

**MEMBANGUN APLIKASI GoSE (GO SERVICE ELECTRONIC)
BERBASIS ANDROID MENGGUNAKAN FRAMEWORK
REACT NATIVE DAN FIREBASE REALTIME DATABASE**

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana



RAHMAD AULIA

NIM. 0702163047

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2022 M / 1443 H**



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

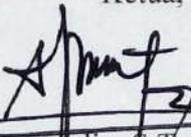
Jl. IAIN No. 1 Medan, Kode Pos 20235
Telp. (061) 6615683-6622925, Fax. (061)6615683
Url: www.saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

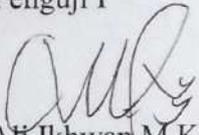
Nomor : B. 127/ST/ST.V.2/PP.01/06/2022

Judul : Membangun Aplikasi GOSE (*Go Service Electronic*) Berbasis *Android* Menggunakan *Framework React Native* dan *Realtime Database*
Nama : Rahmad Aulia
Nomor Induk Mahasiswa : 0702163047
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 : 14 Maret 2022
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

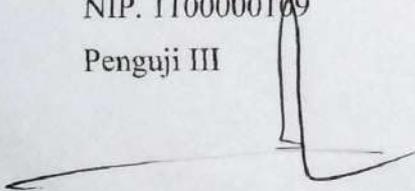
Tim Ujian Munaqasyah,
Ketua,


Samsudin, S.T., M.Kom
NIP. 197612272011011002

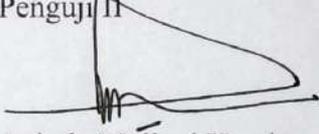
Penguji I


Ali Ikhwan, M.Kom
NIP. 1100000109

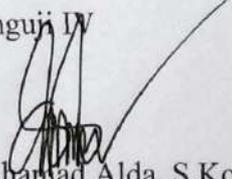
Penguji III


Suendri, M.Kom
NIP. 198712082015031003

Penguji II


Aninda Muliani Harahap, M.Kom
NIP. 198611292019032009

Penguji IV


Mulhanad Alda, S.Kom, M.S.I
NIP. 198908302020121007

Mengesahkan
Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan

(Dr. Mhd. Syahnan, MA)
NIP.196609051991031002

**MEMBANGUN APLIKASI GoSE (GO SERVICE ELECTRONIC)
BERBASIS ANDROID MENGGUNAKAN FRAMEWORK
REACT NATIVE DAN FIREBASE REALTIME DATABASE**

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana

*Ace Sidang Munqashid
21/2 - 2022
Suendri*



*Ace Sidang Munqashid
25/2 - 2022
Muhamad Alidw*

RAHMAD AULIA

NIM. 0702163047

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA**

MEDAN

2022 M / 1443 H

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp :-

Kepada Yth :

Dekan Fakultas Sains dan Teknologi

UIN Sumatera Utara Medan

Assalamu'alaikum Wr,Wb.

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

Nama : Rahmad Aulia

Nomor Induk Mahasiswa : 0702163047

Program Studi : Sistem Informasi

Judul : *Membangun Aplikasi GOSE (Go Service Electronic) Berbasis Android Menggunakan Framework React Native dan Realtime Database*

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

Wa'alaikum salam Wr, Wb

Medan, Februari 2022

Komisi Pembimbing

Pembimbing I

Suendri, M.Kom

NIP.198712082015031003

Pembimbing II

Muhammad Alda, S.Kom, M.S.I

NIP. 198908302020121007

SURAT PERNYATAAN SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : Rahmad Aulia
Nomor Induk Mahasiswa : 0702163047
Program Studi : Sistem Informasi
Judul : Membangun Aplikasi GOSE (*Go Service Electronic*) Berbasis *Android* Menggunakan *Framework React Native* dan *Realtime Database*

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

Medan, 22 Februari 2022



Rahmad Aulia
NIM : 0702163047

**MEMBANGUN APLIKASI GOSE (*GO SERVICE ELECTRONIC*)
BERBASIS *ANDROID* MENGGUNAKAN *FRAMEWORK REACT NATIVE*
DAN *FIREBASE REALTIME DATABASE***

ABSTRAK

Banyak usahawan yang menyediakan jasa *service* elektronik terutama pada ponsel dan PC namun pemasaran mereka sangat terbatas dikarenakan posisi toko maupun *banner* yang kurang strategis sehingga sulit untuk mengetahui dimana tempat jasa *service*, disisi lain pelanggan juga sulit untuk mengetahui estimasi biaya perbaikan perangkat elektronik serta seringkali calon pelanggan merasa kecewa ketika sudah sampai dilokasi namun jasa *service* tersebut sedang tutup. Dengan permasalahan tersebut maka penulis membangun aplikasi *Go Service Electronic* berbasis android dengan memanfaatkan teknologi *Framework React Native* untuk membangun aplikasi berbasis *android* serta menggunakan *Firebase Realtime Database* sebagai penyimpanan data. Aplikasi tersebut juga dilengkapi fitur konsultasi, sistem *booking*, menghitung estimasi biaya jasa perbaikan perangkat, serta memberikan informasi jasa servis terbaik melalui sistem penilaian yang telah di berikan oleh pelanggan yang sudah melakukan *booking* jasa servis sebelumnya. Dengan adanya aplikasi *Go Service Electronic* maka masyarakat lebih mudah untuk mengetahui dimana lokasi penyedia jasa *service*, mengetahui estimasi biaya perbaikan, dan dapat berkonsultasi secara langsung melalui aplikasi tersebut, serta mampu memperbesar peluang bisnis bagi para penyedia jasa *service* elektronik.

Kata Kunci : *Service Electronic, Framework React Native, Firebase.*

**BUILD AN ANDROID-BASED GOSE (GO SERVICE ELECTRONIC)
APPLICATION USING THE REACT NATIVE FRAMEWORK AND
FIREBASE REALTIME DATABASE**

ABSTRACT

Many entrepreneurs provide electronic services, especially on cellphones and PCs, but their marketing is very limited due to the location of the store and banners that are not strategic so it is difficult to know where the service is located, on the other hand customers are also difficult to know the estimated cost of repairing electronic devices and often potential customers I was disappointed when I arrived at the location but the service was closed. With these problems, the author built an Android-based Go Service Electronic application by utilizing React Native Framework technology to build Android-based applications and using the Firebase Realtime Database as data storage. The application is also equipped with consulting features, a booking system, calculating the estimated cost of equipment repair services, as well as providing information on the best service services through an assessment system that has been provided by customers who have booked previous services. With the Go Service Electronic application, it is easier for the public to find out where the location of service providers is, find out the estimated repair costs, and can consult directly through the application, as well as being able to increase business opportunities for electronic service providers.

Keyword : Electronic, Framework React Native, Firebase.

PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah serta berharap atas rahmat Allah SWT, saya mempersembahkan karya tulis ini kepada :

1. Kedua orang tua penulis, Ayah M. Sukhri S.Pd dan Almarhumah Ibu Siti Aminah Nasution atas tulusnya hati dalam mendidik, mengasuh, memberi bimbingan dan selalu mendoakan serta memenuhi kebutuhan materi penulis dengan kasih sayang dan ketulusan hingga mengantarkan penulis lulus dari UIN Sumatera Utara.
2. Kepada Kakak dan adik kandung saya yang selalu menjadi alasan bagi saya untuk tetap tegar serta memberikan motivasi untuk tetap terus berjalan dan menyelesaikan apa yang telah dimulai, terima kasih atas do`a dan semangat dukungannya selama ini.
3. Kepada teman saya atas nama Teguh Kurniawan S.Kom, Ahmad Syarif S.Kom, Heru Irfansyah S.Kom, Rusdin Halomoan S.Kom, Muhammad Ichsan S.Kom, Bagus Aji Pratama S.Kom, Rozi Marzuki Panggabean S.Kom, serta teman-teman dan sahabat lainnya yang selalu memberikan semangat, motivasi, serta memberikan segala solusi untuk menghadapi permasalahan dalam menulis skripsi ini.
4. Almamater tercinta UIN Sumatera Utara Medan yang saya banggakan.

KATA PENGANTAR

Alhamdulillahirabbil'alamin. Puji syukur kepada Allah SWT yang telah memberikan banyak rahmat dan karuniaNya kepada penulis, sehingga penulis dapat menyelesaikan proposal skripsi ini dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Strata-1 (S-1) pada Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara dengan Mengangkat judul skripsi “Membangun Aplikasi GoSE (*Go Service Electronic*) Berbasis *Android* Menggunakan *Framework React Native* dan *Firebase Realtime Database*.”

Selama penyusunan proposal makalah ini, penulis menyadari bahwa banyak pihak yang telah berkontribusi secara signifikan dalam penyusunan proposal makalah ini. Pada kesempatan ini penulis mengucapkan terimakasih kepada seluruh pihak yang sudah membantu untuk pembuatan proposal makalah ini, penulis ucapkan terhadap :

1. Bapak Prof. Dr. Syahrin Harahap, MA sebagai Rektor Universitas Islam Negeri Sumatera Utara-Medan.
2. Bapak Dr. Mhd. Syahnan, MA sebagai Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara-Medan.
3. Bapak Samsudin, S.T, M.Kom sebagai Ketua Program Studi Sistem Informasi, Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
4. Bapak Suendri M.Kom sebagai Sekretaris Prodi Studi Sistem Informasi, Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara dan selaku Dosen Pembimbing satu yang telah berkenan memberikan tambahan ilmu dan solusi dalam penulisan skripsi ini.
5. Bapak Muhammad Alda, S.Kom, M.S.I sebagai dosen pembimbing dua yang telah bersedia membimbing dan memberikan banyak ilmu dalam penulisan skripsi ini.
6. Pak Ilka Zufria, M.Kom sebagai Dosen Pembimbing Akademik yang selalu membimbing penulis dalam menjalankan selama jenjang perkuliahan.

7. Bapak/Ibu Dosen Program Studi Sistem Informasi Universitas Islam Negeri Sumatera Utara-Medan telah memberikan ilmu, pengalaman dan masukan dalam penyusunan karya ilmiah ini.
8. Kedua orang tua, ayahanda M. Sukhri dan ibunda Siti Aminah yang telah memberikan dukungan baik moril maupun materil serta doa yang tiada henti kepada penulis.
9. Seluruh pejuang Angkatan 2016 khususnya Sistem Informasi-2 yang telah memberikan dukungan dan motivasi dalam penyelesaian skripsi kepada penulis.

Penulis mohon maaf atas kesalahan yang pernah dilakukan. Semoga skripsi ini bermanfaat untuk kemudian hari dan mendorong penelitian-penelitian berikutnya.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh.

Medan, 21 September 2021
Penyusun,



Kahmad Aulia
0702163047

DAFTAR ISI

PERSETUJUAN SKRIPSI	i
SURAT PERNYATAAN SKRIPSI	ii
ABSTRAK	iii
ABSTRACT	iv
PERSEMBAHAN	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xvi
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. Aplikasi	6
2.1.1. Pengertian Aplikasi	6
2.1.2. Jenis-Jenis Aplikasi	7
2.2. <i>Electronic</i>	8
2.3. <i>Online</i>	9
2.4. <i>Android</i>	9
2.4.1. Pengertian <i>Android</i>	9
2.4.2. Sejarah <i>Android</i>	9
2.4.3. Fitur-Fitur <i>Android</i>	10
2.4.4. Arsitektur <i>Android</i>	11
2.4.5. Versi-versi <i>Android</i>	12
2.5. <i>Javascript</i>	19
2.5.1. Pengertian <i>JavaScript</i>	19
2.5.2. Sejarah <i>Javascript</i>	20
2.6. <i>Framework</i>	23
2.7. <i>React Native</i>	23

2.7.1. Pengertian <i>React Native</i>	24
2.7.2. Keuntungan <i>React Native</i>	24
2.7.3. Cara Kerja <i>React Native</i>	25
2.8. <i>Database SQL</i>	26
2.8.1. Pengertian <i>Database SQL</i>	26
2.8.2. Sejarah <i>Database SQL</i>	27
2.9. <i>Cloud Computing</i>	27
2.9.1. Sejarah <i>Cloud Computing</i>	28
2.9.2. Perkembangan <i>Cloud Computing</i>	30
2.9.3. Layanan <i>Cloud Computing</i>	32
2.9.4. Fungsi dan Manfaat <i>Cloud Computing</i>	34
2.9.5. Kelebihan dan Kekurangan <i>Cloud Computing</i>	34
2.9.6. Sistem Kerja <i>Cloud Computing</i>	35
2.10. <i>Firebase</i>	36
2.10.1. <i>Firebase Realtime Database</i>	37
2.10.2. <i>Firebase Auth</i>	37
2.11. <i>Unified Modelling Language</i>	39
2.12. <i>Rapid Application Development (RAD)</i>	46
2.12.1. Defenisi Metode <i>RAD</i>	46
2.12.2. Sejarah Metode <i>RAD</i>	47
2.12.3. Tahapan-tahapan dalam <i>RAD</i>	48
2.13. Penelitian Terkait	48
BAB III METODE PENELITIAN	51
3.1. Tempat dan Waktu Penelitian	51
3.1.1. Tempat Penelitian	51
3.1.2. Waktu Penelitian	52
3.2. Kebutuhan Sistem	54
3.2.1. Perangkat Keras	54
3.2.2. Perangkat Lunak	54
3.3. Cara Kerja	55
3.3.1. Metode Pengumpulan Data	55
3.3.2. Metode Pengembangan Sistem.....	58
3.3.3. Kerangka Berfikir	61
BAB IV HASIL DAN PEMBAHASAN	63

4.1. <i>Requirements Planning</i>	63
4.1.1. Struktur Organisasi Dalam Usaha Jasa Servis Ini	63
4.1.2. <i>Job Description</i> dari struktur organisasi.....	64
4.1.3. Hasil Observasi Dan Wawancara	65
4.1.4. Analisis Sistem Berjalan.....	65
4.1.5. Analisis Sistem Usulan.....	66
4.2. <i>Workshop Design</i>	71
4.3. Implementasi dan Pengujian Sistem	122
BAB V PENUTUP	157
5.1. Kesimpulan.....	157
5.2. Saran.....	157
DAFTAR PUSTAKA	158
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 : Arsitektur Android (Supardi 2017)	12
Gambar 2.2 : <i>Android</i> Versi 1.1 (Adikara 2013)	12
Gambar 2.3 : Versi 1.5 (Cupcake) (Adikara 2013).....	13
Gambar 2.4 : <i>Android</i> Versi 1.6 (<i>Donut</i>) (Adikara 2013).....	13
Gambar 2.5 : <i>Android</i> Versi 2.0/2.1 (<i>Eclair</i>) (Adikara 2013).....	14
Gambar 2.6 : <i>Android</i> Versi 2.2 (<i>Froyo: Frozen Yoghurt</i>) (Adikara 2013)	14
Gambar 2.7 : <i>Android</i> Versi 2.3 (<i>Gingerbread</i>) (Adikara 2013)	15
Gambar 2.8 : <i>Android</i> Versi 3.0/3.1 (<i>Honeycomb</i>) (Adikara 2013).....	15
Gambar 2.9 : <i>Android</i> Versi 4.0 (<i>ICS: Ice Cream Sandwich</i>).....	16
Gambar 2.10 : <i>Android</i> Versi 4.1 (<i>Jelly Bean</i>) (Adikara 2013).....	16
Gambar 2.11 : <i>Android</i> Versi 4.4 (<i>Kit Kat</i>) (Adikara 2013)	17
Gambar 2.12 : <i>Android</i> Versi 5.0.2 (<i>Lollipop</i>) (Adikara 2013).....	17
Gambar 2.13 : <i>Android</i> Versi 6.0 (<i>Marshmallow</i>) (Adikara 2013)	18
Gambar 2.14 : Logo <i>Javascript</i> (www.javascript.com)	19
Gambar 2.15 : Logo <i>React Native</i> (www.reactnative.dev)	23
Gambar 2.16 : Cara Kerja <i>DOM</i> (Eisenman 2017).....	25
Gambar 2.17 : Perbedaan <i>Rendering React Native</i> (Eisenman 2017)	26
Gambar 2.18 : <i>Cloud Computing</i> (Ahmad and Setiawan 2011)	27
Gambar 2.19 : Perkembangan <i>Cloud Computing</i> (Husain et al. 2018).....	30
Gambar 2.20 : Arsitektur <i>Software as a Service</i> (SaaS)	32
Gambar 2.21 : Arsitektur <i>Platform as a Service</i> (PaaS)	33
Gambar 2.22 : Arsitektur <i>Infrastructure as a Service</i> (IaaS) (Herwanto et al. 2021)	33
Gambar 2.23 : Logo <i>Firebase</i> (firebase.google.com).....	36
Gambar 2.24 : <i>Firebase Realtime Database</i> (console.firebase.google.com).....	37
Gambar 2.25 : <i>Console Firebase Auth</i> (console.firebase.google.com).....	38
Gambar 2.26 : Contoh <i>Diagam Usecase</i> (Suendri 2018)	41
Gambar 2.27 : Contoh <i>Class Diagram</i> (Suendri 2018)	43
Gambar 2.28 : Contoh <i>Activity Diagram</i> (Samsudin 2019)	44

Gambar 2.29 : Contoh Sequence Diagram (Suendri 2018)	46
Gambar 2.30 : Langkah Penerapan Metode RAD (Habibi and Aprilian 2020)....	46
Gambar 3. 1 : Peta lokasi penelitian pada Kota Medan	51
Gambar 3. 2 : Kerangka Berfikir	61
Gambar 4. 1: Struktur Organisasi Pada Toko Maju Jaya Computer	63
Gambar 4. 2: Arsitektur sistem usulan	71
Gambar 4. 3: Usecase Diagram sistem Go Service Electronic	72
Gambar 4. 4: Activity Diagram Registrasi dan Login (Customer)	73
Gambar 4. 5: Activity Diagram Registrasi dan Login (Admin).....	74
Gambar 4. 6: Activity Diagram Manajemen Profil (Customer)	75
Gambar 4. 7: Activity Diagram Manajemen Profil (Admin).....	76
Gambar 4. 8: Activity Diagram History Message (Customer)	77
Gambar 4. 9: Activity Diagram History Message (Admin).....	78
Gambar 4. 10: Activity Diagram Chatting (Customer).....	79
Gambar 4. 11: Activity Diagram Manajemen Biaya Jasa Servis (Admin)	80
Gambar 4. 12: Activity Diagram Booking Jasa Servis (Customer).....	81
Gambar 4. 13: Activity Diagram Konfirmasi Customer (Admin)	82
Gambar 4. 14: Activity Diagram Rating Toko (Customer)	83
Gambar 4. 15: Activity Diagram History Service (Customer)	84
Gambar 4. 16: Activity Diagram History Service (Admin).....	85
Gambar 4. 17: Sequence Diagram Register	86
Gambar 4. 18: Sequence Diagram Login.....	87
Gambar 4. 19: Sequence Diagram Manajemen Profil	88
Gambar 4. 20: Sequence Diagram History Message	89
Gambar 4. 21: Sequence Diagram Chatting.....	90
Gambar 4. 22: Sequence Diagram Manajemen Biaya Servis	92
Gambar 4. 23: Sequence Diagram Booking Jasa Servis	93
Gambar 4. 24: Sequence Diagram Rating Toko	94
Gambar 4. 25: Sequence Diagram History Service	95
Gambar 4. 26: Class Diagram	96
Gambar 4. 27: Rancangan Halaman Splash Screen	103

Gambar 4. 28: Rancangan Halaman Login	103
Gambar 4. 29: Rancangan Halaman Register	104
Gambar 4. 30: Rancangan Halaman Location Point	104
Gambar 4. 31: Rancangan Halaman Upload Photo	105
Gambar 4. 32: Rancangan Halaman Menu Home	105
Gambar 4. 33: Rancangan Halaman Pilih Toko Servis.....	106
Gambar 4. 34: Rancangan Halaman Profil Toko Jasa Servis	106
Gambar 4. 35: Rancangan Halaman Cek Estimasi Biaya	107
Gambar 4. 36: Rancangan Halaman Konfirmasi Servis	107
Gambar 4. 37: Rancangan Halaman Perbaikan Telah Selesai	108
Gambar 4. 38: Rancangan Halaman Input Rating	108
Gambar 4. 39: Rancangan Halaman Pembatalan Servis	109
Gambar 4. 40: Rancangan Halaman Menu Message	109
Gambar 4. 41: Rancangan Halaman Chatting.....	110
Gambar 4. 42: Rancangan Halaman Menu Service	110
Gambar 4. 43: Rancangan Halaman Menu Akun	111
Gambar 4. 44: Rancangan Halaman Update Profil	111
Gambar 4. 45: Rancangan Halaman Lihat Titik Lokasi Pada Google Maps	112
Gambar 4. 46: Rancangan Halaman Splash Screen	112
Gambar 4. 47: Rancangan Halaman Login	113
Gambar 4. 48: Rancangan Halaman Registrasi.....	113
Gambar 4. 49: Rancangan Halaman Upload Data Toko.....	114
Gambar 4. 50: Rancangan Halaman Maps Location Coordinate.....	114
Gambar 4. 51: Rancangan Halaman Upload Logo Toko	115
Gambar 4. 52: Rancangan Halaman Data Foto Toko	115
Gambar 4. 53: Rancangan Halaman Jadwal Toko Buka.....	116
Gambar 4. 54: Rancangan Halaman Data Estimasi Jasa Servis.....	116
Gambar 4. 55: Rancangan Halaman Konfirmasi Pengambilan Barang	117
Gambar 4. 56: Rancangan Halaman Konfirmasi Barang Telah Tiba	117
Gambar 4. 57: Rancangan Halaman Konfirmasi Kerusakan Barang.....	118
Gambar 4. 58: Rancangan Halaman Konfirmasi Perbaikan Selesai	118

Gambar 4. 59: Rancangan Halaman Pembatalan Servis	119
Gambar 4. 60: Rancangan Halaman Melakukan Pengembalian Barang	119
Gambar 4. 61: Rancangan Halaman Utama	120
Gambar 4. 62: Rancangan Halaman Detail Ulasan.....	120
Gambar 4. 63: Rancangan Halaman Halaman Akun	121
Gambar 4. 64: Rancangan Halaman Edit Profile	121
Gambar 4. 65: Tampilan Android Studio.....	123
Gambar 4. 66: Tampilan Edit User Variable	124
Gambar 4. 67: Tampilan Home React Native.....	125
Gambar 4. 68: Halaman Project Environment dari Firebase	126
Gambar 4. 69: Menu dari Project Layanan Firebase.....	126
Gambar 4. 70: Layanan Penyedia Authentication Firebase	127
Gambar 4. 71: Aturan Realtime Database.....	127
Gambar 4. 72: Kunci Server Pada Cloud Messaging.....	128
Gambar 4. 73: Tampilan Splash Screen	128
Gambar 4. 74: Tampilan login	129
Gambar 4. 75: Tampilan Registrasi	129
Gambar 4. 76: Tampilan Location Point.....	130
Gambar 4. 77: Tampilan Upload Photo	130
Gambar 4. 78: Tampilan Home.....	131
Gambar 4. 79: Tampilan Pilih Toko Servis	131
Gambar 4. 80: Tampilan Profil Toko Jasa Servis	132
Gambar 4. 81: Tampilan Cek Estimasi Biaya	133
Gambar 4. 82: Tampilan Konfirmasi Servis	133
Gambar 4. 83: Tampilan Perbaikan Telah Selesai	134
Gambar 4. 84: Tampilan Input Rating	134
Gambar 4. 85: Tampilan Pembatalan Servis.....	135
Gambar 4. 86: Tampilan Menu Message	135
Gambar 4. 87: Tampilan Chatting.....	136
Gambar 4. 88: Tampilan Menu Service	136
Gambar 4. 89: Tampilan Menu Akun	137

Gambar 4. 90: Tampilan Update Profile	137
Gambar 4. 91: Tampilan Lihat Titik Lokasi Pada Google Maps.....	138
Gambar 4. 92: Tampilan Splash Screen	138
Gambar 4. 93: Tampilan Login	139
Gambar 4. 94: Tampilan Registrasi	139
Gambar 4. 95: Tampilan Upload Data Toko.....	140
Gambar 4. 96: Tampilan Upload Maps Location Coordinate.....	140
Gambar 4. 97: Tampilan Upload Logo Toko.....	141
Gambar 4. 98: Tampilan Data Foto Toko	141
Gambar 4. 99: Tampilan Jadwal Toko Buka	142
Gambar 4. 100: Tampilan Data Estimasi Jasa Servis	142
Gambar 4. 101: Tampilan Konfirmasi Pengambilan Barang.....	143
Gambar 4. 102: Tampilan Konfirmasi Barang Telah Tiba	143
Gambar 4. 103: Tampilan Konfirmasi Kerusakan Barang.....	144
Gambar 4. 104: Tampilan konfirmasi Perbaikan Selesai.....	144
Gambar 4. 105: Tampilan Pembatalan Servis.....	145
Gambar 4. 106: Tampilan Melakukan Pengembalian Barang	145
Gambar 4. 107: Tampilan Halaman Utama	146
Gambar 4. 108: Tampilan Detail Ulasan.....	147
Gambar 4. 109: Tampilan Akun.....	147
Gambar 4. 110: Tampilan Edit Profile	148

DAFTAR TABEL

Tabel 2. 1 : Edisi ECMAScript (Supardi 2021)	20
Tabel 2. 2 : Simbol Dalam Diagram Use case(Hendini 2016).....	40
Tabel 2. 3 : Simbol Pada Digram Class (Hendini 2016).....	42
Tabel 2. 4 : Simbol Pada Activity Diagram (Hendini 2016).....	44
Tabel 2. 5 : Simbol Pada Sequence Diagram (Muhamad Alda 2021)	45
Tabel 2. 6 : Penelitian Terdahulu	48
Tabel 3. 1 : Waktu dan Jadwal Pelaksanaan Penelitian	52
Table 4. 1: Job Description	64
Table 4. 2: Flowmap Sistem Berjalan	66
Table 4. 3: Flowmap Sistem Usulan Bagian 1	69
Table 4. 4: Flowmap Sistem Usulan Bagian 2	70
Table 4. 5: Tabel User	97
Table 4. 6: Tabel Toko	98
Table 4. 7: Tabel Kategori Servis	99
Table 4. 8: Tabel Kategori Kerusakan	99
Table 4. 9: Tabel Rating	99
Table 4. 10: Table Riwayat Message	100
Table 4. 11: Tabel Chatting.....	100
Table 4. 12: Tabel Data Servis	101
Table 4. 13: Macam-macam Widget Android.....	102
Table 4. 14: Data Diri Validator	149
Table 4. 15: Pengujian Sistem Pada Aplikasi Customer	149
Table 4. 16: Pengujian Sistem Pada Aplikasi Admin	152

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dengan munculnya sebuah teknologi, membawa pengaruh baik pada perkembangan zaman yang semakin cepat dan efisien dalam melakukan seluruh aktifitas pekerjaan berat, pekerjaan kantor hingga kegiatan sehari-hari. Peralatan elektronik yang berada disekitar kita saat ini semakin cepat berkembang serta sudah menjadi kebutuhan setiap masyarakat. Seiring berjalannya waktu, peralatan elektronik ini akan mengalami kerusakan diakibatkan terlalu sering digunakan maupun diakibatkan pemakaian yang tidak sesuai dengan prosedur pabrik, sehingga masyarakat harus memperbaiki peralatan elektronik agar dapat digunakan kembali. Salah satu langkah mudah dalam hal melakukan *service* elektronik tanpa harus mencari jasa *service* kelapangan yaitu menggunakan aplikasi GoSE (*Go Service Electronic*) untuk melakukan *booking* jasa *service* pada ponsel *android*.

usahawan dibidang jasa *service* terutama di bidang jasa *service* elektronik ponsel dan PC pada beberapa toko *service* di Kota Medan, banyak masyarakat yang menyediakan jasa *service*, namun pemasaran mereka sangat terbatas, karena letak yang kurang strategis dikarenakan posisi toko maupun *banner* kurang terlihat sehingga calon pelanggan sulit untuk mengetahui dimana tempat jasa *service* serta sulit mengetahui estimasi biaya perbaikan kerusakan perangkat elektronik. Para pelanggan juga sering terjadi memiliki masalah untuk membawa peralatan elektronik yang ingin di *service* seperti perangkat CPU dikarenakan peralatan elektronik yang ukurannya terlalu besar atau beban terlalu berat maupun peralatan elektronik yang diharuskan diperbaiki ditempat. Seperti calon pelanggan harus datang ketoko untuk melakukan diskusi permasalahan perbaikan peralatan elektronik, dan seringkali calon pelanggan merasa kecewa ketika sudah sampai dilokasi namun jasa *service* tersebut sedang tutup atau sudah banyak menumpuk peralatan elektronik yang ingin di-*service* sehingga tidak dapat menerima *service* untuk sementara waktu. Permasalahan lain yang dialami bagi calon pelanggan

ialah sedikitnya informasi terhadap penyedia jasa *service*, khususnya pada kota Medan, masih banyak warga sekitar yang tidak mengetahui dimana penyedia jasa *service* yang dekat dengan warga setempat.

Dalam membangun aplikasi ini nantinya penulis akan menggunakan *Framework React Native* dan menggunakan *firebase* sebagai *database* yang akan digunakan untuk membangun sebuah aplikasi *GoSE (Go Service Electronic)* berbasis android, keunggulan dari *Framework React Native* ini yaitu mampu membangun suatu aplikasi pada ponsel yang menggunakan sistem operasi *android* maupun pada sistem operasi IOS seperti pada ponsel *Apple* cukup dengan menggunakan satu bahasa program yaitu menggunakan bahasa *javascript*. Sedangkan keunggulan menggunakan *database firebase* ini yaitu dapat melakukan apengambilan maupun pengiriman data secara *realtime* serta menggunakan *firebase auth* untuk melakukan identifikasi login setiap user maupun login sebagai admin jasa *service*. Sistem ini juga akan mempermudah masyarakat untuk mengetahui tempat lokasi *service* elektronik tanpa harus mencari informasi tempat jasa elektronik secara manual kelapangan atau situs jasa *service*. Aplikasi tersebut juga dilengkapi fitur konsultasi seputar perangkat elektronik, sistem *booking* agar penyedia jasa langsung cek peralatan elektronik sekaligus menghitung estimasi biaya jasa perbaikan perangkat, serta memberikan informasi jasa *service* terbaik melalui sistem penilaian yang telah di berikan oleh pelanggan yang sudah melakukan *booking* jasa *service* sebelumnya.

Dari masalah tersebut penulis melakukan penelitian untuk membuat inovasi membangun aplikasi yang dapat membantu untuk memudahkan setiap permasalahan elektronik kepada konsultan yang sesuai di bidangnya. Dengan adanya sistem ini maka masyarakat tidak perlu lagi pusing untuk melakukan perbaikan pada perangkat elektronik, mengetahui rentang harga *service* perangkat elektronik serta dengan mudah untuk mencari dimana saja lokasi penyedia jasa *service* peralatan elektronik. Berdasarkan penelitian ini maka penulis tertarik untuk mengambil judul “MEMBANGUN APLIKASI *GoSE (GO SERVICE ELECTRONIC)* BERBASIS ANDROID MENGGUNAKAN *FRAMEWORK*

REACT NATIVE DAN FIREBASE REALTIME DATABASE". Dengan mengangkat judul ini penulis berharap aplikasi ini mampu memudahkan masyarakat dalam mencari penyedia jasa *service* sekaligus bias berkonsultasi dengan para konsultan *service*.

