	50	75	80	85	90	.	90	70	75	80	100	70	75	80	100	75	100	70	.	75	90	90	90	90	90	90	90	90
10	ANDRIP	90	70	75	80	100	70	75	80	100	75	100	70	90	.	70	75	80	100	70	75	80	100	75	100	70	90	90	90	90	90	
11	ABRYAYU	78	70	75	80	100	70	75	80	100	75	100	70	70	75	80	100	70	75	80	100	75	100	70	80	100	75	100	70	88	90	
12	HERMAN	100	67	88	.	90	70	75	80	100	70	75	80	100	75	100	70	50	75	80	85	90	90	70	75	80	100	70	75	80	100	
13	BINO	88	70	75	80	100	70	75	80	100	75	100	70	85	90	67	78	98	90	88	70	75	80	100	70	75	80	100	75	100	70	
14	KATNO	90	90	67	85	80	50	75	80	85	90	77	55	85	80	50	75	80	85	90	90	67	78	98	90	88	56	65	76	88	90	
15	REJA	70	70	75	80	100	70	75	80	100	75	100	70	98	78	88	90	67	67	56	87	65	88	56	56	45	45	90	90	78	97	
16	SUMARNO	88	85	80	50	75	80	85	90	70	75	80	100	70	75	80	100	75	70	75	80	100	70	75	80	100	75	100	70	90	90	
17	BAMBANG	70	70	75	80	100	70	75	80	100	75	100	70	50	75	80	85	90	90	90	90	70	75	80	100	70	75	80	100	75	100	70

LAMPIRAN VI
DATA KARYAWAN PANEN WILAYAH AFDELING IV

PTP. Nusantara II Perk Bandar Klippa
Data Karyawan Lapangan Wilayah Afdeling IV

NO	Nama	Alamat	NO HP	Status Karyawan
1	Bambang	Jln. Batang Kuis gg Raja Wali no 3	085265738494	Karyawan Lapangan
2	Sumardi	Jln. Batang Kuis Pasra 10	081374719475	Karyawan Lapangan
3	Reja	Jln. Batang Jambu gg Mawar 24	081246800764	Karyawan Lapangan
4	Katino	Jln. Batang Kuis no 33	081274748405	Karyawan Lapangan
5	Bimo	Jln. Batang Jambu no 33	081264738495	Karyawan Lapangan
6	Herman	Jln. Batang Kuis gg Mawar 3	081273748488	Karyawan Lapangan
7	Abybayu	Jln. Batang Kuis Pasar 10 No10	081264738292	Karyawan Lapangan
8	Andri P	Jln. Batang Kuis Pasar 10	081264738203	Karyawan Lapangan
9	Andri K	Jln. Batang Kuis gg Mawar 23	081364748292	Karyawan Lapangan
10	Hasrul	Jln. Batang Jambu gg Mawar 7	081253739273	Karyawan Lapangan
11	Misdi	Jln. Tanjung Morawa No. 25	082366126754	Karyawan Lapangan
12	Legiman	Jln. Batang Jambu Dan I	082373648401	Karyawan Lapangan
13	Poniran	Jln. Batang Jambu Dan V	085252737736	Karyawan Lapangan
14	Gato	Jln. Batang Jambu gg Mawar 24	085726385927	Karyawan Lapangan
15	Iwan	Jln. Tanjung Morawa	087752648198	Karyawan Lapangan
16	Jarwo	Jln. Batang Kuis Per 10	081256837303	Karyawan Lapangan
17	Suriadi	Jln. Batang Jambu gg Mawar 08	082576137565	Karyawan Lapangan

Mandor Afdeling IV



SOFYAN

LAMPIRAN VII

Source Code

AbsenAktivity.kt (Mandor)

```
esponse(message: String) {
    alertIOS.setSubtitle(message)
    alertIOS.build().show()
}package
com.heru.mandorptpn.ui.absen
import
android.annotation.SuppressLin
t
import android.content.Intent
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import android.util.Log
import
android.widget.ImageView
import
androidx.recyclerview.widget.R
ecyclerView
import
com.gdacciaro.iOSDialog.iOSDia
log
import
com.gdacciaro.iOSDialog.iOSDia
logBuilder
import
com.gmail.samehadar.iosdialog.
IOSDialog
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.adapter.Ab
sensiAdapter
import
com.heru.mandorptpn.adapter.Ab
sensiItemListener
import
com.heru.mandorptpn.models.Abs
ensi
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class AbsenActivity :
AppCompatActivity(),
AbsensiView,
AbsensiItemListener {

    private lateinit var
absensiPresenter:
AbsensiPresenter
    private lateinit var
rvDataAbsensi: RecyclerView
    private lateinit var
adapterAbsensi: AbsensiAdapter
    private lateinit var
loadingDialog: IOSDialog
    private lateinit var
alertIOS: IOSDialogBuilder
    private lateinit var
toolbarBackArrow: ImageView
    companion object {
        const val
EXTRA_DETAIL_ABSEN =
"extra_detail_absen"
    }
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceSt
ate)
        setContentView(R.layout.activi
ty_absen)
        // layouting
        rvDataAbsensi =
findViewById(R.id.absen_recycl
er_view)
        toolbarBackArrow =
findViewById(R.id.toolbar_back
_arrow)
        // toolbar listener
        toolbarBackArrow.setOnClickLis
tener {
            super.onBackPressed()
        }
        // insialisasi loading ios
dialog
        loadingDialog =
IOSDialog.Builder(this)
            .setTitle("Memuat
Data")
            .setTitleColorRes(R.color.gray
)
            .setCancelable(false)
            .build()
        // insialisasi alert
dialog ios
        alertIOS =
IOSDialogBuilder(this)
            .setTitle("Perhatian")
```

```

.setBoldPositiveLabel(true)
.setCancelable(false)
.setPositiveListener("Oke")
{dialog: iAlertDialog? ->
dialog?.dismiss()
super.onBackPressed()
}
// presenter
    absensiPresenter =
AbsensiPresenter(this)
    // default adapter
    adapterAbsensi=
AbsensiAdapter(arrayListOf(),
this)
}
@SuppressLint("SimpleDateForma
t")
    override fun onResume() {
        super.onResume()
        val date = Date()
        val sdf =
SimpleDateFormat("yyyy-MM-dd")
        val tanggal =
sdf.format(date)
        Log.d("SDF", "tanggal:
$tanggal")
absensiPresenter.showAbsensi(t
anggal)
    }
    // implements absensi view
override fun showLoading()
{
    loadingDialog.show()
}
    // implements absensi view
override fun hideLoading()
{
    loadingDialog.hide()
}
    // implements absensi view
override fun
setResponseSuccess(listAbsensi
: ArrayList<Absensi>?) {
        // adapter dan rv
adapterAbsensi.resetListAbsens
i(listAbsensi ?:
arrayListOf())
        rvDataAbsensi.adapter
= adapterAbsensi
    }
    // implements absensi view
override fun setFailedR
// implements adapter
absensi item clicked
override fun
onClicked(data: Absensi) {

```

```

        val intent =
Intent(this,
DetailAbsensiActivity::class.j
ava)
intent.putExtra(EXTRA_DETAIL_A
BSEN, data)
        startActivity(intent)
    }
}

```

AbsensiPresenter.kt (Mandor)

```

package
com.heru.mandorptpn.ui.absen
import
com.heru.mandorptpn.models.Res
ponseAbsensi
import
com.heru.mandorptpn.models.Res
ponseAction
import
com.heru.mandorptpn.networking
.ApiClient
import
com.heru.mandorptpn.networking
.BackendEndpoint
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class AbsensiPresenter(val
view: AbsensiView) {
    fun showAbsensi(tanggal:
String) {
        view.showLoading()
        val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.getAbsensiByDate(tan
ggal).enqueue(object :
Callback<ResponseAbsensi> {
            override fun
onFailure(call:
Call<ResponseAbsensi>, t:
Throwable) {
                view.hideLoading()
                view.setFailedResponse(t.messa
ge ?: "")
            }
            override fun
onResponse(
                call:
Call<ResponseAbsensi>,
                response:
Response<ResponseAbsensi>
            ) {

```



```

        if
        (response.isSuccessful) {
        view.hideLoading()
            val status
        = response.body()?.isSuccess
            if (status
        != null && status == true) {
            val
        listAbsen =
        response.body()?.data
        view.setResponseSuccess(listAbsen)
            } else if
        (status != null && status ==
        false) {
        view.setFailedResponse(response
        e.body()?.message ?: "")
            }
            } else {

        view.hideLoading()
        view.setFailedResponse(response
        e.body()?.message ?: "")
            }
        })
    }
    fun
    storeAbsensi(statusKehadiran:
    String, idAbsensi: String) {
        view.showLoading()
        val getClient =
        ApiClient.getClient().create(B
        ackendEndpoint::class.java)

        getClient.updateAbsensiByStatu
        s(idAbsensi,
        statusKehadiran).enqueue(objec
        t : Callback<ResponseAction> {
            override fun
        onFailure(call:
        Call<ResponseAction>, t:
        Throwable) {
        view.hideLoading()
        view.setFailedResponse(t.messa
        ge ?: "")
            }
            override fun
        onResponse(
            call:
        Call<ResponseAction>,
            response:
        Response<ResponseAction>
        ) {
            if
        (response.isSuccessful) {

```

```

        view.hideLoading()
        view.setFailedResponse(response
        e.body()?.message ?: "")
            } else {
        view.hideLoading()
        view.setFailedResponse(response
        e.body()?.message ?: "")
            }
        })
    }
}

```

DetailAbsensiAktivty.kt (Mandor)

```

package
com.heru.mandorptpn.ui.absen
import android.Manifest
import
android.annotation.SuppressLin
t
import
android.content.pm.PackageMana
ger
import android.graphics.Color
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import android.util.Log
import android.widget.Button
import
android.widget.ImageView
import android.widget.TextView
import
androidx.appcompat.app.AlertDi
alog
import
androidx.core.app.ActivityComp
at
import
com.gdacciaro.iOSDialog.iOSDia
log
import
com.gdacciaro.iOSDialog.iOSDia
logBuilder
import
com.github.florent37.runtimepe
rmission.kotlin.askPermission
import
com.gmail.samehadar.iosdialog.
IOSDialog
import
com.google.android.gms.maps.Ca
meraUpdateFactory
import
com.google.android.gms.maps.Go

```

```

ogleMap
import
com.google.android.gms.maps.Ma
pView
import
com.google.android.gms.maps.On
MapReadyCallback
import
com.google.android.gms.maps.mo
del.LatLng
import
com.google.android.gms.maps.mo
del.MarkerOptions
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.helper.Str
ingToTimestamp
import
com.heru.mandorptpn.models.Abs
ensi
class DetailAbsensiActivity :
AppCompatActivity(),
OnMapReadyCallback,
AbsensiView {
    private lateinit var
txtTanggal: TextView
    private lateinit var
txtWaktu:TextView
    private lateinit var
txtStatusKehadiran: TextView
    private lateinit var
toolbarArrowBack: ImageView
    private var dataAbsensi:
Absensi? = null
    private var mMap:
GoogleMap? = null
    private lateinit var
mapView: MapView
    private lateinit var
loadingIOS: IOSDialog
    private lateinit var
alertIOS: IOSDialogBuilder
    private lateinit var
btnKonfirmasi: Button
    private lateinit var
absensiPresenter:
AbsensiPresenter
    private lateinit var
btnTidakHadir: Button
    companion object {
        const val
MAP_VIEW_BUNDLE_KEY =
"map_view_bundle_key"
        const val DEFAULT_ZOOM
= 17f
    }
}

@SuppressLint("SetTextI18n")
override fun
onCreate(savedInstanceState:
Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.acti
vity_detail_absensi)
    // check permission
    askPermissionLocation()
    // insialisasi
    layouting
        txtTanggal =
findViewById(R.id.detail_tangg
al_absen)
        txtWaktu =
findViewById(R.id.detail_waktu
_absen)
        txtStatusKehadiran =
findViewById(R.id.detail_statu
s_absen)
        toolbarArrowBack =
findViewById(R.id.detail_absen
_back_arrow)
        mapView =
findViewById(R.id.detail_absen
_mapview)
        btnKonfirmasi =
findViewById(R.id.detail_absen
si_btn)
        btnTidakHadir =
findViewById(R.id.detail_btn_t
idak_hadir)
    // insialisasi
    presenter
        absensiPresenter =
AbsensiPresenter(this)
    // insialisasi loading
    dialog
        loadingIOS =
IOSDialog.Builder(this)
            .setTitle("Mohon
Tunggu")
            .setTitleColorRes(R.color.gray
)
            .setCancelable(false)
            .build()
    // insialisasi alert
    dialog ios
        alertIOS =
IOSDialogBuilder(this)
            .setBoldPositiveLabel(true)
    // toolbar arrow back
    toolbarArrowBack.setOnClickListener {
        super.onBackPressed()
    }
}