1.2. Rumusan Masalah

Dari latar belakang diatas akan dilakukan sebuah penelitian dengan rumusan masalah yaitu :

1. Bagaimanakah membangun aplikasi GoSE (*Go Service Electronic*) menggunakan teknologi *Framework React Native* berbasis android ?
2. Bagaimanakah menerapkan teknologi *firebase realtime database* pada aplikasi GoSE (*Go Service Electronic*) ?

1.3. Batasan Masalah

Agar penelitian yang dilakukan lebih terarah dan tidak menyimpang, maka perlu dibuat batasan masalah yaitu :

1. Penggunaan sistem ini hanya dilakukan pada Kota Medan.
2. Penggunaan aplikasi hanya dilakukan pada delapan toko yaitu di toko "*Maju Jaya Computer*" dan "*Mandiri Jaya Computer*" yang terletak di Kecamatan Medan Barat, "*Hardware Laptop*" yang terletak di Kecamatan Medan Sunggal, "*Horas Computer*" yang terletak di Kecamatan Medan Kota, "*Ananda 217 Ponsel*" yang terletak di Kecamatan Medan Johor, "*399 Tech Shop*" yang terletak di Kecamatan Medan Perjuangan, "*Anchi Alfashop*" yang terletak di Kecamatan Medan Timur, serta "*DMS Service Center*" yang terletak di Kecamatan Medan Area
3. Sistem ini hanya dapat dijalankan pada OS *android* minimal versi 6.0 (*Marshmallow*).
4. Sistem ini hanya berguna untuk jasa konsultasi seputar perangkat elektronik ponsel dan PC, melakukan *booking* jasa *service*, mengetahui rentang harga *service* elektronik, serta mengetahui dimana saja lokasi penerima jasa

5. Sistem ini diimplementasikan menggunakan *Framework React Native* versi 6.0 dan menggunakan *Firebase* sebagai menyimpan data pengguna, data harga serta data laporan kegiatan *booking* jasa *service*.
6. Pendaftaran maupun login ke aplikasi hanya menggunakan email dan password.
7. Sistem ini hanya menyediakan pembayaran uang tunai untuk biaya perbaikan perangkat elektronik.
8. Sistem ini tidak menyediakan jaminan garansi elektronik melainkan secara manual yang akan diberikan oleh penyedia jasa *service*.

1.4. Tujuan Penelitian

Berikut tujuan dari penelitian saya adalah :

1. Membangun aplikasi GoSE (*Go Service Electronic*) menggunakan teknologi *Framework React Native* berbasis android.
2. Menerapkan teknologi *firebase realtime database* pada aplikasi GoSE (*Go Service Electronic*).

1.5. Manfaat Penelitian

Berikut manfaat dari penelitian saya adalah :

1. Bagi Penulis

- a. Meningkatkan pengalaman serta wawasan dalam membangun sebuah aplikasi berbasis *android* menggunakan *Framework React Native* dan *database Firebase*.
- b. mengaplikasikan ilmu akademis yang didapat saat jenjang perkuliahan.
- c. Sebagai salah satu syarat lulus sarjana (S1) Program Studi Sistem Informasi Fakultas Sains dan Teknologi.
- d. Sebagai suatu karya bagi penulis dalam hal membangun aplikasi berbasis *android* untuk meningkatkan jenjang karir penulis dalam membuat portofolio.

2. Bagi Universitas

- a. Universitas mampu meningkatkan kualitas para lulusannya terutama dalam hal membangun aplikasi berbasis *android*.
- b. Mengukur tingkat kesanggupan mahasiswa saat menguasai teori yang diterima selama jenjang perkuliahan dalam hal membangun aplikasi *GoSE (Go Service Electronic)* untuk mempermudah masyarakat.
- c. Menjadi bahan rujukan terhadap pembaca atau peneliti selanjutnya terutama dalam hal membangun aplikasi berbasis *android* menggunakan *Framework React Native* serta menggunakan *database Firebase*.

3. Bagi Penyedia Jasa Service

- a. Para usahawan penyedia jasa *service* elektronik dapat memasarkan jasanya lebih baik lagi sehingga mampu memperbesar peluang bisnis.
- b. Dapat lebih mudah diketahui dimana saja letak para penyedia jasa *service* elektronik berada kepada para pelanggan.

4. Bagi Masyarakat

- a. Bagi para pelanggan aplikasi *GoSE (Go Service Electronic)* menjadi lebih mudah dalam melakukan *booking* jasa *service* elektronik untuk melakukan perbaikan peralatan elektronik yang rusak.
- b. Menjadi lebih mudah dalam mendapatkan informasi dimana saja letak para penyedia jasa *service* elektronik.

BAB II

TINJAUAN PUSTAKA

2.1. Aplikasi

2.1.1. Pengertian Aplikasi

Aplikasi berasal dari kata *application*, yang artinya penggunaan. Ketika datang ke aplikasi, ini adalah program siap pakai yang menjalankan fungsi pengguna dan aplikasi lain dan dapat digunakan untuk tujuan apa pun. Aplikasi juga dapat didefinisikan sebagai unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan berbagai aktivitas dan pekerjaan, seperti: Kegiatan komersial, periklanan, kegiatan komunitas, permainan, dan berbagai kegiatan manusia lainnya. (Muhamad Alda 2020)

Aplikasi adalah perangkat lunak atau program komputer yang berjalan pada sistem tertentu dan ditulis serta dirancang untuk menjalankan perintah tertentu. Istilah aplikasi sendiri berasal dari bahasa Inggris *application* dan dapat diartikan sebagai aplikasi atau penggunaan. Aplikasi secara harfiah adalah bagian dari perangkat lunak yang dirancang untuk melakukan tugas tertentu. (Pane, Zamzam, and Fadillah 2020)

Setelah dikembangkan, aplikasi dapat dibagi menjadi tiga kelompok, diantaranya :

1. Aplikasi *desktop*, yaitu aplikasi yang hanya berjalan di perangkat PC komputer atau laptop.
2. Aplikasi *web*, yaitu aplikasi yang dijalankan dengan komputer dan koneksi internet.
3. Aplikasi seluler, yaitu aplikasi yang dijalankan di perangkat seluler. Kategori ini banyak kegunaannya. (Pane et al. 2020)

2.1.2. Jenis-Jenis Aplikasi

Berikut ini adalah berbagai macam jenis Aplikasi :

1. Aplikasi internet, yaitu aplikasi yang memfasilitasi akses ke internet, seperti *Yahoo, Gmail, Google*, dan lainnya.
2. Aplikasi *office*, yaitu aplikasi yang digunakan di area perkantoran untuk mempercepat pekerjaan, seperti *MS Office*.
3. Aplikasi *grafis*, yakni aplikasi untuk mengedit foto dan pembuatan desain, contohnya *adobe photoshop*.
4. Aplikasi *programming*, ini adalah aplikasi khusus yang digunakan untuk membuat program atau perangkat lunak tertentu, seperti *visual basic, Delphi 7*, dan lainnya.
5. Aplikasi *multimedia*, yaitu aplikasi yang mendukung pembuatan informasi dalam format teks, video, dan audio. Contohnya termasuk *winamp*, pemutar media musik, dan lain-lain.
6. Aplikasi *game*, yaitu aplikasi yang berisi sebuah permainan, dan digunakan hanya untuk tujuan hiburan. (Pane et al. 2020)

Anda dapat mengelompokkan aplikasi Anda ke dalam tujuh jenis untuk membantu Anda mengenali jenis aplikasi. Pengelompokan aplikasi adalah sebagai berikut:

1. *System Software*, yaitu aplikasi yang dapat mengatur dan mengontrol operasi internal sistem komputer.
2. *Real Time Software*, yaitu aplikasi dengan kemampuan untuk memantau, menganalisa, dan mengontrol suatu peristiwa aktual yang sedang berlangsung.
3. *Business Software*, yaitu aplikasi yang dibuat dan dikembangkan untuk tujuan bisnis seperti mengelola sistem keuangan.
4. *Engineering and Scientific Software*, aplikasi yang dibuat dan dikembangkan untuk membantu manusia memecahkan masalah *non algoritmik*.
5. *Web Based Software*. ini adalah aplikasi yang bertindak sebagai media penghubung antara pengguna dengan internet. (Pane et al. 2020)

2.2. *Electronic*

Elektronik adalah alat yang didasarkan pada prinsip-prinsip elektronik, dan benda-benda dan benda-benda yang menggunakan alat-alat ini, (Pusat Bahasa, 2005). Dalam Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik. (Ilham 2021)

Berikut beberapa definisi yang kaitannya dengan elektronik, sebagai berikut:

1. Informasi Elektronik, tulisan, *audio*, gambar, peta, *design*, foto, *Electronic Data Interchange* (EDI), surat elektronik (*email*), *telegram*, *teleks*, telekopi atau satu atau sekumpulan data elektronik yang sejenisnya. Demikian pula dengan olahan huruf, tanda, angka, kode akses, simbol, atau *perforasi* yang dapat dimengerti atau bermakna bagi siapa saja yang memahaminya.
2. Transaksi Elektronik, perbuatan hukum yang dilakukan dengan menggunakan perdagangan elektronik, jaringan komputer, dan/atau media elektronik lainnya.
3. Teknologi Informasi, teknologi untuk mengumpulkan, menyiapkan, menyimpan, memproses, mengumumkan, menganalisis, dan/atau menyebarkan informasi.
4. Informasi elektronik yang dapat dibuat, ditransmisikan, ditransmisikan, diterima, atau disimpan dalam dokumen elektronik, elektronik, analog, digital, elektromagnetik, optik, atau format serupa, dan dilihat, ditampilkan, dan/atau didengar melalui komputer atau sistem elektronik. atau makna yang dipahami atau dapat dipahami oleh orang yang dapat memahaminya.
5. Penyelenggaraan Sistem Elektronik, penggunaan sistem elektronik oleh penyelenggara pemerintahan, perorangan, dunia usaha, dan/atau masyarakat.
6. Jaringan Sistem Elektronik, hubungan dua atau lebih sistem elektronik, tertutup maupun terbuka.
7. Agen elektronik adalah perangkat dalam sistem elektronik yang dibuat untuk melakukan tindakan pada informasi elektronik tertentu dan secara otomatis dikendalikan oleh manusia. (Ilham 2021)

2.3. *Online*

Pada dasarnya, *online* terhubung dengan internet, dan aktif, sehingga anda dapat berkomunikasi dengan atau dikendalikan oleh komputer. *Online* juga dapat didefinisikan sebagai situasi di mana suatu perangkat atau komputer dapat terhubung ke komputer atau perangkat lain, biasanya melalui perangkat terbaru. Pengertian *online* juga dapat digambarkan sebagai suatu keadaan dimana seorang pengguna dapat menggunakan dan terhubung dalam suatu jaringan, menghubungkan satu perangkat dengan perangkat lainnya, dan menjalin komunikasi antara satu perangkat dengan perangkat lainnya. (Almuttaqin 2016)

2.4. *Android*

2.4.1. Pengertian *Android*

Android adalah sistem operasi berbasis *linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* juga memberi wadah terbuka untuk setiap pengembang supaya membuat aplikasi mereka sendiri untuk digunakan pada berbagai perangkat seluler. *Android* menyediakan pengembang dengan sistem operasi terbuka untuk membangun aplikasi. *Android* dirilis ke publik pada musim gugur di tahun 2008. *Android* berkembang pesat di industri karena beberapa aspek utama yaitu bersifat *open source* dan model arsitektur. (Muhamad Alda 2020)

Android yaitu sebuah sistem operasi untuk perangkat *mobile* berbasis sistem operasi *linux* yang dapat mencakup sistem operasi, *middleware* dan *aplikasi*. *Android* menyediakan platform terbuka ataupun *opensource* bagi para pengembang, dimana para pengguna ataupun pengembangnya dapat dengan bebas menciptakan bahkan memodifikasi aplikasi mereka sesuai keinginan dan kebutuhan. (Pratama and Hermawan 2016)

2.4.2. Sejarah *Android*

Android awalnya dikembangkan oleh *Android Inc*, dipimpin oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White, untuk menciptakan sistem

operasi untuk kamera digital, tetapi pasar untuk perangkat ini tidak sebesar itu, jadi tidak untuk perkembangan *smartphone* sudah bergeser. Kompatibel dengan *Symbian* dan *Windows Mobile*. Dikatakan untuk mengembangkan perangkat lunak untuk ponsel, meskipun oleh seorang ahli teknis yang memiliki pengalaman dalam mengembangkan sistem operasi. (Gunawan et al. 2021)

Selama proses pengembangan, tim *Android Inc* kehabisan dana dan akhirnya pada 17 Agustus 2005, *Google* mengakuisisi *Android Inc*, menjadikannya anak perusahaan yang dimiliki sepenuhnya oleh *Google*. Namun, para pendiri *Android Inc*. masih bekerja secara internal. Pada tahun 2007, sistem operasi *Android* dirilis dan organisasi *Open Handset Alliance* didirikan. Selain *Google Open Handset Alliance*, ada juga perusahaan ternama seperti *Samsung*, *Motorola*, *LG*, *Toshiba*, *Intel*, *T-Mobile*, *Vodafone* dan *Sony Ericsson*. (Gunawan et al. 2021)

2.4.3. Fitur-Fitur *Android*

Android adalah sistem operasi seluler *open source* dan dapat dikustomisasi, sehingga tidak ada konfigurasi perangkat lunak atau perangkat keras khusus. Namun, secara umum, *Android* sendiri mendukung fitur-fitur berikut: (Zikria, 2016).

1. Storage: Mendukung SQL Lite. SQL Lite adalah database relasional ringan (versi ringan) yang digunakan untuk penyimpanan data.
2. Konektivitas: mampu koneksi *GSM/EDGE*, *IDEN*, *CDMA*, *EVDO*, *UMTS*, *Bluetooth*, *Wifi*, *LTE*, dan *Wimax*.
3. Messaging: bisa melakukan *SMS* dan *MMS*.
4. Menggunakan *browser* berbasis *open source Webkit*, dengan *engine javascript Chrome V8*.
5. *Media Support*: Mendukung file media bertipe: *H.263*, *H.264* (dalam bentuk *3GP* atau *MP4 container*), *MP3*, *MIDI*, *Ogg*, *Vorbis*, *WAV*, *JPEG*, *PNG*, *GIF*,

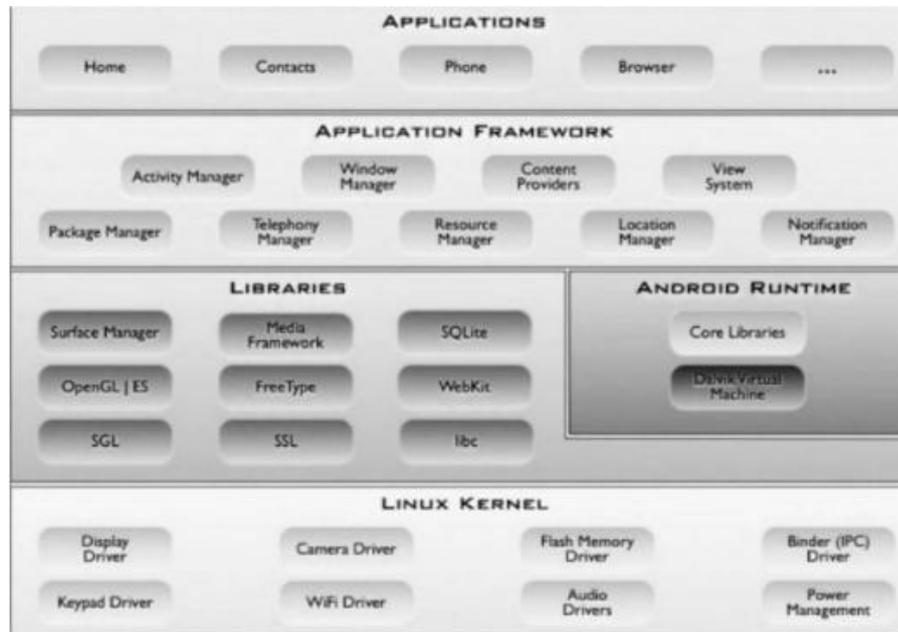
dan *BMP*, *MPEG-4 SP*, *AMR*, *AMR-WB (3GP container)*, *AAC*, *HE-AAC* (dalam bentuk *MP4* atau *3GP container*).

6. Dukungan *hardware*: *GPS (Global Positioning System)*, *camera*, kompas digital, *sensor proximity*, *Sensor accelerometer*.
7. *Multi Touch*: dukungan *multi touch*.
8. *Multitasking*: melaksanakan tugas secara bersamaan.
9. *Flash*: animasi *Flash*.
10. *Tethering*: mendukung berbagi koneksi internet.

2.4.4. Arsitektur Android

Secara garis besar arsitektur Android terdiri dari aplikasi dan widget, *framework* aplikasi, *library*, *runtime Android*, dan *kernel Linux*. (Supardi 2017)

1. Aplikasi dan widget adalah lapisan yang hanya berhubungan dengan aplikasi.
2. *Application Framework* merupakan *Open Development Platform* yang ditawarkan Android untuk dapat dikembangkan guna membangun aplikasi. Pengembang memiliki akses penuh menuju *API Frameworks* seperti yang dilakukan oleh aplikasi kategori inti. Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.
3. Pustaka adalah lapisan tempat fitur Android ada.
4. *Android Run Time* adalah lapisan yang menggunakan implementasi Linux untuk menjalankan aplikasi Android.
5. *Kernel Linux* adalah lapisan inti dari sistem operasi Android. Lihat gambar arsitektur Android di bawah ini untuk informasi selengkapnya.



Gambar 2.1 : Arsitektur Android (Supardi 2017)

2.4.5. Versi-versi *Android*

Menurut (Kusniyati 2016), Sejak diperkenalkan pada tahun 2007, *Android* telah mengalami perkembangan pesat yang menampilkan banyak versi sistem operasi *Android*.

1. *Android* versi 1.1



Gambar 2.2 : *Android* Versi 1.1 (Adikara 2013)

Android pertama kali diluncurkan pada tahun 2007, namun sistem operasinya dirilis pada tanggal 9 Maret 2009 dan diterapkan pada berbagai macam gadget.

Android versi 1.1 adalah Android pertama yang versi ini hanya menyentuh beberapa aplikasi seperti sistem antarmuka pengguna yang lebih baik dan beberapa aplikasi lainnya.

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



Gambar 2.3 : *Versi 1.5 (Cupcake) (Adikara 2013)*

Pada Mei 2009, Android menerima perubahan versi lain. Android versi 1.1 kemudian diperpanjang dengan Android versi 1.5 atau biasa disebut *Android* cupcakes.

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



Gambar 2.4 : *Android Versi 1.6 (Donut) (Adikara 2013)*

Donat (versi 1.6) dirilis dalam waktu empat bulan sejak peluncuran *cupcake* Android pertama pada September 2009.

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



Gambar 2.5 : Android Versi 2.0/2.1 (Eclair) (Adikara 2013)

Pada tahun yang sama, Android merilis versi *Eclair*, Android versi 2.0 / 2.1 *Eclair*. Android *Eclair* dirilis oleh Google tiga bulan setelah dirilis.

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



Gambar 2.6 : Android Versi 2.2 (Froyo: Frozen Yoghurt) (Adikara 2013)

Butuh lima bulan bagi *Google* buat melakukan regenerasi menurut *Android Eclair* versi sebelumnya ke versi *Froyo Frozen Yoghurt*. Pada lepas 20 Mei 2010, *Android* versi dua titik dua alias *Android Froyo* ini dirilis.

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



Gambar 2.7 : *Android* Versi 2.3 (*Gingerbread*) (Adikara 2013)

Tujuh bulan kemudian, *Android* mencapai terobosan lain yang dikenal sebagai *Android* versi 2.3 atau *Android Gingerbread* yang baru dirilis.

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



Gambar 2.8 : *Android* Honeycomb Versi 3.0/3.1 (Adikara 2013)

Pada versi ini *HoneyComb* dirilis pada Mei 2011. *Android Honeycomb* adalah *operating system* yang dirancang khusus untuk digunakan dengan tablet basis *Android*.

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



Gambar 2.9 : *Android* Versi 4.0 (*ICS: Ice Cream Sandwich*)
(Adikara 2013)

Android ICS atau *Ice Cream Sandwich* jua dirilis dalam tahun yg sama menggunakan *Honeycomb*, yaitu dalam bulan Oktober 2011.

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



Gambar 2.10 : *Android* Versi 4.1 (*Jelly Bean*) (Adikara 2013)

Android Jelly Bean saat ini merupakan versi terbaru dari *Android*. Salah satu gadget yang menggunakan sistem operasi *Jelly Bean* adalah *Google Nexus 7* yang dirilis oleh ASUS, vendor asal Taiwan yang juga merupakan sahabat Acer di kampung halamannya.

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



Gambar 2.11 : *Android* Versi 4.4 (*Kit Kat*) (Adikara 2013)

Keberadaan *Android KitKat* merupakan pengenalan produk sistem operasi baru yang dirilis pada 4 September 2013. Sebelumnya, beredar rumor bahwa *Android* akan meluncurkan sistem operasi baru yang disebut *Android Key Lime Pie*, tetapi analisis menunjukkan bahwa itu tidak cocok dengan ejaan rata-rata orang, sehingga nama itu diberi nama OS *Android KitKat*, yang akrab dengan kebanyakan orang. Ini telah berubah.

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



Gambar 2.12 : *Android* Versi 5.0.2 (*Lollipop*) (Adikara 2013)

Android Lollipop adalah hadirnya sistem operasi *Android* yang menjadi tren baru di industri *smartphone* saat ini. Hal ini tidak terlepas dari keunikan dan manfaat yang banyak didapatkan orang dari sistem operasinya. Keberadaan *Android* versi ini dinantikan banyak pihak, berharap sistem operasi *Lollipop* akan lebih baik dari versi sebelumnya.

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



Gambar 2.13 : *Android* Versi 6.0 (*Marshmallow*) (Adikara 2013)

Android 6.0 Marshmallow adalah versi sistem operasi seluler *Android*. Ini pertama kali diperkenalkan dengan *codename Android M* di *Google I/O* pada Mei 2015 dan resmi dirilis pada Oktober 2015. *Android Marshmallow* memperkenalkan model izin aplikasi yang didesain ulang. Saat ini hanya ada 8 kategori izin dan aplikasi tidak lagi otomatis.

13. *Android* 7.0 (*Nougat*)

Versi ini dirilis pada Oktober 2016. Versi ini menambahkan dukungan mode malam, dukungan untuk beberapa panggilan titik akhir, *keyboard* standar yang memungkinkan Anda mengirim GIF animasi secara langsung, dan bisa

multitasking sehingga pengguna dapat membuka aplikasi tanpa menutup aplikasi yang lain. (Gunawan et al. 2021)

14. *Android 8.0 (Oreo)*

Versi ini dirilis pada Agustus 2017. Versi ini memiliki fitur *autofill* yang memudahkan pengisian formulir. (Gunawan et al. 2021)

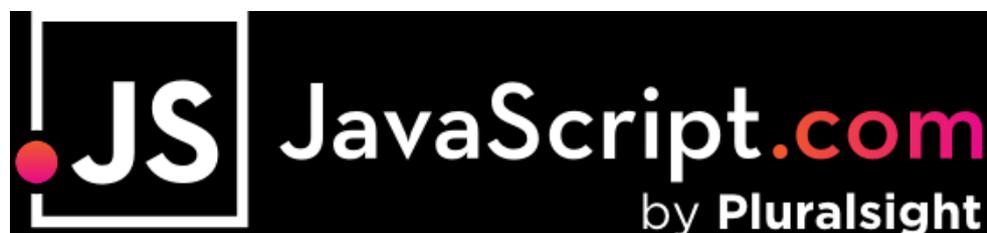
15. *Android 9.0 (Pie)*

Versi ini dirilis pada Agustus 2018. Fitur utama dari versi ini adalah kecerdasan buatan atau fitur AI. Dengan fungsi ini, Anda dapat secara otomatis mempelajari pola penggunaan di smartphone Anda. Ini juga mendukung ponsel tanpa bezel dan kecerahan adaptif yang secara otomatis menyesuaikan kecerahan layar. (Gunawan et al. 2021)

16. *Android 10*

Versi ini dirilis pada September 2019. Versi ini memiliki keunggulan seperti mode fokus, tema gelap, balasan cerdas, mengontrol lokasi, teks langsung, notifikasi intuitif, proyek arus utama, pembaruan tautan keluarga, navigasi gerakan baru, dan banyak lagi. (Gunawan et al. 2021)

2.5. *Javascript*



Gambar 2.14 : Logo *JavaScript* (www.javascript.com)

2.5.1. *Pengertian JavaScript*

JavaScript yaitu pemrograman basis *web* yang diproses di sisi klien. *JavaScript* dijalankan di sisi klien, jadi *JavaScript* hanya dijalankan menggunakan *browser*. Tidak seperti PHP, yang berjalan di sisi *server*, Anda tidak perlu

memperbarui *browser* untuk menjalankan skrip *JavaScript*. *JavaScript* biasanya dijalankan ketika peristiwa tertentu terjadi di halaman *web*. Kedua peristiwa yang dipicu oleh pengguna dan peristiwa yang dipicu oleh perubahan halaman di situs. *JavaScript* mulai sangat populer sejak munculnya *query* sebagai salah satu *library* yang memudahkan penggunaan *JavaScript*. Namun seiring munculnya *library* baru dan *framework* baru dengan berbagai kelebihan, popularitas *jQuery* mulai tergeser. (Abdulloh 2020)

2.5.2. Sejarah *Javascript*

JavaScript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di *browser*. Namun, seiring waktu, *JavaScript* tidak hanya berjalan di *browser* Anda. *Javascript* dapat digunakan pada *server*, game, IoT, desktop, dan lainnya. *Javascript* awalnya disebut *Mocha*, tetapi diubah menjadi *LiveScript* ketika *Netscape Navigator V2.0* dirilis dalam versi *beta* (September 1995). Namun, kemudian diubah namanya menjadi *Javascript*. Terinspirasi oleh kesuksesan *Javascript*, *Microsoft* membuat teknologi serupa. *Microsoft* telah membuat versi "Javascript"-nya sendiri yang disebut *JScript*. Itu kemudian ditanam di *Internet Explorer 3.0*. Hal ini menyebabkan "perang *browser*" karena *JScript Microsoft* berbeda dari *Javascript Netscape*. Akhirnya, pada tahun 1996 *Netscape* menyerahkan standar ECMA-262 ke *Ecma International*. Sehingga muncullah standarisasi kode *Javascript* bernama *ECMAScript* atau *ES* (Supardi 2021). Saat ini *ECMAScript* sudah mencapai versi 10 (*ES10*), lihat table di bawah ini:

Tabel 2. 1 : Edisi *ECMAScript* (Supardi 2021)

Edisi	Tanggal Terbit	Nama	Perubahan Dari Edisi Sebelum	Editor
1	Juni 1997		Edisi Pertama	Guy L. Steele Jr.
2	Juni 1998		Perubahan editorial untuk menjaga spesifikasi sepenuhnya	Mike Cowlishaw

			selaras internasional ISO/IEC 16262	
3	Desember 1999		Menambahkan ekspresi <i>regular</i> , penanganan <i>string</i> yang lebih baik, pernyataan kontrol baru, penanganan pengecualian coba/tangkapan, defenisi kesalahan yang lebih ketat, <i>pemformatan</i> untuk <i>output numeric</i> dan peningkatan lainnya	Mike Cowlshaw
4	Ditinggalkan		Edisi Keempat ditinggalkan, karena perbedaan politik mengenai kompleksitas bahasa. Banyak fitur yang diusulkan untuk Edisi keempat telah sepenuhnya dihapus; beberapa dimasukkan ke dalam edisi keenam.	
5	Desember 2009		Menambahkan “mode ketat”, bagian yang dimaksudkan untuk memberikan pemeriksaan kesalahan yang lebih menyeluruh dan menghindari konstruksi yang cenderung kesalahan. Mengklarifikasi banyak	Pratap Lakshman, Allen Wirfs- Brock

			ambiguitas dalam spesifikasi edisi ke-3, dan mengakomodasi perilaku implementasi dunia nyata yang berbeda secara konsisten dari spesifikasi itu. Menambahkan beberapa fitur baru, seperti <i>getter</i> dan <i>setter</i> , dukungan perpustakaan untuk <i>JSON</i> , dan refleksi yang lebih lengkap tentang properti objek.	
5.1	Juni 2011		Edisi 5.1 dari standar <i>ECMAScript</i> ini sepenuhnya selaras dengan edisi ketiga standar internasional ISO / IEC 16262: 2011.	Allen Wirfs-Brock
6	Juni 2015 [11]	<i>ECMAScript</i> 2015 (ES2015)	Lihat Edisi 6 – <i>ECMAScript</i> 2015	Allen Wirfs-Brock
7	Juni 2016 [12]	<i>ECMAScript</i> 2016 (ES2016)	Lihat Edisi 7 – <i>ECMAScript</i> 2016	Brian Terlson
8	Juni 2017 [13]	<i>ECMAScript</i> 2017 (ES2017)	Lihat Edisi 8 – <i>ECMAScript</i> 2017	Brian Terlson
9	Juni 2018 [14]	<i>ECMAScript</i> 2018 (ES2018)	Lihat Edisi 9 – <i>ECMAScript</i> 2018	Brian Terlson
10	Juni 2019 [9]	<i>ECMAScript</i> 2019 (ES2019)	Lihat Edisi 10 – <i>ECMAScript</i> 2019	Brian Terlson, Bradley Farias, Jordan Harband

2.6. *Framework*

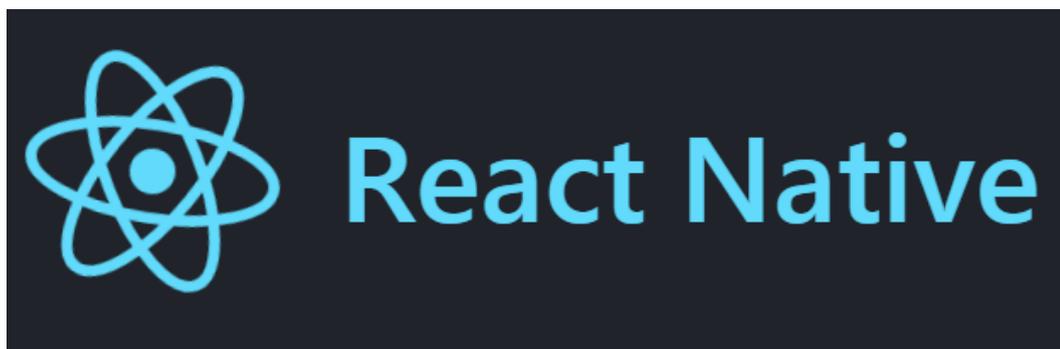
Framework adalah kumpulan direktif yang dikumpulkan dengan masing-masing fungsi dalam kelas dan fungsi, sehingga memudahkan pengembang/programmer untuk memanggilnya tanpa harus menulis sintaks program yang sama berkali-kali. Ini menghemat waktu dan membantu Anda menghindari penulisan ulang sintaks untuk membuat kode sumber yang bersih dan terstruktur. (Habibi and Sandi 2020)

Framework adalah kerangka kerja yang digunakan untuk mengembangkan aplikasi berbasis desktop atau berbasis *web*. Kerangka kerja ini sengaja dibuat, terstruktur dan terstruktur untuk membantu pengembang mengembangkan aplikasi mereka lebih cepat. Kerangka kerja menyederhanakan pembuatan aplikasi hanya dengan mengkompilasi komponen pemrograman yang tersedia. Mencegah pengembang dan pemrogram dari keharusan memprogram berulang kali. (Nurhidayah, Fauzan, and Rahayu 2020)

Framework memiliki fungsi berikut ini (Nurhidayah et al. 2020) :

- a. Program Menjadi Lebih Terstruktur dan Tersusun
- b. Praktis untuk Developer
- c. Memiliki Keamanan yang Lebih Unggul

2.7. *React Native*



Gambar 2.15 : Logo React Native (www.reactnative.dev)

2.7.1. Pengertian *React Native*

React Native adalah kerangka kerja *Javascript* untuk membuat aplikasi seluler asli untuk iOS dan Android. Ini didasarkan pada perpustakaan *javascript Facebook* untuk membangun antarmuka pengguna, Bereaksi, tetapi tidak untuk *browser*, tetapi untuk *platform* seluler. Dengan kata lain, pengembang web kini dapat membuat aplikasi seluler yang terlihat dan terasa benar-benar "asli" dari kenyamanan perpustakaan *Javascript* yang kita semua kenal dan sukai. Sebagian besar kode plat nomor yang Anda buat dapat dibagikan, jadi *React Native* memudahkan pengembangan Android dan iOS secara bersamaan. (Eisenman 2017)

React Native sendiri ditulis dengan menggabungkan *JavaScript* dan *markup* gaya XML yang disebut *JSX*, dan *React Native* ditulis dengan memanggil API rendering asli di *Objective-C* (untuk iOS) atau *Java* (untuk *Android*). "Bridge". Dengan cara ini, aplikasi yang dibuat atau dibuat dirender menggunakan komponen UI asli daripada antarmuka *web* dan terlihat seperti aplikasi seluler lainnya. Selain itu, API *platform React Native* memperlihatkan antarmuka *JavaScript*, sehingga aplikasi *React Native* dapat mengakses fitur *platform* seperti kamera dan lokasi pengguna. Proyek inti *React Native* mendukung pembuatan aplikasi seluler untuk iOS dan Android. (Eisenman 2017)

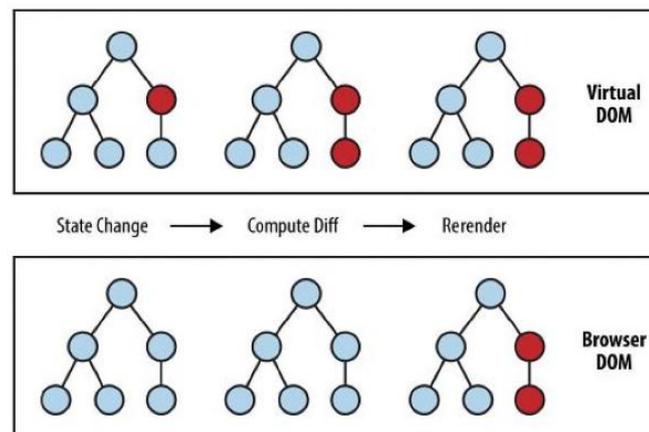
2.7.2. Keuntungan *React Native*

Fakta bahwa *React Native* merender menggunakan API rendering standar *platform* membuat *React Native* berbeda dari kebanyakan kerangka kerja pengembangan *multi-platform* lainnya seperti *Cordova* dan *Ionic*. Kerangka kerja aplikasi seluler yang menggabungkan *JavaScript*, *HTML*, dan *CSS* biasanya ditampilkan melalui antarmuka *web*. Ini berfungsi, tetapi memiliki kekurangan, terutama dalam hal kinerja. Selain itu, kerangka kerja biasanya tidak memiliki akses ke elemen UI asli. Ketika kerangka kerja ini mencoba meniru UI asli, hasilnya biasanya sangat kabur. Detail rekayasa terbalik seperti animasi juga bisa sangat padat karya, dan aplikasi yang menggunakan kerangka kerja ini dapat dengan cepat menjadi usang. Sebaliknya, *React Native* sebenarnya mengubah

markup yang dihasilkan menjadi elemen UI asli, memberikan *platform* yang Anda kerjakan tampilan asli. Siklus pembaruan *React Native* sama dengan *React*. *React Native* merender tampilan saat *prop* atau *state* berubah. Perbedaan utama antara *React Native* dan *Reactjs* di browser adalah bahwa *React Native* melakukan ini dengan memanfaatkan perpustakaan UI dari *platform* asli (Android atau iOS) daripada menggunakan markup HTML dan CSS. Pengembang yang terbiasa bekerja di *web* dapat menggunakan alat dan bahasa yang dikuasai untuk memperoleh pengalaman membangun aplikasi seluler dengan kinerja dan tampilan serta nuansa *platform* seluler asli mereka. (Eisenman 2017)

2.7.3. Cara Kerja *React Native*

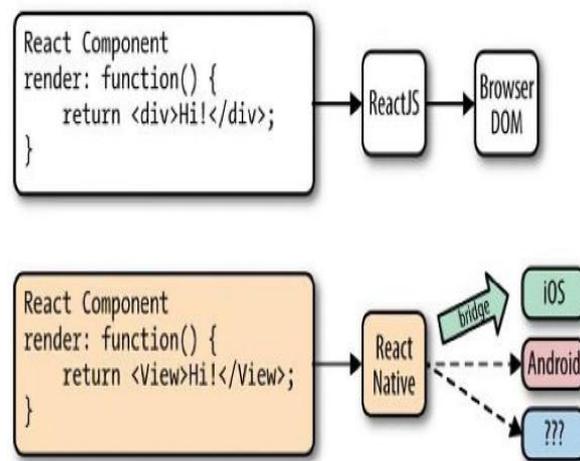
Untuk memahami teknologi yang mendasari *React Native*, Anda harus terlebih dahulu memahami salah satu konsep *React*, *Virtual-DOM.web*. Untuk membuat antarmuka pengguna interaktif di browser, pengembang perlu mengedit DOM atau model objek dokumen. Prosedur ini sulit dibuat karena memiliki dampak yang signifikan terhadap kinerja. Gambar 2.16 akan menunjukkan cara kerjanya. (Eisenman 2017)



Gambar 2.16 : Cara Kerja DOM (Eisenman 2017)

Di Bereaksi di *Web*, sebagian besar pengembang melihat DOM *virtual* sebagai pengoptimalan kinerja. DOM virtual memang memiliki manfaat kinerja, tetapi potensi sebenarnya terletak pada kekuatan abstraksi. Tambahkan lapisan

abstraksi bersih antara kode pengembang dan render. Ini menunjukkan bagaimana *React Native* bekerja, seperti yang ditunjukkan pada Gambar 2.16. Alih-alih merender di DOM browser, *React* memanggil *Objective-C* API secara *native* untuk merender dalam komponen iOS, atau memanggil *Java* API untuk merender di komponen *Android*. Ini membuat *React Native* berbeda dari opsi pengembangan aplikasi lintas *platform* lainnya yang sering membuat tampilan berbasis *web*. (Eisenman 2017)



Gambar 2.17 : Perbedaan Rendering *React Native* (Eisenman 2017)

Ini dimungkinkan oleh jembatan yang menghubungkan *React* ke antarmuka elemen UI asli *platform host*. Komponen *React* mengembalikan *markup* yang menjelaskan bagaimana tampilannya di *platform host*. Jadi sementara *React for the web* menerjemahkan langsung ke DOM browser, *React Native* menerjemahkan *markup* ini agar sesuai dengan *platform host*, jadi ini bisa menjadi tampilan UI khusus iOS. Proyek inti *React Native* juga mendukung iOS dan *Android*. (Eisenman 2017)

2.8. Database SQL

2.8.1. Pengertian Database SQL

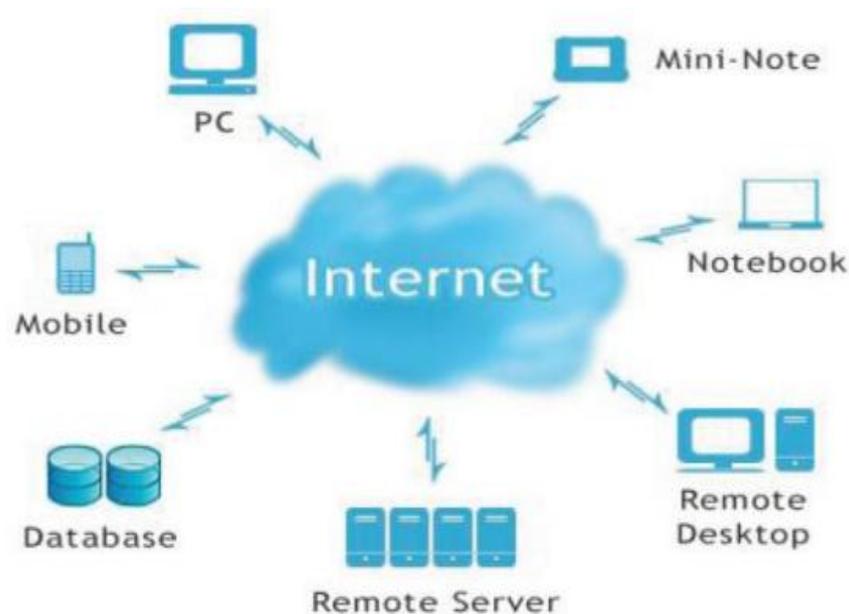
Basis data relasional (*Database SQL*) adalah kumpulan dari item-item yang distrukturkan sebagai tabel yang terdeskripsi secara formal dimana data bisa di-akses dan disusun kembali tanpa harus mengorganisasikan kembali tabel-tabel

tersebut. Basis data relasional mengorganisasikan data dari satu tabel ketabel lainnya (relasi) yang berisi kolom dan baris, dengan *key* yang unik untuk setiap barisnya. (Silalahi 2018)

2.8.2. Sejarah Database SQL

Sejarah SQL dimulai pada bulan Juni 1970 dengan sebuah artikel oleh seorang peneliti IBM bernama EFCod membahas ide untuk membuat basis data relasional. Artikel ini juga menjelaskan kemungkinan membuat bahasa standar untuk mengakses data dalam basis data. Bahasa ini kemudian disebut SEQUEL (*Structured English Query Language*). Setelah artikel tersebut diterbitkan, IBM memulai sebuah proyek untuk membuat basis data relasional berdasarkan bahasa SEQUEL. Namun, karena masalah hukum dengan penamaan SEQUEL, IBM mengubah SEQUEL menjadi SQL. Implementasi database relasional dikenal sebagai *System / R*. Pada akhir 1970-an, sebuah perusahaan bernama Oracle muncul untuk membuat *server* basis data populer dengan nama yang sama. Kami juga memperhatikan bahwa ketika *Oracle* menjadi lebih populer. (Eri Mardiani 2016)

2.9. Cloud Computing



Gambar 2.18 : *Cloud Computing* (Ahmad and Setiawan 2011)

Cloud computing adalah teknologi dimana berita disimpan disuatu *server* yg bisa diakses sang client melalui jaringan internet, & menggunakan menggabungkan teknologi komputer menggunakan pengembangan internet sebagai akibatnya sebagai insfrastruktur komplek yg tak berbentuk & tersembunyi. (Cancer and Alim 2016)

Komputasi awan berkembang secara paralel dengan adanya perkembangan *Internet* dan *Web*. Contoh komputasi awan adalah *Google Cloud Firebase* , *Microsoft Azure*, *Amazon Web Services*. *Cloud Computing* dibagi menjadi dua bagian, ujung depan dan ujung belakang, yang terhubung melalui jaringan *Internet*. *Front end* berada di sisi pengguna komputer (dalam hal ini *client*), dan *back end* adalah sistem itu sendiri berupa *server* dan sistem penyimpanan data di *cloud* atau layanan komputasi. Beberapa manfaat yang dapat Anda capai melalui komputasi awan adalah:

1. Bagi para pelaku yang bergerak pada industri IT, hal ini merupakan peluang yang besar karena dengan meningkatnya penggunaan layanan dan meningkatkan penggunaan internet.
2. Integrasi aplikasi diberbagai *device* (perangkat).
3. Memungkinkan pengembangan dan implementasi dengan cepat sehingga meningkatkan produktifitas.
4. Minimalisir biaya infrastruktur dalam dunia industri IT. (Mutia 2016a)

2.9.1. Sejarah *Cloud Computing*

Evolusi yang disebut dengan *alort computing* atau yang biasa disebut dengan *cloud computing* ini awalnya lahir pada tahun 1960-an. Berbagi waktu dan *multitasking*, yang merupakan fitur umum di sistem operasi saat ini, adalah fitur utama yang memungkinkan pengembangan teknologi komputasi awan. Pertama, revolusi komputasi ini telah menjadikan komputasi awan seperti IBM sebagai pengembang terbesar teknologi ini. IBM *Mainframe* adalah sistem yang sepenuhnya tersedia secara komersial yang memungkinkan pengguna untuk

menangani hanya beberapa perintah sederhana. Pada 1970-an, sebagai penyedia layanan, *Timeshare* mengembangkan sistem berbagi ruang *mainframe* yang dapat diakses melalui saluran telepon. Perusahaan ini menawarkan layanan tingkat tinggi pada komputer yang ditargetkan pada perusahaan besar pada saat itu. (Husain et al. 2018)

Pada dekade 1980-an dan 1990-an, perusahaan lahir untuk memproduksi *mainframe* besar dan komputer kecil. Penyewa dari perusahaan-perusahaan ini terhubung ke pusat data melalui jalur ISDN dengan daya komputasi yang tersedia untuk pelanggan mereka, sehingga perangkat keras yang mereka buat dapat dikirimkan ke perusahaan dengan biaya bulanan atau tahunan untuk memenuhi kebutuhan pusat data. disewa. Data pelanggan dicadangkan ke *tape* dan dapat digunakan jika terjadi kegagalan sistem. Selama ini, pengembangan sistem persewaan sumber daya mengarah pada pengembangan sistem SLA atau perjanjian tingkat layanan. Dekade 1980 dan 1990 juga merupakan kelahiran penyedia layanan aplikasi (ASP). ASP juga dikenal sebagai layanan pihak ketiga yang bekerja dalam arti delegasi pada saat itu.. (Husain et al. 2018)

Data, pengelolaan data, dan *hosting* perangkat lunak dalam layanan berbasis rental. ASP membuka peluang bagi UKM untuk memanfaatkan teknologi terkini dengan biaya yang relatif murah. Harga, layanan pelanggan, dan RAS (*reliability, availability, maintainability*) adalah parameter ASP. Dekade 2000 adalah tahun pertumbuhan pesat dalam penggunaan ASP. Ketika sebuah perusahaan besar seperti AT&T atau *Oracle* sedang mengintegrasikan layanan ASP ke dalam bisnis mereka. Namun, layanan ini terbatas untuk pengguna perusahaan dan tidak menjangkau pengguna akhir sebagai konsumen. Ini karena ASP sangat sulit dan mahal untuk dipasang dan dirawat. Sejumlah besar orang diperlukan untuk mendukung dan memelihara sistem ini bagi pengguna akhir. Oleh karena itu, banyak perusahaan ASP bangkrut dan hanya sedikit yang dapat bertahan. Dan selama seluruh evolusi sistem komputer, dua konsep dasar pada akhirnya akan muncul, dan sebuah teknologi baru, yang kemudian disebut komputasi awan, akan

lahir. Konsepnya adalah sebagai berikut: *Hosting web* dan aplikasi *web* asli. (Husain et al. 2018)

2.9.2. Perkembangan *Cloud Computing*



Gambar 2.19 : Perkembangan *Cloud Computing* (Husain et al. 2018)

Perkembangan komputasi awan merupakan teknologi masa depan dan tentunya akan berguna baik dalam kehidupan maupun teknologi banyak orang di masa depan, atau di tahun-tahun mendatang. *Siakad Online System* mengimplementasikan model *SaaS* komputasi awan. Dengan sistem ini, Anda tidak perlu membeli *server* atau membuat aplikasi yang sangat kompleks dan membutuhkan keahlian yang sangat khusus serta biaya operasional yang sangat tinggi. Dengan koneksi internet, Anda tidak perlu khawatir tentang pemeliharaan server atau aplikasi atau menghabiskan banyak uang. Anda dapat mengakses *Siakad* secara online. Oleh karena itu, aplikasi *cloud computing* dapat digunakan oleh institusi pendidikan sebagai solusi murah untuk meningkatkan kualitas pendidikan, namun tidak murah. Saya sendiri bangga *Eresha School* menerapkan teknologi ini untuk mengembangkan IT di Indonesia. Tentunya kita mengetahui pentingnya belajar di dunia IT dari beberapa aplikasi *Cloud Computing for the Future* di atas. Ini bagus untuk banyak orang dan juga bangga pada kami.

Tentunya masa depan IT *cloud computing* akan berkembang pesat, dan kita harus mampu berperan memajukan dunia IT di Indonesia. (Husain et al. 2018)

a) *Cloud Computing* untuk Pemerintahan

Karena perkembangan teknologi yang pesat, banyak industri yang mengikuti perkembangan zaman. Seperti halnya pemerintah, tidak dapat dipungkiri bahwa penerapan TIK untuk memperkuat program pengabdian kepada masyarakat akan membangun infrastruktur TIK (Teknologi Informasi dan Komunikasi). Pemerintah berbasis Internet saat ini diimplementasikan baik di G2G (*Government to Government*), G2B (*Government to Enterprise*), dan G2C (*Government to Citizen*) di kedua sistem *e-government*. 2018 Semua pemerintah daerah baik di kota maupun kabupaten telah mendirikan kantor baru untuk TIK. Namanya Komunikasi dan Layanan TI. [14]. Sistem ini bertujuan untuk memberikan pelayanan dan menerapkan tata kelola yang transparan, efisien dan efektif. Masyarakat diharapkan dapat merasakan langsung implikasi sosial, ekonomi dan budaya dari keberadaan TIK, seperti peningkatan layanan masyarakat dan nilai investasi lokal. Sinombor dan Taslim menunjukkan bahwa implementasi *e-government* memiliki implikasi yang signifikan untuk semua bidang [15]. Dalam model ini, layanan *e-government* dimulai dari yang paling dasar dan memiliki empat tingkatan:

1. Publikasi informasi lewat *website* sehingga masyarakat dapat mengakses secara *online* informasi dasar dan relevan yang disediakan (*information*).
2. Menyediakan interaksi sederhana pemerintah dengan masyarakat, misalnya melalui email (*interaction*).
3. Menyediakan layanan terintegrasi yang bisa mendukung transaksi nilai pemerintah-masyarakat (*transaction and vertical integration*).
4. Mengintegrasikan layanan - layanan antar - kantor pemerintahan (*transformation and horizontal integration*). (Husain et al. 2018)

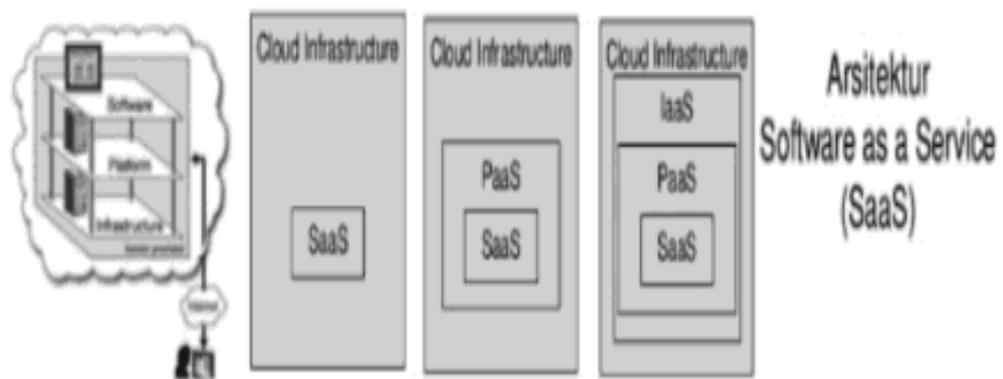
b) Pemanfaatan *Cloud Computing* untuk *startup* dan pariwisata

Peran Teknologi *Cloud Computing* dalam sektor pariwisata berbasis teknologi di segala bidang mulai diimplementasikan seperti halnya telah banyak teknologi *cloud* dalam membantu suksesnya pariwisata mulai dari promosi, hingga kedatangan wisatawan dengan berbagai *platform* aplikasi yang dapat membantu wisatawan menuju lokasi wisata yang di inginkan. (Husain et al. 2018)

2.9.3. Layanan *Cloud Computing*

Secara umum, terdapat tiga bentuk layanan *Cloud Computing*, yaitu :

a. *Software as a Service* (SaaS)

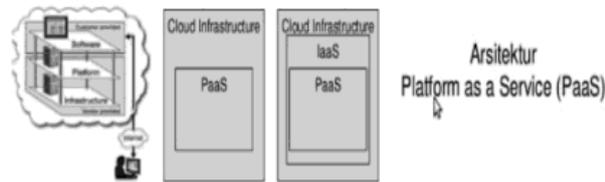


Gambar 2.20 : Arsitektur *Software as a Service* (SaaS)

(Herwanto, Purbo, and Aziz 2021)

Software as a Service adalah layanan komputasi awan yang memungkinkan pengguna untuk langsung menggunakan aplikasi yang disediakan. Pengguna layanan *SaaS* hanya memerlukan aplikasi untuk menghubungkan pengguna ke aplikasi yang mendukung internet. Contoh layanan *SaaS* populer adalah *Gmail*, *Google+*, dan *aplikasi Google*. *SaaS* memiliki banyak keunggulan. Salah satunya adalah pengguna tidak perlu membeli lisensi untuk mengakses aplikasi. Sebuah laporan yang diterbitkan oleh *Microsoft Corporation* menekankan bahwa salah satu manfaat terbesar menggunakan layanan *SaaS* adalah investasi awal yang rendah dalam perangkat lunak dan perangkat keras. (Herwanto et al. 2021)

b. *Platform as a Service (PaaS)*

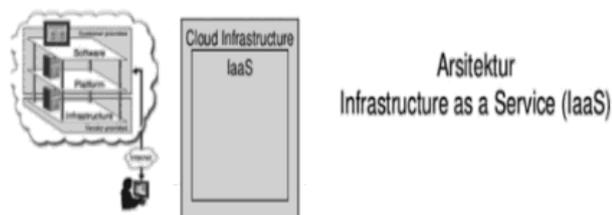


Gambar 2.21 : Arsitektur *Platform as a Service (PaaS)*

(Herwanto et al. 2021)

Platform as a Service adalah layanan yang menyediakan *platform* komputasi. Biasanya berupa desain aplikasi, proses pengujian dan penerapan, dan *hosting*. Layanan ini memungkinkan pengguna untuk membangun aplikasi dengan virtualisasi perangkat keras, redundansi data, dan ketersediaan tinggi. Setelah pengembangan selesai, pengguna dapat mengirimkan aplikasi melalui Internet. Keuntungan dari layanan ini adalah pengguna dapat lebih fokus dalam membangun aplikasi mereka tanpa harus khawatir dengan pemeliharaan *platform* komputasi. *Google AppEngine*, *Microsoft Azura*, *Salesforce.com* adalah contoh layanan. (Herwanto et al. 2021)

c. *Infrastructure as a Service (IaaS)*



Gambar 2.22 : Arsitektur *Infrastructure as a Service (IaaS)* (Herwanto et al. 2021)

Infrastruktur sebagai Layanan adalah layanan komputasi awan yang menyediakan infrastruktur dan perangkat keras kepada pengguna seperti *server*, media penyimpanan, *bandwidth*, *virtualisasi*, dan utilitas lain yang dapat dikonfigurasi. Keuntungan dari layanan IaaS adalah pengguna tidak perlu membeli komputer fisik, sehingga lebih menghemat biaya. Anda juga dapat mengubah konfigurasi mesin virtual sesuai dengan kebutuhan Anda. Misalnya, Anda dapat langsung menambahkan media penyimpanan saat hampir penuh.

Perusahaan yang menawarkan layanan LaaS adalah Amazon EC2, TelkomCloud, dan BizNet Cloud. (Herwanto et al. 2021)

2.9.4. Fungsi dan Manfaat *Cloud Computing*

Komputasi awan menggunakan *server* komputer yang sangat cepat dan cepat yang saat ini ditawarkan oleh banyak penyedia *server* media. Yang penting pihak universitas bisa meminimalisir sarana dan lokasi untuk menyiapkan *server*. Contoh aplikasi berbasis *cloud computing* adalah *Dropbox*, *Google Drive*, dan *iCloud*. Dalam bidang pendidikan, teknologi informasi berperan sangat penting dalam menyediakan fasilitas pendidikan dan pembelajaran yang lebih efisien seperti pembelajaran jarak jauh, *e-learning*, *e-library*, dan multimedia. (Mutia 2016b)

2.9.5. Kelebihan dan Kekurangan *Cloud Computing*

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

1. Manfaat bagi pelaku bisnis adalah biaya investasi dalam infrastruktur publik diminimalkan, memungkinkan perusahaan untuk lebih fokus pada fungsionalitas.
2. Bagi pengembang pelaksanaan, layanan PaaS (*Platform as a service*) memungkinkan pengembangan & implementasi pelaksanaan menggunakan cepat sebagai akibatnya menaikkan produktivitas.
3. Bagi praktisi yang bekerja di industri TI, ini berarti membuka pasar baru di industri jasa pengembangan TI.
4. Bagi perusahaan yang bergerak di bidang infrastruktur, hal ini merupakan peluang besar karena penggunaan layanan PaaS meningkat dan jumlah bandwidth internet yang digunakan meningkat.
5. Saling koneksi antar aplikasi dengan berbagai perangkat.

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

1. Tingkat layanan mengacu pada potensi kinerja layanan yang tidak konsisten oleh penyedia. Inkonsistensi antara penyedia cloud ini mencakup perlindungan data dan pemulihan data.
2. Kerahasiaan (*privacy*), yaitu karena *hosting* digunakan bersama, ada risiko data pengguna akan terlihat oleh pengguna lain.
3. kepatuhan. Hal ini mengacu pada risiko menyimpang dari tingkat kepatuhan penyedia terhadap peraturan yang berlaku pengguna.
4. Kepemilikan data mengacu pada risiko kehilangan kepemilikan data setelah disimpan di *cloud*.
5. Mobilitas data. Ini mengacu pada kemampuan untuk bertukar data antara layanan *cloud* dan mengambil data ketika pengguna keluar dari layanan *cloud computing* di beberapa titik.

2.9.6. Sistem Kerja Cloud Computing

Sistem komputasi awan dibagi menjadi 2 yaitu ujung depan dan ujung belakang. Ini saling berhubungan melalui jaringan (seperti Internet). Ujung depan adalah pengguna komputer (pengguna) atau klien dan ujung belakang adalah bagian "awan" dari sistem. Ujung depan berisi komputer klien (atau jaringan) dan aplikasi yang diperlukan sebagai akses sistem komputasi awan. Tidak semua sistem komputasi awan memiliki antarmuka pengguna yang sama. Layanan seperti program *email* berbasis *web* menggunakan *browser web* yang ada seperti *Internet Explorer* dan *Firefox*. Sistem lain memiliki aplikasi mereka sendiri yang menyediakan akses jaringan ke klien. Bagian belakang sistem berisi berbagai komputer, *server*, dan sistem penyimpanan data yang membentuk "awan" layanan komputer. Setiap aplikasi biasanya memiliki *server* sendiri. *Server* pusat mengelola sistem dan memantau lalu lintas dan permintaan klien untuk memastikan semuanya berjalan lancar. Sistem ini mengikuti sebuah aturan yang disebut protokol dan jenis perangkat lunak yang digunakan khusus disebut *middleware*. *Middleware* jaringan memungkinkan komputer untuk berkomunikasi

satu sama lain. Sebagian besar *server* tidak berjalan dengan kapasitas penuh. Ini berarti komputasinya kuat dan *outputnya* tetap tidak digunakan. Selanjutnya, kita membutuhkan sebuah metode. Teknik ini disebut *virtualisasi server*. Dengan memaksimalkan kinerja setiap *server*, *virtualisasi server* mengurangi kebutuhan untuk menjalankan mesin. (Mutia 2016b)

2.10. Firebase



Gambar 2.23 : Logo Firebase (firebase.google.com)

Firebase adalah penyedia layanan awan *back-end* sebagai layanan yang berlokasi di *San Francisco, California*. *Firebase* menciptakan sekumpulan produk atau layanan untuk pengembangan aplikasi seluler atau *web*. Didirikan pada tahun 2011 oleh *Andrew Lee* dan *James Display*, *Firebase* meluncurkan database *cloud real-time* pada tahun 2012. Perusahaan ini diakuisisi oleh *Google* pada Oktober 2014. (Sonita and Fardianitama 2018)

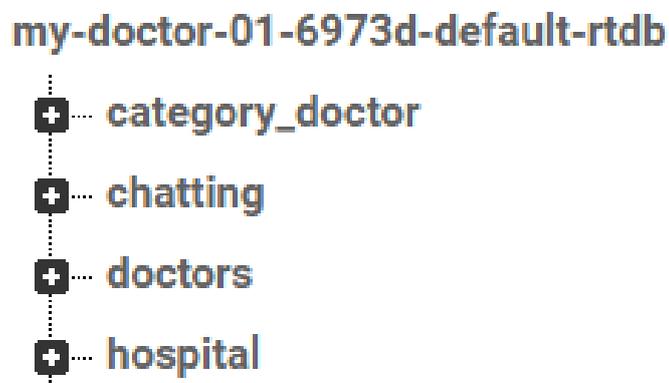
Firebase adalah penyedia layanan *database real-time* dan *back end* sebagai layanan. Aplikasi yang memberi pengembang membuat API (antarmuka pemrograman aplikasi) yang disinkronkan untuk klien yang berbeda dan disimpan di *cloud Firebase*. *Firebase* mempunyai banyak *library* yang memungkinkan Anda mengintegrasikan layanan ini dengan *Android, iOS, JavaScript, Java, Objective-C*, dan *Node.JS*. Beberapa fitur lain yang ditawarkan *Firebase* adalah:

1. *Responsive* walaupun saat *offline*. *Database Realtime* dilengkapi SDK untuk menyimpan data ke disk lokal. Sehingga saat *offline* pengguna dapat melakukan akses data.
2. *Firebase* memiliki tingkat keamanan yang mumpuni, karena diakuisi langsung oleh *Google*.