```

```

}

// bundle data mapView
var mapViewBundle:
Bundle? = null
    if (savedInstanceState
!= null)
        mapViewBundle =
savedInstanceState.getBundle(M
AP_VIEW_BUNDLE_KEY)
        // generate map
viewMapView.onCreate(mapViewBu
ndle)
mapView.getMapAsync(this)
    // get parcelable data
    dataAbsensi =
intent.getParcelableExtra<Abse
nsi>(AbsenActivity.EXTRA_DETAI
L_ABSEN)
        txtTanggal.text =
StringToTimestamp.setLongDate(
dataAbsensi?.tanggal)
        txtWaktu.text =
dataAbsensi?.waktu + " WIB"
txtStatusKehadiran.text =
dataAbsensi?.statusKehadiran
        if
(dataAbsensi?.tipeAbsen ==
"pengajuan")
btnKonfirmasi.setText("Konfirm
asi Izin")
        // kondisi ketika
status != pending
        if
(dataAbsensi?.statusKehadiran
!= "pending") {
btnKonfirmasi.isEnabled =
false
btnKonfirmasi.setBackgroundColor(
Color.GRAY)
        }
        // listener button
konfirmasi
btnKonfirmasi.setOnClickListener{
if (dataAbsensi?.tipeAbsen ==
"pengajuan") {
alertIOS.setTitle("Konfirmasi
Izin")
.setBoldPositiveLabel(true)
.setSubtitle("Apakah anda
yakin karyawan dengan nama
${dataAbsensi?.nama} mengambil
izin kerja ?")
.setPositiveListener("Iya"){di
alog: iOSDialog? ->
absensiPresenter.storeAbsensi(
"izin", dataAbsensi?.idAbsensi
?: "")
dialog?.dismiss()
}
.setNegativeListener("Batal"){
dialog: iOSDialog? ->
dialog?.dismiss()
}
.build()
                .show()
            } else {
alertIOS.setTitle("Konfirmasi
Kehadiran")
.setBoldPositiveLabel(true)
.setSubtitle("Apakah anda
yakin karyawan dengan nama
${dataAbsensi?.nama} hadir
ditempat ?")
.setPositiveListener("Iya"){di
alog: iOSDialog? ->
absensiPresenter.storeAbsensi(
"hadir",
dataAbsensi?.idAbsensi ?: "")
dialog?.dismiss()
}
.setNegativeListener("Batal"){
dialog: iOSDialog? ->
dialog?.dismiss()
}
.build()
                .show()
            }
        }
// listener button tidak hadir
btnTidakHadir.setOnClickListener {
alertIOS.setTitle("Konfirmasi
Kehadiran")
.setBoldPositiveLabel(true)
.setSubtitle("Apakah anda
yakin karyawan dengan nama
${dataAbsensi?.nama} TIDAK
HADIR ?")
.setPositiveListener("Iya"){di
alog: iOSDialog? ->
absensiPresenter.storeAbsensi(
"tidak hadir",
dataAbsensi?.idAbsensi ?: "")
dialog?.dismiss()
}
.setNegativeListener("Batal"){
dialog: iOSDialog? ->
dialog?.dismiss()
}
.build()

```

```

        .show()
    }
}
    override fun onResume() {
        mapView.onResume()
        super.onResume()
    }
    // saved instance state
    override fun
onSaveInstanceState(bundle:
Bundle) {
super.onSaveInstanceState(bund
le)
askPermissionLocation()
    var mapviewBundle =
bundle.getBundle(MAP_VIEW_BUND
LE_KEY)
    if (mapviewBundle ==
null){
        // masukkan bundle
data
        mapviewBundle =
Bundle()
bundle.putBundle(MAP_VIEW_BUND
LE_KEY, mapviewBundle)
    }
    // set ke mapView
mapView.onSaveInstanceState(ma
pviewBundle)
    }
    // fungsi permission
lokasi
    private fun
askPermissionLocation() {
askPermission(Manifest.permiss
ion.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COA
RSE_LOCATION) {
Log.d("PERMISSION", "lokasi di
izinkan : true ")
}.onDeclined {
// Apabila permission ditolak
    if
(it.hasDenied()) {
it.denied.forEach{
}
// Menbuka alertdialog
AlertDialog.Builder(this@DetailAbsensiActivity)
.setMessage("Aplikasi ini
membutuhkan izin lokasi")
.setPositiveButton("Iya") { _,
_ ->
it.askAgain()
}
.setNegativeButton("Tidak") {
dialog, _ ->
dialog.dismiss()
}
.show()
}
// Apabila
permissions lokasi di tolak
selamanya
    if
(it.hasForeverDenied()) {
it.foreverDenied.forEach{
// menuju settings
it.goToSettings()
}
}
}
    override fun
onMapReady(googleMap:
GoogleMap?) {
mapView.onResume()
mMap = googleMap
// Tanyakan lagi ask
permission
askPermissionLocation()
    val permissiOne =
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FIN
E_LOCATION)
    val permissiTwo =
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COA
RSE_LOCATION)
    val pckgMgr =
PackageManager.PERMISSION_GRAT
ED
// Tanyakan permission
    if (permissiOne !=
pckgMgr && permissiTwo !=
pckgMgr ) {
        return
    }
    // set marker
berdasarkan lat dan lng
    val markerOptions =
MarkerOptions()
markerOptions.position(LatLng(
dataAbsensi?.lat?.toDouble()
?: 0.0,
dataAbsensi?.lng?.toDouble()?:
0.0))
markerOptions.title(dataAbsens
i?.nama ?: "")

```

```

markerOptions.snippet("Absen
di wilayah
${dataAbsensi?.kodeWilayah}")
mMap?.addMarker(markerOptions)
    // Move camera ke posisi
    ini
mMap?.moveCamera(CameraUpdateF
actory.newLatLngZoom(LatLng(da
taAbsensi?.lat?.toDouble() ?:
0.0,
dataAbsensi?.lng?.toDouble()
?: 0.0), DEFAULT_ZOOM))
    }
    override fun showLoading()
    {
        loadingIOS.show()
    }
    override fun hideLoading()
    {
        loadingIOS.hide()
    }
    override fun
setResponseSuccess(listAbsensi
: ArrayList<Absensi>?) {
        // DO NOTHING
    }
    override fun
setFailedResponse(message:
String) {
        val x =
iOSDialogBuilder(this)
        .setTitle("Perhatian!")
        .setSubtitle(message)
        .setBoldPositiveLabel(true)
        .setPositiveListener("Oke") {
            it.dismiss()
            finish()
        }
    }
x.build().show()
    }
}

```

AktifitasAktivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.aktifit
as
import
android.annotation.SuppressLin
t
import android.content.Intent
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import

```

```

android.widget.ImageView
import
androidx.recyclerview.widget.R
ecyclerView
import
com.gdacciaro.iOSDialog.iOSDia
log
import
com.gdacciaro.iOSDialog.iOSDia
logBuilder
import
com.gmail.samehadar.iosdialog.
IOSDialog
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.adapter.Ak
tifitasAdapter
import
com.heru.mandorptpn.adapter.Ak
tifitasItemListener
import
com.heru.mandorptpn.models.Akt
ifitas
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class AktifitasActivity :
AppCompatActivity(),
AktifitasView,
AktifitasItemListener {
    private lateinit var
aktifitasPresenter:
AktifitasPresenter
    private lateinit var
rvAktifitas: RecyclerView
    private lateinit var
adapterAktifitas:
AktifitasAdapter
    private lateinit var
loadingDialog: IOSDialog
    private lateinit var
alertIOS: iOSDialogBuilder
    private lateinit var
toolbarBackArrow: ImageView
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceSt
ate)
        setContentView(R.layout.activi
ty_aktifitas)
        // insialisasi
        layouting
        rvAktifitas =

```

```

findViewById(R.id.aktifitas_re
cycler_view)
        toolbarBackArrow =
findViewById(R.id.toolbar_back
_arrow)
        // toolbar listener
toolbarBackArrow.setOnClickListener {
super.onBackPressed()
}
// inialisasi loading ios
dialog
        loadingDialog =
IOSDialog.Builder(this)
        .setTitle("Memuat
Data")
        .setTitleColorRes(R.color.gray
)
        .setCancelable(false)
        .build()
        // inialisasi alert
dialog ios
        alertIOS =
IOSDialog.Builder(this)
        .setTitle("Perhatian")
        .setBoldPositiveLabel(true)
        .setCancelable(false)
        .setPositiveListener("Oke")
        {dialog: IOSDialog? ->
dialog?.dismiss()
super.onBackPressed()
}
// inialisasi presenter
        aktifitasPresenter =
AktifitasPresenter(this)
        // inialisasi adapter
        adapterAktifitas =
AktifitasAdapter(arrayListOf()
, this)
    }
@SuppressLint("SimpleDateForma
t")
        override fun onResume() {
            super.onResume()
            val date = Date()
            val sdf =
SimpleDateFormat("yyyy-MM-dd")
            val tanggal =
sdf.format(date)
            aktifitasPresenter.showAktfifi
tasToday(tanggal)
        }
        // implementas view
aktifitas
            override fun showLoading()
{
            loadingDialog.show()
        }
        // implementas view
aktifitas
            override fun hideLoading()
{
            loadingDialog.hide()
        }
        // implementas view
aktifitas
            override fun
responseDataAktifitas(response
Data: ArrayList<Aktifitas>) {
                // adapter dan rv
                adapterAktifitas.resetListAktif
itas(responseData)
                rvAktifitas.adapter =
                adapterAktifitas
            }
        // implementas view
aktifitas
            override fun
failedResponse(failedMessage:
String) {
                alertIOS.setSubtitle(failedMes
sage) alertIOS.build().show()
            }
        // implements item clicked
aktifitas
            override fun
onItemAktifitasClicked(aktifit
as: Aktifitas) {
                val intent =
Intent(this,
DetailAktifitasActivity::class
.java)
                intent.putExtra(EXTRA_DETAIL_A
KTIFITAS, aktifitas)
                startActivity(intent)
            }
        companion object {
            const val
EXTRA_DETAIL_AKTIFITAS =
"extra_detail_aktifitas"
        }
    }
}

```

AktivitasPresenter.kt (Mandor)

```

package
com.heru.mandorptpn.ui.aktifit
as
import
com.heru.mandorptpn.models.Res
ponseAktifitas
import
com.heru.mandorptpn.networking

```



```

itas_tanggal)
        txtWaktu =
findViewById(R.id.detail_aktifitas_waktu)
        txtKeterangan =
findViewById(R.id.detail_aktifitas_keterangan)
        backButton =
findViewById(R.id.detail_aktifitas_back_arrow)
backArrow.setOnClickListener {
super.onBackPressed()
}
val aktifitas =
intent.getParcelableExtra<Aktifitas>(AktifitasActivity.EXTRA_DETAIL_AKTIFITAS)
        Glide.with(this)
.load(ApiClient.getImgURL() +
"${aktifitas?.fileFoto}")
.placeholder(R.drawable.bg_placeholder)
.into(imgAktifitas)
        txtKeterangan.text =
aktifitas?.keterangan
        txtWaktu.text =
aktifitas?.waktu + " WIB"
        txtTanggal.text =
StringToTimestamp.setLongDate(
aktifitas?.tanggal)
    }
}

```

FormKaryawanActivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.karyawan
import
android.annotation.SuppressLint
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.*
import
com.gdacciaro.iOSDialog.iOSDialog
import
com.gdacciaro.iOSDialog.iOSDialogBuilder
import
com.gmail.samehadar.iosdialog.iOSDialog

```

```

import com.heru.mandorptpn.R
import
com.heru.mandorptpn.models.Karyawan
class FormKaryawanActivity :
AppCompatActivity(),
KaryawanView {
    private var karyawan:
Karyawan?= null
    private lateinit var
edtNama: EditText
    private lateinit var
edtUsername: EditText
    private lateinit var
edtPassword: EditText
    private lateinit var
edtAlamat: EditText
    private lateinit var
edtNomorTelepon: EditText
    private lateinit var
imgBack: ImageView
    private lateinit var
titleLabel: TextView
    private lateinit var
btnSimpan: Button
    private lateinit var
btnHapus: Button
    private lateinit var
karyawanPresenter:
KaryawanPresenter
    private lateinit var
loadingDialog: IOSDialog
    private lateinit var
alertIOS: IOSDialogBuilder
    @SuppressWarnings("SetTextI18n")
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_form_karyawan)
// layouting
edtNama =
findViewById(R.id.form_karyawan_nama)
        edtUsername =
findViewById(R.id.form_karyawan_username)
        edtPassword =
findViewById(R.id.form_karyawan_password)
        edtAlamat =
findViewById(R.id.form_karyawan_alamat)
        edtNomorTelepon =

```



```

findViewById(R.id.form_karyawa
n_nohp)
        imgBack =
findViewById(R.id.detail_karya
wan_back_arrow)
            titleToolbar =
findViewById(R.id.detail_karya
wan_txt_toolbar)
                btnSimpan =
findViewById(R.id.form_btn_sim
pan)
                    btnHapus =
findViewById(R.id.form_btn_hap
us)
                        // inialisasi loading
ios dialog
                            loadingDialog =
IOSDialog.Builder(this)
                                .setTitle("Memuat
Data").setTitleColorRes(R.col
or.gray)
                                    .setCancelable(false)
                                        .build()

                                // inialisasi alert
dialog ios
                                    alertIOS =
iOSDialogBuilder(this)
                                        .setTitle("Perhatian")
                                        .setBoldPositiveLabel(true)
                                        .setCancelable(false)
                                        .setPositiveButton("Oke")
                                        {dialog: iOSDialog? ->
dialog?.dismiss()
super.onBackPressed()
}
// inialisasi img back
imgBack.setOnClickListener {
super.onBackPressed() }
// inialisasi presenter
        karyawanPresenter =
KaryawanPresenter(this)
            // inialisai karyawan
            karyawan =
intent.getParcelableExtra<Kary
awan>(KaryawanActivity.EXTRA_D
ETAIL_KARYAWAN)
                val
isTambahKaryawanState =
intent.getBooleanExtra("TAMBAH
_KARYAWAN", false)
                    if
(isTambahKaryawanState) {
                        titleToolbar.text
= "Tambah Karyawan"
                            btnSimpan.text =
"Simpan"
                                btnHapus.visibility =
View.GONE
                                    }
                                        // Jika karyawan tidak
null
                                            if (karyawan != null)
{edtNama.setText(karyawan?.nam
a)
edtUsername.setText(karyawan?.
username)
edtPassword.setText(karyawan?.
password)
edtAlamat.setText(karyawan?.al
amat)
edtNomorTelepon.setText(karyaw
an?.noTelepon)
}
// simpan listener
btnSimpan.setOnClickListener {
val idKaryawan =
karyawan?.idKaryawan
        val nama =
edtNama.text.toString()
            val username =
edtUsername.text.toString()
                val password =
edtPassword.text.toString()
                    val alamat =
edtAlamat.text.toString()
                        val noHp =
edtNomorTelepon.text.toString(
)
                            if (isTambahKaryawanState)
{
karyawanPresenter.addKaryawan(
nama, username, password,
alamat, noHp)
} else {
karyawanPresenter.ubahKaryawan
(idKaryawan, nama, username,
password, alamat, noHp)
}
}
// hapus listener
btnHapus.setOnClickListener {
val idKaryawan =
karyawan?.idKaryawan ?: ""
iOSDialogBuilder(this)
    .setTitle("Konfirmasi")
    .setSubtitle("Yakin ingin
hapus data ini ?")
    .setBoldPositiveLabel(true)
    .setCancelable(false)
    .setPositiveButton("Oke")
    {dialog: iOSDialog? ->