3. *Firebase* memberikan layanan gratis kepada pengembang aplikasi. (Sonita and Fardianitama 2018)

2.10.1. *Firebase Realtime Database*

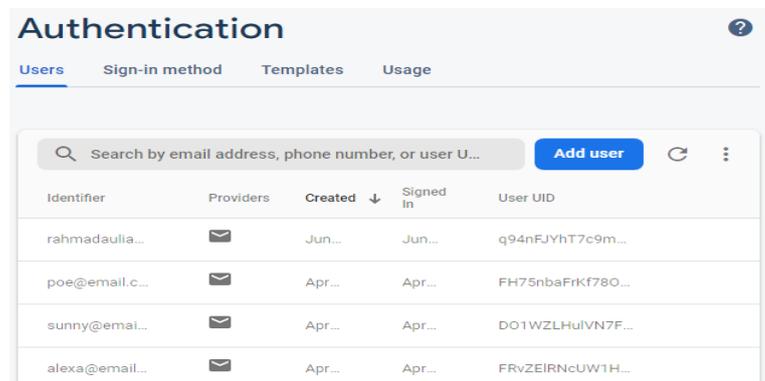
Firebase Realtime Database adalah layanan basis data *real-time* yang disediakan oleh *Firebase*. Pengguna pengembang perangkat lunak juga dapat menggunakan database ini untuk membuat cadangan data mereka menggunakan *server Firebase* dengan aturan yang ada. Di situs resminya, *database Firebase* disebut *Cloud Realtime Database* dan menyimpan data dalam format JSON (*Javascript Object Notation*). *Firebase* melakukan sinkronisasi data otomatis untuk aplikasi yang terhubung. Dengan *database real-time Firebase*, *database* dapat mengurutkan dan memfilter 15 data menggunakan kueri *NoSQL*. Basis data *NoSQL* sendiri terdiri dari empat karakteristik: nilai kunci, berbasis dokumen, berbasis kolom, dan berbasis grafik. *Firebase* sendiri dapat mengoptimalkan akses data dalam mikrodetik daripada menggunakan layanan *web*. (Sudiartha, Indrayana, and Suasnawa 2018)



Gambar 2.24 : *Firebase Realtime Database* (console.firebase.google.com)

2.10.2. *Firebase Auth*

Layanan yang menarik di *Firebase* adalah *Firebase auth*. Ini adalah layanan yang memungkinkan pengembang dengan mudah membuat otentikasi untuk berinteraksi dengan data *email*, nomor telepon, dan akun media sosial yang ada di aplikasi mereka. Layanan ini biasanya digunakan sehubungan dengan pendaftaran akun dan login sistem.



The screenshot shows the 'Authentication' section of the Firebase console. It has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. Below the tabs is a search bar with the text 'Search by email address, phone number, or user U...' and an 'Add user' button. Below the search bar is a table with the following columns: 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. The table contains four rows of user data.

Identifier	Providers	Created	Signed In	User UID
rahmadaulia...	✉	Jun...	Jun...	q94nFJYhT7c9m...
poe@email.c...	✉	Apr...	Apr...	FH75nbaFrkF78O...
sunny@emai...	✉	Apr...	Apr...	DO1WZLHulVN7F...
alexa@email...	✉	Apr...	Apr...	FRvZEIRNcUW1H...

Gambar 2.25 : Console Firebase Auth (console.firebase.google.com)

Cara kerja otentikasi *Firestore* adalah dengan mengetahui identitas Anda dan menyimpannya di *cloud*. Pada fase ini, pengguna harus mendaftar terlebih dahulu dengan *email* dan kata sandi, mengautentikasi dengan nomor ponsel, atau membuat data statis (untuk pengembangan sistem). Setelah pengguna mendaftar, pengguna dapat masuk dan keluar dari sistem. (Ilhami 2017)

Berikut adalah langkah-langkah penggunaan *Firestore Auth* dalam pengembangan aplikasi android :

1. Pertama untuk menggunakan basis data *Firestore Realtime* ini adalah dengan melakukan koneksi akun *email (Gmail)* kita dengan *Firestore*.
2. Selanjutnya memperkenalkan SHA-1 (Secure Hash Algorithm – 1) aplikasi kita kepada *Firestore*, hal ini sebagai identitas aplikasi dalam penggunaan *database*. Contoh SHA-1 yang dimaksud adalah seperti kode berikut :

SHA-1

06:BA:DD:97:7A:36:E8:52:BC:BB:45:81:15:4E:C4:BD:23:45:AA:DB

3. Setelah mendaftarkan SHA-1 aplikasi, lalu kita melakukan penambahan *source code* pada aplikasi android kita yang disediakan oleh *Firestore*. Adapun kode *dependency* yang akan ditambah yaitu sebagai berikut :

com.google.firebase:firebase-auth:18.0.0

2.11. Unified Modelling Language

Unified Modeling Language (UML) adalah bahasa berbasis grafik atau gambar untuk menampilkan, menetapkan, merancang, dan mendokumentasikan sistem pengembangan perangkat lunak berdasarkan pemrograman berorientasi objek. UML sendiri menyediakan standar untuk membuat sistem cetak biru yang mencakup konsep proses bisnis, deskripsi kelas dalam bahasa pemrograman tertentu, skema *database*, dan komponen yang diperlukan untuk perangkat lunak sistem. (Suendri 2018)

Unified Modeling Language (UML) adalah keluarga dari notasi grafis yang didukung oleh sistem perangkat lunak, terutama sistem yang dibangun menggunakan konsep pemrograman berorientasi objek (OOP), metamodel tunggal. Berguna untuk mengeksplorasi sistem yang ada. (Irawan and Herviana 2019)

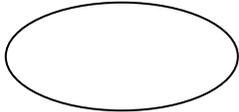
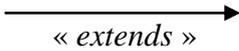
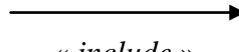
UML saat ini distandarisasi oleh *Object Management Group (OMG)*, dan semua perubahan dan revisi spesifikasi UML adalah tanggung jawab OMG. Versi UML terakhir yang dirilis oleh OMG pada bulan Desember 2017 adalah UML 2.5.1. UML memiliki dua jenis diagram utama, diagram struktural dan diagram operasional, yang berfungsi sebagai model untuk membuat UML. Model juga merupakan cara untuk memberikan gambaran tentang struktur dan fungsionalitas sistem tertentu. Diagram struktural menunjukkan struktur statis sistem, berbagai bagian dari lapisan abstraksi dan implementasi, dan bagaimana bagian-bagian ini berhubungan satu sama lain. Diagram perilaku menunjukkan perilaku dinamis objek dalam sistem. Ini dapat digambarkan sebagai serangkaian perubahan dalam sistem dari waktu ke waktu. (Akil and Kom 2018). Di bawah ini adalah diagram yang biasa digunakan oleh para pengembang atau praktisi TI saat menggunakan UML, yang merupakan bagian dari dua model utama diagram UML, yaitu sebagai berikut :

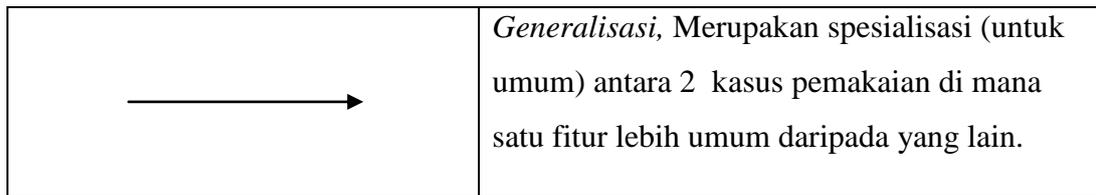
1. Use Case Diagram

Use case adalah entitas eksternal (dalam bentuk antarmuka) dari sistem yang menerima perintah dari aktor dalam bentuk peristiwa. *Use case* ini mengacu pada implementasi urutan untuk menyampaikan pesan antara objek terkait. (Akil and Kom 2018). *Use case* juga dapat diartikan sebagai grafik interaksi antara beberapa aktor yang mewakili sistem dan aktor. *Use case* adalah aktor atau pengguna yang melakukan operasi pada sistem dan mewakili fungsi proses yang terlibat dalam transformasi sistem. (Samsudin 2019)

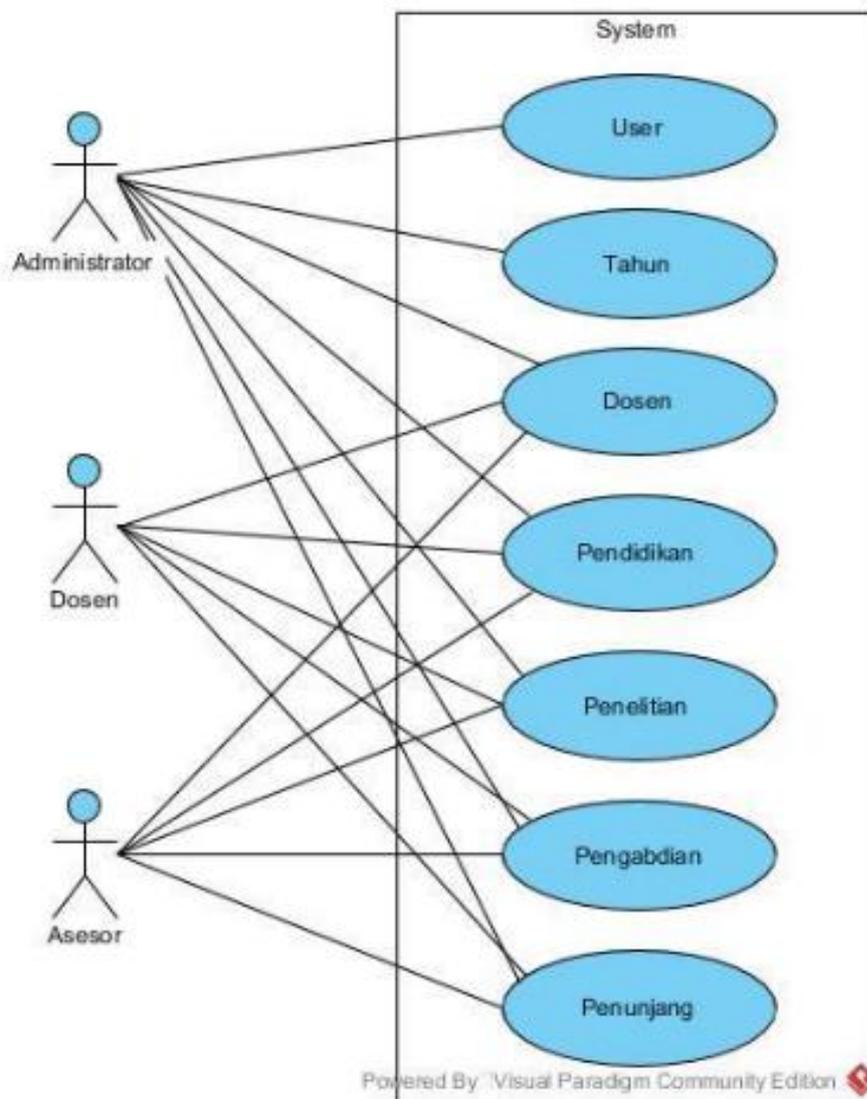
Use Case diagram adalah bagian dari behavior diagram. Berikut sebuah tabel simbol-simbol yang digunakan untuk pembuatan diagram *Use Case* :

Tabel 2. 2 : Simbol Dalam Diagram Use case (Hendini 2016)

Simbol	Deskripsi
	<i>Aktor</i> , Seseorang atau proses yang berinteraksi dengan sistem dan berada di luar lingkungan sistem informasi.
	<i>Use case</i> , Sebuah fitur yang disediakan oleh sistem sebagai entitas yang bertukar pesan.
	<i>Asosiasi</i> , Komunikasi antara <i>use case</i> aktor yang berpartisipasi dan <i>use case</i>
	<i>Extended</i> , Gunakan hubungan kasus dengan kasus penggunaan mandiri tambahan
	<i>Include</i> , Gunakan hubungan kasus saat Anda perlu menjalankan fungsi dalam kasus penggunaan lain, dan sebagai kondisi untuk menjalankan kasus penggunaan.



Saat Anda membuat *use case*, Anda perlu mengidentifikasi kebutuhan Anda. Pemeran yang menjadi aktor dalam sistem. Berikut ini adalah contoh *use case* diagram setelah dilakukan *need assessment*.

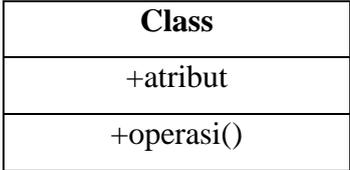
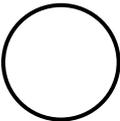


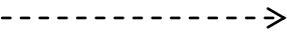
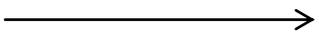
Gambar 2.26 : Contoh Diagram Usecase (Suendri 2018)

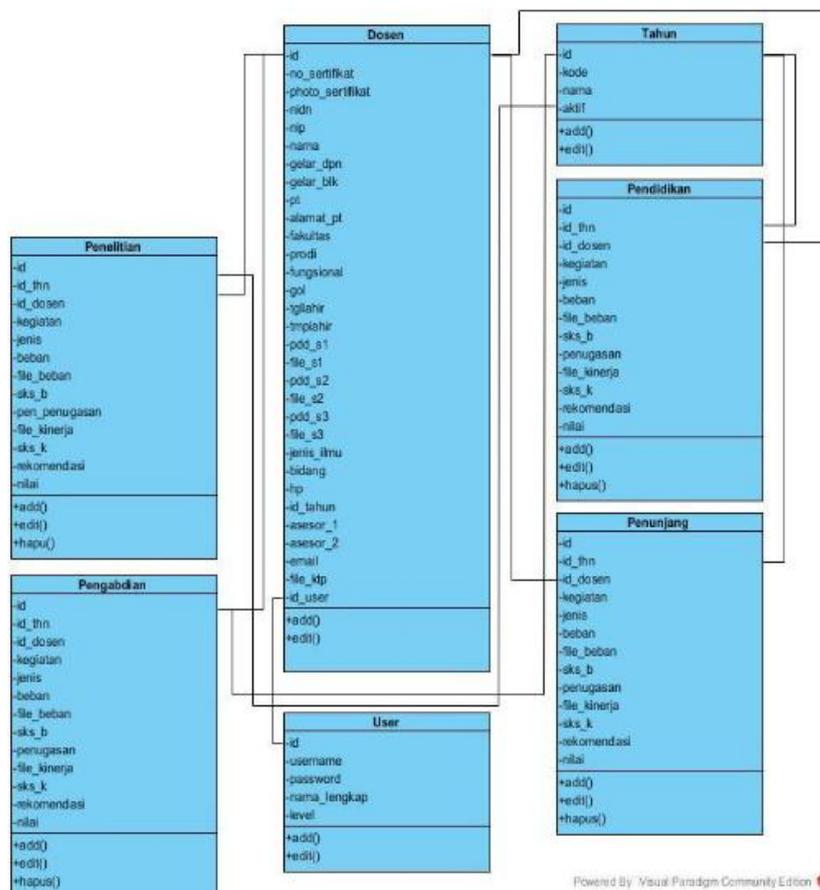
2. *Class Diagram*

Diagram kelas yang menggambarkan struktur statis dan hubungannya. Diagram kelas adalah diagram yang menunjukkan kumpulan suatu kelas, gambaran, dan hubungannya, dan diagram kelas adalah bagian dari diagram struktur model. (Akil and Kom 2018). Diagram kelas yang menggambarkan struktur statis dan hubungannya. Diagram kelas adalah diagram yang menunjukkan sekumpulan kelas, tampilan, dan hubungannya, dan diagram kelas adalah bagian dari diagram struktur model. (Samsudin 2019). Berikut yaitu simbol yang digunakan dalam pembuatan class diagram :

Tabel 2. 3 : Simbol Pada Digram Class (Hendini 2016)

Simbol	Deskripsi
	<p><i>Class</i>, Sebuah struktur sistem yang menunjukkan atribut dan operasi. Ini sangat berguna dalam tahap implementasi konstruksi sistem.</p>
	<p><i>Interface</i>, Simbol yang mendefinisikan konsep OOP (<i>Object Oriented Programming</i>) dalam suatu sistem.</p>
	<p><i>Aggregation</i>, Sebuah simbol yang menghubungkan kelas-kelas yang memiliki arti untuk semua bagian. Oleh karena itu, simbol ini digunakan ketika suatu kelas merupakan bagian dari kelas lain.</p>

	<p><i>Dependency</i>, Ikon ini digunakan untuk menampilkan operasi untuk kelas yang menggunakan kelas lain.</p>
	<p><i>Generalisasi</i>, Digunakan untuk mengasosiasikan kelas dengan arti umum tertentu.</p>



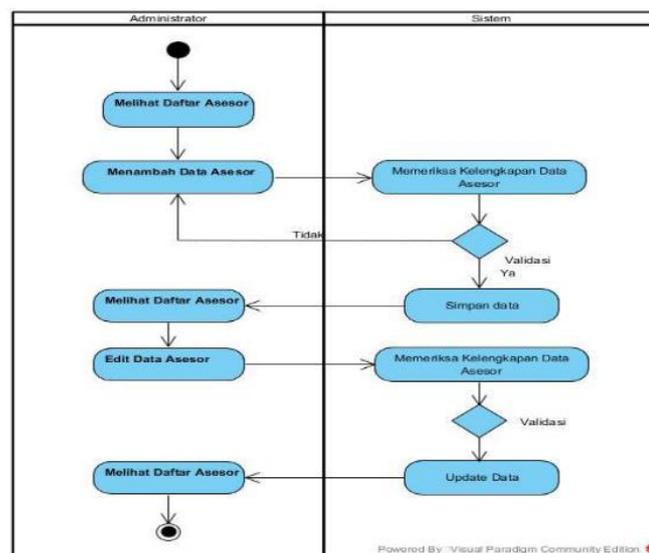
Gambar 2.27 : Contoh Class Diagram (Suendri 2018)

3. Activity Diagram

Diagram aktivitas, seperti diagram alur, fokus pada penggambaran proses logis dari sistem komputer. Ini juga dikenal sebagai diagram yang menggambarkan alur kerja dan proses bisnis. Keuntungan dari gambar ini dibandingkan *flowchart* adalah mendukung proses paralel. (Akil and Kom 2018). Adapun Simbol yang digunakan dalam diagram aktivitas adalah:

Tabel 2. 4 : Simbol Pada Activity Diagram (Hendini 2016)

Simbol	Deskripsi
	<i>Start</i> , Menunjukkan awal dari proses
	<i>Stop</i> , Menyatakan akhir dari proses
	<i>Decision</i> , Digunakan untuk mewakili keadaan suatu proses
	<i>Action</i> , Merupakan tindakan yang dilakukan dalam arsitektur sistem.

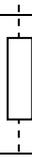
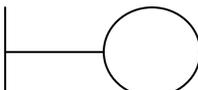
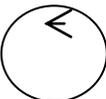
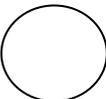


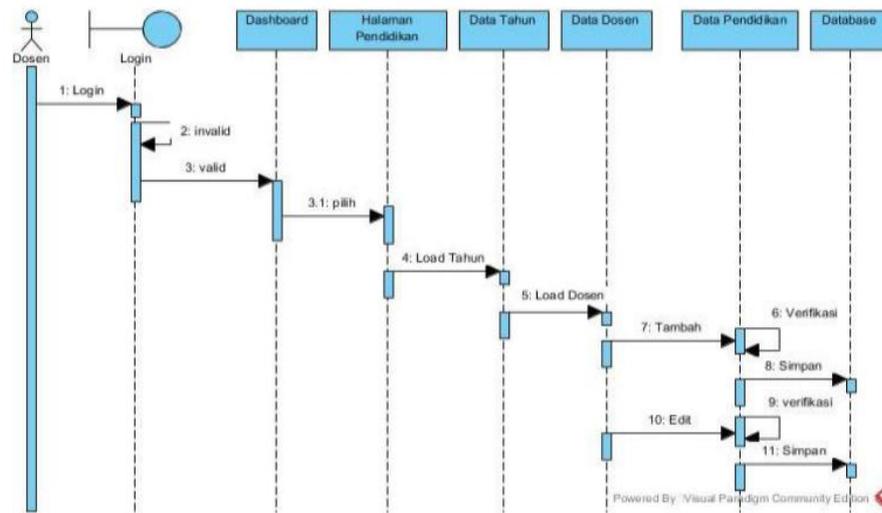
Gambar 2.28 : Contoh Activity Diagram (Samsudin 2019)

4. *Sequence Diagram*

Diagram urutan menunjukkan perilaku skenario tunggal. Gambar ini menunjukkan objek yang ditampilkan oleh proses dan urutan pesan yang dikirimkan di antara mereka. Diagram urutan biasanya ditampilkan untuk menjelaskan kasus penggunaan eksekusi. (Akil and Kom 2018). Diagram urutan adalah deskripsi langkah demi langkah yang berisi garis waktu (urutan) perubahan logis yang perlu dibuat untuk menghasilkan kecocokan untuk diagram kasus penggunaan. (Irawan and Herviana 2019). Adapun simbol-simbol yang digunakan dalam membuat sequence diagram adalah sebagai berikut :

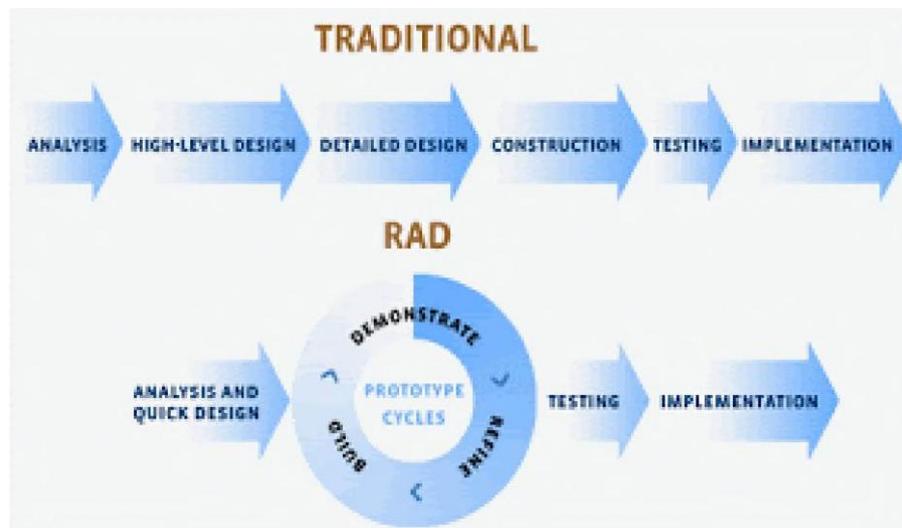
Tabel 2. 5 : Simbol Pada Sequence Diagram (Muhamad Alda 2021)

Simbol	Deskripsi
	<i>Aktor</i> , Digunakan untuk mendeskripsikan pengguna
	<i>Lifeline</i> . Objek entitas, antarmuka yang berinteraksi.
	<i>Boundary</i> , Digunakan untuk menggambarkan bentuk
	<i>Message</i> , Sebuah spesifikasi komunikasi antara objek yang berisi informasi tentang apa yang terjadi.
	<i>Control Class</i> , Digunakan untuk menghubungkan perbatasan ke tabel.
	<i>Entity Class</i> , Digunakan untuk menggambarkan hubungan antara aktivitas yang Anda lakukan.



Gambar 2.29 : Contoh Sequence Diagram (Suendri 2018)

2.12. Rapid Application Development (RAD)



Gambar 2.30 : Langkah Penerapan Metode RAD (Habibi and Aprilian 2020)

2.12.1. Defenisi Metode RAD

Metode pengembangan sistem adalah aktivitas otomatis, metode, praktik terbaik, dan alat yang digunakan untuk mengembangkan dan terus meningkatkan sistem informasi dan perangkat lunak. Pengembangan sistem informasi adalah persiapan suatu sistem untuk menggantikan keseluruhan sistem yang lama atau memperbaiki sistem yang sudah ada. (Habibi and Aprilian 2020)

2.12.2. Sejarah Metode RAD

Rapid Application Development (RAD) suatu istilah yang digunakan untuk menunjukkan proses pengembangan perangkat lunak yang pertama kali dikembangkan dan berhasil digunakan pada pertengahan tahun 1970-an oleh Pusat Pengembangan Sistem *Verizon New York* yang dipimpin oleh *Dan Gielan*. Setelah implementasi proses ini dengan sangat sukses, *Gielan* memberikan presentasi ekstensif di berbagai forum tentang metodologi praktis dan manfaat dari proses ini.

RAD mencakup pengembangan berulang dan pembuatan prototipe. Pada tahun 1990, James Martin mencatat interpretasi metodologis pada buku RAD. Baru-baru ini, istilah dan akronim telah digunakan dalam arti yang lebih luas dan lebih umum, termasuk berbagai cara yang ditujukan untuk mempercepat pengembangan aplikasi, seperti: Penggunaan berbagai jenis kerangka kerja perangkat lunak. B. Kerangka aplikasi web.

Pengembangan aplikasi yang cepat membahas proses berikut yang dikembangkan pada 1970-an dan 1980-an: B. Metode analisis dan desain sistem terstruktur dan model air terjun lainnya. Masalah dengan metodologi sebelumnya adalah aplikasi membutuhkan waktu terlalu lama untuk dibangun, mengubah persyaratan sebelum sistem selesai, membuat sistem tidak memadai atau tidak dapat digunakan. Isu lainnya adalah asumsi bahwa semua kebutuhan penting diidentifikasi hanya dalam tahap analisis kebutuhan sistematis. Bukti bahwa ini jarang terjadi, bahkan dalam proyek yang melibatkan profesional yang sangat berpengalaman di semua tingkatan.

James Martin menggunakan ide Brian Gallagher, Alex Baltin, Barry Bame, dan Scott Schultz untuk mengembangkan pendekatan pengembangan aplikasi cepat di IBM pada 1980-an, berjudul *Rapid Application Development* pada 1991. Buku itu diterbitkan dan diresmikan. (Habibi and Aprilian 2020)

2.12.3. Tahapan-tahapan dalam RAD

RAD digunakan pada aplikasi sistem konstruksi, Ada tiga fase dalam RAD yaitu :

1. *Requirement Planning*, Pada tahap ini, Anda akan tahu apa kebutuhan sistem Anda. Yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi dalam menentukan tujuan, batasan sistem, keterbatasan, dan alternatif pemecahan masalah. Analisis juga digunakan untuk mengetahui perilaku sistem dan untuk mengetahui aktivitas apa yang sedang dilakukan pada sistem.
2. *Design Workshop*, Artinya, identifikasi solusi alternatif dan pilih salah satu yang paling sesuai untuk Anda. Selanjutnya membuat desain proses bisnis dan desain pemrograman untuk data yang diperoleh dan dimodelkan dalam arsitektur sistem informasi. Alat-alat yang digunakan dalam pemodelan sistem biasanya menggunakan *Unified Modeling Language (UML)*.
3. *Implentation*, Setelah lokakarya desain dilakukan, sistem diimplementasikan (dikodekan) dengan cara yang dapat dipahami mesin dan diwujudkan dalam bentuk program atau unit program. Fase implementasi sistem adalah fase dimana sistem menjadi operasional. (Habibi and Aprilian 2020)

2.13. Penelitian Terkait

Penelitian-penelitian sebelumnya merupakan penelitian-penelitian yang telah dilakukan sebelumnya dan selanjutnya digunakan oleh penulis sebagai acuan untuk melakukan penelitian. Tabel berikut menunjukkan penelitian-penelitian sebelumnya yang berkaitan dengan judul penelitian penulis.

Tabel 2. 6 : Penelitian Terdahulu

NO	Judul	Tahun	Hasil Penelitian	Perbedaan
1	Rancang Bangun Aplikasi Jasa <i>Service</i> Peralatan	2020	Dalam penelitian ini telah dibuat aplikasi jasa <i>service</i> peralatan elektronik	Perbedaan sistem yang akan dibuat dengan sistem sebelumnya adalah perangkat lunak untuk

	Elektronik Berbasis <i>Android</i> , (L. Iqbal, Y.Septiana, R. Setiawan)		berbasis <i>Android</i> yang dapat menangani masalah yang terdapat pada penelitian yang dilakukan	membangun aplikasi berbeda, <i>Framework React Native</i> mampu membangun aplikasi <i>android</i> dan IOS cukup dengan bahasa pemograman <i>Javascript</i> .
2	Rancang Bangun Aplikasi Pemesanan Jasa <i>Service Komputer Berbasis Web</i> , (M. Dharma, N. Ananda, M. Alzaki)	2018	Dapat melakukan <i>booking service</i> , pencatatan laporan <i>service</i> , riwayat <i>service</i> dan juga update status <i>service</i>	Perbedaan sistem yang akan dibuat dengan sistem sebelumnya yaitu hasil sistem yang dirancang masih berbasis <i>website</i> , sedangkan sistem yang akan dibuat berbasis <i>android</i> menggunakan <i>Framework React Native</i> .
3	Aplikasi Jasa <i>Service AC (Air Conditioner)</i> Pada CV Teknik Makmur, (Adhitya Ilham Ramdhani, Zaenal Mutaqin Subekti, Ishak Husein, Esmi Imrohatun)	2021	CV Teknik Makmur sebelumnya belum terkomputerisasi dan masih menggunakan manual. Dengan adanya aplikasi jasa <i>service AC</i> diharapkan menjadi lebih tertib administrasi, mudah dalam pencarian data.	Perbedaan sistem yang akan dibangun dengan sistem sebelumnya yaitu pada fungsi penggunaan aplikasi, aplikasi yang akan dibangun nantinya dapat digunakan oleh para penyedia jasa <i>service</i> dan masyarakat umum.
4	Aplikasi Online Berbasis <i>Android</i> “SI TekO” (Sistem Informasi Teknisi Online) Sebagai Solusi Mempermudah Masyarakat Dalam Mendapatkan	2021	Aplikasi "SiTekO" (Sistem Informasi Teknisi Online) ini memberikan layanan teknisi online kepada masyarakat dan memudahkan untuk mendapatkan layanan tanpa keluar rumah.	Perbedaan sistem yang akan dibangun dengan sistem sebelumnya yaitu kelengkapan fitur, aplikasi yang akan dibangun memiliki fitur <i>chatting</i> .

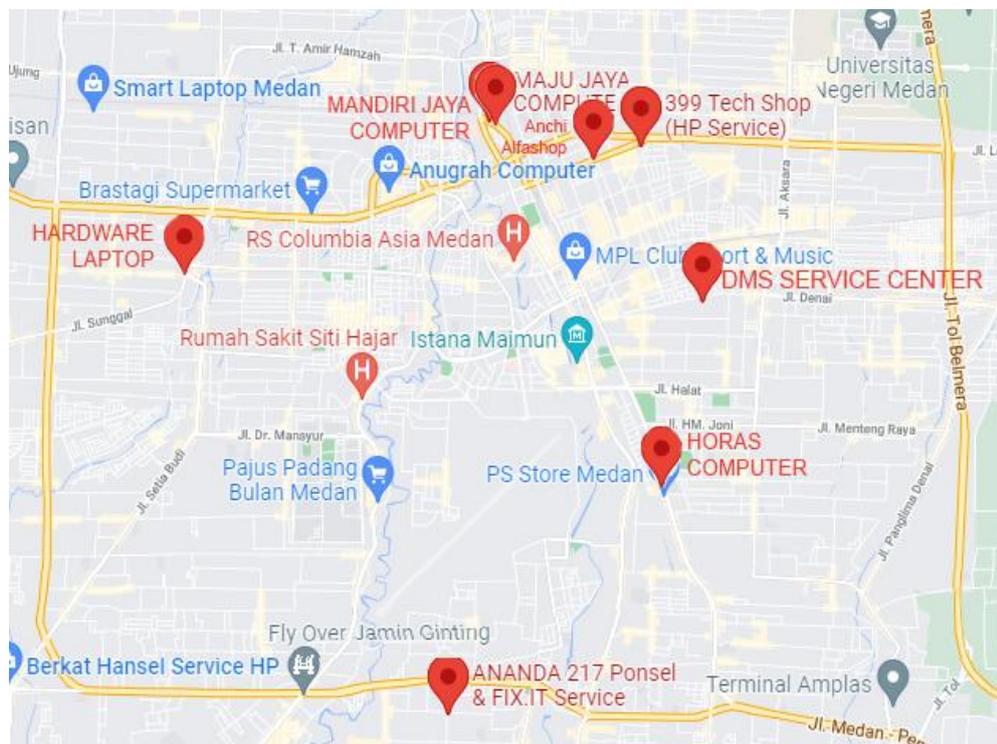
	Jasa <i>Service</i> , (Akbar, RN asfi, Indiarto, F., Aristiantoro, A., & Utomo, Y)			
5	Rancang Bangun Aplikasi Pemesanan Jasa Perbaikan Komputer, Laptop, dan <i>Smartphone</i> Berbasis <i>Android</i> , (Yoko Andramawan, Khairul Ummi, Alfa Saleh)	2018	Membuat aplikasi pemesanan jasa reparasi komputer, laptop, dan smartphone berbasis android yang sudah terkoneksi dengan server melalui koneksi internet.	Perbedaan sistem yang akan dibangun dengan sistem sebelumnya yaitu pada <i>database</i> , dimana aplikasi yang akan dibangun menggunakan <i>Firebase</i> dengan memiliki kelebihan transfer data yang lebih cepat disebabkan konsep <i>database Firebase</i> berbentuk JSON.

BAB III METODE PENELITIAN

3.1. Tempat dan Waktu Penelitian

Sub-bab ini menjelaskan lokasi survei dan waktu spesifik survei. Di bawah ini adalah presentasi dari sub-bab ini.

3.1.1. Tempat Penelitian



Gambar 3. 1 : Peta lokasi penelitian pada Kota Medan

Dalam penelitian ini, penulis mengambil tempat penelitian kepada beberapa pengusaha penyedia jasa perbaikan elektronik yang meliputi wilayah Kecamatan Medan Barat, Kecamatan Medan Sunggal, Kecamatan Medan Kota, Kecamatan Medan Johor, Kecamatan Medan Perjuangan, Kecamatan Medan Timur, dan Kecamatan Medan Area. Beberapa toko yang dimaksud adalah sebagai berikut :

1. Kecamatan Medan Barat
 - a. *Maju Jaya Computer (Service PC)*
 - b. *Mandiri Jaya Computer (Service PC)*

2. Kecamatan Medan Sunggal
 - a. Hardware Laptop (*Service PC*)
3. Kecamatan Medan Kota
 - a. Horas Computer (*Service PC*)
4. Kecamatan Medan Johor
 - a. Ananda 217 Ponsel (*Service Ponsel*)
5. Kecamatan Medan Perjuangan
 - a. 399 Tech Shop (*Service Ponsel*)
6. Kecamatan Medan Timur
 - a. Anchi Alfashop (*Service Ponsel*)
7. Kecamatan Medan Area
 - a. DMS Service Center (*Service Ponsel*)

3.1.2. Waktu Penelitian

Sub-bab ini menjelaskan lokasi dan waktu spesifik survei. Di bawah ini adalah presentasi dari sub-bab ini, penelitian ini dimulai dari Juli 2021 – November 2021 yang dirancang sebagai berikut :

Tabel 3. 1 : Waktu dan Jadwal Pelaksanaan Penelitian

Tahapan	2021																								2022							
	Jul				Ags				Sep				Okt				Nov				Des				Jan				Feb			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																																
Pengumpulan Data																																
Pengajuan Proposal																																
Seminar Proposal																																

6. Perancangan Sistem

Pada fase ini penulis memulai perancangan dengan membuat alur sistem melalui *Entity-Relationship* (UML), perancangan basis data, dan perancangan antarmuka.

7. Pembuatan *Coding*

Pada fase ini penulis menuliskan kode program ke dalam sebuah aplikasi.

8. *Testing*

Pada fase ini, aplikasi yang sudah jadi diuji. Tujuan dari pengujian adalah untuk memastikan bahwa aplikasi berjalan sesuai dengan yang diharapkan pada perancangan sistem. Pada tahap pengujian ini, penulis menggunakan pengujian *black-box*. Ini adalah cara untuk menguji sistem Anda dengan memperhatikan arsitektur dasar sistem Anda, tanpa memperhatikan kode sumbernya. (Cholifah, Yulianingsih, and Sagita 2018)

3.2. Kebutuhan Sistem

3.2.1. Perangkat Keras

Berikut merupakan Spesifikasi *hardware* penulis gunakan untuk membangun aplikasi :

1. *Processor* Intel ® *Core*TM i5-450M
2. AMD Radeon 5450
3. RAM 4GB DDR3 *Memory*
4. SSD 120GB
5. HDD 500GB

3.2.2. Perangkat Lunak

Berikut merupakan Spesifikasi *software* penulis gunakan untuk membangun aplikasi :

1. Sistem Operasi *Windows 7*
2. *Visual Studio Code 1.6*
3. *Node Js 14.17.6 LTS*

4. *Android Studio IDE*
5. *Java Standart Edition 8.1*
6. *Figma*
7. *Microsoft Office 2007*
8. *Diagram.Net*

3.3. Cara Kerja

Metode pengerjaan penelitian ini menggunakan metode R&D, R&D, atau metode pengumpulan data yang dilakukan dalam bahasa Inggris. R&D adalah metode penelitian yang digunakan untuk membuat produk tertentu dan menguji keefektifan produk tersebut. Pengertian tersebut adalah “R&D merupakan strategi ampuh untuk meningkatkan praktik (Hanafi, 2017). Di dalamnya juga dijelaskan bagaimana mengembangkan sistem menggunakan *Rapid Application Development* (RAD).

3.3.1. Metode Pengumpulan Data

Metode penelitian dalam penelitian ini menggunakan metode penelitian R&D (*Research and Development*). Berikut adalah Langkah-langkah metode penelitian R&D :

1. Penelitian dan Pengumpulan Informasi, dalam hal ini penulis melakukan observasi, wawancara dan studi pustaka untuk pengumpulan data. Observasi dilakukan pada tujuh kecamatan di Kota Medan seperti dengan cara melihat langsung kelapangan seperti tempat-tempat penyedia jasa *service* elektronik pada tujuh kecamatan tersebut, pengamatan yang dilihat antara lain adalah data biaya jasa *service*, data pelanggan toko *service*, lokasi toko *service*, serta pelayanan yang didapatkan, dan juga bagaimana sistem yang ada berjalan. Wawancara adalah sebuah kegiatan dialog yang dilakukan oleh dua individu, dialog tersebut bersifat tanya jawab. Dalam hal ini penulis selaku pewawancara, dan yang menjadi narasumber ialah para pengelola jasa *service* elektronik serta para customer yang biasa memakai jasa *service* tersebut. Pada toko jasa *service* elektronik Laptop/PC, penulis melakukan wawancara kepada

seorang narasumber bernama Ahwat, beliau merupakan pemilik toko “Maju Jaya *Computer*” di Kecamatan Medan Barat. Selanjutnya penulis melakukan wawancara kepada seorang narasumber bernama Asrullah Daulay yang merupakan teknisi Laptop/PC di toko “Mandiri *Jaya Computer*” di Kecamatan Medan Barat. Selanjutnya, penulis melakukan wawancara kepada seorang narasumber bernama Arjoko Siahaan, S.Kom yang merupakan salah seorang pengelola toko “*Hardware Laptop*” di kecamatan Medan Sunggal. Selanjutnya, penulis melakukan wawancara kepada seorang narasumber bernama Zepi Lingga yang merupakan salah seorang pemilik toko “*Horas Computer*” di kecamatan Medan Kota. Pada toko jasa *service* elektronik Ponsel, penulis melakukan wawancara kepada seorang narasumber bernama Ananda yang merupakan salah seorang pemilik toko “Ananda 217 Ponsel” di kecamatan Medan Johor. Selanjutnya, penulis melakukan wawancara kepada seorang narasumber bernama Raja Mulia yang merupakan salah seorang teknisi toko “399 *Tech Shop*” di kecamatan Medan Perjuangan. Selanjutnya, penulis melakukan wawancara kepada seorang narasumber bernama Ardi Saputra yang merupakan salah seorang teknisi toko “*Anchi Alfashop*” di kecamatan Medan Johor. Selanjutnya, penulis melakukan wawancara kepada seorang narasumber bernama Muhammad Rajab yang merupakan salah seorang teknisi toko “DMS *Service Center*” di kecamatan Medan Area. Dan terakhir penulis melakukan wawancara dengan narasumber bernama Aryanto selaku pelanggan di toko “DMS *Service Center*”, kecamatan Medan Area. Secara umum isi wawancara yang diajukan oleh penulis hampir sama yaitu untuk mendapatkan informasi teknis bagaimana sistem berjalan saat ini, informasi harga yang ditawarkan untuk perbaikan elektronik, keuntungan apa saja yang diberikan, dan jadwal operasional. Studi Pustaka dilaksanakan dengan melihat banyak kajian/penelitian terdahulu, seperti jurnal, skripsi, dan lainnya. Serta juga dengan melihat buku-buku terkait permasalahan yang ingin diselesaikan dengan teknologi yang ingin diangkat. Adapun penelitian terdahulu yang dimaksud seperti Jurnal L. Iqbal, Y.Septiana, R. Setiawan. “Rancang Bangun Aplikasi Jasa *Service* Peralatan Elektronik Berbasis *Android*.” Jurnal

Algoritma Volume 17 No.1, 2020. Jurnal M. Dharma, N. Ananda, M. Alzaki. “Rancang BAngun Aplikasi Pemesanan Jasa *Service* Komputer Berbasis Web”. Jurnal Teknik Informatika Volume 10, No. 3, 2018.

2. Perencanaan, pada tahap ini dibuat rencana desain pengembangan produk. Aspek-aspek penting dalam rencana tersebut meliputi produk tentang apa, tujuan dan manfaatnya apa, siapa pengguna produknya. Dalam hal ini penulis membuat produk Aplikasi GoSE (*Go Service Electronic*) Berbasis *Android* yang dapat digunakan oleh masyarakat yang tujuannya agar mempermudah masyarakat untuk melakukan perbaikan perangkat elektronik, mengetahui rentang harga *service* perangkat elektronik, mudah dalam hal mengetahui dimana saja lokasi penyedia jasa *service* serta membantu para jasa *service* elektronik dapat memasarkan jasanya lebih baik lagi sehingga mampu memperbesar peluang bisnis, sistem ini nantinya dapat digunakan oleh masyarakat dan para jasa *service* elektronik.
3. Pengembangan Sistem, tahap ini dilakukan pembuatan desain produk yang akan dibuat.
4. Uji Coba Awal, uji coba awal dilakukan dengan melakukan pengujian awal terhadap desain produk.
5. Merivisi hasil uji coba, dalam hal ini revisi dilakukan oleh para penyedia jasa *service* elektronik pada tujuh kecamatan pada kota Medan seperti yang sudah dipaparkan sebelumnya. Apakah desain produk telah sesuai atau tidak.
6. Uji kelayakan, tahap ini berkaitan dengan pengujian efektifitas pemakaian produk. Dalam tahap ini sistem diuji apakah sudah layak untuk digunakan atau masih ada kekurangan dalam sistem.
7. Revisi produk akhir, revisi didasarkan atas masukan dari uji kelayakan. Apabila kelayakan produk sudah tepat tidak perlu dilakukan revisi produk akhir lagi.
8. Implementasi, mempublikasi hasil dari produk yang telah dibuat dan dapat digunakan oleh pengguna.

3.3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD). Fase RAD terdiri dari tiga fase di mana fase-fase ini terstruktur dan saling bergantung. Fase-fasenya adalah sebagai berikut:

1. *Requirements Planning* (Perencanaan Persyaratan)

Pada tahap ini, pengguna dan analitik bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem, dan tahap ini lebih fokus pada pemecahan masalah bisnis. Pada fase ini, penulis menjalankan rencana kebutuhan sebagai berikut:

- a. Pengumpulan data dan syarat-syarat informasi yang akan digunakan untuk tahap berikutnya, data tersebut ialah berupa lokasi-lokasi toko *service* elektronik di tempat kecamatan yang ada pada Kota Medan. Pengumpulan data ini dilakukan dengan observasi ketempat dan wawancara. Adapun data yang didapatkan penulis disini adalah data biaya jasa *service*, jumlah pelanggan dalam satu hari, data pelayanan terhadap pelanggan dan lain sebagainya. Data tersebut didapatkan dari satu tempat toko jasa *service* yang mewakili masing-masing toko jasa *service* elektronik.
- b. Identifikasi sistem dilakukan dalam tahap ini dilakukan untuk mengembangkan sistem yang sudah ada. Identifikasi sistem dalam penelitian ini ialah identifikasi sistem berjalan/lama dan identifikasi sistem usulan. Adapun sistem lama/berjalan saat ini menggunakan cara manual dimana pelanggan datang kelokasi jasa *service* untuk melakukan perbaikan peralatan elektronik yang mana hal ini kadang memiliki beberapa kendala, seperti sulitnya membawa peralatan elektronik dikarenakan terlalu besar maupun beban terlalu berat seperti CPU maupun diwajibkan *service* ditempat, toko jasa *service* elektronik terkadang tutup, sedang tidak menerima *service* sementara dikarenakan terlalu banyak peralatan yang ingin di *service* ataupun rentang biaya *service* yang tidak diketahui saat ingin memperbaiki peralatan elektronik, serta sedikitnya informasi terhadap penyedia layanan jasa *service* elektronik terutama di Kota Medan.

2. Design Workshop

Fase ini adalah fase desain dan penyempurnaan, yang menggunakan kelompok pendukung keputusan sistem untuk membantu pengguna menyetujui sistem yang mereka bangun. Pengguna biasanya melihat representasi visual dari desain dan pengoperasian sistem. Dalam penelitian ini, penulis membagi tahap desain menjadi beberapa bagian sebagai berikut:

a. Desain Proses

Pada fase desain proses ini, penulis mengidentifikasi aktor yang terlibat dalam sistem layanan elektronik dari mana data dari fase sebelumnya diambil. Berikut cara menggunakan bahasa model terpadu pada diagram model yang digunakan:

1) Membuat Use Case

Pada gambar ini, penulis menggambarkan hubungan sistem-aktor yang diusulkan dan apa yang dapat dilakukan aktor sebagai pengguna dalam aliran sistem. Dalam use case ini terdapat 2 aktor yang menjadi end user yaitu pengelola toko jasa *service* dan pelanggan. Pengelola toko jasa *service* dapat melakukan *registrasi* ke sistem, melakukan login, melakukan pendaftaran toko jasa *service*, mengisi data profil toko, melakukan penginputan biaya jasa *service* berdasarkan kerusakan. Sedangkan pada pelanggan, dapat melakukan registrasi ke sistem, melakukan login, memilih kategori jasa *service* yang diinginkan, memilih toko jasa *service* yang diinginkan, melakukan *booking* jasa *service* serta melakukan diskusi dengan admin dari toko jasa *service*.

2) Membuat *Class Diagram*

Setelah mengetahui aliran sistem selanjutnya dilakukan pembuatan *Class Diagram* untuk mengetahui *class-class* yang nantinya akan dipakai dalam implementasi kode secara *Object Oriented Programming*. Adapun *class diagram* pada sistem ini terdapat 5 *class* yaitu *class* pelanggan, *class sparepart*, *class* pengelola/penyedia jasa *service*, *class booking*, *class* jasa biaya *service*.

3) Membuat Activity Diagram

Dalam diagram ini berisikan aliran sistem usulan atau bisa dikatakan sebagai proses bisnis dalam sistem. Terdapat 2 proses bisnis (2 Activity diagram) yang

terjadi dalam sistem ini yaitu pada sisi pelanggan dan juga sisi pengelola jasa *service*. Aktifitas tersebut adalah proses bisnis *booking* jasa *service*, proses bisnis pendaftaran usaha/ jasa *service*.

4) Membuat Sequence Diagram

Disini penulis mendeskripsikan interaksi objek-objek yang tersusun dalam deret waktu dan interaksinya dengan sistem. Ada dua diagram urutan, diagram urutan aktor pelanggan/pelanggan dan diagram urutan aktor manajemen layanan.

b. Desain *Database*

1) Memenentukan potensial objek

Penulis membuat daftar potensial objek dengan cara menemukan objek yang penting berlandaskan use case. Dari diagram tersebut terdapat 5 object yaitu, data pelanggan, data toko jasa *service*, data *booking* jasa *service*, data *chatting*, data biaya jasa *service*.

2) Membuat rancangan Database

Penulis membuat rancangan database dengan membuat dari nama database yang diberi nama *db_gose*. Dalam database tersebut terdapat beberapa tabel antara lain ialah *tb_pelanggan*, *tb_toko*, *tb_booking*, *tb_biaya_jasa*, *tb_chatting* *tb_transaksi*.

c. Desain *Interface*

1) Rancangan Struktur Menu

Penulis mulai merancang struktur menu untuk dijadikan landasan pembuatan Interface. Adapun struktur menu pada sisi pengelola toko adalah, register, login, menu mengelola biaya jasa *service*, menu data *booking service*, menu *chatting*, menu riwayat Transaksi. Sedangkan pada sisi end user pelanggan adalah register, login, menu *chatting*, menu lokasi jasa *service*, menu kategori jasa *service*, menu *booking*, menu riwayat transaksi, menu informasi.

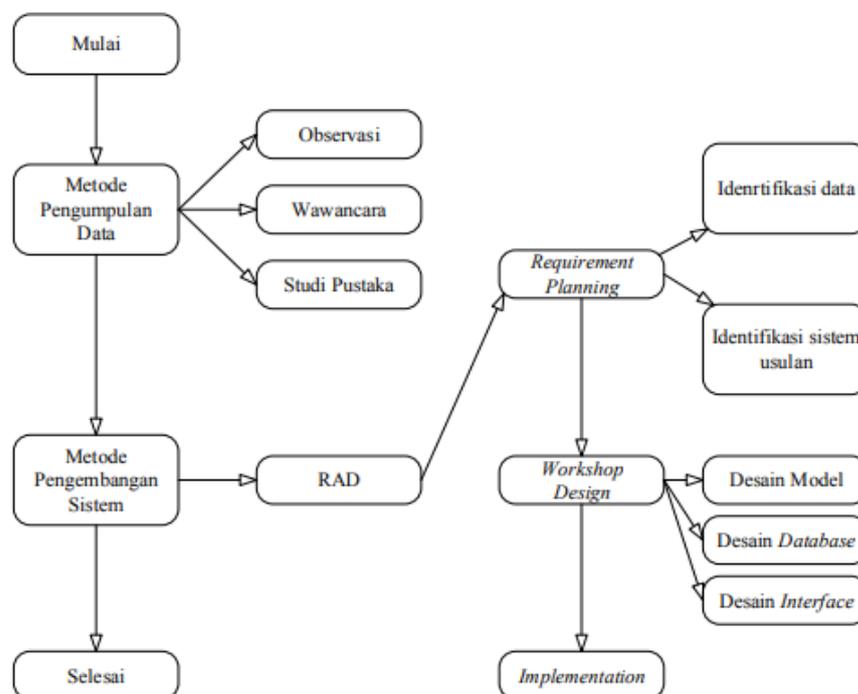
2) Rancangan Interface

Penulis merancang Interface sistem yang akan dibuat untuk menggambarkan tampilan sistem. Dalam hal ini penulis menggunakan Microsoft Visio 2019 untuk melakukan desain antarmuka sistem yang dapat dilihat pada bab berikutnya.

3. Implementation

Sistem yang baru dibangun, sistem baru atau parsial diuji dengan memperkenalkan kepada pengguna dalam hal ini menggunakan metode blackbox sebagai tahap testing sistem, dan sistem yang lama tidak perlu dijalankan secara beriringan. Dan pada tahap ini jugalah melakukan pengkodean sistem berdasarkan tahapan pada *Requirements Planning* dan *Workshop Design* serta implementasi dari *React Native* dan *Firebase*.

3.3.3. Kerangka Berfikir



Gambar 3. 2 : Kerangka Berfikir

Kesimpulan penelitian ini memiliki beberapa tahapan dalam pengerjaannya, langkah pertama dimulai dengan metode pengumpulan data. Dengan cara tersebut, penulis penelitian ini memperoleh data dari tahap observasi di lokasi penelitian, dalam hal ini beberapa penyedia jasa toko jasa di tujuh kecamatan yang ada di kota Medan. Untuk menunjang hasil yang optimal, penulis melakukan wawancara dengan pengelola bisnis jasa serta membaca dan mengumpulkan referensi dari berbagai penelitian terkait makalah dan buku akademik untuk mengkaji literatur terkait penelitian yang ditugaskan. Tahap selanjutnya adalah

tahap metode pengembangan sistem, dimana penulis menggunakan metode pengembangan sistem RAD dalam penelitian ini. Ini adalah metode pengembangan sistem yang menekankan waktu manufaktur yang efisien dalam pengembangan sistem. RAD sendiri memiliki tiga fase yang perlu dilakukan: perencanaan kebutuhan, desain bengkel, dan implementasi. Perencanaan kebutuhan adalah kegiatan atau fase di mana seorang penulis mengidentifikasi kebutuhan sistem dengan menganalisis data yang dibutuhkan. Hal ini dilakukan setelah sebelumnya penulis memiliki ide untuk membangun sistem yang diusulkan dengan mengumpulkan data dan mengimplementasikan *React Native* dan *Firebase* pada sistem yang mereka kembangkan. Kemudian dilanjutkan ke workshop desain. Pada tahap ini, penulis mulai merancang aliran atau model sistem, menyajikan data yang diperoleh dalam bentuk desain *database*, dan merancang struktur menu utama dan antarmuka sistem. Selanjutnya, tahap terakhir dari RAD adalah implementasi. Di sinilah implementasi *Firebase* diimplementasikan dan pengkodean sistem dilakukan. Setelah melakukan dua hal ini, pengujian unit dilakukan pada pengguna.

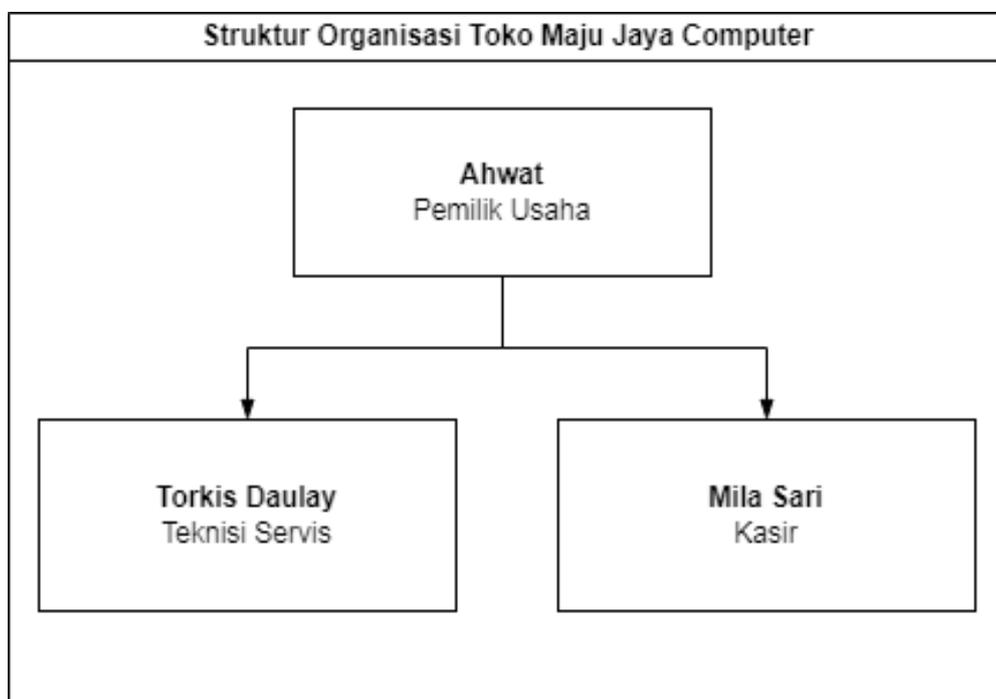
BAB IV HASIL DAN PEMBAHASAN

4.1. *Requirements Planning*

Seperti yang telah dijelaskan pada bab-bab sebelumnya tentang survei ini, penulis melakukan survei ke tujuh kecamatan dan beberapa layanan untuk mendapatkan data yang dibutuhkan untuk membuat sistem atau aplikasi. Selama observasi, penulis mendapatkan informasi sebagai berikut:

4.1.1. Struktur Organisasi Dalam Usaha Jasa Servis Ini

Pada usaha jasa servis ini penulis menyimpulkan bahwasanya struktur organisasi dalam jasa servis ini salah satunya pada toko *Maju Jaya Computer* tidaklah kompleks atau rumit. Strukturnya sangat sederhana, yaitu hanya diisi oleh pemilik usaha dan karyawannya, dalam hal ini karyawan bisa lebih dari satu tergantung tempat usaha tersebut. Dibawah ini merupakan gambaran struktur organisasi yang penulis katakan.



Gambar 4. 1: Struktur Organisasi Pada Toko Maju Jaya Computer

Struktur organisasi di atas merupakan struktur yang paling umum digunakan dalam bisnis jasa ini, namun ada kemungkinan pemilik bisnis tidak memiliki karyawan sama sekali, yaitu pemilik bisnis menjalankan bisnisnya sendiri.

4.1.2. *Job Description* dari struktur organisasi

Adapun *job description* dari struktur organisasi pada usaha jasa servis elektronik dapat dilihat pada table berikut ini :

Table 4. 1: Job Description

Jabatan	<i>Job Description</i>
1. Pemilik Usaha Jasa Servis	Jika pemilik bisnis jasa memiliki karyawan, tugas pemilik hanyalah meninjau tinjauan keuangan dan mengembangkan strategi pemasaran untuk membuat bisnisnya dikenal publik. Sebaliknya, jika pemilik toko tidak memiliki karyawan, maka pelayanannya meliputi seluruh proses bisnis jasa.
2. Bagian Servis	Jabatan ini bertugas untuk memeriksa dan memperbaiki peralatan elektronik, bertugas sebagai konsultasi dalam usaha ini apabila ada pelanggan yang ingin bertanya terkait kerusakan pada peralatan elektronik untuk mengetahui seberapa berat kerusakan elektronik tersebut.
3. Lainnya	Posisi ini ditugaskan secara kondisional dan secara hierarkis adalah satu karyawan.

4.1.3. Hasil Observasi Dan Wawancara

Berikut adalah hasil observasi pada beberapa toko dari setiap kecamatan di kota Medan, Observasi yang dilakukan peneliti pada toko “Maju Jaya *Computer*” berlokasi di Jl. Putri Merak Jingga, Kecamatan Medan Barat, toko “*Hardware Laptop*” berlokasi Jl. Sunggal No. 64, Sei Sikambing B, Kecamatan Medan Sunggal, dan toko “*Horas Computer*” berlokasi Jl. Sisingamangaraja No. 154, Teladan Barat, Kecamatan Medan Kota. Hasil observasi dan wawancara pada beberapa toko tersebut yaitu sulit untuk mendapatkan informasi toko dikarenakan belum memiliki media informasi *online* seperti situs *website* maupun sosial media, adapun informasi lokasi toko hanya bisa diakses melalui layanan *Google Maps*, kemudian terkait untuk mengetahui estimasi biaya servis pada toko tersebut mengharuskan pelanggan membawa peralatan elektronik ke toko untuk mengetahui berapa estimasi biaya servis yang harus dibayar, dan untuk mengetahui toko tersebut sedang buka atau tutup, pelanggan harus mengunjungi langsung ke lokasi toko servis.

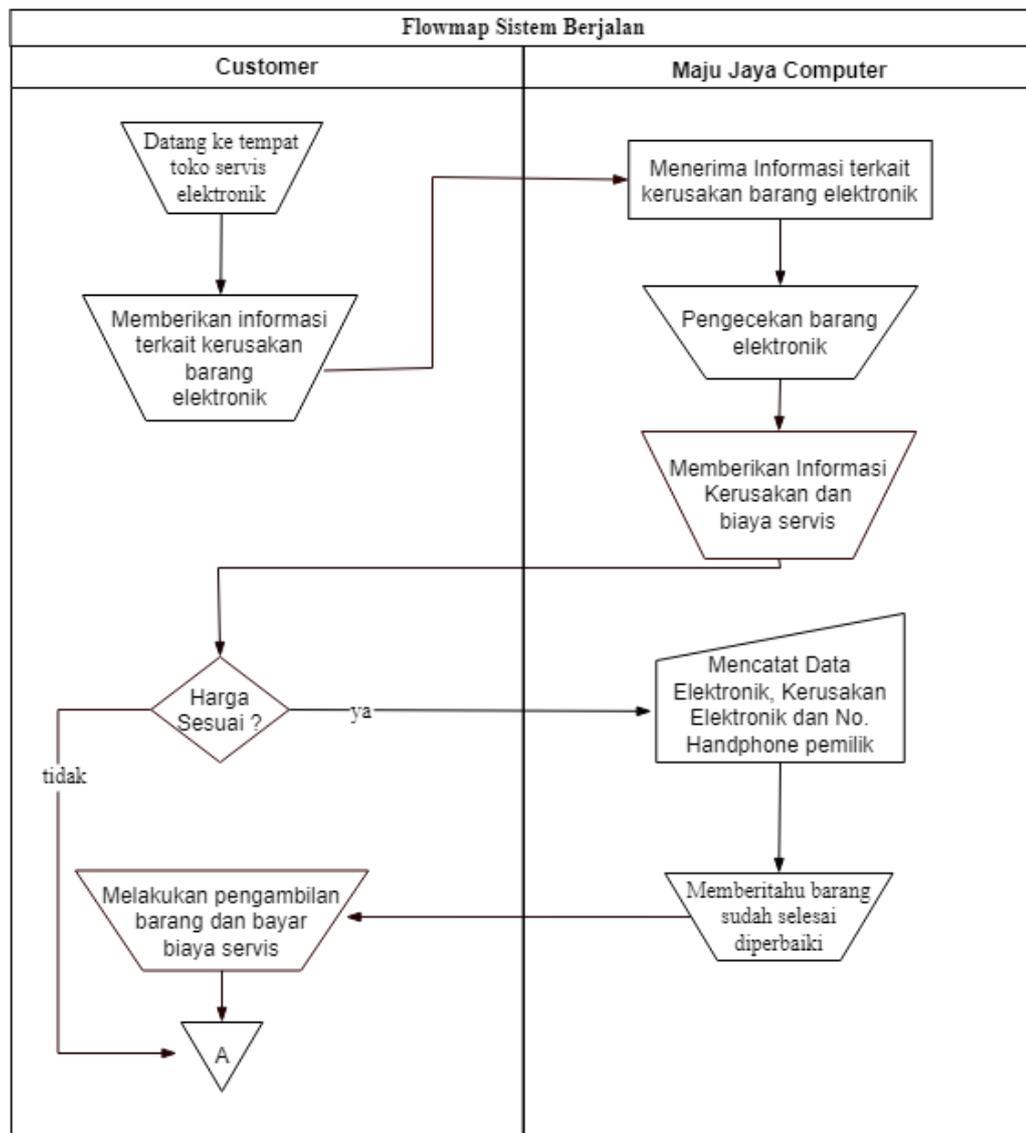
4.1.4. Analisis Sistem Berjalan

Pada sub-bab ini menjelaskan bagaimana *flow* atau alur dari sistem berjalan terhadap usaha jasa servis yang dilakukan secara konvensional. *Flowmap* dibawah tersebut dimulai dari *customer* yang datang toko jasa servis elektronik guna melakukan perbaikan perangkat elektronik tersebut, selanjutnya *customer* akan memberikan informasi kerusakan perangkat elektronik kepada pihak toko jasa servis dan pihak toko jasa servis akan menyerap informasi tersebut lalu memprosesnya untuk melakukan cek kondisi fisik serta kerusakan barang yang ingin di servis, kemudian pihak toko jasa servis menentukan biaya servis perangkat elektronik kepada *customer*, apabila biaya yang ditentukan tidak mencukupi dana *customer*, maka servis tidak dilakukan, apabila biaya servis yang ditentukan disetujui oleh *customer* maka pihak jasa servis memberikan kertas faktur yang terdiri dari identitas barang elektronik serta biaya jasa servis kepada *customer*, setelah barang elektronik telah selesai diperbaiki, maka pihak jasa

servis mengkonfirmasi *customer* untuk mengambil barang tersebut serta melakukan pembayaran jasa servis tersebut.