```

```

karyawanPresenter.hapusKaryawa
n(idKaryawan)
}
.setNegativeButton("Batal")
{ dialog: iosDialog ->
dialog.dismiss()
}
.build()
                .show()
}
}
override fun showLoading()
{
    loadingDialog.show()
}
override fun hideLoading()
{
    loadingDialog.hide()
}
override fun
karyawanResponse(list:
ArrayList<Karyawan>) {
    // do nothing
}
override fun
messageResponse(message:
String) {

alertIOS.setSubtitle(message)

alertIOS.build().show()
}
}

```

KaryawanAktivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.karyawa
n
import android.content.Intent
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import
android.widget.ImageView
import
androidx.recyclerview.widget.R
ecyclerView
import
com.gdacciaro.iOSDialog.iOSDia
log
import
com.gdacciaro.iOSDialog.iOSDia
logBuilder
import
com.gmail.samehadar.iosdialog.

```

```

IOSDialog
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.adapter.Ka
ryawanAdapter
import
com.heru.mandorptpn.adapter.Ka
ryawanItemListener
import
com.heru.mandorptpn.models.Kar
yawan
class KaryawanActivity :
AppCompatActivity(),
KaryawanView,
KaryawanItemListener {
    private lateinit var
arrowBack: ImageView
    private lateinit var
addKaryawanImg: ImageView
    private lateinit var
karyawanAdapter:
KaryawanAdapter
    private lateinit var
karyawanPresenter:
KaryawanPresenter
    private lateinit var
rvKaryawan: RecyclerView
    private lateinit var
loadingDialog: IOSDialog
    private lateinit var
alertIOS: iosDialogBuilder
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceSt
ate)
        setContentView(R.layout.activi
ty_karyawan)
        // layouting
        arrowBack =
findViewById(R.id.toolbar_back
_arrow)
        addKaryawanImg =
findViewById(R.id.toolbar_add_
karyawan)
        rvKaryawan =
findViewById(R.id.karyawan_rec
ycler_view)
        // insialisasi loading
ios dialog
        loadingDialog =
IOSDialog.Builder(this)
            .setTitle("Memuat
Data")
            .setTitleColorRes(R.color.gray
)

```

```

.setCancelable(false)
        .build()
        // inialisasi alert
dialog ios
    alertIOS =
iOSDialogBuilder(this)
    .setTitle("Perhatian")
    .setBoldPositiveLabel(true)
    .setCancelable(false)
    .setPositiveButton("Oke")
{dialog: iOSDialog? ->
dialog?.dismiss()
super.onBackPressed()
}
// listener arrow back
arrowBack.setOnClickListener {
super.onBackPressed()
}
// listener add btn karyawan
toolbar
addKaryawanImg.setOnClickListener {
val intent = Intent(this,
FormKaryawanActivity::class.java)
intent.putExtra("TAMBAH_KARYAWAN", true)
startActivity(intent)
}
// presenter
    karyawanPresenter =
KaryawanPresenter(this)
    // adapter
    karyawanAdapter =
KaryawanAdapter(arrayListOf(),
this)
    }
    override fun onResume() {
super.onResume()karyawanAdapter
r.resetKaryawan()
karyawanPresenter.showAllKaryawan()
    }
    override fun showLoading()
{
    loadingDialog.show()
    }
    override fun hideLoading()
{
    loadingDialog.hide()
    }
    override fun
karyawanResponse(list:
ArrayList<Karyawan>) {
karyawanAdapter.setKaryawan(li

```

```

st)
        rvKaryawan.adapter =
karyawanAdapter
    }
    override fun
messageResponse(message:
String) {
    alertIOS.setSubtitle(message)
    alertIOS.build().show()
    }
    override fun
onItemClicked(karyawan:
Karyawan) {
        val intent =
Intent(this,
FormKaryawanActivity::class.java)
intent.putExtra(EXTRA_DETAIL_KARYAWAN, karyawan)
startActivity(intent)
    }
    companion object {
        const val
EXTRA_DETAIL_KARYAWAN =
"extra_detail_karyawan"
    }
}

```

KaryawanPresenter.kt (Mandor)

```

package
com.heru.mandorptpn.ui.karyawan
import
com.heru.mandorptpn.models.ResponseAction
import
com.heru.mandorptpn.models.ResponseKaryawan
import
com.heru.mandorptpn.networking.ApiClient
import
com.heru.mandorptpn.networking.BackendEndpoint
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class KaryawanPresenter(val
view: KaryawanView) {
    // show all karyawan
    fun showAllKaryawan() {
        // show loading
        view.showLoading()
        val getClient =
ApiClient.getClient().create(B

```

```

ackendEndpoint::class.java)
getClient.getAllKaryawan().enqueue(object :
Callback<ResponseKaryawan> {
    override fun
onFailure(call:
Call<ResponseKaryawan>, t:
Throwable) {
        // hide loading
view.hideLoading()
view.messageResponse(t.localizedMessage ?: "")
    }
    override fun
onResponse(
        call:
Call<ResponseKaryawan>,
        response:
Response<ResponseKaryawan>
    ) {
        val status =
response.body()?.isSuccess
        if (status !=
null && status == true) {
view.hideLoading()
view.karyawanResponse(response
.body()?.data ?:
arrayListOf())
        } else if
(status != null && status ==
false){
view.hideLoading()
view.messageResponse(response
.body()?.message ?: "")
        }
    }
})
// add karyawan
fun addKaryawan(
    namaKaryawan: String,
    username: String,
    password: String,
    alamat: String,
    noHp: String
) {
    //check is empty ?
    if
(namaKaryawan.isEmpty() ||
username.isEmpty()
||
password.isEmpty()
||
alamat.isEmpty()
||
noHp.isEmpty())
        {
view.messageResponse("Lengkapi
seluruh form terlebih dahulu")
return
        }
        view.showLoading()
        val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.tambahKaryawan(namaK
aryawan, username, password,
alamat, noHp)
            .enqueue(object :
Callback<ResponseAction> {
                override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()
view.messageResponse(t.localized
edMessage ?: "")
                }
                override fun
onResponse(
                    call:
Call<ResponseAction>,
                    response:
Response<ResponseAction>
                ) {
                    if
(response.isSuccessful) {
view.hideLoading()
view.messageResponse(response
.body()?.message ?: "")
                    } else {
view.hideLoading()
view.messageResponse(response
.body()?.message ?: "")
                    }
                }
            })
        // update karyawan
fun ubahKaryawan(
    idKaryawan: String?,
    namaKaryawan: String,
    username: String,
    password: String,
    alamat: String,
    noHp: String
) {
        //check is empty ?
        if
(namaKaryawan.isEmpty() ||
username.isEmpty())

```

```

||
        password.isEmpty()
||
        alamat.isEmpty()
||
        noHp.isEmpty())
    {
view.messageResponse("Lengkapi
seluruh form terlebih dahulu")
        return
    }
    view.showLoading()
    val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.ubahKaryawan(idKarya
wan, namaKaryawan, username,
password, alamat, noHp)
        .enqueue(object :
Callback<ResponseAction> {
            override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()
view.messageResponse(t.localiz
edMessage ?: "")
            }
            override fun
onResponse(
                call:
                Call<ResponseAction>,
                response:
                Response<ResponseAction>
            ) {
                if
                (response.isSuccessful) {
view.hideLoading()
view.messageResponse(response.
body()?.message ?: "")
                } else {
view.hideLoading()
view.messageResponse(response.
body()?.message ?: "")
                }
            }
        })
    }

    // hapus karyawan
    fun
hapusKaryawan(idKaryawan:
String) {
if(idKaryawan.isEmpty()) {
view.messageResponse("Id
Karyawan Empty")
        return
    }
    view.showLoading()
    val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.hapusKaryawan(idKary
awan)
        .enqueue(object :
Callback<ResponseAction>{
            override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()
view.messageResponse(t.localiz
edMessage ?: "")
            }
            override fun
onResponse(
                call:
                Call<ResponseAction>,
                response:
                Response<ResponseAction>
            ) {
                if
                (response.isSuccessful) {
view.hideLoading()
view.messageResponse(response.
body()?.message ?: "")
                } else {
view.hideLoading()
view.messageResponse(response.
body()?.message ?: "")
                }
            }
        })
    }
}

CetakLaporanActivity.kt
(Mandor)
package
com.heru.mandorptpn.ui.laporan
import android.Manifest
import
android.annotation.SuppressLin
t
import
android.app.DownloadManager
import
android.content.BroadcastRecei
ver
import android.content.Context

```

```

import android.content.Intent
import
android.content.IntentFilter
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import
android.widget.ImageView
import android.widget.Toast
import
androidx.appcompat.app.AlertDialog
import
com.gdacciaro.iOSDialog.iOSDialog
import
com.gdacciaro.iOSDialog.iOSDialogBuilder
import
com.github.florent37.runtimepermission.kotlin.askPermission
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.helper.DatePickerDialog
import
com.heru.mandorptpn.helper.DatePickerListener
import
com.heru.mandorptpn.models.Karyawan
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class CetakLaporanActivity :
AppCompatActivity(),
DatePickerListener,
LaporanView {
    private lateinit var
edtTanggalAwal: EditText
    private lateinit var
edtTanggalAkhir: EditText
    private lateinit var
btnCetak: Button
    private lateinit var
arrowBack: ImageView
    private lateinit var
presenter: LaporanPresenter
    private lateinit var
alertIOS: iOSDialogBuilder
    private var downloadID:
Long? = null
    // broadcast receiver
    private val
onDownloadComplete:
BroadcastReceiver = object :
BroadcastReceiver() {
        override fun
onReceive(context: Context,
intent: Intent) {
            //Fetching the
download id received with the
broadcast
            val id =
intent.getLongExtra(DownloadMa
nager.EXTRA_DOWNLOAD_ID, -1)
            //Checking if the
received broadcast is for our
enqueued download by matching
download id
            if (downloadID !=
null && downloadID == id) {
                Toast.makeText(this@CetakLapor
anActivity, "Download
Completed",
Toast.LENGTH_SHORT).show()
            }
        }
    }
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activi
ty_cetak_laporan)
        //layout
        edtTanggalAwal =
findViewById(R.id.cetak_edt_tg
l_awal)
        edtTanggalAkhir =
findViewById(R.id.cetak_edt_tg
l_akhir)
        btnCetak =
findViewById(R.id.cetak_btn_la
poran)
        arrowBack =
findViewById(R.id.toolbar_back
_arrow)

        // insialisasi alert
dialog ios
        alertIOS =
iOSDialogBuilder(this)
        .setTitle("Perhatian")
        .setBoldPositiveLabel(true)

```

```

.setCancelable(false)
.setPositiveButton("Oke")
{dialog: iOSDialog? ->
dialog?.dismiss()
}
// broadcast receiver
registerReceiver(onDownloadComplete,
IntentFilter(DownloadManager.ACTION_DOWNLOAD_COMPLETE))
    val karyawan =
intent.getParcelableExtra<Karyawan>("EXTRA_CETAK_DETAIL_KARYAWAN")
        presenter =
LaporanPresenter(this)
arrowBack.setOnClickListener {
super.onBackPressed()
}
edtTanggalAwal.setOnClickListener {
val datePickerDialog =
DatePickerDialog(this)
datePickerDialog.show(supportFragmentManager,
"TANGGAL_AWAL_PICKER")
}
edtTanggalAkhir.setOnClickListener {
val datePickerDialog =
DatePickerDialog(this)
datePickerDialog.show(supportFragmentManager,
"TANGGAL_AKHIR_PICKER")
}
btnCetak.setOnClickListener {
if
(edtTanggalAwal.text.toString().isEmpty() ||
edtTanggalAkhir.text.toString().isEmpty()) {

Toast.makeText(this, "Lengkapi tanggal awal dan akhir",
Toast.LENGTH_SHORT).show()
return@setOnClickListener
}
// cek permission
checkRuntimePermission(karyawan)
}
}

override fun onDestroy() {
super.onDestroy()
unregisterReceiver(onDownloadComplete)
}

}
private fun
checkRuntimePermission(karyawan: Karyawan?) {
askPermission(
Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE
) {

presenter.getUrlLaporan(
karyawan?.idKaryawan ?: "",
karyawan?.nama
?: "",
startDate,
endDate
)
}.onDeclined {
if (it.hasDenied()) {
it.denied.forEach{
Toast.makeText(this, it,
Toast.LENGTH_SHORT).show() }
// Menmbuka alerdialog
AlertDialog.Builder(this@CetakLaporanActivity)
.setMessage("Aplikasi ini membutuhkan izin storage")
.setPositiveButton("Izinkan")
{ _, _ ->
it.askAgain()
}
.setNegativeButton("Tidak") {
dialog, _ ->
dialog.dismiss()
}
.show()
} else if
(it.hasForeverDenied()) {
it.foreverDenied.forEach{
Toast.makeText(this, it,
Toast.LENGTH_SHORT).show() }
it.goToSettings()
}
}
}

@SuppressLint("SetTextI18n")
override fun
onDateSelected(tag: String,
year: Int, month: Int, day:
Int) {
if (tag ==
"TANGGAL_AWAL_PICKER") {

```

```

        val calendar =
Calendar.getInstance()
        calendar.set(year,
month, day)
        val dateFormat =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault())
        val dateFormat2 =
SimpleDateFormat("dd/M/yyyy",
Locale.getDefault())

edtTanggalAwal.setText(datefor
mat2.format(calendar.time))
        startDate =
dateFormat.format(calendar.tim
e)
        } else if (tag ==
"TANGGAL_AKHIR_PICKER") {
        val calendar =
Calendar.getInstance()
        calendar.set(year,
month, day)
        val dateFormat =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault())
        val dateFormat2 =
SimpleDateFormat("dd/M/yyyy",
Locale.getDefault())
edtTanggalAkhir.setText(datefo
rmat2.format(calendar.time))
        endDate =
dateFormat.format(calendar.tim
e)
        }
        else
Toast.makeText(this, "Gagal
parsing onDateSelected
Listener",
Toast.LENGTH_SHORT).show()
        }
        companion object {
            var startDate = ""
            var endDate = ""
        }
        override fun
isLoading(status: Boolean) {
            if (status) {
                btnCetak.setText("Mohon Tunggu
                ...")
            } else {
                btnCetak.setText("Cetak
                Laporan")
            }
        }
        override fun

```