Adapun alurnya dijelaskan pada diagram berikut :

Table 4. 2: Flowmap Sistem Berjalan



4.1.5. Analisis Sistem Usulan

Sistem yang diusulkan untuk sistem yang dilakukan pada penelitian ini adalah mengembangkan aplikasi yang dapat dijalankan sebagai client pada sistem operasi *Android* dan menggunakan layanan *Firestore* *Google*. Usulan pada sistem ini akan dibuat dua aplikasi yaitu untuk *customer* dan pemilik toko jasa

servis elektronik, alasan kenapa harus memakai dua aplikasi agar aplikasi tidak mengalami *crash* saat permintaan data antara toko jasa servis dan *customer* serta menghindari peretasan data penting. Berikut adalah *flowmap* serta ilustrasi arsitektur sistem yang akan diusulkan:

Pada sistem usulan gambar 4.3, usulan disisi *customer*, beberapa proses sebagai berikut :

1. *Customer* memulai membuka aplikasi
2. Lalu melakukan registrasi/login menggunakan akun yang telah ada ke sistem.
3. Sistem akan menampilkan menu utama kepada *Customer*.
4. *Customer* akan memilih kategori barang yang akan di servis.
5. Sistem akan menampilkan ke *interface* pengguna toko yang tersedia dan terdekat dengan *Customer*.
6. *Customer* memilih toko jasa servis yang akan menjadi tempat melakukan servis.
7. *Customer* memilih tombol cek biaya servis.
8. *Customer* mengisi *form* data servis dan kerusakan, secara otomatis sistem akan menampilkan estimasi biaya servis.
9. *Customer* memilih tombol pengecekan lebih lanjut.
10. Sistem memberikan informasi pengambilan barang elektronik.
11. Apabila pihak toko jasa servis sudah menjemput barang, maka *customer* melakukan konfirmasi barang telah diberikan.
12. Sistem memberikan informasi data kerusakan dan biaya jasa servis dari toko jasa servis.
13. Jika *customer* menyetujui biaya servis perangkat elektronik, maka *customer* melakukan konfirmasi perbaikan barang, namun apabila tidak disetujui, maka *customer* melakukan konfirmasi pembatalan servis perangkat elektronik.
14. Setelah toko jasa servis sudah memperbaiki perangkat elektronik, maka pihak toko akan mengkonfirmasi pengembalian perangkat elektronik dan menampilkan rincian biaya yang harus dibayarkan.

15. Setelah barang diterima, maka *customer* melakukan konfirmasi barang telah diterima.

Selanjutnya pada sisi Jasa servis. Dimana penjelasannya adalah sebagai berikut :

1. Jasa servis memulai aplikasi
2. Selanjutnya melakukan registrasi akun/login menggunakan akun yang sudah ada.
3. Lalu sistem akan menampilkan menu utama.
4. Kondisi notifikasi muncul merupakan ada perbaikan perangkat elektronik yang masuk dari *customer*, apabila kondisi bernilai benar maka memilih menus servis, jika bernilai salah maka alur selesai.
5. Dari poin nomor empat jika dilanjutkan maka jasa servis memilih data servis terbaru.
6. Setelah sistem menampilkan informasi perangkat elektronik dan data kerusakan, maka pihak rook melakukan konfirmasi pengambilan perangkat elektronik
7. Apabila *customer* telah mengkonfirmasi perangkat elektronik telah diambil, maka pihak toko memberikan konfirmasi bahwa perangkat elektronik telah diterima.
8. Selanjutnya setelah toko jasa servis melakukan pengecekan kerusakan perangkat elektronik secara detail, kemudian pihak toko melakukan pengisian data kerusakan serta biaya jasa perbaikan dan menekan tombol konfirmasi biaya servis.
9. Apabila pihak *customer* melakukan pembatalan servis, maka toko jasa servis akan melakukan pengembalian perangkat elektronik kepada pelanggan, namun apabila perbaikan elektronik disetujui, melakukan konfirmasi perbaikan perangkat elektronik setelah perangkat sudah diperbaiki dan akan melakukan pengiriman perangkat elektronik kembali sekaligus mengambil dana perbaikan dari *customer*.

10. Setelah calon jasa servis melakukan konfirmasi barang diterima, maka sistem akan menampilkan data penerimaan perangkat elektronik pada toko jasa servis

Table 4. 3: Flowmap Sistem Usulan Bagian 1

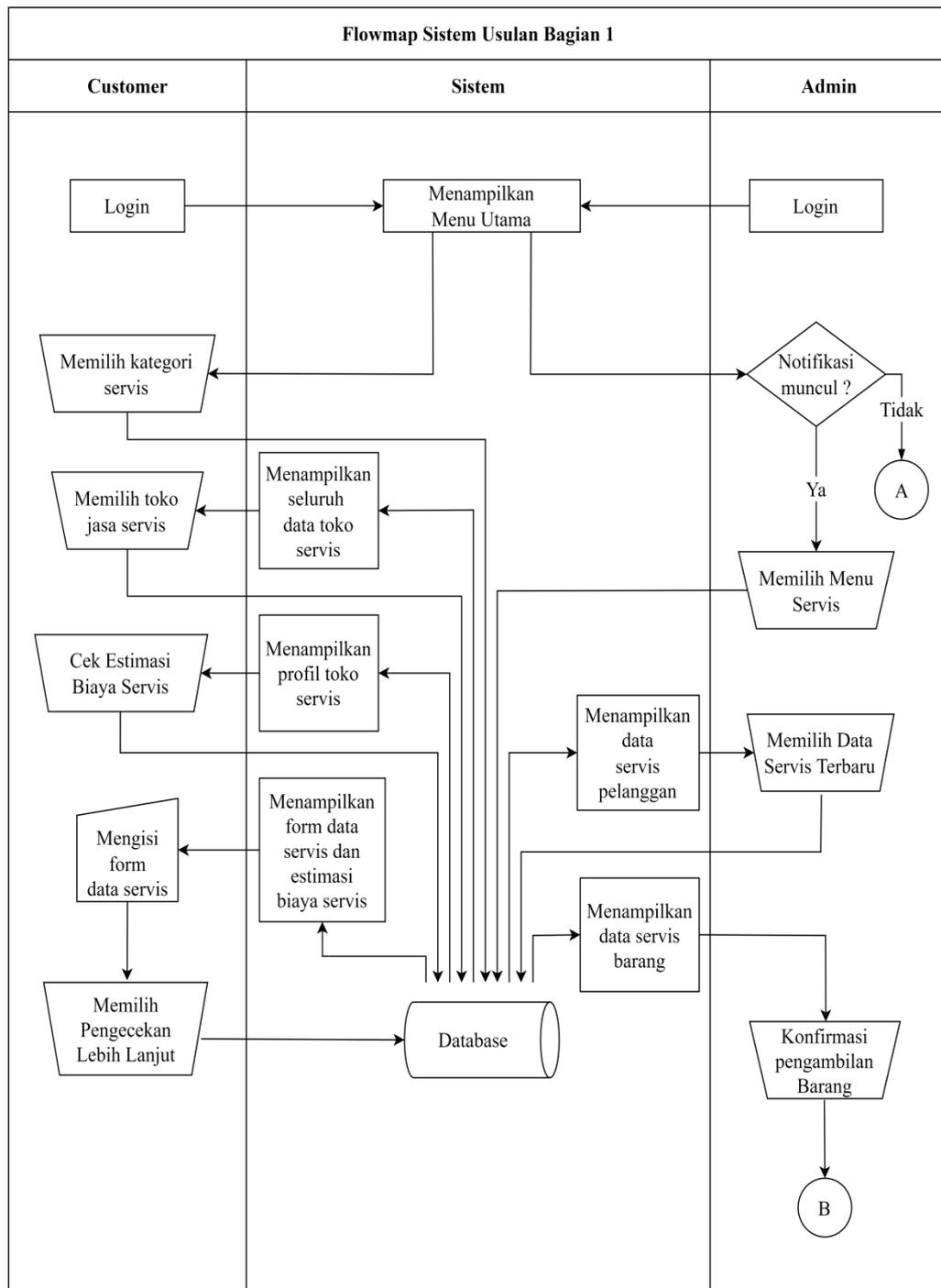
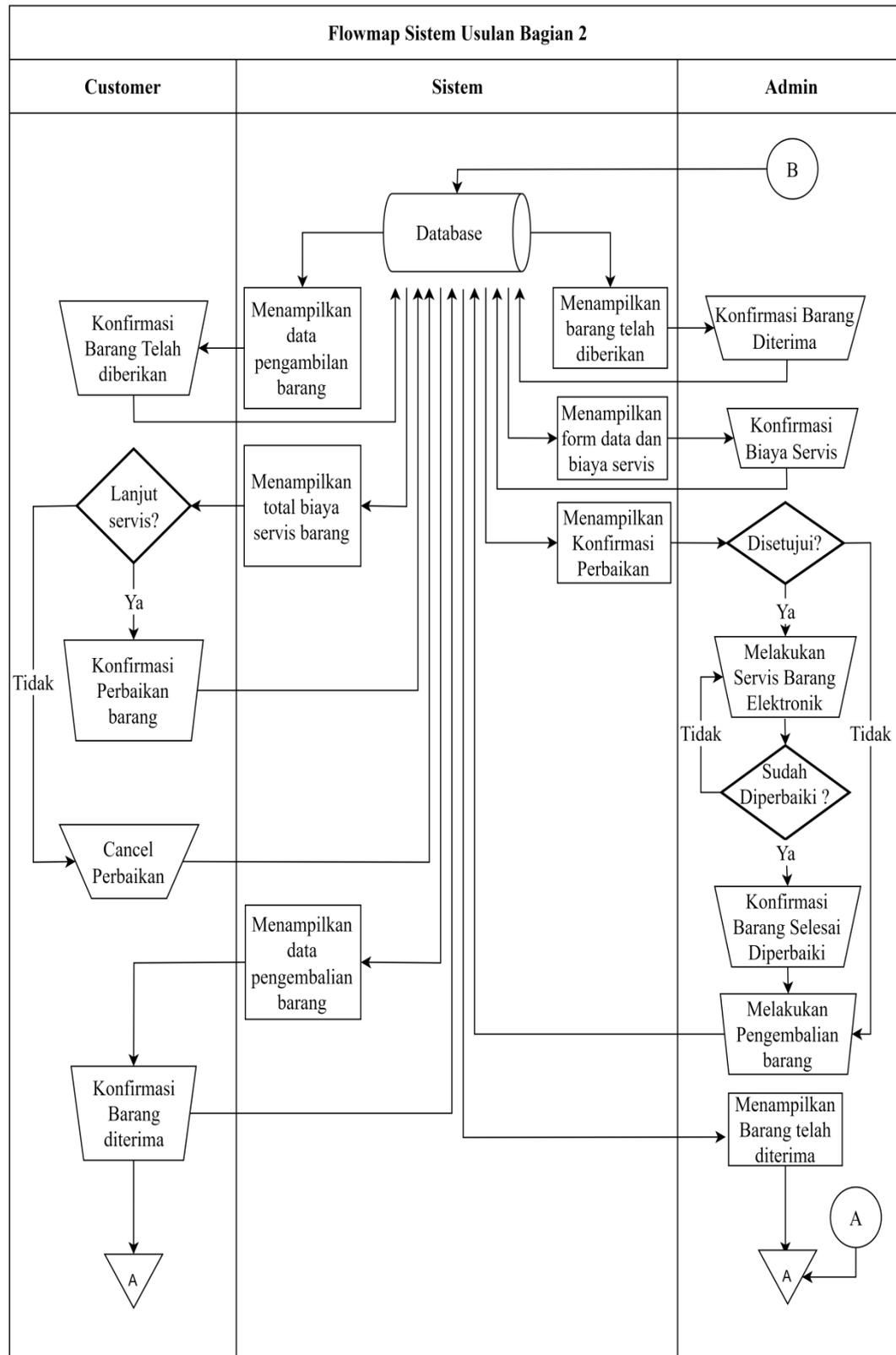
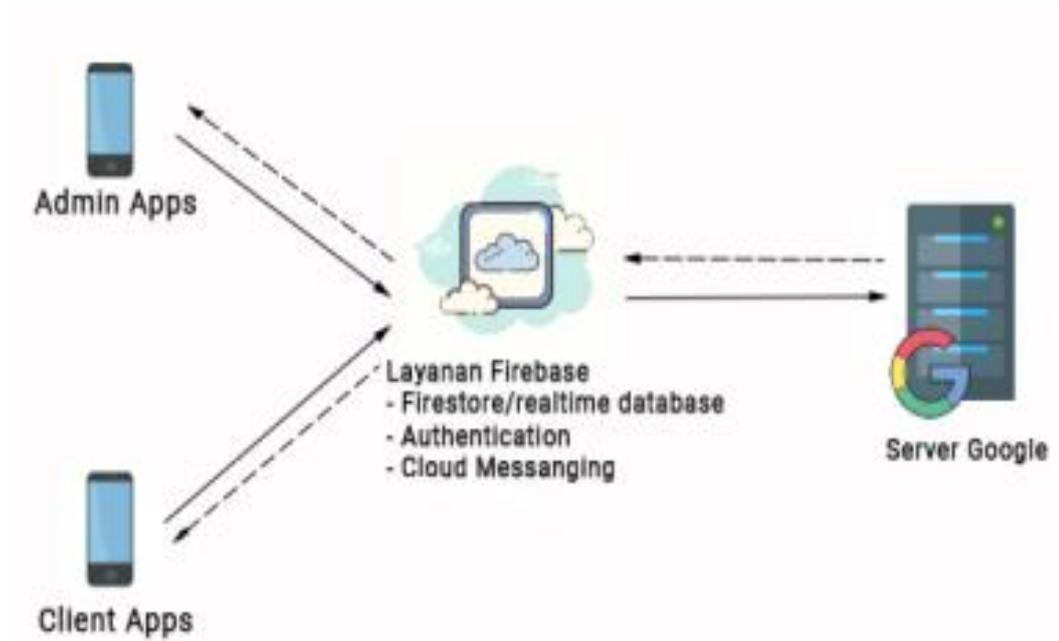


Table 4. 4: Flowmap Sistem Usulan Bagian 2



Berikut ini adalah arsitektur cara membangun aplikasi menggunakan dua aplikasi yang terhubung dengan layanan *Firebase*.



Gambar 4. 2: Arsitektur sistem usulan

Dari ilustrasi diatas dapat dilihat bahwa sistem ini melakukan *request* data ke layanan *firebase* dalam hal ini layanan yang dipakai ada dua yaitu *database realtime*, *authentication*, *cloud messaging* dan dengan *google* sebagai infrastruktur servernya.

4.2. Workshop Design

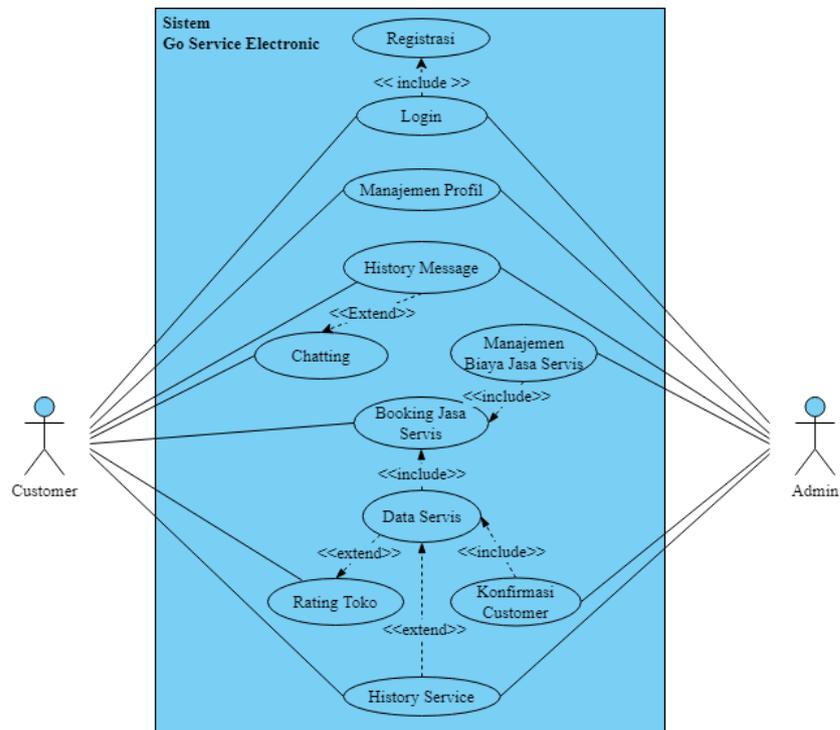
Pada bagian *Workshop Design* ini membahas tentang analisis yang lebih detail terkait jalan sistem usulan yang telah di bahas pada bagian sebelumnya, bagian ini akan membahas tiga hal yaitu tentang analisis sistem dengan melalui diagram UML, analisa *database*, dan analisa *interface* yang dibuat.

4.2.1. Design Model

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

4.2.1.1. Use Case Diagram

Secara singkat *usecase* diagram memaparkan tentang hal-hal berupa aktifitas apa saja yang bisa dilakukan oleh aktor selaku *user* dan *admin* dalam sistem yang akan dibangun atau dikembangkan. Pada gambar dibawah dapat dilihat bahwa yang berperan sebagai aktor adalah *customer* dan *Admin*, dari diagram terlihat kedua aktor dapat melakukan *usecase* *registrasi*, *login*, *chatting*, manajemen profil, *history message*, dan *history service*. Aktor *customer* dapat melakukan *booking* jasa servis, memberikan *rating* pada toko, histori pesan, dan histori perbaikan. Sementara *admin* dapat melakukan konfirmasi *customer* dan manajemen toko dan manajemen kerusakan.



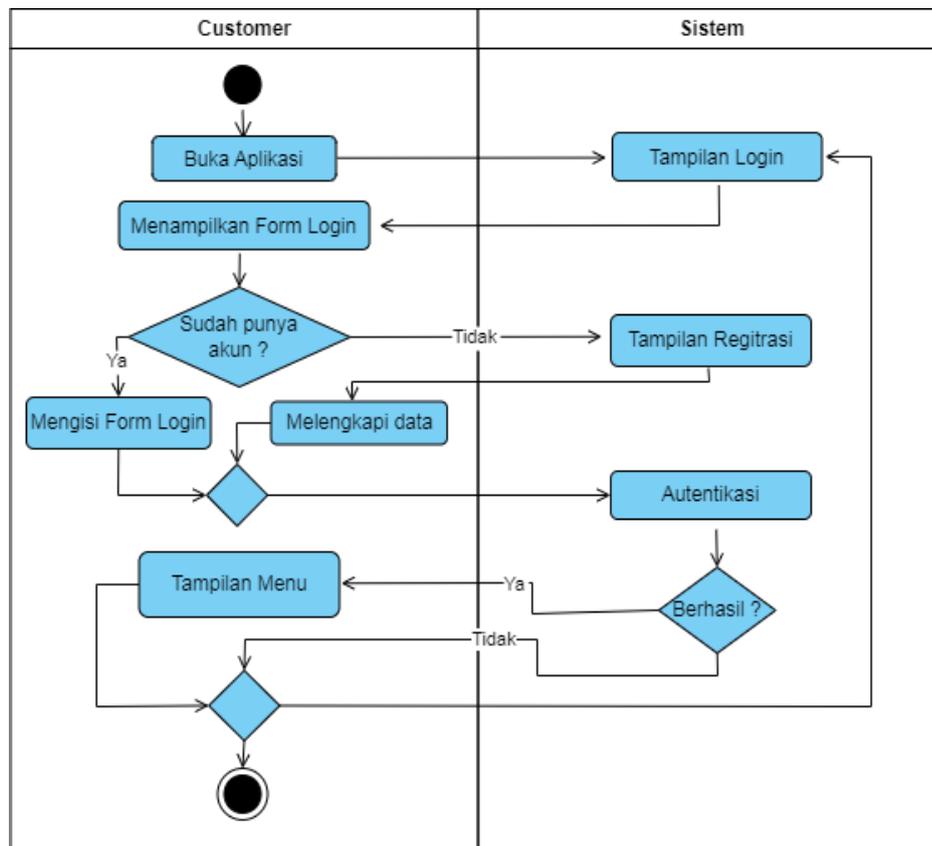
Gambar 4. 3: Usecase Diagram sistem Go Service Electronic

4.2.1.2. Activity Diagram

Activity Diagram ialah sebuah *flow* tentang aktifitas apa saja yang bisa di lakukan oleh *customer* dan *Admin*. *Activity* yang akan dipaparkan terbagi dua yaitu *activity* untuk sisi *customer* dan *activity* untuk sisi *Admin*. Berikut ialah beberapa *activity* yang akan dibahas pada aplikasi *Go Service Electronic*.

1. Activity Diagram Registrasi dan Login (Customer)

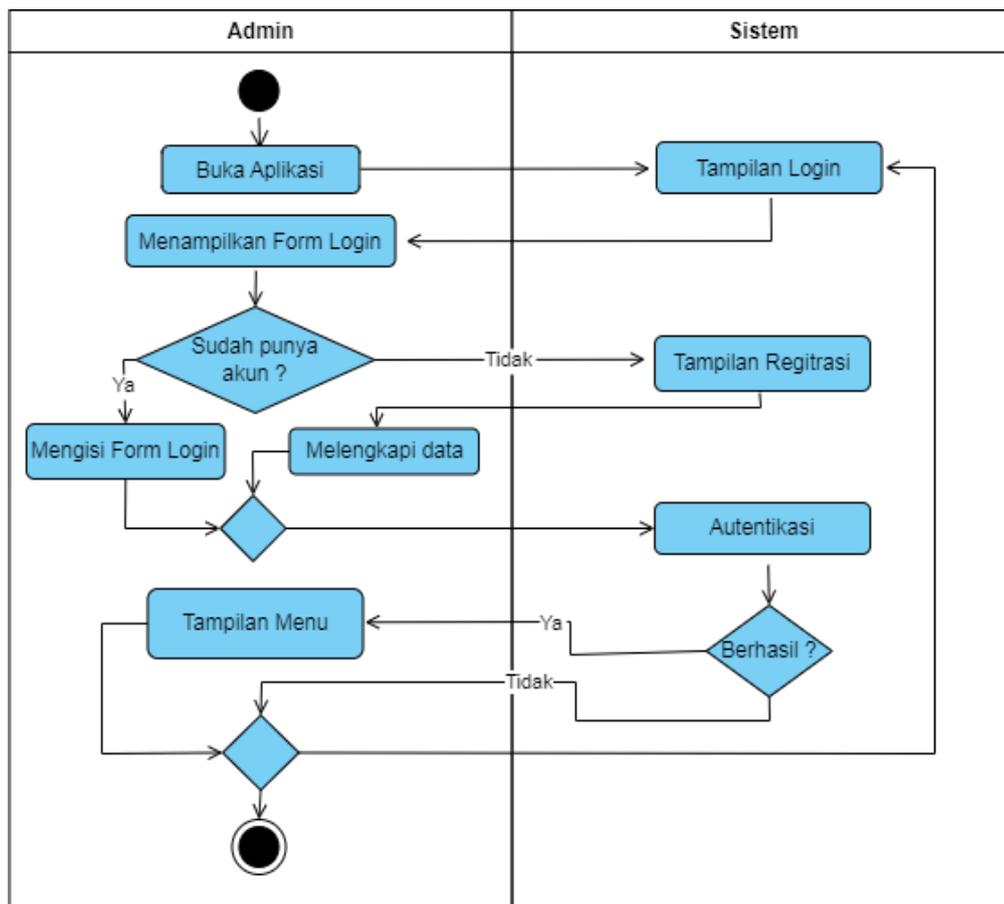
Pada gambar dibawah adalah proses melakukan login, pada saat melakukan login, *customer* akan membuka aplikasi setelah itu sistem akan menampilkan halaman login lalu akan menampilkan *form* login, apabila *customer* sudah memiliki akun, kedua pihak akan mengisi *form* login yang sudah di daftarkan sebelumnya, setelah itu sistem akan memproses data login *customer* apa bila *email* atau *password* benar maka sistem akan langsung menampilkan menu utama, apabila salah maka sistem akan menampilkan halaman login kembali. Apabila *customer* belum memiliki akun, maka kedua pihak melakukan registrasi terlebih dahulu, kemudian *customer* mengisi data untuk registrasi, apabila sudah melakukan registrasi, maka sistem akan memproses data registrasi *customer*, apabila data tersebut salah, maka sistem akan kembali ke halaman login, namun apabila data benar, maka sistem akan menampilkan menu utama.



Gambar 4. 4: Activity Diagram Registrasi dan Login (Customer)

2. Activity Diagram Registrasi dan Login (Admin)

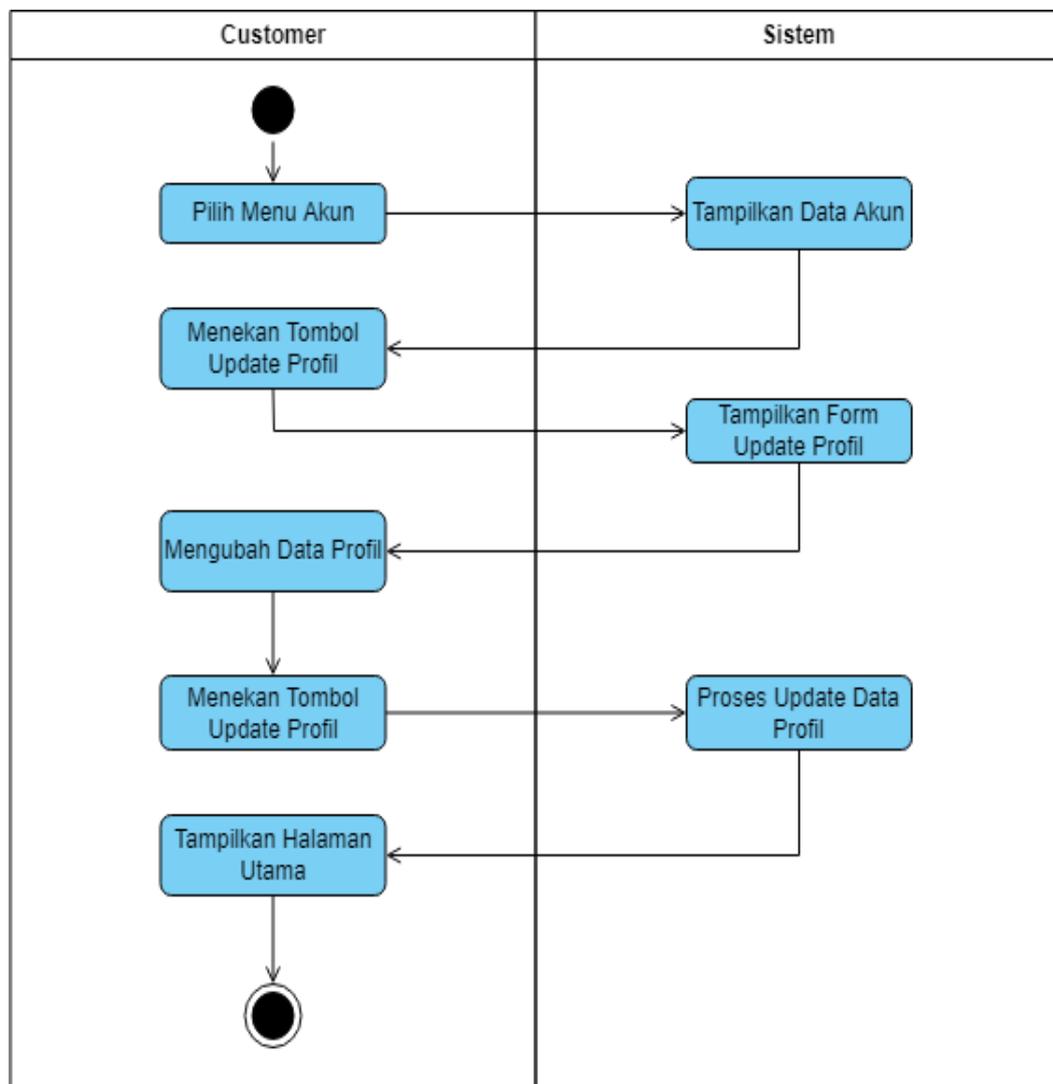
Pada gambar dibawah adalah proses melakukan login, pada saat melakukan login, kedua pihak akan membuka aplikasi setelah itu sistem akan menampilkan halaman login lalu di karyawan akan menampilkan form login, apabila kedua pihak sudah memiliki akun, kedua pihak akan mengisi form login yang sudah di daftarkan sebelumnya, setelah itu sistem akan memproses data login mereka apa bila *email* atau *password* benar maka sistem akan langsung menampilkan menu utama, apabila salah maka sistem akan menampilkan halaman login kembali. Apabila kedua pihak belum memiliki akun, maka kedua pihak melakukan registrasi terlebih dahulu, kemudian kedua pihak mengisi data untuk registrasi, apabila sudah melakukan registrasi, maka sistem akan memproses data regitrasinya mereka, apabila data tersebut salah, maka sistem akan kembali ke halaman login, namun apabila data benar, maka sistem akan menampilkan menu utama.