```

responseKaryawan(listKaryawan:
ArrayList<Karyawan>) {
            // do nothing
        }
        override fun
messageResponse(message:
String) {
            alertIOS.setSubtitle(message)
            alertIOS.build().show()
        }
        override fun
pdfResponse(url: String) {
            startDownloading(url)
        }
        private fun
startDownloading(url: String)
        {
            //val request =
presenter.startDownloadLaporan
(url)
            val request =
presenter.startDownloadingForP
ie(url)
            // get download
service, dan antian file
            val manager =
getSystemService(Context.DOWNL
OAD_SERVICE) as
DownloadManager
            downloadID =
manager.enqueue(request)
        }
    }
}

```

LaporanActivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.laporan
import android.Manifest
import
android.annotation.SuppressLin
t
import
android.app.DownloadManager
import
android.content.BroadcastRecei
ver
import android.content.Context
import android.content.Intent
import
android.content.IntentFilter
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText

```



```

import
android.widget.ImageView
import android.widget.Toast
import
androidx.appcompat.app.AlertDialog
import
import com.gdacciaro.iOSDialog.iOSDialog
import
import com.gdacciaro.iOSDialog.iOSDialogBuilder
import
import com.github.florent37.runtimepermission.kotlin.askPermission
import com.heru.mandorptpn.R
import
import com.heru.mandorptpn.helper.DatePickerDialog
import
import com.heru.mandorptpn.helper.DatePickerListener
import
import com.heru.mandorptpn.models.Karyawan
import
import java.text.SimpleDateFormat
import java.util.*
import
import kotlin.collections.ArrayList
class CetakLaporanActivity :
AppCompatActivity(),
DatePickerListener,
LaporanView {
    private lateinit var
edtTanggalAwal: EditText
    private lateinit var
edtTanggalAkhir: EditText
    private lateinit var
btnCetak: Button
    private lateinit var
arrowBack: ImageView
    private lateinit var
presenter: LaporanPresenter
    private lateinit var
alertIOS: iOSDialogBuilder
    private var downloadID:
Long? = null
    // broadcast receiver
    private val
onDownloadComplete:
BroadcastReceiver = object :
BroadcastReceiver() {
        override fun
onReceive(context: Context,
intent: Intent) {
            //Fetching the
download id received with the
broadcast
            val id =
intent.getLongExtra(DownloadMa
nager.EXTRA_DOWNLOAD_ID, -1)
            //Checking if the
received broadcast is for our
enqueued download by matching
download id
            if (downloadID !=
null && downloadID == id) {
                Toast.makeText(this@CetakLapor
anActivity, "Download
Completed",
Toast.LENGTH_SHORT).show()
            }
        }
    }
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activi
ty_cetak_laporan)
        //layout
        edtTanggalAwal =
findViewById(R.id.cetak_edt_tg
l_awal)
        edtTanggalAkhir =
findViewById(R.id.cetak_edt_tg
l_akhir)
        btnCetak =
findViewById(R.id.cetak_btn_la
poran)
        arrowBack =
findViewById(R.id.toolbar_back
_arrow)
        // insialisasi alert
dialog ios
        alertIOS =
iOSDialogBuilder(this)
        .setTitle("Perhatian")
        .setBoldPositiveLabel(true)
        .setCancelable(false)
        .setPositiveButton("Oke")
        {dialog: iOSDialog? ->
dialog?.dismiss()
        }
        // broadcast receiver
        registerReceiver(onDownloadCom
plete,
IntentFilter(DownloadManager.A
CTION_DOWNLOAD_COMPLETE))
    }
}

```

```

        val karyawan =
intent.getParcelableExtra<Kary
awan>("EXTRA_CETAK_DETAIL_KARY
AWAN")
        presenter =
LaporanPresenter(this)
arrowBack.setOnClickListener {
super.onBackPressed()
}
edtTanggalAwal.setOnClickListener {
val datePickerDialog =
DatePickerDialog(this)
datePickerDialog.show(supportF
ragmentManager,
"TANGGAL_AWAL_PICKER")
}
edtTanggalAkhir.setOnClickListener {
val datePickerDialog =
DatePickerDialog(this)
datePickerDialog.show(supportF
ragmentManager,
"TANGGAL_AKHIR_PICKER")
}
btnCetak.setOnClickListener {

if
(edtTanggalAwal.text.toString(
).isEmpty() ||
edtTanggalAkhir.text.toString(
).isEmpty()) {
Toast.makeText(this, "Lengkapi
tanggal awal dan akhir",
Toast.LENGTH_SHORT).show()

return@setOnClickListener
}
// cek permission
checkRuntimePermission(karyawa
n)
}
}
override fun onDestroy() {
super.onDestroy()
unregisterReceiver(onDownloadC
omplete)
}
private fun
checkRuntimePermission(karyawa
n: Karyawan?) {
askPermission(
Manifest.permission.WRITE_EXTE
RNAL_STORAGE,
Manifest.permission.READ_EXTE
RNAL_STORAGE

```

```

        startDate =
dateFormat.format(calendar.time)
    } else if (tag ==
"TANGGAL_AKHIR_PICKER") {
        val calendar =
Calendar.getInstance()
        calendar.set(year,
month, day)
        val dateFormat =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault())
        val dateFormat2 =
SimpleDateFormat("dd/M/yyyy",
Locale.getDefault())
edtTanggalAkhir.setText(datefo
rmat2.format(calendar.time))
        endDate =
dateFormat.format(calendar.time)
    }
    else
Toast.makeText(this, "Gagal
parsing onDateSelected
Listener",
Toast.LENGTH_SHORT).show()
    }
    companion object {
        var startDate = ""
        var endDate = ""
    }
    override fun
isLoading(status: Boolean) {
        if (status) {
btnCetak.setText("Mohon Tunggu
...")
        } else {
btnCetak.setText("Cetak
Laporan")
        }
    }
    override fun
responseKaryawan(listKaryawan:
ArrayList<Karyawan>) {
        // do nothing
    }
    override fun
messageResponse(message:
String) {
        alertIOS.setSubtitle(message)
        alertIOS.build().show()
    }
    override fun
pdfResponse(url: String) {
        startDownloading(url)
    }

```

```

private fun
startDownloading(url: String)
{
    //val request =
presenter.startDownloadLaporan
(url)
    val request =
presenter.startDownloadingForP
ie(url)
    // get download
service, dan antian file
    val manager =
getSystemService(Context.DOWNL
OAD_SERVICE) as
DownloadManager
        downloadID =
manager.enqueue(request)
    }
}

```

LaporanPresenter.kt(Mandor)

```

package
com.heru.mandorptpn.ui.laporan
import
android.app.DownloadManager
import android.net.Uri
import android.os.Environment
import
com.heru.mandorptpn.models.Res
ponseKaryawan
import
com.heru.mandorptpn.models.Res
ponseLaporan
import
com.heru.mandorptpn.networking
.ApiClient
import
com.heru.mandorptpn.networking
.BackendEndpoint
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class LaporanPresenter(val
view: LaporanView) {
    // show all karyawan
    fun showKaryawan() {
        view.isLoading(true)
        val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)

        getClient.getAllKaryawan()
            .enqueue(object :
Callback<ResponseKaryawan> {
                override fun
onFailure(call:

```

```

Call<ResponseKaryawan>, t:
Throwable) {
view.isLoading(false)
view.messageResponse(t.localizedMessage ?: "")
}
override fun
onResponse(
call:
Call<ResponseKaryawan>,
response:
Response<ResponseKaryawan>
) {
val status
= response.body()?.isSuccess
if (status
!= null && status == true) {
view.isLoading(false)
view.responseKaryawan(response
.body()?.data ?:
arrayListOf())
} else if
(status != null && status ==
false) {
view.isLoading(false)
view.messageResponse(response.
body()?.message ?: "")
}
}
}
// cetak laporan
fun getUrlLaporan(
idKaryawan: String,
namaKaryawan: String,
starDate: String,
endDate: String
) {
view.isLoading(true)
val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.cetakLaporan(
idKaryawan,
namaKaryawan, starDate,
endDate
).enqueue(object
:Callback<ResponseLaporan> {
override fun
onFailure(call:
Call<ResponseLaporan>, t:
Throwable) {
view.isLoading(false)
view.messageResponse(t.localizedMessage ?: "")
}
override fun
onResponse(
call:
Call<ResponseLaporan>,
response:
Response<ResponseLaporan>
) {
val status =
response.body()?.isSuccess
if (status !=
null && status == true) {
view.isLoading(false)
view.pdfResponse(response.body
()?.urlPdf ?: "")
} else if (status != null &&
status == false) {
view.isLoading(false)
view.messageResponse(response.
body()?.message ?: "")
}
})
// start download laporan
fun
startDownloadLaporan(namafile:
String) :
DownloadManager.Request{
val request =
DownloadManager.Request(Uri.pa
rse(ApiClient.getPdfURL() +
namafile))
// izinkan dalam
bentuk jaringan berikut
request.setAllowedNetworkTypes
(DownloadManager.Request.NETWO
RK_WIFI or
DownloadManager.Request.NETWOR
K_MOBILE)
request.setTitle("Laporan
Panen")
request.setDescription("File
sedang di download ...")
request.allowScanningByMediaSc
anner()
request.setNotificationVisibil
ity(DownloadManager.Request.VI
SIBILITY_VISIBLE_NOTIFY_COMPLE
TED)
// set direktori
penyimpanan di folder
download/nama_file.pdf
//request.setDestinationInExte
rnalPublicDir(Environment.DIRE
CTORY_DOWNLOADS,
"Hasil_Download.pdf")
}
}
}
}

```

```

        // set direktori
        penyimpanan dengan folder
        // pertama set dulu
        folder nya, apabila foldernya
        gak ada ya dibuat
        /* val folder =
        File(Environment.getExternalStorageDirectory().toString() +
        "/TestingFolder")
        if (!folder.exists())
        {
            folder.mkdir()
            }*/
        // baru arahkan path
        hasil downloadnya kemana
        // android 7.0
        return
        request.setDestinationInExternalPublicDir("/TestingFolder/LaporanBulanan",
        "laporan_ku.pdf")
    }
    fun
    startDownloadingForPie (namafile: String) :
    DownloadManager.Request{
        // request
        val uri : Uri =
        Uri.parse("${ApiClient.getPdfURL()}$namafile")
        Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS)
            .mkdirs()
        val downloadManager =
        DownloadManager.Request(uri)
        downloadManager.setAllowedNetworkTypes (DownloadManager.Request.NETWORK_WIFI or
        DownloadManager.Request.NETWORK_MOBILE)

        downloadManager.setAllowedOverRoaming (false)

        downloadManager.setTitle ("Mendownload Laporan")

        downloadManager.setDescription ("mohon tunggu ya..")
        downloadManager.setDestinationInExternalPublicDir (Environment.DIRECTORY_DOWNLOADS,
        "laporankaryawan.pdf")
        return downloadManager

```

```

    }
}

```

LoginActivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.login
import android.content.Intent
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import
com.gdacciaro.iOSDialog.iOSDialog
import
com.gdacciaro.iOSDialog.iOSDialogBuilder
import
com.gmail.samehadar.iosdialog.IOSDialog
import
com.heru.mandorptpn.MainActivity
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.helper.UserPreferences
import
com.heru.mandorptpn.models.Admin
class LoginActivity :
AppCompatActivity(), LoginView
{
    private lateinit var
    btnLogin: Button
    private lateinit var
    dialogIOS: IOSDialog
    private lateinit var
    alertDialog: IOSDialogBuilder
    private lateinit var
    loginPresenter: LoginPresenter
    private lateinit var
    txtUsername: EditText
    private lateinit var
    txtPassword: EditText
    private lateinit var
    userPreferences:
    UserPreferences
    companion object {
        const val EXTRA_ADMIN
        = "extra_admin"
    }
    override fun
    onCreate (savedInstanceState:

```

```

Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_login)
    // inialisasi
    layoutInflater =
        LayoutInflater.from(this)
    btnLogin =
        findViewById(R.id.login_btn)
    txtUsername =
        findViewById(R.id.login_edt_username)
    txtPassword =
        findViewById(R.id.login_edt_password)
    // inialisasi pref
    userPreferences =
        UserPreferences(this)
    // inialisasi
    presenter =
        LoginPresenter(this)
    // inialisasi dialog
    loading ios
        dialogIOS =
            IOSDialog.Builder(this)
                .setTitle("Mohon
                Tunggu")

                .setTitleColorRes(R.color.gray)
                )

                .setCancelable(false)
                .build()

                // inialisasi alert
                dialog ios
                    alertDialog =
                        IOSDialog.Builder(this)
                            .setTitle("Gagal
                            Login")
                            .setBoldPositiveLabel(true)
                            .setCancelable(false)
                            .setPositiveButton("Oke")
                            {dialog: IOSDialog? ->
                                dialog?.dismiss()
                            }
                            btnLogin.setOnClickListener {
                                // Cara simulasi pakai
                                coroutines
                                /*CoroutineScope(Dispatchers.Main)

                                .launch {
                                    dialogIOS.show()
                                    delay(3000)
                                    dialogIOS.dismiss()
                                }
                            }
                    alertDialog.build().show()
                }*/
                val username =
                    txtUsername.text.toString()
                val password =
                    txtPassword.text.toString()
                loginPresenter.executeLogin(username, password)
            }
        }
    // Implements loginview
    override fun showLoading()
    {
        dialogIOS.show()
    }
    // Implements loginview
    override fun hideLoading()
    {
        dialogIOS.dismiss()
    }
    // Implements loginview
    override fun
    setSuccessResponse(admin:
    Admin?) {
        // set pref
        status_login
        userPreferences.setStatusLogin(true)

        val intent =
        Intent(this@LoginActivity,
        MainActivity::class.java)
        intent.putExtra(EXTRA_ADMIN,
        admin)

        startActivity(intent)
        finish()
    }
    // implements loginviews
    override fun
    setFailedResponse(message:
    String) {
        alertDialog.setSubtitle(message)
        alertDialog.build().show()
    }
}

```

LoginPresenter.kt (Mandor)

```

package
com.heru.mandorptpn.ui.login
import
com.heru.mandorptpn.models.ResponseAdmin
import
com.heru.mandorptpn.networking
.ApiClient

```

```

import
com.heru.mandorptpn.networking
.BackendEndpoint
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class LoginPresenter(val view:
LoginView) {

    // presenter admin login
    fun executeLogin(username:
String, password: String) {
        // show loading
        view.showLoading()
        val getClient =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
getClient.loginAdmin(username,
password).enqueue(object :
Callback<ResponseAdmin> {
            override fun
onFailure(call:
Call<ResponseAdmin>, t:
Throwable) {
                view.hideLoading()
                view.setFailedResponse(t.messa
ge?: "")
            }

            override fun
onResponse(call:
Call<ResponseAdmin>, response:
Response<ResponseAdmin>) {
                if
(response.isSuccessful){
                    view.hideLoading()
                    val status =
response.body()?.isSuccess
                    if
(status!!) {
                        val admin =
response.body()?.data
                        view.setSuccessResponse(admin)
                    } else {
                        view.setFailedResponse(respons
e.body()?.message ?: "")
                    }
                } else {
                    view.hideLoading()
                    view.setFailedResponse("Respon
d Failed")
                }
            }
        })
    }
}

```

DetailPanenActivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.panen
import
android.annotation.SuppressLin
t
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import
android.widget.ImageView
import android.widget.TextView
import
com.bumptech.glide.Glide
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.helper.Str
ingToTimestamp
import
com.heru.mandorptpn.models.Pan
en
import
com.heru.mandorptpn.networking
.ApiClient
class DetailPanenActivity :
AppCompatActivity() {
    private lateinit var
imgPanen: ImageView
    private lateinit var
arrowBack: ImageView
    private lateinit var
txtTanggal: TextView
    private lateinit var
txtWaktu: TextView
    private lateinit var
txtKeterangan: TextView
    private lateinit var
txtKuantitas: TextView
    @SuppressWarnings("SetTextI18n")
    override fun
onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activi
ty_detail_panen)

        // layouting
        imgPanen =
findViewById(R.id.detail_panen
_img)

        arrowBack =
findViewById(R.id.detail_panen
_back_arrow)

```

```

        txtTanggal =
findViewById(R.id.detail_panen
_tanggal)
        txtWaktu =
findViewById(R.id.detail_panen
_waktu)
        txtKeterangan =
findViewById(R.id.detail_panen
_keterangan)
        txtKuantitas =
findViewById(R.id.detail_panen
_kuantitas)
arrowBack.setOnClickListener {
super.onBackPressed() }
val panen =
intent.getParcelableExtra<Pane
n>(PanenActivity.EXTRA_PANEN_D
ETAIL)
        Glide.with(this)
.load(ApiClient.getImgURL() +
"${panen?.fileFoto}")
.placeholder(R.drawable.bg_pla
ceholder)
        .into(imgPanen)
        txtTanggal.text =
StringToTimestamp.setLongDate(
panen?.tanggal)
        txtWaktu.text =
panen?.waktu + " WIB"
        txtKuantitas.text =
panen?.kuantitas + " Kg"
        txtKeterangan.text =
panen?.keterangan
    }
}

```

PanenActivity.kt (Mandor)

```

package
com.heru.mandorptpn.ui.panen
import
android.annotation.SuppressLin
t
import android.content.Intent
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import
android.widget.ImageView
import
androidx.recyclerview.widget.R
ecyclerView
import
com.gdacciaro.iOSDialog.iOSDia
log
import

```

```

com.gdacciaro.iOSDialog.iOSDia
logBuilder
import
com.gmail.samehadar.iosdialog.
IOSDialog
import com.heru.mandorptpn.R
import
com.heru.mandorptpn.adapter.Pa
nenAdapter
import
com.heru.mandorptpn.adapter.Pa
nenItemListener
import
com.heru.mandorptpn.models.Pa
nen
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class PanenActivity :
AppCompatActivity(),
PanenView, PanenItemListener {
    private lateinit var
panenPresenter: PanenPresenter
    private lateinit var
panenAdapter: PanenAdapter
    private lateinit var
rvPanen: RecyclerView
    private lateinit var
loadingDialog: IOSDialog
    private lateinit var
alertIOS: iOSDialogBuilder
    private lateinit var
toolbarBackArrow: ImageView

```

```

        override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.acti
vity_panen)
        // layout
        rvPanen =
findViewById(R.id.panen_recycl
er_view)
        toolbarBackArrow =
findViewById(R.id.toolbar_back
_arrow)
toolbarBackArrow.setOnClickLis
tener {
super.onBackPressed()
}
// insialisasi loading ios
dialog

```



```

        loadingDialog =
        IOSDialog.Builder(this)
            .setTitle("Memuat
Data")
            .setTitleColorRes(R.color.gray
)
            .setCancelable(false)
                .build()
            // inialisasi alert
dialog ios
            alertIOS =
        IOSDialogBuilder(this)
            .setTitle("Perhatian")
            .setBoldPositiveLabel(true)
            .setCancelable(false)
            .setPositiveListener("Oke")
        {dialog: IOSDialog? ->
        dialog?.dismiss()
        super.onBackPressed()
        }
// inialisasi presenter
        panenPresenter =
        PanenPresenter(this)
            // inialisasi adapter
        panenAdapter =
        PanenAdapter(arrayListOf(),
        this)
        }
@SuppressLint("SimpleDateForma
t")
        override fun onResume() {
            super.onResume()
            val date = Date()
            val sdf =
SimpleDateFormat("yyyy-MM-dd")
            val tanggal =
sdf.format(date)
        panenPresenter.showPanenToday(
tanggal)
        }
        override fun showLoading()
{
            loadingDialog.show()
        }
        override fun hideLoading()
{
            loadingDialog.hide()
        }
        override fun
responseSuccess(list:
ArrayList<Panen>) {
        panenAdapter.resetListPanen(li
st)
            rvPanen.adapter =
        panenAdapter
        }

```

```

        override fun
failedResponse(message:
String) {
        alertIOS.setSubtitle(message)
        alertIOS.build().show()
        }
        override fun
onItemClicked(panen: Panen) {
            val intent =
Intent(this,
DetailPanenActivity::class.jav
a)

            intent.putExtra(EXTRA_PANEN_DE
TAIL, panen)
                startActivity(intent)
            }

            companion object {
                const val
EXTRA_PANEN_DETAIL =
"extra_panen_detail"
            }
        }

```

PanenPresenter.kt (Mandor)

```

package
com.heru.mandorptpn.ui.panen
import
com.heru.mandorptpn.models.Res
ponsePanen
import
com.heru.mandorptpn.networking
.ApiClient
import
com.heru.mandorptpn.networking
.BackendEndpoint
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class PanenPresenter(val view:
PanenView) {
    // show all panen
    fun
showPanenToday(tanggal:String)
{
        view.showLoading()
        val getCliet =
ApiClient.getClient().create(B
ackendEndpoint::class.java)
        getCliet.getPanenToday(tanggal
).enqueue(object :
Callback<ResponsePanen> {
            override fun
onFailure(call:
Call<ResponsePanen>, t:

```



```

Fragment(), AutoCompleteView,
OnMapReadyCallback,
GoogleMap.OnCameraIdleListener
{
    private lateinit var
    autoCompleteTxtLoading:
    AutoCompleteWithLoading
    private lateinit var
    progressDialog: ProgressBar
    private lateinit var
    absensiPresenter:
    AbsensiPresenter
    private lateinit var
    absensiPref: UserPreferences
    private lateinit var
    btnAbsensi: Button
    private lateinit var
    btnPengajuan: Button
    private lateinit var
    mapView: MapView
    private var
    fusedLocProviderClient:
    FusedLocationProviderClient? =
    null
    private var mMap:
    GoogleMap? = null
    companion object {
        private var
        kodeWilayahSearch = ""
        private var latitude:
        Double? = 0.0
        private var longitude:
        Double? = 0.0
        const val
        MAP_VIEW_BUNDLE_KEY =
        "map_view_bundle_key"
        const val DEFAULT_ZOOM
        = 17f
    }
    override fun
    onCreate(savedInstanceState:
    Bundle?) {
        super.onCreate(savedInstanceState)
    }
    override fun onCreateView(
        inflater:
        LayoutInflater, container:
        ViewGroup?,
        savedInstanceState:
        Bundle?
    ): View? {
        // Inflate the layout
        for this fragment
        return
        inflater.inflate(R.layout.frag
ment_absensi, container,
false)
    }
    @SuppressWarnings("ClickableViewAc
cessibility",
"SimpleDateFormat")
    override fun
    onCreateView(view: View,
savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // Inisialisasi
        layouting binding
            autoCompleteTxtLoading
        =
        view.findViewById(R.id.absensi
_txt_kode_wilayah)
        progressDialog =
        view.findViewById(R.id.loading
_indicator)
        btnAbsensi =
        view.findViewById(R.id.btn_abs
ensi)
        btnPengajuan =
        view.findViewById(R.id.btn_pen
gajuan_izin)
        mapView =
        view.findViewById(R.id.map_vie
w)
        // tanya perizinan lokasi
        GPS
        askPermissionLocation()
        // bundle data
        var mapViewBundle:
        Bundle? = null
        if (savedInstanceState
        != null)
            mapViewBundle =
            savedInstanceState.getBundle(M
            AP_VIEW_BUNDLE_KEY)
        // Generate map
        mapView.onCreate(mapViewBu
        ndle)
        mapView.getMapAsync(this)
        // inisialisai absen
        prefrencas
            absensiPref =
            UserPreferences(requireActivit
            y())
        // get status absen
        /*val statusAbsen =
        absensiPref.getStatusAbsensi()
        if (!statusAbsen){
            Toast.makeText(context, "Belum
            melakukan absensi",
            Toast.LENGTH_SHORT).show()

```

```

        }*/
        // absensi presenter
        absensiPresenter =
AbsensiPresenter(this)
        // auto complete
dengan indicator loading
autoCompleteTxtLoading.setLoad
ingIndicator(progressDialog)
        // Textwatcher ketika
textfield sedang diketik
autoCompleteTxtLoading.addText
ChangedListener(object :
TextWatcher{
            override fun
afterTextChanged(p0: Editable)
{
absensiPresenter.getWilayah()
            }
            override fun
beforeTextChanged(p0:
CharSequence?, p1: Int, p2:
Int, p3: Int) {
            }

            override fun
onTextChanged(p0:
CharSequence, p1: Int, p2:
Int, count: Int) {

kodeWilayahSearch =
p0.toString()
            }
        })

        // Ketika dropdown
autoComplete di klik

autoCompleteTxtLoading.setOnIt
emClickListener { adapterView,
view, i, l ->
val wilayah =
adapterView.getItemAtPosition(
i) as Wilayah
            kodeWilayahSearch
= wilayah.idWilayah ?: ""
        }
btnAbsensi.setOnClickListener
{
val sdfDate =
SimpleDateFormat("yyyy-MM-dd")
val sdfTime =
SimpleDateFormat("HH:mm:ss")
val tanggal =
sdfDate.format(Date())
val waktu =
sdfTime.format(Date())

absensiPresenter.storeAbsensi(
kodeWilayahSearch,
absensiPref.getDataKaryawan().
idKaryawan ?: "",
latitude.toString(),
longitude.toString(),
tanggal,
waktu,
"absensi",
"pending"
        )
    }
btnPengajuan.setOnClickListener {
val sdfDate =
SimpleDateFormat("yyyy-MM-dd")
val sdfTime =
SimpleDateFormat("HH:mm:ss")
val tanggal =
sdfDate.format(Date())
val waktu =
sdfTime.format(Date())

absensiPresenter.storeAbsensi(
kodeWilayahSearch,

absensiPref.getDataKaryawan().
idKaryawan ?: "",
latitude.toString(),
longitude.toString(),
tanggal,
waktu,
"pengajuan",
"pending"
        )
    }
}
        override fun onResume() {
            getLocation()
            mapView.onResume()
            super.onResume()
        }
        override fun onLowMemory()
{
            super.onLowMemory()
            mapView.onLowMemory()
        }
        override fun onDestroy() {
            mapView.onDestroy()
            super.onDestroy()
        }
        override fun
onSaveInstanceState(outState:
Bundle) {
            super.onSaveInstanceState(outS

```

```

tate)
askPermissionLocation()
    var mapViewBundle =
outState.getBundle(MAP_VIEW_BUN
DLE_KEY)
    if (mapViewBundle ==
null){
        mapViewBundle =
Bundle()

outState.putBundle(MAP_VIEW_BUN
DLE_KEY, mapViewBundle)
    }
mapView.onSaveInstanceState(ma
pViewBundle)
    }
    // Implements auto
complete view
    override fun onLoading() {
autoCompleteTxtLoading.displayI
ndicator()
    }
    // Implements auto
complete view
    override fun
onHideLoading() {

autoCompleteTxtLoading.removeI
ndicator()
    }
    // Implements auto
complete view
    override fun
onSuccess(listWilayah:
ArrayList<Wilayah>) {
        //val adapter =
ArrayAdapter(requireActivity()
,
android.R.layout.simple_list_i
tem_1, listWilayah ?:
arrayListOf() )
        if
(listWilayah.isNotEmpty()){
            val adapter =
AutoCompleteAdapter(requireAct
ivity(),
R.layout.custom_item_auto_comp
lete, listWilayah ?:
arrayListOf())
adapter.filter.filter(kodeWila
yahSearch)
autoCompleteTxtLoading.setAdap
ter(adapter)
        } else {
            val adapter =
AutoCompleteAdapter(requireAct
ivity(),
R.layout.custom_item_auto_comp
lete, arrayListOf())
adapter.notifyDataSetChanged()
autoCompleteTxtLoading.setAdap
ter(adapter)
        }
    }
    // Implements auto
complete view
    override fun
onFailure(throwable:
Throwable) {
        Toast.makeText(context,
"Error: ${throwable.message}",
Toast.LENGTH_SHORT).show()
    }
    // Implements auto
complete view
    override fun
absenSukses(m: String) {
        AlertDialog.Builder(requireCon
text())
            .setMessage(m)
            .setPositiveButton("Oke") {
dialog, _ ->
dialog.dismiss()
            }
            .show()
    }
    // Implement
OnMapReadyCallback
    override fun
onMapReady(googleMap:
GoogleMap?) {
        // agar maps nya tetap
tampil
        mapView.onResume()

        // insialisai maps
mMap = googleMap

// Tanyakan lagi ask
permission
askPermissionLocation()
        val permissiOne =
ActivityCompat.checkSelfPermission(
requireActivity(),