Gambar 4. 5: Activity Diagram Registrasi dan Login (Admin)

3. Activity Diagram Manajemen Profil (Customer)

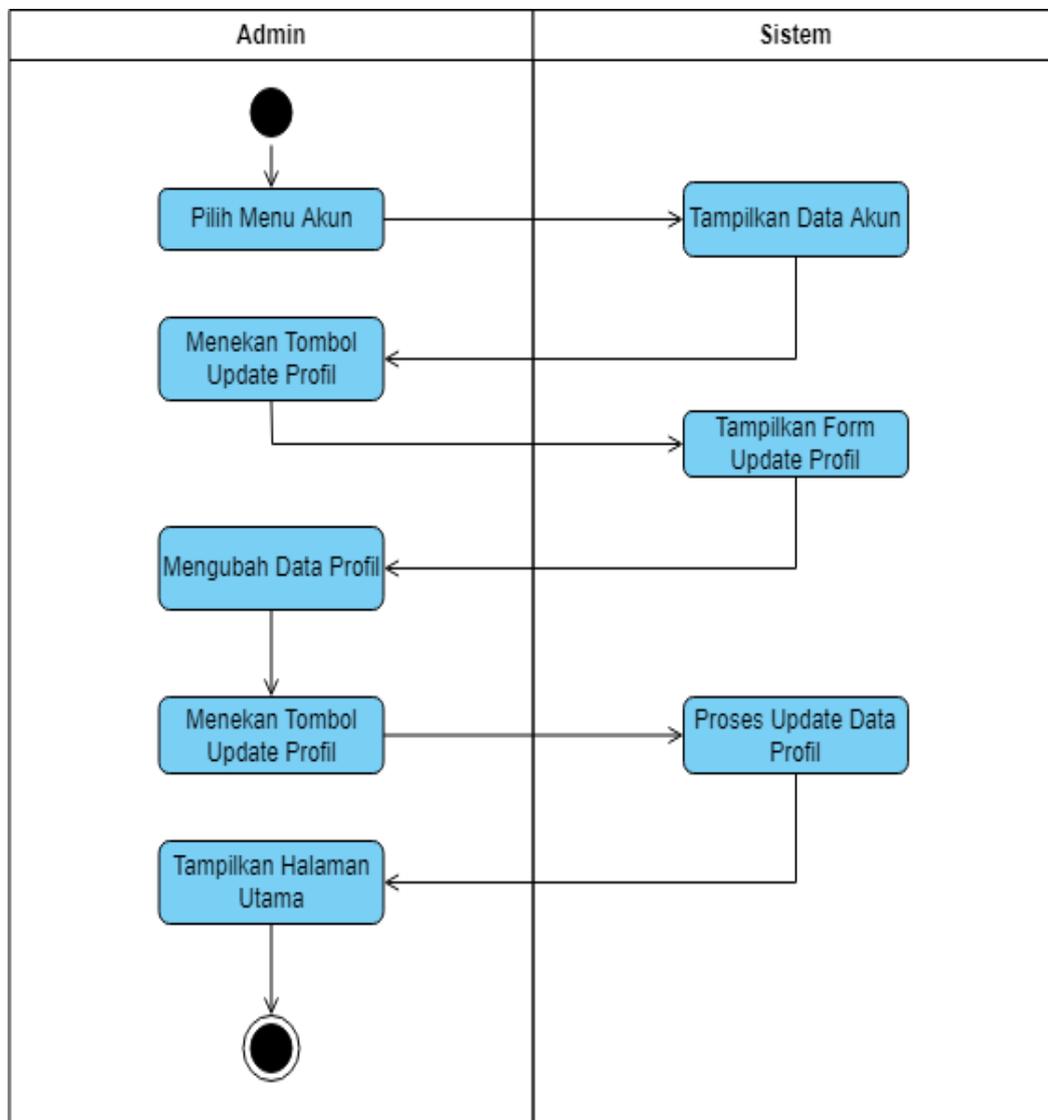
Pada gambar dibawah adalah Activity diagram manajemen profil, dimana *customer* memilih menu profil, kemudian sistem menampilkan data profil, selanjutnya untuk melakukan *update* profil, maka *customer* menekan tombol *update* profil, kemudian sistem akan menampilkan *form update* profil, kemudian *customer* mengubah data profil yang ingin diubah, setelah selesai mengubah data, maka *customer* menekan tombol *update* profil sehingga sistem memproses *update* data profil, setelah di *update*, maka pada sisi *customer* akan tampil ke halaman utama.



Gambar 4. 6: Activity Diagram Manajemen Profil (Customer)

4. Activity Diagram Manajemen Profil (Admin)

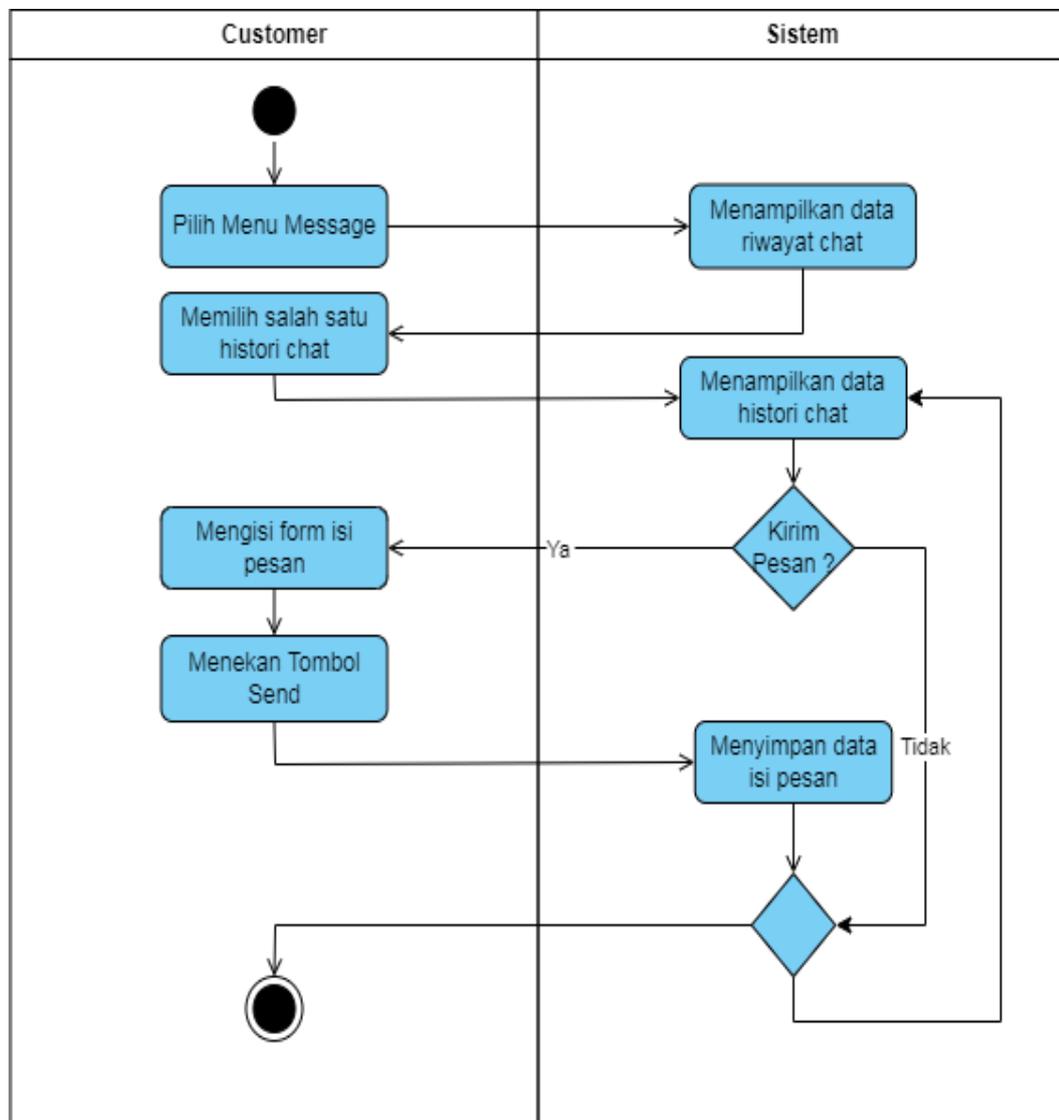
Pada gambar dibawah adalah *Activity* diagram manajemen profil, dimana *admin* memilih menu profil, kemudian sistem menampilkan data profil, selanjutnya untuk melakukan *update* profil, maka *admin* menekan tombol *update* profil, kemudian sistem akan menampilkan *form update* profil, kemudian *admin* mengubah data profil yang ingin diubah, setelah selesai mengubah data, maka *admin* menekan tombol *update* profil sehingga sistem memproses *update* data profil, setelah di *update*, maka pada sisi *admin* akan tampil ke halaman utama.



Gambar 4. 7: Activity Diagram Manajemen Profil (Admin)

5. Activity Diagram *History Message (Customer)*

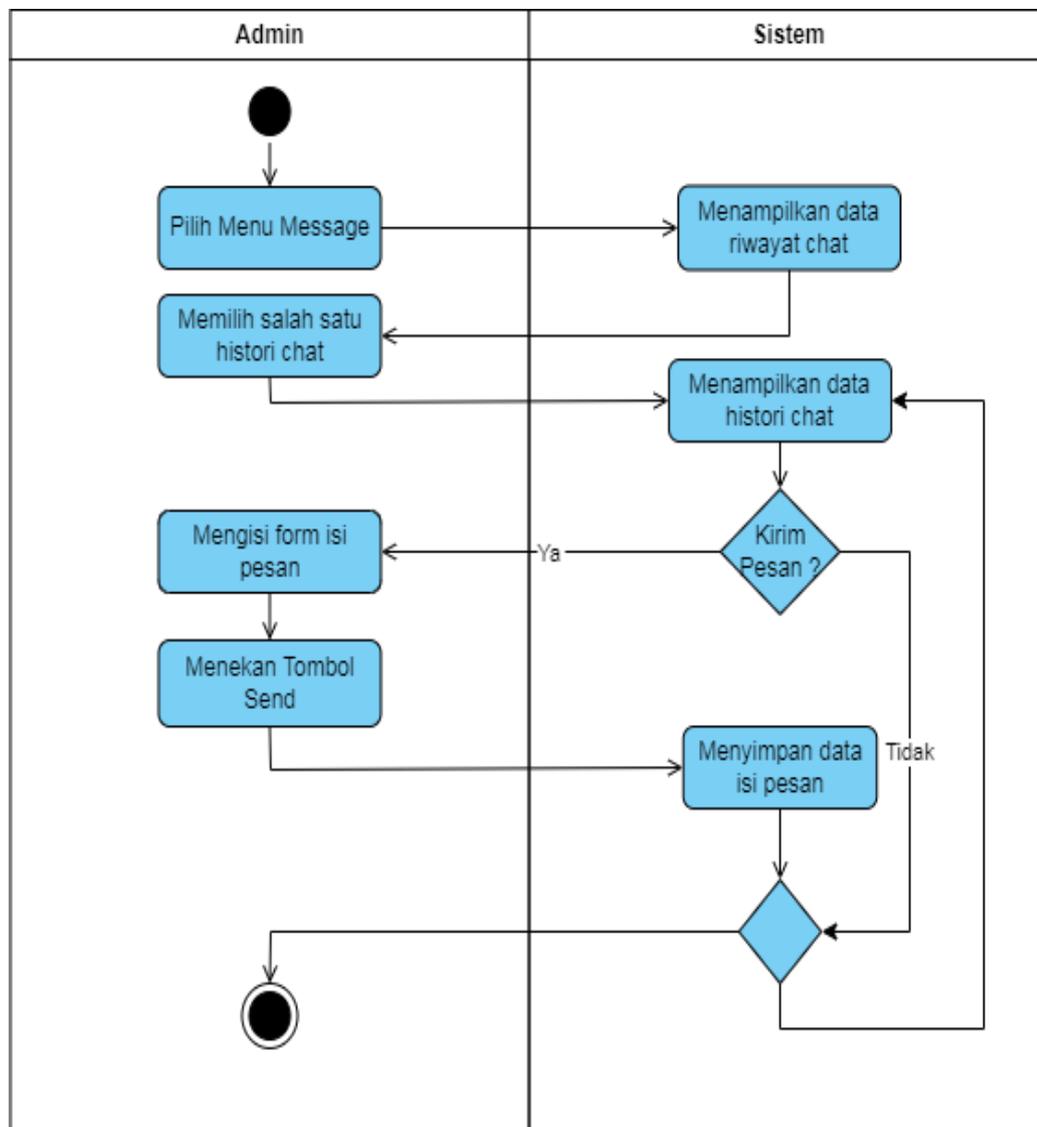
Pada gambar dibawah adalah Activity Diagram *History Message*, dimana *customer* memilih menu *message*, kemudian sistem akan menampilkan data riwayat *chat*, setelah itu *customer* memilih salah satu histori *chat* sehingga sistem menampilkan seluruh data histori *chat*, kemudian *customer* mengisi *form* isi pesan, setelah mengisi *form* pesan, kemudian kedua pihak menekan tombol *send* sehingga sistem akan menyimpan data isi pesan, kemudian pada bagian *admin* akan tampil kembali data histori *chat*.



Gambar 4. 8: Activity Diagram *History Message (Customer)*

6. Activity Diagram *History Message (Admin)*

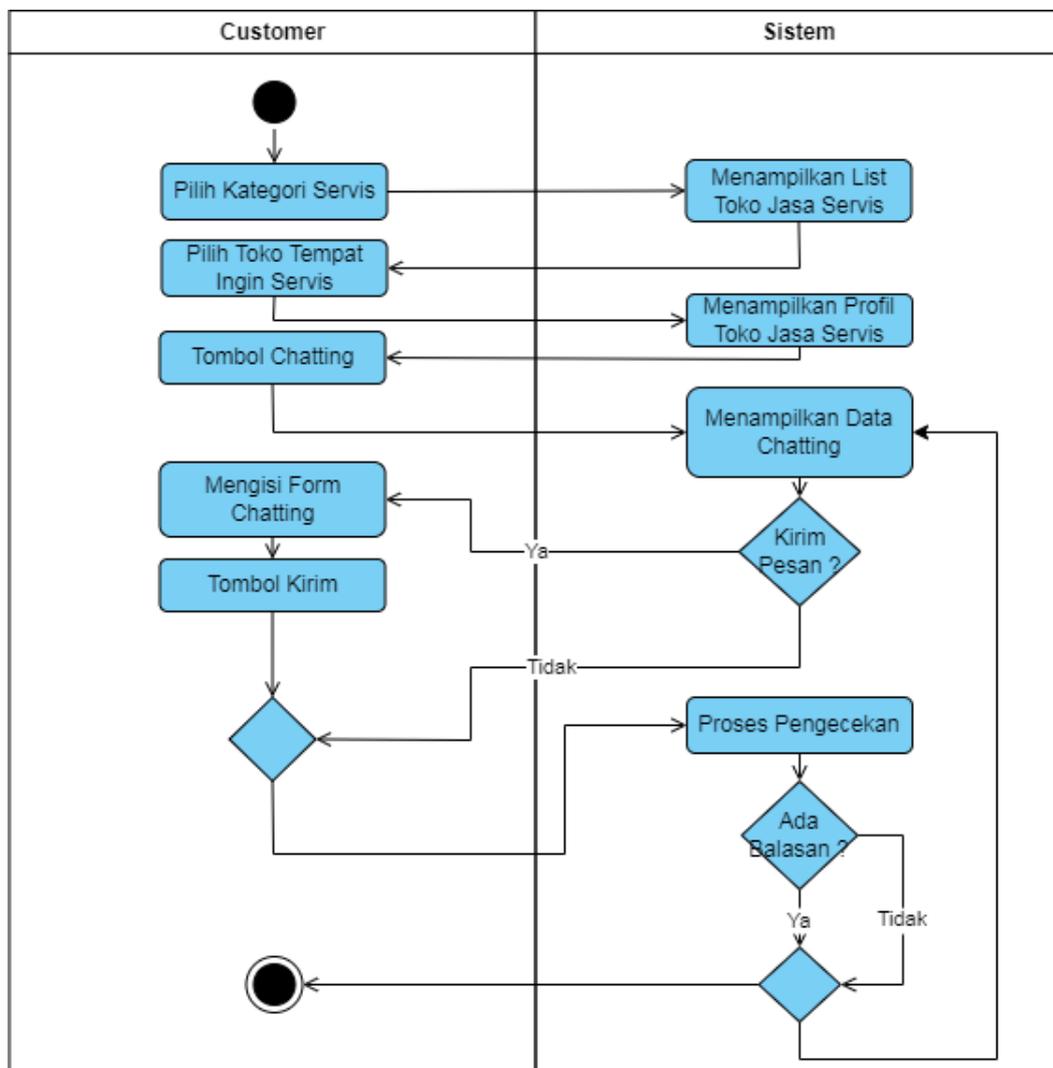
Pada gambar dibawah adalah *Activity Diagram History Message*, dimana *admin* memilih menu *message*, kemudian sistem akan menampilkan data riwayat *chat*, setelah itu *admin* memilih salah satu histori *chat* sehingga sistem menampilkan seluruh data histori *chat*, kemudian *admin* mengisi *form* isi pesan, setelah mengisi *form* pesan, kemudian *admin* menekan tombol *send* sehingga sistem akan menyimpan data isi pesan, kemudian pada bagian *customer* akan tampil kembali data histori *chat*.



Gambar 4. 9: Activity Diagram *History Message (Admin)*

7. Activity Diagram Chatting (Customer)

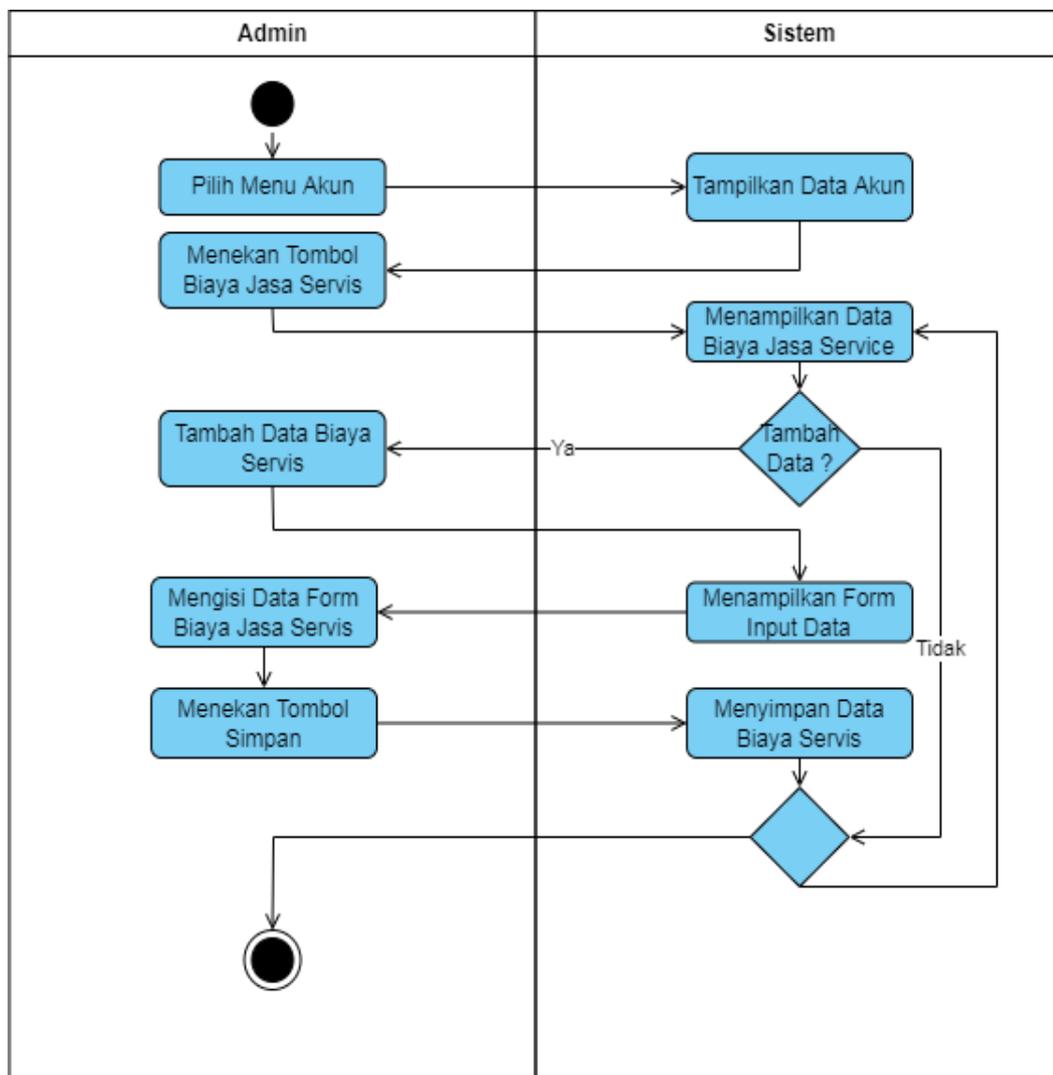
Pada gambar dibawah adalah proses melakukan *chatting*, pada saat melakukan *chatting*, *customer* akan memilih kategori jasa servis terlebih dahulu, setelah itu sistem akan menampilkan *Admin* sesuai dengan kategori yang dipilih serta menampilkan jarak dan rating toko tersebut. Kemudian *customer* memilih salah satu *Admin* untuk tempat servis barang elektronik, setelah itu sistem akan menampilkan data profil *Admin*, setelah itu *customer* menekan tombol *chatting*, kemudian sistem menampilkan data *chatting*, selanjutnya *customer* mengisi *form chat* menekan tombol kirim. Setelah sistem menampilkan data balasan *chat*, maka sistem akan menampilkan data *chatting*.



Gambar 4. 10: Activity Diagram Chatting (Customer)

8. Activity Diagram Manajemen Biaya Jasa Servis (Admin)

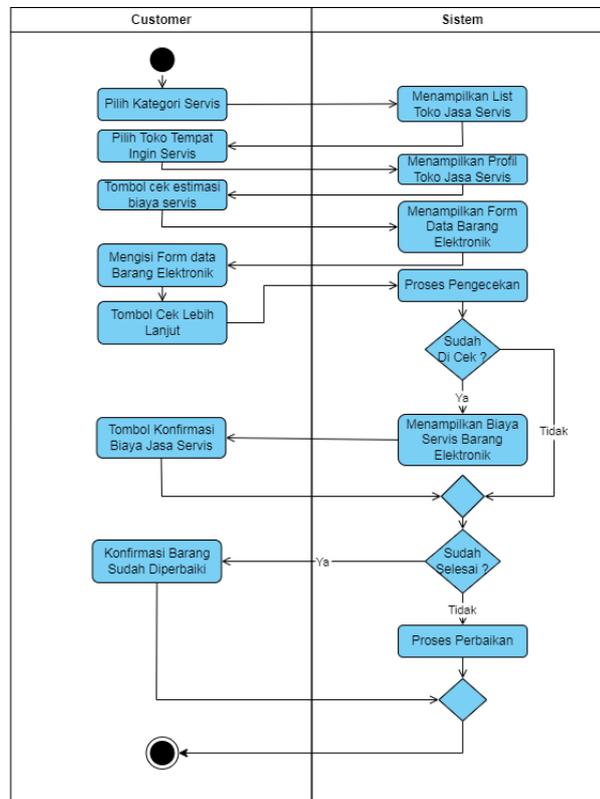
Pada gambar dibawah adalah *activity* diagram manajemen biaya jasa servis, pada saat *customer* ingin melakukan pengecekan estimasi biaya jasa servis, sistem memerlukan data biaya servis dari setiap toko, maka setiap toko melakukan input data biaya servis pada aplikasi bagian *Admin*, dengan memilih menu akun, kemudian sistem menampilkan data akun, kemudian *Admin* menekan tombol biaya jasa servis, kemudian sistem data biaya jasa servis, selanjutnya menekan tombol data biaya servis, selanjutnya *Admin* mengisi data servis baru, kemudian menekan tombol simpan data.



Gambar 4. 11: Activity Diagram Manajemen Biaya Jasa Servis (Admin)

9. Activity Diagram *Booking Jasa Servis (Customer)*

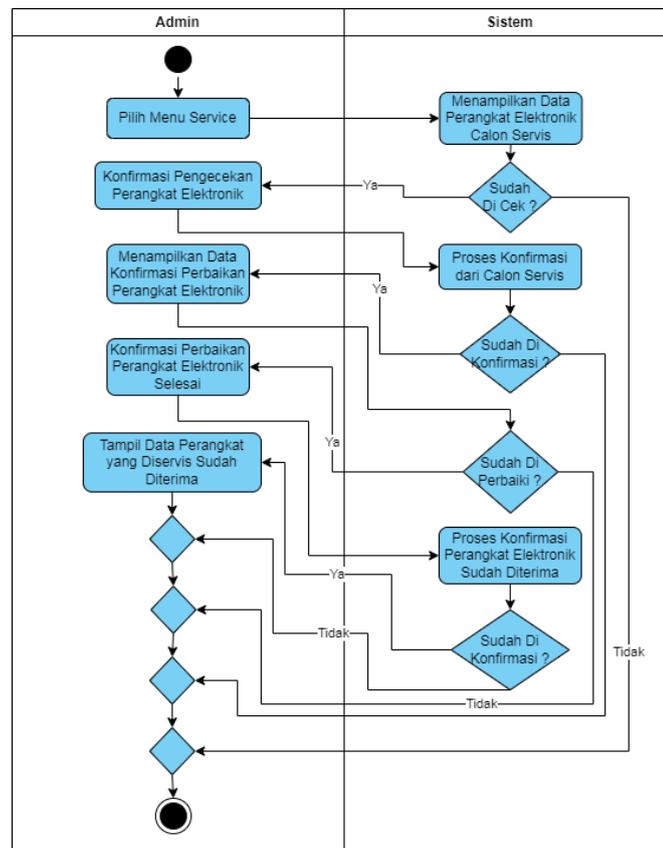
Pada gambar diatas adalah proses melakukan booking jasa servis, pada saat melakukan booking jasa servis, *customer* akan memilih kategori jasa servis terlebih dahulu, setelah itu sistem akan menampilkan *Admin* sesuai dengan kategori yang dipilih serta menampilkan jarak dan rating toko tersebut. Kemudian *customer* memilih salah satu *Admin* untuk tempat servis barang elektronik, setelah itu sistem akan menampilkan data profil *Admin*, setelah itu *customer* menekan tombol cek lebih lanjut, kemudian sistem menampilkan *form* data barang elektronik, selanjutnya *customer* mengisi data yang diperlukan untuk melakukan servis kemudian menekan tombol cek barang. Setelah sistem menampilkan data biaya pengecekan barang elektronik, kemudian *customer* menekan tombol konfirmasi biaya perbaikan, apabila sistem menampilkan data barang elektronik telah selesai, maka calon penyewa melakukan konfirmasi barang elektronik sudah diterima.



Gambar 4. 12: Activity Diagram *Booking Jasa Servis (Customer)*

10. Activity Diagram Konfirmasi Customer (Admin)

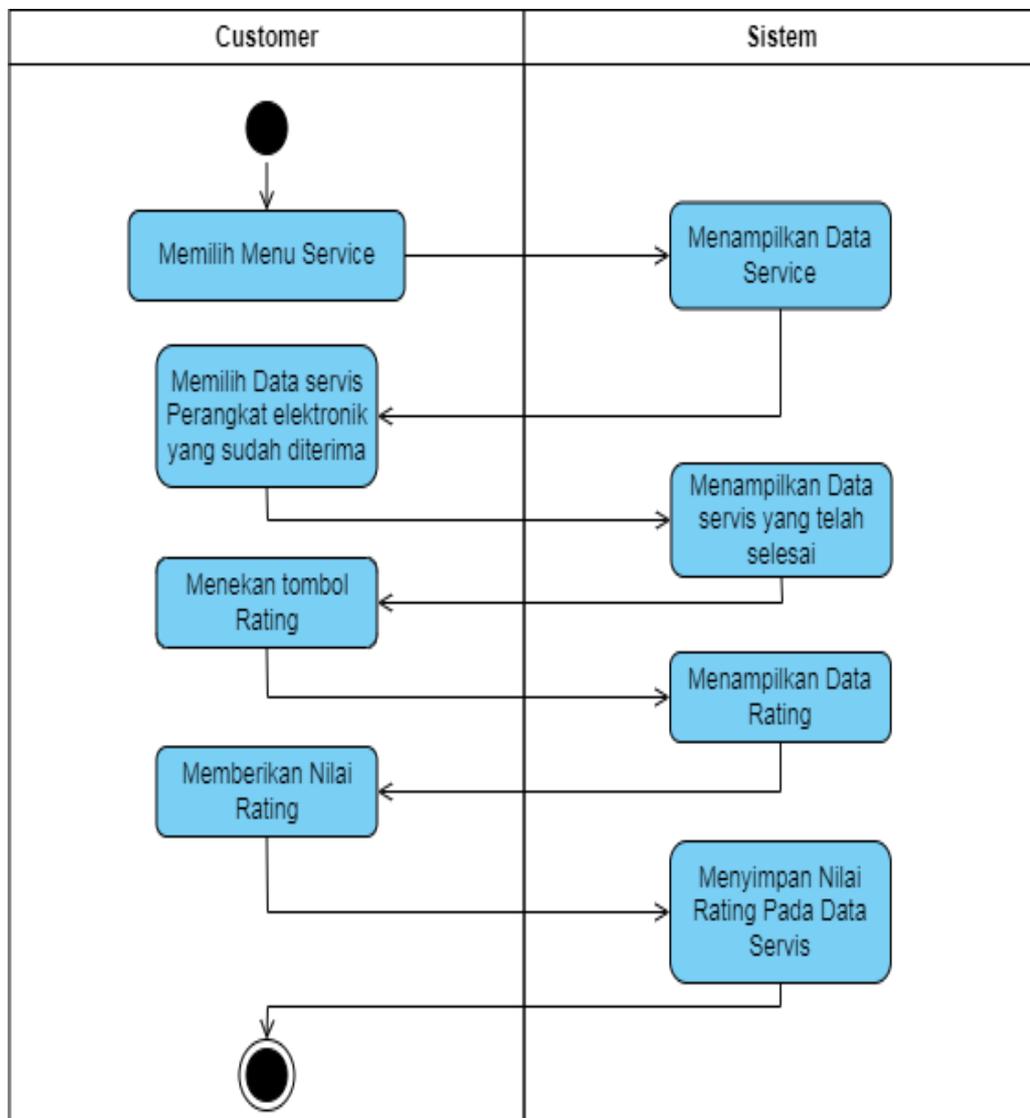
Pada gambar dibawah adalah Activity Diagram Konfirmasi Customer, dimana Admin memilih menu service kemudian sistem akan menampilkan data perangkat elektronik customer, apabila perangkat elektronik sudah di cek, maka Admin menekan tombol konfirmasi pengecekan perangkat elektronik, kemudian sistem akan memproses konfirmasi dari customer, apabila sistem telah menyatakan bahwa sudah dikonfirmasi, maka sistem akan menampilkan data konfirmasi perbaikan perangkat elektronik, kemudian apabila Admin telah selesai memperbaiki perangkat elektronik, maka Admin akan menekan tombol konfirmasi perbaikan perangkat elektronik telah selesai, setelah itu sistem akan memproses konfirmasi perangkat elektronik sudah diterima, apabila sistem telah mendapatkan konfirmasi dari customer, maka sistem akan menampilkan data perangkat yang diservis sudah diterima pada sisi Admin.