```

```

Manifest.permission.ACCESS_FINE_LOCATION)
        val permissiTwo =
ActivityCompat.checkSelfPermission(
requireActivity(),
Manifest.permission.ACCESS_COARSE_LOCATION)
        val pckgMgr =
PackageManager.PERMISSION_GRANTED
        // Tanyakan permission
        if (permissiOne !=
pckgMgr && permissiTwo !=
pckgMgr ) {
            return
        }
        // Lokasi saya saat ini
/*mMap?.isMyLocationEnabled =
true*/
mMap?.setOnCameraIdleListener(
this)
    }
    // fungsi permission
lokasi
    private fun
askPermissionLocation() {
askPermission(Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION) {
// Get lokasi saat ini

getCurrentLocation()
}.onDeclined {
// Apabila permission lokasi
ditolak
        if (it.hasDenied())
        {
it.denied.forEach{
// Menbuka alerdialog
AlertDialog.Builder(requireAct
ivity())
.setMessage("Aplikasi ini
membutuhkan izin lokasi")
.setPositiveButton("Iya") { _,
_ ->
it.askAgain()
}
.setNegativeButton("Tidak") {
dialog, _ ->
dialog.dismiss()
}
.show()
        }
        // Apabila
permissions lokasi di tolak
selamanya
        if (it.hasForeverDenied())
        {
it.foreverDenied.forEach{
// menuju settings
it.goToSettings()
        }
    }}
    // fungsi mendapatkan
lokasi saat ini, via SDK
google maps
    private fun
getCurrentLocation() {
        fusedLocProviderClient
=
LocationServices.getFusedLocat
ionProviderClient(requireActiv
ity())
        try {
@SuppressLint("MissingPermissi
on")
            val location =
fusedLocProviderClient?.lastLo
cation

location?.addOnCompleteListener(
object :
OnCompleteListener<Location> {
                override fun
onComplete(task:
Task<Location>) {
                    if
(task.isSuccessful){
// Lokasi saat ini
val currentLoc = task.result
// set ke latitude dan
longitud global scopelatitude
= currentLoc?.latitude
longitude =
currentLoc?.longitude
val markerOptions =
MarkerOptions()
markerOptions.position(LatLng(
currentLoc?.latitude ?: 0.0,
currentLoc?.longitude ?: 0.0))
markerOptions.title("Posisi
Kamu")
mMap?.addMarker(markerOptions)
// Pindahkan kamera sesuai lat
dan lng
mMap?.moveCamera(
CameraUpdateFactory.newLatLngZ
oom(
LatLng(task.result?.latitude
?: 0.0, task.result?.longitude
?: 0.0),

```

```

DEFAULT_ZOOM
                                ))
                                } else {
askPermissionLocation()
                                }
                                }
                                })
                                } catch (e: Exception)
{
    Log.d("ERROR",
"getCurrentLocation:
${e.message}")
    }
    // Implements onMapIdle
    override fun
onCameraIdle() {
    }
}

```

AbsensiPresenter.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.absensi
import
com.heru.ptpnapps.models.Respo
nseAction
import
com.heru.ptpnapps.models.Respo
nseWilayah
import
com.heru.ptpnapps.networking.A
bsensiService
import
com.heru.ptpnapps.networking.A
piClient
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class AbsensiPresenter(private
val autoCompleteView:
AutoCompleteView) {
    fun getWilayah() {
autoCompleteView.onLoading()
        val getClient =
ApiClient.getClient().create(A
bsensiService::class.java)
getClient.showWilayah().enqueu
e(object :
Callback<ResponseWilayah> {
            override fun
onFailure(call:
Call<ResponseWilayah>, t:
Throwable) {
autoCompleteView.onHideLoading
()

```

```

autoCompleteView.onFailure(t)
        }
        override fun
onResponse(
            call:
Call<ResponseWilayah>,
            response:
Response<ResponseWilayah>
        ) {
            if
(response.isSuccessful) {
autoCompleteView.onHideLoading
()
                val
listWilayah =
response.body()?.data
autoCompleteView.onSuccess(lis
tWilayah ?: arrayListOf())
            } else {
autoCompleteView.onHideLoading
()
autoCompleteView.onFailed("Res
ponse Failed")
            }
        })
        // store data absen ke
backend
        fun storeAbsensi(
            idWilayah: String,
            idKaryawan: String,
            latitude: String,
            longitude: String,
            tanggal: String,
            waktu: String,
            tipeAbsen: String,
            statusKehadiran:
String
        ) {
            // show progressbar
/*autoCompleteView.onLoading()
*/
            val getClient =
ApiClient.getClient().create(A
bsensiService::class.java)
            getClient.addAbsen(
                idWilayah,
                idKaryawan, latitude,
                longitude, tanggal, waktu,
                tipeAbsen, statusKehadiran
            ).enqueue(object :
Callback<ResponseAction> {
                override fun

```

```

onFailure(call:
Call<ResponseAction>, t:
Throwable) {
autoCompleteView.onFailure(t)
}
        override fun
onResponse(
            call:
Call<ResponseAction>,
            response:
Response<ResponseAction>
        ) {
            if
                (response.isSuccessful) {
autoCompleteView.absenSukses(r
esponse.body()?.message ?: "")
                } else {
autoCompleteView.onFailed("Res
ponse Failed")
                }
            }
        })
    }
}

```

AktifitasPresenter.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.aktifitas
import
com.heru.ptpnapps.models.Respo
nseAction
import
com.heru.ptpnapps.models.Respo
nseAktifitas
import
com.heru.ptpnapps.networking.A
ktifitasService
import
com.heru.ptpnapps.networking.A
piClient
import okhttp3.MultipartBody
import okhttp3.RequestBody
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class
AktifitasPresenter(private val
view: AktifitasView){
    fun addAktifitas(foto:
MultipartBody.Part,
keterangan: RequestBody,
tanggal: RequestBody, waktu:
RequestBody, idKaryawan:
RequestBody) {
        view.showLoading()

```

```

        val getClient =
ApiClient.getClient().create(A
ktifitasService::class.java)
getClient.addAktifitas(foto,
keterangan, tanggal, waktu,
idKaryawan).enqueue(object :
Callback<ResponseAction> {
            override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()view.onFailu
re(t)
            }
        })
        override fun
onResponse(
            call:
Call<ResponseAction>,
            response:
Response<ResponseAction>
        ) {
            if
                (response.isSuccessful){
view.hideLoading()
                    val msg =
response.body()?.message
view.succesResponse(msg ?: "")
                } else {
view.hideLoading()
                    val msg =
response.body()?.message
view.succesResponse(msg ?:
"Response Gagal")
                }
            })
        }
        fun
getAllAktifitas(idKaryawan:
String) {
            view.showLoading()
            val getClient =
ApiClient.getClient().create(A
ktifitasService::class.java)
getClient.showAktifitas(idKary
awan).enqueue(object :
Callback<ResponseAktifitas>{
                override fun
onFailure(call:
Call<ResponseAktifitas>, t:
Throwable) {
view.hideLoading()
view.onFailure(t)

```



```

        }
        override fun
onResponse(
            call:
Call<ResponseAktifitas>,
            response:
Response<ResponseAktifitas>
        ) {
            if
(response.isSuccessful) {
view.hideLoading()
                if
(response.body()?.isSuccess ==
true) {
                    val
data = response.body()?.data
view.onResponse(data ?:
arrayListOf())
view.succesResponse("")
                } else {
view.succesResponse(response.b
ody()?.message ?: "")
                }
            }
        })
    }
    fun deleteAktifitas(id:
String) {
        view.showLoading()
        val getClient =
ApiClient.getClient().create(A
ktifitasService::class.java)
getClient.deleteAktifitas(id).
enqueue(object :
Callback<ResponseAction>{
            override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()view.onFailu
re(t)
            }
            override fun
onResponse(
                call:
Call<ResponseAction>,
                response:
Response<ResponseAction>
            ) {
                if
(response.isSuccessful &&
response.body()?.isSuccess ==

```

```

true){
view.hideLoading()
view.succesResponse(response.b
ody()?.message ?: "")
            } else if
(response.isSuccessful &&
response.body()?.isSuccess ==
false){
view.hideLoading()
view.succesResponse(response.b
ody()?.message ?: "")
            } else {
view.hideLoading()
view.succesResponse("Response
Failed")
            }
        })
    }
}

```

Aktifitasragment.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.aktifitas
import android.content.Intent
import android.os.Bundle
import
androidx.fragment.app.Fragment
import
android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import
androidx.appcompat.app.AlertDi
alog
import
androidx.recyclerview.widget.D
ividerItemDecoration
import
androidx.recyclerview.widget.R
ecyclerView
import
com.github.loadingview.Loading
Dialog
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.adapter.akti
fitas.AktifitasAdapter
import
com.heru.ptpnapps.adapter.akti
fitas.AktifitasItemClick
import
com.heru.ptpnapps.helper.UserP

```

```

referances
import
com.heru.ptpnapps.models.Aktif
itas
class Aktifitasragment :
Fragment(), AktifitasView,
AktifitasItemClick {

    private lateinit var
rvAktifitas: RecyclerView
    private lateinit var
btnAddAktifitas: Button
    private lateinit var
adapterAktifitas:
AktifitasAdapter
AktifitasAdapter
    private lateinit var
loadingProgress: LoadingDialog
    private lateinit var
aktifitasPresenter:
AktifitasPresenter
    private lateinit var
txtEmptyData: TextView
    private lateinit var
userPref: UserPreferences
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
    }
    override fun onCreateView(
inflater:
LayoutInflater, container:
ViewGroup?,
savedInstanceState:
Bundle?
): View? {
    // Inflate layout
    return
inflater.inflate(R.layout.frag
ment_aktifitas, container,
false)
    }
    override fun
onViewCreated(view: View,
savedInstanceState: Bundle?) {
super.onViewCreated(view,
savedInstanceState)
    // binding data
aktifitas fragment
rvAktifitas =
view.findViewById(R.id.aktifita
s_rv)
    btnAddAktifitas =
view.findViewById(R.id.aktifit
as_add_btn)

        loadingProgress =
LoadingDialog[requireActivity(
)]
        txtEmptyData =
view.findViewById(R.id.aktifit
as_empty_txt)
        // insialisasi
presenter
        aktifitasPresenter =
AktifitasPresenter(this)
        // insialisasi
preferances
        userPref =
UserPreferences(requireActivit
y())
        // set adapter
        adapterAktifitas =
AktifitasAdapter(arrayListOf(
, this)
, btnAddAktifitas.setOnClickListener {
// pindah ke tambah aktifitas
startActivity(Intent(context,
TambahAktifitasActivity::class
.java))
}
}
        override fun onResume() {
super.onResume()
adapterAktifitas.resetAdapter()
val karyawan =
userPref.getDataKaryawan()
aktifitasPresenter.getAllAktif
itas(karyawan.idKaryawan ?:
"")
}
        override fun showLoading()
{
loadingProgress.show()
}
        override fun hideLoading()
{
loadingProgress.hide()
}

        // generate recyclerview
        override fun
onResponse(dataAktifitas:
ArrayList<Aktifitas>) {
        adapterAktifitas =
AktifitasAdapter(dataAktifitas,
this)
rvAktifitas.adapter =
adapterAktifitas
rvAktifitas.addItemDecoration(

```

```

DividerItemDecoration(context,
DividerItemDecoration.VERTICAL
))
    }
    override fun onFailure(t:
Throwable) {
Toast.makeText(context,
t.message,
Toast.LENGTH_SHORT).show()
    }
    override fun
succesResponse(message:
String) {
    if
(message.isNotEmpty()){
rvAktifitas.visibility =
View.GONE
    txtEmptyData.text =
message
txtEmptyData.visibility =
View.VISIBLE
    } else {
rvAktifitas.visibility =
View.VISIBLE
txtEmptyData.visibility =
View.GONE
    }
    }
    // implements dari adapter
yang di klik
    override fun
onLongClickListener(item:
Aktifitas) {
    // buat alert dialog
AlertDialog.Builder(context ?:
requireContext()).apply {
setTitle("Hapus Aktifitas")
    setMessage("Apakah
anda yakin ingin menghapus
aktifitas ini ?")
setPositiveButton("Iya") {
dialog, _ ->
aktifitasPresenter.deleteAktif
itas(item.id ?: "")
dialog.dismiss()
Toast.makeText(context,
"Berhasil hapus data",
Toast.LENGTH_SHORT).show()
    onResume()
}
setNegativeButton("Batal") {
dialog, _ ->
dialog.dismiss()
}
    }.show()
}
}

```

```

// implements dari adapter
yang di klik
    override fun
onClickListener(item:
Aktifitas) {
    val intent =
Intent(requireActivity(),
DetailAktifitasActivity::class
.java)
intent.putExtra(AKTIFITAS_EKST
RA, item)
startActivity(intent)
}
    companion object {
    const val
AKTIFITAS_EKSTRA =
"aktifitas_extra_data"
    }
}

```

DetailAktifitas.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.aktifitas
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import android.widget.Button
import
android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import
androidx.appcompat.app.AlertDi
alog
import
com.bumptech.glide.Glide
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.URLCo
nstServer
import
com.heru.ptpnapps.models.Aktif
itas
class DetailAktifitasActivity
: AppCompatActivity(),
AktifitasView {
    private lateinit var
txtKeterangan: TextView
    private lateinit var
txtWaktu: TextView
    private lateinit var
txtTanggal: TextView
    private lateinit var
ivAktifitas: ImageView
    private lateinit var

```

```

btnHapus: Button
    private lateinit var
aktifitasPresenter:
AktifitasPresenter
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activi
ty_detail_aktifitas)
    // data parcelable
    val dataAktifitas =
intent.getParcelableExtra<Akti
fitas>(Aktifitasragment.AKTIFI
TAS_EKSTRA)
    // inialisai
presenter
    aktifitasPresenter =
AktifitasPresenter(this)

    // inialisai binding
view
    txtKeterangan =
findViewById(R.id.detail_keter
angan_aktifitas)
    txtWaktu =
findViewById(R.id.detail_waktu
_aktifitas)
    txtTanggal =
findViewById(R.id.detail_tangg
al_aktifitas)
    ivAktifitas =
findViewById(R.id.detail_iv_ak
tifitas)
    btnHapus =
findViewById(R.id.btn_hapus_de
tail)
    txtKeterangan.text =
dataAktifitas?.keterangan
    txtTanggal.text =
dataAktifitas?.tanggal
    txtWaktu.text =
dataAktifitas?.waktu
    Glide.with(this)
.load(URLConstServer.IMAGE_URL
+ dataAktifitas?.fileFoto)
.placeholder(R.drawable.ic_ima
ge)
        .into(ivAktifitas)
btnHapus.setOnClickListener {
// buat alert dialog
AlertDialog.Builder(this).appl
y {
setTitle("Hapus Aktifitas")
setMessage("Apakah anda yakin
ingin menghapus aktifitas ini
?")
setPositiveButton("Iya") {
dialog, _ ->
aktifitasPresenter.deleteAktif
itas(dataAktifitas?.id ?: "")
dialog.dismiss()
}
setNegativeButton("Batal") {
dialog, _ ->
dialog.dismiss()
}
        }.show()
}
}

override fun showLoading()
{
    // do nothing
}

override fun hideLoading()
{
    // do nothing
}
override fun
onResponse(dataAktifitas:
ArrayList<Aktifitas>) {
    // do nothing
}
override fun onFailure(t:
Throwable) {
    Toast.makeText(this,
t.message,
Toast.LENGTH_SHORT).show()
}
override fun
succesResponse(message:
String) {
    Toast.makeText(this,
message,
Toast.LENGTH_SHORT).show()
    finish()
}
}
}

TambahAktifitasActivity.kt
(Karyawan)
package
com.heru.ptpnapps.ui.aktifitas
import android.Manifest
import
android.annotation.SuppressLin
t
import android.app.Activity
import android.content.Intent

```

```

import android.net.Uri
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.StrictMode
import
android.provider.MediaStore
import android.widget.Button
import android.widget.EditText
import
android.widget.ImageView
import android.widget.Toast
import
com.github.loadingview.LoadingDialog
import
com.livinglifetechway.quickpermissions_kotlin.runWithPermissions
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.MediaHelper
import
com.heru.ptpnapps.helper.UserPreferences
import
com.heru.ptpnapps.models.Aktifitas
import
com.heru.ptpnapps.models.Karyawan
import
okhttp3.MediaType.Companion.toMediaTypeOrNull
import okhttp3.MultipartBody
import okhttp3.RequestBody
import java.io.File
import java.lang.Exception
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class TambahAktifitasActivity : AppCompatActivity(),
AktifitasView {
    private lateinit var ivAddAktifitas: ImageView
    private lateinit var simpanBtn: Button
    private lateinit var edtketerangan: EditText
    private lateinit var dialog : LoadingDialog

    private lateinit var mediaHelper: MediaHelper
    private lateinit var aktifitasPresenter: AktifitasPresenter
    private lateinit var userPreferences: UserPreferences

    private var fileUri: Uri = Uri.parse("")
    @SuppressWarnings("SimpleDateFormat")
    override fun
    onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_tambah_aktifitas)
        // binding layout
        ivAddAktifitas = findViewById(R.id.iv_add_aktifitas)
        simpanBtn = findViewById(R.id.add_btn_simpan)
        edtketerangan = findViewById(R.id.add_edt_keterangan)
        // pref user init
        userPreferences = UserPreferences(this)
        // dialog init
        dialog = LoadingDialog[this]
        // insialisasi presenter
        aktifitasPresenter = AktifitasPresenter(this)
        // kalau gak ada ini error
        try {
            val m = StrictMode::class.java.getMethod("disableDeathOnFileUriExposeure")
            m.invoke(null)
        } catch (e: Exception) {
            e.printStackTrace()
        }
        // insialisasi media helper
        mediaHelper = MediaHelper()
        ivAddAktifitas.setOnClickListener {

```

```

requestPermissions()
}
simpanBtn.setOnClickListener {
if
(edtketerangan.text.toString()
.isEmpty()){
Toast.makeText(this,
"Keterangan tidak boleh
kosong",
Toast.LENGTH_SHORT).show()
return@setOnClickListener
}
    val keterangan =
RequestBody.create("text/plain"
.toMediaTypeOrNull(),
edtketerangan.text.toString())
    // simple date
format
    val sdf =
SimpleDateFormat("yyyy-MM-dd")
    val sdfTime =
SimpleDateFormat("HH:mm:ss")
    val currentDate =
sdf.format(Date())
    val currentTime =
sdfTime.format(Date())
    val tanggal =
RequestBody.create("text/plain"
.toMediaTypeOrNull(),
currentDate)
    val waktu =
RequestBody.create("text/plain"
.toMediaTypeOrNull(),
currentTime)
    val image =
File(fileUri.path ?: "")
    val reqBody =
RequestBody.create("multipart/
form-
file".toMediaTypeOrNull(),
image)
    val part =
MultipartBody.Part.createFormD
ata("imageupload", image.name,
reqBody)

    val userKaryawan:
Karyawan =
userPreferences.getDataKaryawa
n()
    val idUserKaryawan
=
RequestBody.create("text/plain"
.toMediaTypeOrNull(),
userKaryawan.idKaryawan ?: "")
aktifitasPresenter.addAktifita

s(part, keterangan, tanggal,
waktu, idUserKaryawan)
}
    private fun
requestPermissions() =
runWithPermissions(Manifest.pe
rmission.WRITE_EXTERNAL_STORAG
E,
Manifest.permission.CAMERA) {
fileUri =
mediaHelper.getOutputMediaFile
Uri()
    val intent =
Intent(MediaStore.ACTION_IMAGE
_CAPTURE)
intent.putExtra(MediaStore.EXT
RA_OUTPUT, fileUri)
startActivityForResult(intent,
mediaHelper.getRcCamera())
}
override fun
onActivityResult(requestCode:
Int, resultCode: Int, intent:
Intent?) {
super.onActivityResult(request
Code, resultCode, intent)
    if (resultCode ==
Activity.RESULT_OK)
        if (requestCode ==
mediaHelper.getRcCamera()){
mediaHelper.getBitmapToString(
ivAddAktifitas, fileUri)
        }
    }
override fun showLoading()
{
    dialog.show()
}
override fun hideLoading()
{
    dialog.hide()
}
    override fun
onResponse(dataAktfitas:
ArrayList<Aktifitas>) {
// tidak di eksekusi
}
    override fun onFailure(t:
Throwable) {
Toast.makeText(this,
t.message,
Toast.LENGTH_SHORT).show()
}
    override fun
succesResponse(message:

```

```
String) {
    Toast.makeText(this,
message,
Toast.LENGTH_SHORT).show()
        finish()
    }
}
```

LoginActivity.kt (Karyawan)

```
package
com.heru.ptpnapps.ui.login
import android.content.Intent
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.EditText
import
android.widget.ProgressBar
import android.widget.Toast
import
androidx.appcompat.app.AlertDialog
import
androidx.cardview.widget.CardView
import
com.heru.ptpnapps.MainActivity
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.UserP
referances
import
com.heru.ptpnapps.models.Karya
wan
class LoginActivity :
AppCompatActivity(), LoginView
{
    private lateinit var
edtUsername: EditText
    private lateinit var
edtPassword: EditText
    private lateinit var
btnLogin: CardView
    private lateinit var
userPreferences:
UserPreferences
    private lateinit var
loginPresenter: LoginPresenter
    private lateinit var
progressBar: ProgressBar
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState
```

```
ate)
setContentViews(R.layout.activi
ty_login)
    // insialisasi user
preferences
    userPreferences =
UserPreferences(this)
    // insialisasi layout
edtUsername =
findViewById(R.id.login_edt_us
ername)
    edtPassword =
findViewById(R.id.login_edt_pa
ssword)
    btnLogin =
findViewById(R.id.login_cv_btn
)
    progressBar =
findViewById(R.id.progress_bar
_login)
    // insialisasi
presenter
    loginPresenter =
LoginPresenter(this)
btnLogin.setOnClickListener {
    val username =
edtUsername.text.toString()
    val password =
edtPassword.text.toString()
    if
(username.isNotEmpty() &&
password.isNotEmpty()) {
loginPresenter.addLogin(userna
me, password)
    } else {
Toast.makeText(this, "Username
dan Password tidak boleh
kosong",
Toast.LENGTH_LONG).show()
    }
}
}
// implements presenter
login
    override fun
showLoading(status: Boolean) {
    if (status) {
progressBar.visibility =
View.VISIBLE
    } else {
progressBar.visibility =
View.GONE
    }
}
// implements presenter
login
```

```

        override fun
        errorMessage(message: String)
        {
        AlertDialog.Builder(this)

        .setMessage(message)

        .setPositiveButton("OK") {
        dialog, _ ->
        dialog.dismiss()
        }
        .show()
        }
        // implements presenter
        login
        override fun onFailure(t:
        Throwable) {
        AlertDialog.Builder(this)
        .setMessage(t.message)
        .setPositiveButton("OK") {
        dialog, _ ->
        dialog.dismiss()
        }
        .show()
        }
        // implements presenter
        login
        override fun
        successLogin(karyawan:
        Karyawan?) {
        userPreferences.setStatusLogin
        (true)
        userPreferences.setDataKaryawa
        n(karyawan?.nama ?: "",
        karyawan?.alamat ?: "",
        karyawan?.noTelepon ?: "",
        karyawan?.username ?: "",
        karyawan?.idKaryawan ?: "")
        startActivity(Intent(this,
        MainActivity::class.java))
        finish()
        }
        }
    }

```

Login.Presenter.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.login
import android.util.Log
import
com.heru.ptpnapps.models.Karya
wan
import
com.heru.ptpnapps.models.Respo
nseUser
import
com.heru.ptpnapps.networking.A

```

```

piClient
import
com.heru.ptpnapps.networking.L
ogAndRegService
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class LoginPresenter(private
val view: LoginView) {
    // check login
    fun addLogin(
        username:String,
        password: String
    ) {
        view.showLoading(true)
        val getClient =
        ApiClient.getClient().create(L
        ogAndRegService::class.java)
        getClient.login(username, passw
        ord).enqueue(object :
        Callback<ResponseUser> {
            override fun
            onFailure(call:
            Call<ResponseUser>, t:
            Throwable) {
                view.showLoading(false) view.on
                Failure(t)
            }
            override fun
            onResponse(call:
            Call<ResponseUser>, response:
            Response<ResponseUser>) {
                if
                (response.isSuccessful){
                    val
                    responseStatus: Boolean? =
                    response.body()?.isSuccess
                    Log.d("STATUS", "onResponse:
                    $responseStatus")
                    if
                    (responseStatus == true){
                        view.showLoading(false)
                    }
                    val
                    karyawan: Karyawan? =
                    response.body()?.data
                    view.successLogin(karyawan)
                } else {
                    view.showLoading(false)
                    view.errorMessage(response.bod
                    y()?.message ?: "")
                }
            }
            } else {
                view.showLoading(false)
                view.errorMessage("Respon

```



```

login failed")
        }
    }
    })
}
}

```

DetailPanenActivity.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.panen
import
android.annotation.SuppressLin
t
import
androidx.appcompat.app.AppComp
atActivity
import android.os.Bundle
import android.widget.Button
import
android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import
androidx.appcompat.app.AlertDi
alog
import
com.bumptech.glide.Glide
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.URLCo
nstServer
import
com.heru.ptpnapps.models.Panen
class DetailPanenActivity :
AppCompatActivity(), PanenView
{
    private lateinit var
imgPanen: ImageView
    private lateinit var
txtKet: TextView
    private lateinit var
txtKuantitas: TextView
    private lateinit var
txtTanggal: TextView
    private lateinit var
txtWaktu: TextView
    private lateinit var
btnHapus: Button
    private lateinit var
panenPresenter: PanenPresenter
@SuppressLint("SetTextI18n")
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState

```

```

ate)
setContentView(R.layout.activi
ty_detail_panen)
        // insialisai binding
layout
        imgPanen =
findViewById(R.id.detail_iv_pa
nen)
        txtKet =
findViewById(R.id.detail_keter
angan_panen)
        txtKuantitas =
findViewById(R.id.detail_kuant
itas_panen)
        txtTanggal =
findViewById(R.id.detail_tangg
al_panen)
        txtWaktu =
findViewById(R.id.detail_waktu
_panen)
        btnHapus =
findViewById(R.id.btn_hapus_de
tail_panen)

        // data parcel dari
fragment panen
        val dataPanen =
intent.getParcelableExtra<Pane
n>(PanenFragment.EXTRA_PANEN_D
ATA)
        // insialisasi
presenter
        panenPresenter =
PanenPresenter(this)
        // set data ke layout
Glide.with(this)
.load(URLConstServer.IMAGE_URL
+ dataPanen?.fileFoto)
.placeholder(R.drawable.ic_ima
ge)
        .into(imgPanen)
        txtKet.text =
dataPanen?.keterangan
        txtKuantitas.text =
"Kuantitas :
${dataPanen?.kuantitas.toStrin
g()} Kg"
        txtTanggal.text =
dataPanen?.tanggal
        txtWaktu.text =
dataPanen?.waktu
        // action button hapus
btnHapus.setOnClickListener {
// buat alert dialog
AlertDialog.Builder(this).appl
y {

```

```

setTitle("Hapus Hasil Panen")
setMessage("Apakah anda yakin
ingin menghapus panen ini ?")
setPositiveButton("Iya") {
dialog, _ ->
panenPresenter.deletePanen(dat
aPanen?.id ?: "")
dialog.dismiss()
}
setNegativeButton("Batal") {
dialog, _ ->
dialog.dismiss()
}
        }.show()
}
}
    override fun showLoading()
{
    }
    override fun hideLoading()
{
    }
    override fun
onResponse(dataPanen:
ArrayList<Panen>) {
    }
    override fun onFailure(t:
Throwable) {
        Toast.makeText(this,
t.message,
Toast.LENGTH_SHORT).show()
    }
    override fun
succesResponse(message:
String) {
        Toast.makeText(this,
message,
Toast.LENGTH_SHORT).show()
        finish()
    }
}

```

PanenFragment.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.panen
import android.content.Intent
import android.os.Bundle
import
androidx.fragment.app.Fragment
import
android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.TextView
import android.widget.Toast

```

```

import
androidx.recyclerview.widget.D
ividerItemDecoration
import
androidx.recyclerview.widget.R
ecyclerView
import
com.github.loadingview.Loading
Dialog
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.adapter.pane
n.PanenAdapter
import
com.heru.ptpnapps.adapter.pane
n.PanenClick
import
com.heru.ptpnapps.helper.UserP
referances
import
com.heru.ptpnapps.models.Panen
class PanenFragment :
Fragment(), PanenView,
PanenClick{
    private lateinit var
rvPanen: RecyclerView
    private lateinit var
btnAddPanen: Button
    private lateinit var
txtEmpty: TextView
    private lateinit var
adapterPanen: PanenAdapter
    private lateinit var
loadingProgress: LoadingDialog
    private lateinit var
panenPresenter: PanenPresenter
    private lateinit var
userSession: UserPreferences
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
inflater:
LayoutInflater, container:
ViewGroup?,
savedInstanceState:
Bundle?
): View? {
        // Inflate fragment
        panen
        return
inflater.inflate(R.layout.frag

```

```

ment_panen, container, false)
    }
    override fun
onViewCreated(view: View,
savedInstanceState: Bundle?) {
super.onViewCreated(view,
savedInstanceState)
    rvPanen =
view.findViewById(R.id.panen_r
v)
    btnAddPanen =
view.findViewById(R.id.panen_a
dd_btn)
    txtEmpty =
view.findViewById(R.id.panen_e
mpty_txt)
    loadingProgress =
LoadingDialog[requireActivity(
)]
    // sharedpref
    userSession =
UserPreferences(requireActivit
y())
    // insialisasi
presenter
    panenPresenter =
PanenPresenter(this)
    // set adapter
recyclerview
    adapterPanen =
PanenAdapter(arrayListOf(),
this)
    // event button add
btnAddPanen.setOnClickListener
{
startActivity(Intent(requireAc
tivity(),
TambahPanenActivity::class.jav
a))
}
}
    override fun onResume() {
super.onResume()
adapterPanen.resetAdapter()
val karyawan =
userSession.getDataKaryawan()

panenPresenter.getAllPanen(kar
yawan.idKaryawan ?: "")
    }

    override fun showLoading()
{
loadingProgress.show()
}
    override fun hideLoading()
{
loadingProgress.hide()
}
    override fun
onResponse(dataPanen:
ArrayList<Panen>) {
adapterPanen =
PanenAdapter(dataPanen, this)
rvPanen.adapter =
adapterPanen
rvPanen.addItemDecoration(Divi
derItemDecoration(context,
DividerItemDecoration.VERTICAL
))
}
    override fun onFailure(t:
Throwable) {
Toast.makeText(context,
t.message,
Toast.LENGTH_SHORT).show()
}
    override fun
succesResponse(message:
String) {
if
(message.isNotEmpty()){
rvPanen.visibility
= View.GONE
txtEmpty.text =
message
txtEmpty.visibility =
View.VISIBLE
} else {
rvPanen.visibility
= View.VISIBLE
txtEmpty.visibility =
View.GONE
}
}
    override fun
onLongListerner(dataPanen:
Panen) {
}
    override fun
onSetListener(dataPanen:
Panen) {
val intent =
Intent(requireActivity(),
DetailPanenActivity::class.jav
a)
intent.putExtra(EXTRA_PANEN_DA
TA, dataPanen)
startActivity(intent)
}
companion object {

```

```

        const val
        EXTRA_PANEN_DATA =
        "extra_panen_data"
    }
}

```

PanenPresenter.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.panen
import
com.heru.ptpnapps.models.Respo
nseAction
import
com.heru.ptpnapps.models.Respo
nsePanen
import
com.heru.ptpnapps.networking.A
piClient
import
com.heru.ptpnapps.networking.H
asilPanenService
import okhttp3.MultipartBody
import okhttp3.RequestBody
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
class PanenPresenter(private
val view: PanenView){

    fun
    getAllPanen(idKaryawan:
    String) {
        view.showLoading()
        val getClient =
        ApiClient.getClient().create(H
        asilPanenService::class.java)
        getClient.showHasilPanen(idKar
        yawan)
            .enqueue(object :
            Callback<ResponsePanen> {
                override fun
                onFailure(call:
                Call<ResponsePanen>, t:
                Throwable) {
                    view.hideLoading()
                    view.onFailure(t)
                }
            })
            override fun
            onResponse(call:
            Call<ResponsePanen>, response:
            Response<ResponsePanen>) {
                if
                (response.isSuccessful) {
                    view.hideLoading()
                    if
                    (response.body()?.isSuccess ==

```

```

true){
                val data
                = response.body()?.data
                view.onResponse(data ?:
                arrayOf())
                view.succesResponse("")
            } else {
                view.succesResponse(response.b
                ody()?.message ?: "")
            }
        })
    }
}

fun deletePanen(id:
String) {
    view.showLoading()
    val getClient =
    ApiClient.getClient().create(H
    asilPanenService::class.java)
    getClient.deletePanen(id).enqu
    eue(object :
    Callback<ResponseAction> {
        override fun
        onFailure(call:
        Call<ResponseAction>, t:
        Throwable) {
            view.hideLoading() view.onFailu
            re(t)
        }
    })
    override fun
    onResponse(
        call:
        Call<ResponseAction>,
        response:
        Response<ResponseAction>
        ) {
        view.hideLoading()
        if
        (response.isSuccessful &&
        response.body()?.isSuccess ==
        true){
            view.succesResponse(response.b
            ody()?.message ?: "")
        } else if
        (response.isSuccessful &&
        response.body()?.isSuccess ==
        false) {
            view.succesResponse(response.b
            ody()?.message ?: "")
        } else {
            view.succesResponse("Response
            Failed")
        }
    }
}

```

```

    })
}
fun addPanen(idKaryawan:
RequestBody ,foto:
MultipartBody.Part,
keterangan: RequestBody,
kuantitas: RequestBody,
tanggal: RequestBody, waktu:
RequestBody) {
view.showLoading()
    val getClient =
ApiClient.getClient().create(H
asilPanenService::class.java)
    getClient.addPanen(
        idKaryawan,
        foto,
        keterangan,
        kuantitas,
        tanggal,
        waktu
    ).enqueue(object :
Callback<ResponseAction> {
        override fun
onFailure(call:
Call<ResponseAction>, t:
Throwable) {
view.hideLoading()view.onFailu
re(t)
        }
        override fun
onResponse(
            call:
Call<ResponseAction>,
            response:
Response<ResponseAction>
        ) {
view.hideLoading()
            if
(response.isSuccessful){
                val msg =
response.body()?.message
view.succesResponse(msg ?: "")
            } else {
                val msg =
response.body()?.message
view.succesResponse(msg ?:
"Response Gagal")
            }
        }
    })
}
}

```

TambahPanenActivity.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.panen
import android.Manifest
import
android.annotation.SuppressLin
t
import android.app.Activity
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.os.StrictMode
import
android.provider.MediaStore
import android.widget.*
import
androidx.appcompat.app.AppComp
atActivity
import
com.github.loadingview.Loading
Dialog
import
com.livinglifetechway.quickper
missions_kotlin.runWithPermiss
ions
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.Media
Helper
import
com.heru.ptpnapps.helper.UserP
referances
import
com.heru.ptpnapps.models.Panen
import
okhttp3.MediaType.Companion.to
MediaTypeOrNull
import okhttp3.MultipartBody
import okhttp3.RequestBody
import java.io.File
import java.lang.Exception
import
java.text.SimpleDateFormat
import java.util.*
import
kotlin.collections.ArrayList
class TambahPanenActivity :
AppCompatActivity(), PanenView
{
    private lateinit var
ivHasilPanen: ImageView
    private lateinit var
edtKeterangan: EditText
    private lateinit var
edtKuantitas: EditText
    private lateinit var
btnSimpan: Button

```

```

        private lateinit var
        dialog: LoadingDialog
        private lateinit var
        panenPresenter: PanenPresenter
        private lateinit var
        mediaHelper: MediaHelper
        private var fileUri: Uri =
        Uri.parse("")
        @SuppressWarnings("SimpleDateForma
        t")
        override
        fun onCreate(savedInstanceState
        : Bundle?) {

        super.onCreate(savedInstanceState)
        setContentView(R.layout.activi
        ty_tambah_panen)
        // inialisai
        layouting xml
        ivHasilPanen =
        findViewById(R.id.iv_add_panen
        )
        edtKeterangan =
        findViewById(R.id.add_panen_ke
        terangan)
        edtKuantitas =
        findViewById(R.id.add_edt_kuan
        titas)
        btnSimpan =
        findViewById(R.id.add_panen_si
        mpan)
        dialog =
        LoadingDialog[this]

        // inialisasi
        presenter
        panenPresenter =
        PanenPresenter(this)
        // kalau gak ada ini
        error
        try {
            val m =
            StrictMode::class.java.getMeth
            od("disableDeathOnFileUriExpos
            ure")
            m.invoke(null)
        } catch (e: Exception)
        {
            e.printStackTrace()
        }
        mediaHelper = MediaHelper()
        // event listener
        imageView
        ivHasilPanen.setOnClickListener {

        reqPermissions()
        }
        // event tambah panen
        btnSimpan.setOnClickListener {
        if
        (edtKeterangan.text.toString()
        .isEmpty() &&
        edtKuantitas.text.toString().i
        sEmpty()) {
            Toast.makeText(this, "Field
            data tidak boleh kosong",
            Toast.LENGTH_SHORT).show()
            return@setOnClickListener
            }
            val sdf =
            SimpleDateFormat("yyyy-MM-dd")
            val sdfTime =
            SimpleDateFormat("HH:mm:ss")
            val currentDate =
            sdf.format(Date())
            val currentTime =
            sdfTime.format(Date())
            val image =
            File(fileUri.path ?: "")
            val reqBody =
            RequestBody.create("multipart/
            form-
            file".toMediaTypeOrNull(),
            image)
            val userPref =
            UserPreferences(this)
            val karyawan =
            userPref.getDataKaryawan()
            val idKaryawan =
            RequestBody.create("text/plain
            ".toMediaTypeOrNull(),
            karyawan.idKaryawan.toString()
            )
            val partImage =
            MultipartBody.Part.createFormD
            ata("imageupload", image.name,
            reqBody)
            val tanggal =
            RequestBody.create("text/plain
            ".toMediaTypeOrNull(),
            currentDate)
            val waktu =
            RequestBody.create("text/plain
            ".toMediaTypeOrNull(),
            currentTime)
            val keterangan =
            RequestBody.create("text/plain
            ".toMediaTypeOrNull(),
            edtKeterangan.text.toString())
            val kuantitas =
            RequestBody.create("text/plain

```

```

".toMediaTypeOrNull(),
edtKuantitas.text.toString())
panenPresenter.addPanen(idKary
awan, partImage, keterangan,
kuantitas, tanggal, waktu)
}
}
    private fun
reqPermissions() =
runWithPermissions(Manifest.pe
rmission.WRITE_EXTERNAL_STORAG
E, Manifest.permission.CAMERA)
{
fileUri =
mediaHelper.getOutputStreamFile
Uri()
        val intent =
Intent(MediaStore.ACTION_IMAGE
_CAPTURE)
intent.putExtra(MediaStore.EXT
RA_OUTPUT, fileUri)
startActivityForResult(intent,
mediaHelper.getRcCamera())
}
override fun
onActivityResult(requestCode:
Int, resultCode: Int, data:
Intent?) {
super.onActivityResult(request
Code, resultCode, data)
    if (resultCode ==
Activity.RESULT_OK) {
        if (requestCode ==
mediaHelper.getRcCamera()) {
mediaHelper.getBitmapToString(
ivHasilPanen, fileUri)
        }
    }
}
    override fun showLoading()
{
        dialog.show()
    }
    override fun hideLoading()
{
dialog.hide()
    }
    override fun
onResponse(dataPanen:
ArrayList<Panen>) {
        // do nothing
    }
    override fun onFailure(t:
Throwable) {

```

```

        Toast.makeText(this,
t.message,
Toast.LENGTH_SHORT).show()
    }
    override fun
succesResponse(message:
String) {
        Toast.makeText(this,
message,
Toast.LENGTH_SHORT).show()
        finish()
    }
}

```

UserFragment.kt (Karyawan)

```

package
com.heru.ptpnapps.ui.user

import
android.annotation.SuppressLin
t
import android.app.AlertDialog
import android.content.Intent
import android.os.Bundle
import
androidx.fragment.app.Fragment
import
android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.TextView
import com.heru.ptpnapps.R
import
com.heru.ptpnapps.helper.UserP
referances
import
com.heru.ptpnapps.ui.login.Log
inActivity

class UserFragment :
Fragment() {
    private lateinit var
userPreferences:
UserPreferences
    private lateinit var
txtNamaLengkap: TextView
    private lateinit var
btnLogout: Button
    private lateinit var
txtStatusAbsen: TextView
    override fun
onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)

```

```

    }
    override fun onCreateView(
        inflater:
        LayoutInflater, container:
        ViewGroup?,
        savedInstanceState:
        Bundle?
    ): View? {
        // Inflate the layout
        for this fragment
        return
        inflater.inflate(R.layout.fragment_user, container, false)
    }
    @SuppressWarnings("SetTextI18n")
    override fun
    onViewCreated(view: View,
        savedInstanceState: Bundle?) {

        super.onViewCreated(view,
            savedInstanceState)
            // inisialisasi
            lateiniit
            userPreferances =
            UserPreferances(requireContext
            ())

            txtNamaLengkap =
            view.findViewById(R.id.user_nama_lengkap)
            btnLogout =
            view.findViewById(R.id.user_btn_logout)
            txtStatusAbsen =
            view.findViewById(R.id.user_status_absen)
            // ambil nilai user
            preferances
            val karyawan =
            userPreferances.getDataKaryawan()
            val statusAbsensi =
            userPreferances.getStatusAbsensi()
            // Set nama karyawan
            txtNamaLengkap.text =
            karyawan.nama
            txtStatusAbsen.text =
            "Karyawan Lapangan"
            // event listener btn
            logout
            btnLogout.setOnClickListener {
            AlertDialog.Builder(requireContext())
            .setTitle("LOGOUT!")
            .setMessage("Yakin ingin
            keluar dari aplikasi ?")

```

```

        .setPositiveButton("Keluar") {
        _, _ ->
        userPreferances.setStatusLogin
        (false)
        userPreferances.setDataKaryawan(
            "",
            "",
            "",
            ""
        )
        startActivity(Intent(requireActivity(),
            LoginActivity::class.java))
        activity?.finish()
        }
        .setNegativeButton("Tidak") {dialog, _ ->
        dialog.dismiss()
        }
        .show()
        }
        }
    }

```

MainActivity.kt

```

package com.heru.ptpnapps
import android.content.Intent
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import
androidx.fragment.app.Fragment
import
androidx.fragment.app.FragmentTransaction
import
com.google.android.material.bottomnavigation.BottomNavigationView
import
com.heru.ptpnapps.ui.absensi.AbsensiFragment
import
com.heru.ptpnapps.ui.aktifitas.Aktifitasragment
import
com.heru.ptpnapps.ui.panen.PanenFragment
import
com.heru.ptpnapps.ui.user.UserFragment
class MainActivity :

```