Gambar 4. 13: Activity Diagram Konfirmasi Customer (Admin)

11. Activity Diagram Rating Toko (Customer)

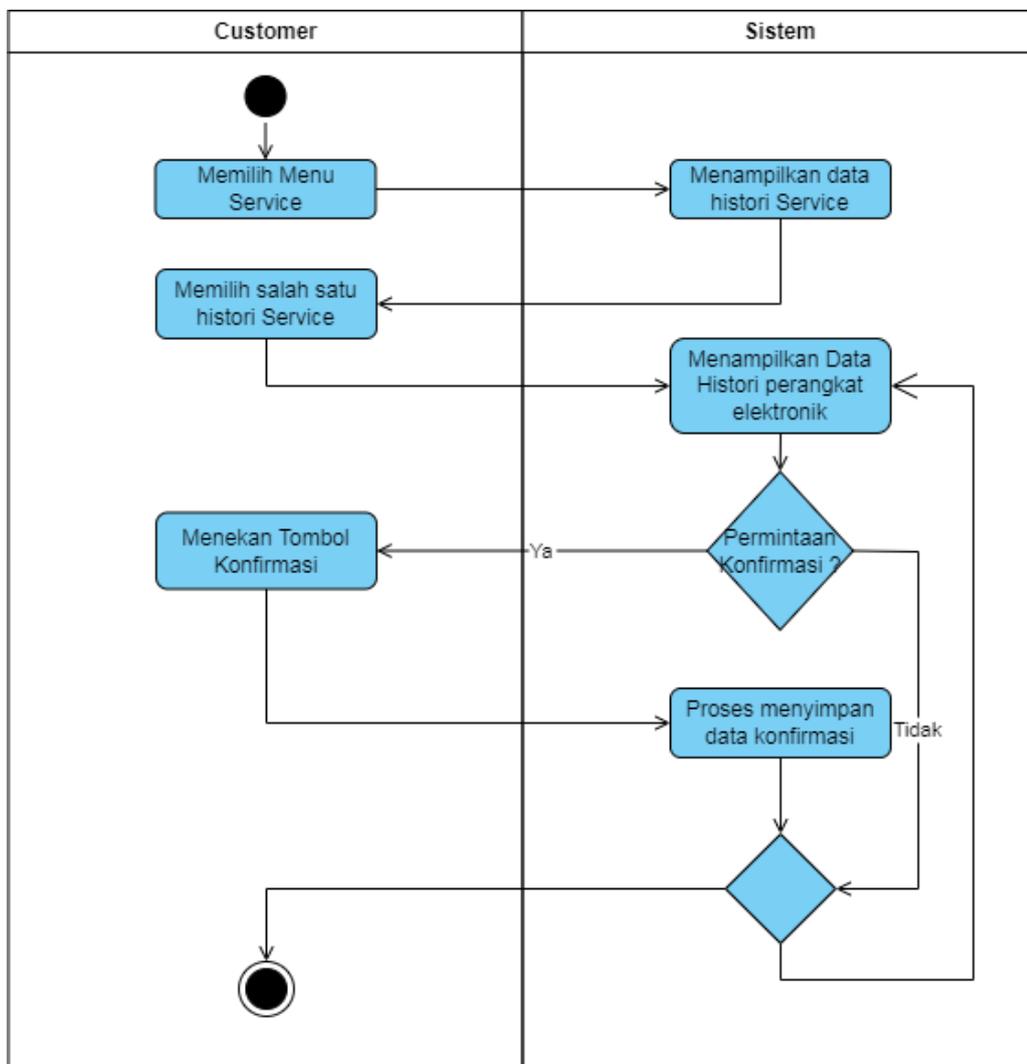
Pada gambar diatas adalah Activity Diagram *Rating Toko* pada pihak *customer*, dimana *customer* memilih menu *service* maka sistem akan menampilkan data *service*, kemudian *customer* memilih data servis perangkat elektronik yang sudah diterima maka sistem akan menampilkan data servis yang telah selesai, setelah itu *customer* menekan tombol *rating* maka sistem akan menampilkan data *Rating*, kemudian *customer* memberikan nilai *rating* kepada *Admin* , maka sistem akan menyimpan nilai *rating* pada data servis.



Gambar 4. 14: Activity Diagram Rating Toko (Customer)

12. Activity Diagram History Service (Customer)

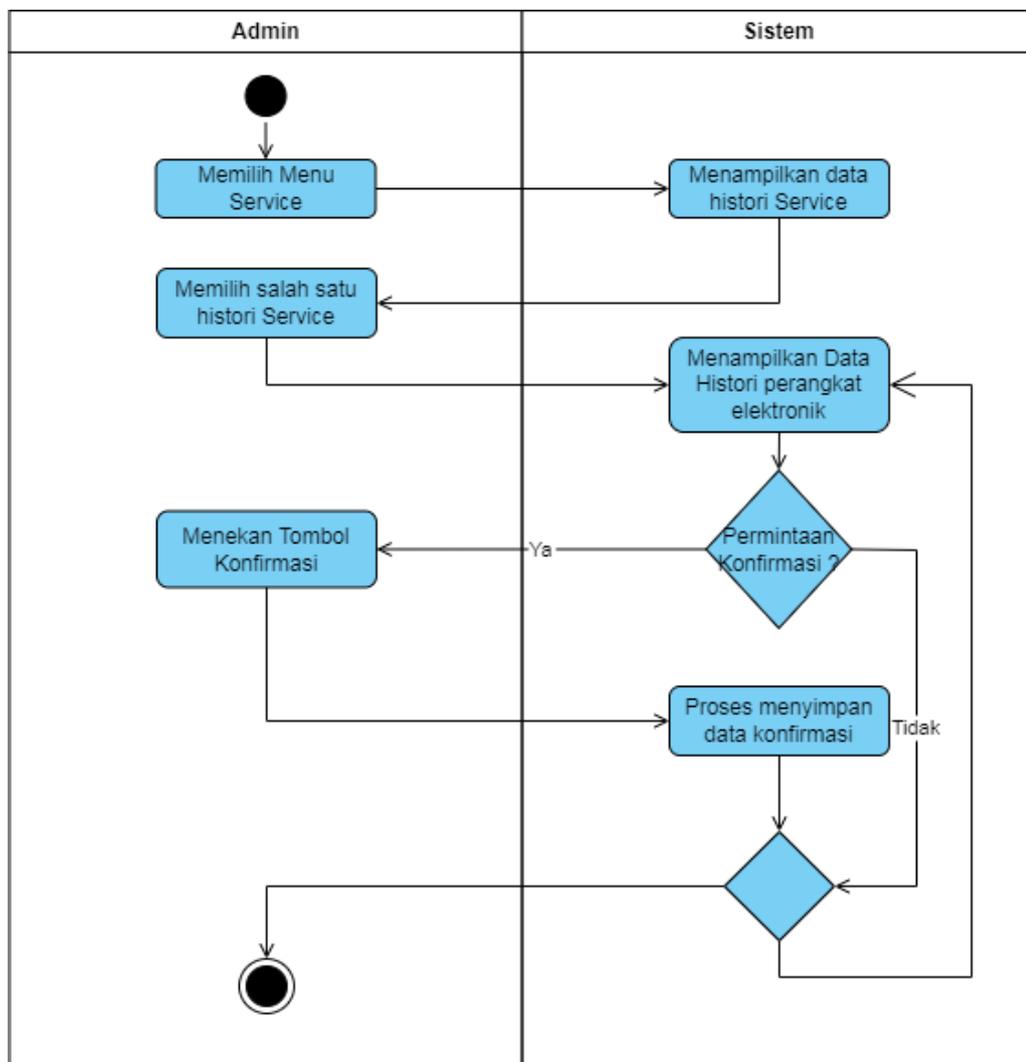
Pada gambar dibawah adalah Activity Diagram *History Service*, dimana *customer* memilih menu *service*, kemudia sistem akan menampilkan data histori *service*. Setelah itu *customer* memilih salah satu histori *service* sehingga sistem akan menampilkan data histori perangkat elektronik, apabila ada permintaan konfirmasi yang ditampilkan oleh sistem, maka *customer* menekan tombol konfirmasi untuk melanjutkan proses, sehingga sistem memproses penyimpanan data konfirmasi dan akan kembali menampilkan data histori perangkat elektronik pada sisi *customer*.



Gambar 4. 15: Activity Diagram History Service (Customer)

13. Activity Diagram History Service (Admin)

Pada gambar dibawah adalah Activity Diagram History Service, dimana *customer* memilih menu *service*, kemudia sistem akan menampilkan data histori *service*. Setelah itu *customer* memilih salah satu histori *service* sehingga sistem akan menampilkan data histori perangkat elektronik, apabila ada permintaan konfirmasi yang ditampilkan oleh sistem, maka *customer* menekan tombol konfirmasi untuk melanjutkan proses, sehingga sistem memproses penyimpanan data konfirmasi dan akan kembali menampilkan data histori perangkat elektronik pada sisi *customer*.



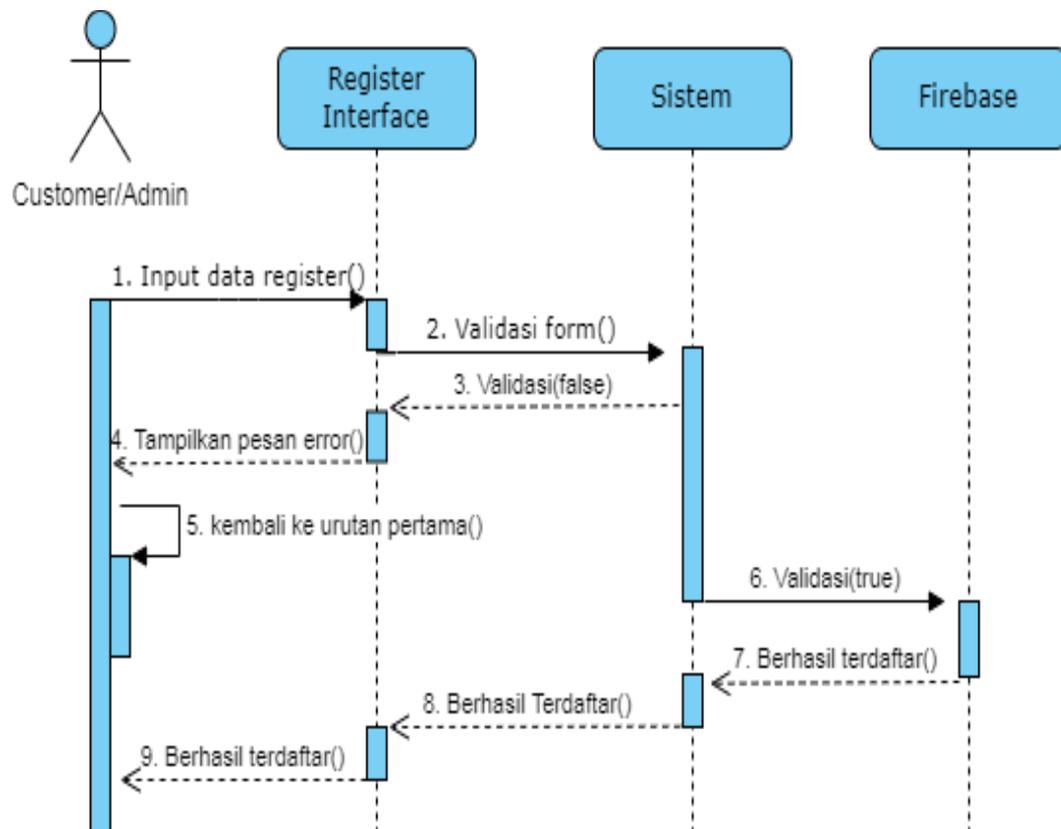
Gambar 4. 16: Activity Diagram History Service (Admin)

4.2.1.3. Sequence diagram

Berikut adalah beberapa diagram *sequence* yang digunakan dalam pengembangan aplikasi ini :

1. *Sequence Diagram Register (Customer dan Admin)*

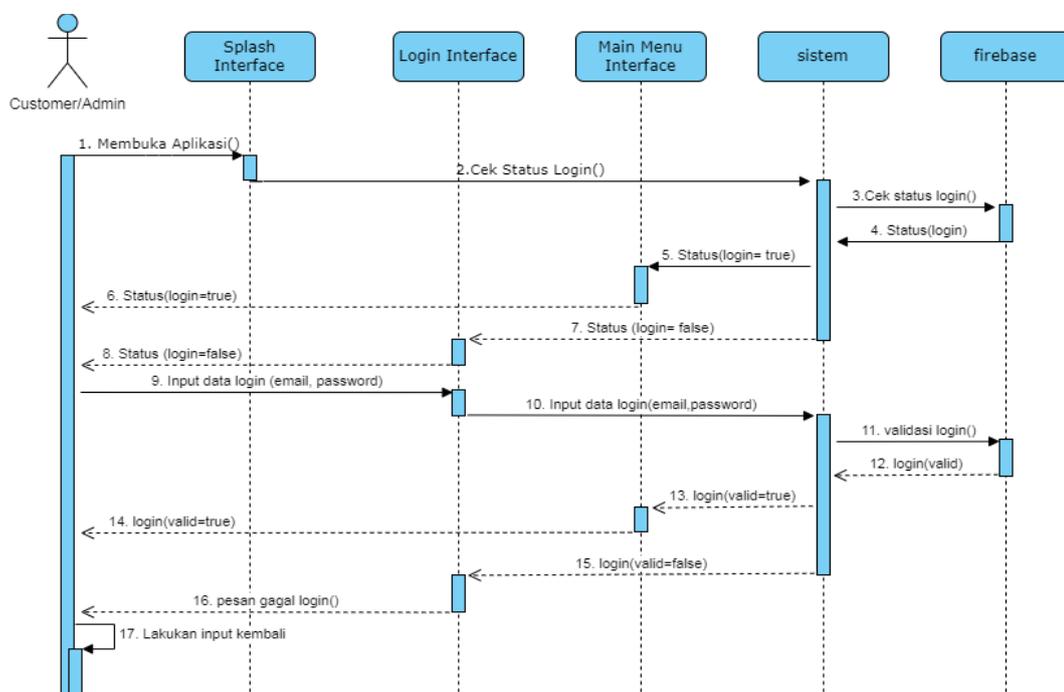
Gambar berikut menunjukkan rangkaian pesan antar objek saat kedua belah pihak menyelesaikan proses registrasi atau registrasi akun untuk menggunakan aplikasi. Grafik dimulai dengan memasukkan data pribadi di antarmuka aplikasi dan, di belakang layar, memvalidasi bahwa sistem benar-benar mengisi formulir. Jika validasi lolos, sistem meminta permintaan data dari *Firebase* dan memberikan respons bahwa data berhasil didaftarkan di *database* kedua belah pihak, tetapi jika validasi ditolak oleh sistem, sistem memanggil kembali / Memberikan panggilan balik. Ini menginformasikan kedua belah pihak bahwa mereka harus terlebih dahulu mengisi data formulir.



Gambar 4. 17: Sequence Diagram Register

2. *Sequence Diagram Login (Customer dan Admin)*

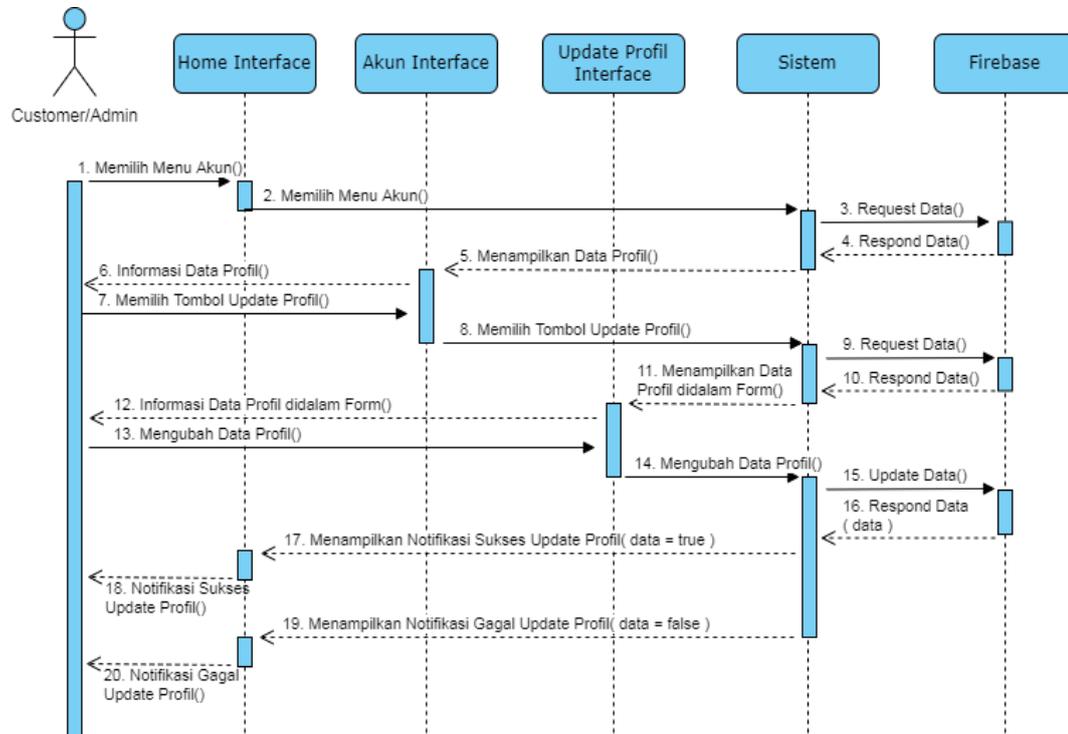
Sequence diagram diatas adalah rangkaian pesan antar objek ketika ke 2 pihak melakukan login. Dimulai saat ke 2 pihak membuka pelaksanaan, ke 2 pihak akan disambut sang sebuah tampilan splash, atau tampilan selamat tiba dimana ketika tampilan ini dibuka, dibalik layar pelaksanaan akan melakukan sebuah aktifitas pengecekan apakah ke 2 pihak pernah masuk sebelumnya pada pelaksanaan ini, proses ini mampu dinamakan menggunakan session. Dalam melakukan pengecekannya sistem melakukan request ke server (pada hal ini dilakukan sang firebase) & server akan memberi response apakah ke 2 pihak telah pernah login atau belum. apabila response yg dikembalikan bernilai true maka ke 2 pihak mampu pribadi memakai pelaksanaan (pelaksanaan menampilkan interface utama), & apabila response nya bernilai false maka ke 2 pihak akan menuju interface login. Lalu ke 2 pihak akan melakukan aktivitas input, melalui inputan ini sistem akan meminta request balik firebase apakah data tadi terdapat, & apabila terdapat maka ke 2 pihak mempunyai hak akses buat memakai pelaksanaan & apabila nir maka ke 2 pihak mengulang balik proses inputan.



Gambar 4. 18: *Sequence Diagram Login*

3. Sequence Diagram Manajemen Profil (Customer dan Admin)

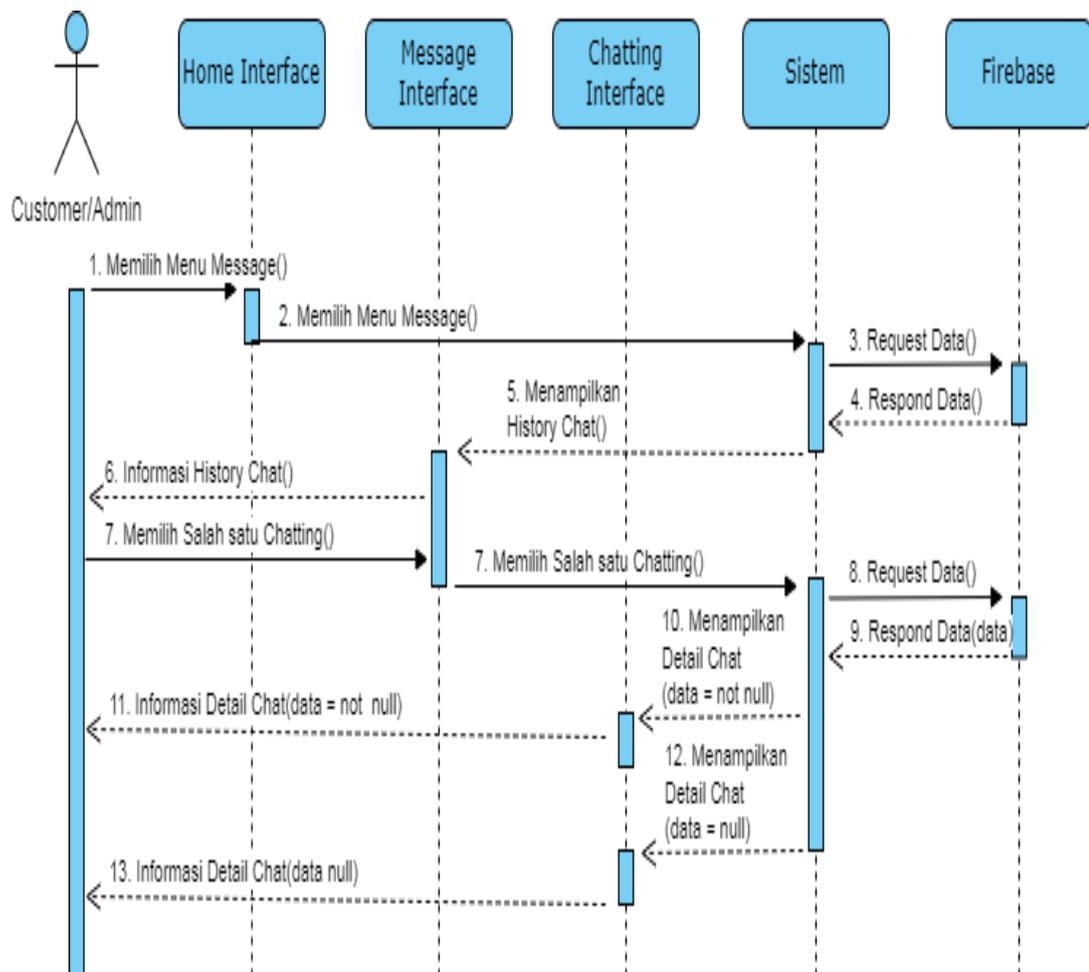
Pada gambar diagram dibawah ini merupakan skenario bagaimana proses mengubah data profil pada manajemen profil. Kedua pihak dimulai dari memilih menu akun pada *home interface*, kemudian sistem akan merespon dan akan meminta data kepada *server* sehingga *firebase* memberikan data kepada sistem, kemudian sistem menampilkan data profil pada akun *interface* sesuai permintaan kedua pihak, selanjutnya kedua pihak memilih tombol *update profile* sehingga sistem kembali meminta data kepada *firebase* kemudian *firebase* memberikan data kembali kepada sistem sehingga sistem menampilkan detail data profil didalam *form update profile interface*. Selanjutnya kedua pihak mengubah data profil yang ingin diubah, setelah selesai diubah, maka sistem akan menyimpan data kepada *firebase* sehingga *firebase* mengirim respond data kepada sistem, kemudian sistem menampilkan *home interface* serta menampilkan notifikasi sukses *update profile*, sehingga pada *home interface* memberikan informasi notifikasi sukses *update profile* kepada kedua pihak



Gambar 4. 19: Sequence Diagram Manajemen Profil

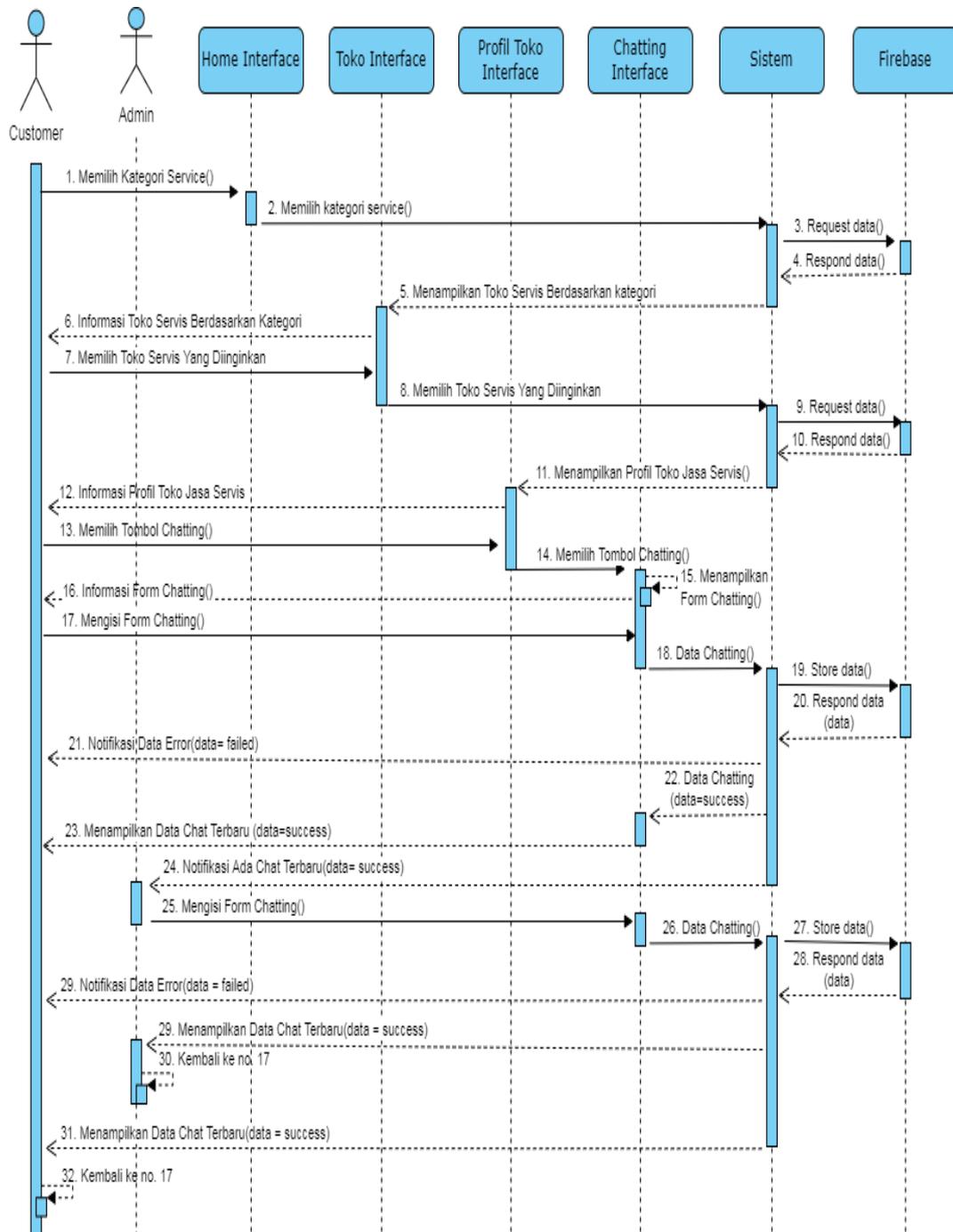
4. Sequence Diagram History Message (Customer dan Admin)

Pada diagram dibawah ini merupakan skenario bagaimana proses mengambil data riwayat pesan pada *History Message*, dimana kedua pihak memilih menu *message* sehingga sistem akan melakukan *request* data kepada *firebase* sehingga *firebase* memberikan *respond* data yang akan dikelola oleh sistem untuk ditampilkan pada *Message interface*, sehingga kedua pihak mendapatkan informasi *history chat*, selanjutnya kedua pihak memilih salah satu histori *chatting* sehingga sistem melakukan permintaan data kepada *firebase*, selanjutnya sistem menerima data dari *firebase* dan sistem menampilkan data *chatting* pada *chatting interface* dan kedua pihak mendapatkan informasi data *chatting*.



Gambar 4. 20: Sequence Diagram History Message

5. Sequence Diagram Chatting (Customer dan Admin)



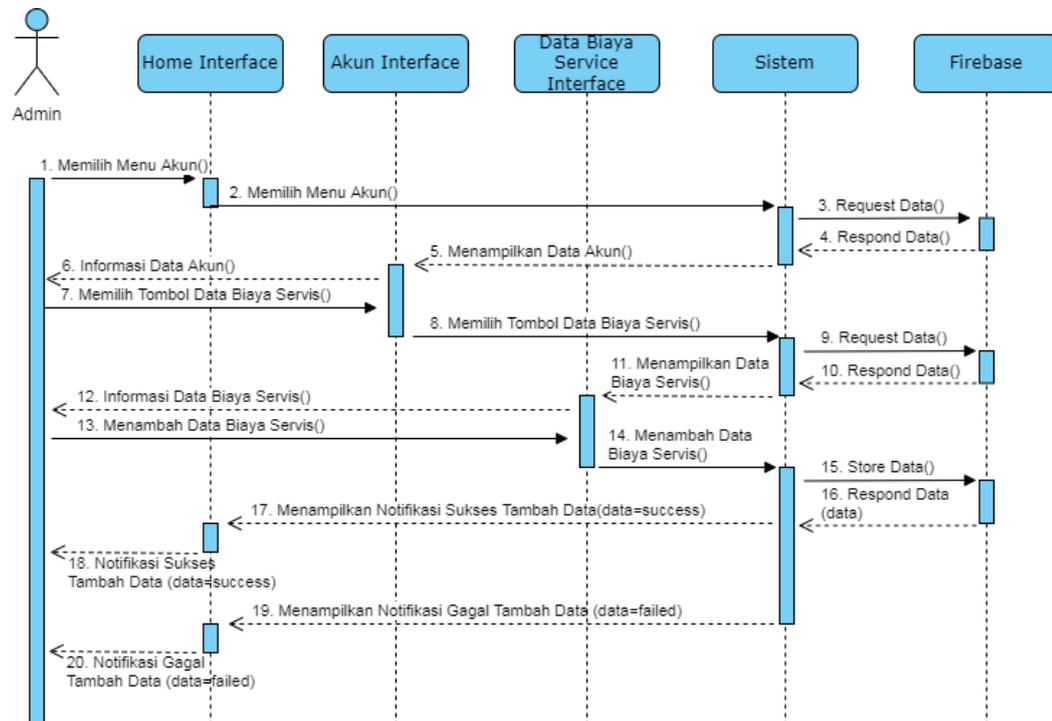
Gambar 4. 21: Sequence Diagram Chatting

Pada diagram diatas ini menjelaskan skenario tentang bagaimana proses *chatting* dalam sistem ini terjadi, mulanya *customer* akan memilih kategori jasa

servis yang diinginkan. Setelah *customer* memilih kategori yang mereka inginkan maka dibelakang layar sistem akan melakukan permintaan data *Admin* berdasarkan kategori yang dipilih oleh *customer* dan jika permintaan berhasil maka akan menampilkan sesuai permintaan sistem yang kemudian data ini diinfokan kedalam *interface* toko, proses selanjutnya *customer* bisa memilih *Admin* yang disukai nya, dan untuk melakukan hal tersebut *customer* melihat detail profil dari toko yang disukainya tadi, kemudian permintaan oleh sistem kembali dilakukan kepada *firebase* untuk menampilkan data detail toko yang ada di database dan menampilkannya kembali kepada *customer* agar bisa mendapatkan informasi toko tersebut. Setelahnya *customer* dapat melakukan *chatting* pada toko tersebut, sehingga diarahkan ke *chatting interface*, maka *customer* mengisi *form chatting*, kemudian sistem kembali melakukan pengiriman data *chatting* ke *firebase* dan memberikan *push notification* ke *Admin* bahwa ada informasi *chatting* terbaru terbaru dari *customer*, kemudian *Admin* melakukan *input* data *chatting* dan sistem kembali melakukan pengiriman data ke *firebase* dan mengirim notifikasi kepada *customer* bahwa ada *chat* terbaru.

6. *Sequence* Diagram Manajemen Biaya Servis

Pada diagram dibawah adalah *sequence diagram* manajemen biaya servis, dimana pihak *Admin* melakukan input data estimasi biaya jasa servis agar *customer* mengetahui estimasi biaya kerusakan pada perangkat mereka, *Admin* melakukan input data dimulai dari memilih menu akun, kemudian sistem akan melakukan *request* data dan akan ditampilkan pada menu akun, kemudian *Admin* memilih tombol data biaya servis, kemudian sistem akan melakukan *request* data kembali sehingga menampilkan data biaya servis yang sudah di input sebelumnya, kemudian untuk melakukan tambah data *Admin* memilih tombol data servis sehingga sistem menampilkan *interface form* data servis sehingga *Admin* mengisi data *form* yang diperlukan kemudian menekan tombol tambah data, selanjutnya sistem akan melakukan *store* data kepada *firebase* sehingga *firebase* mengirim *respond* data kepada sistem, sehingga sistem akan menampilkan keseluruhan data dan data yang baru ditambah.



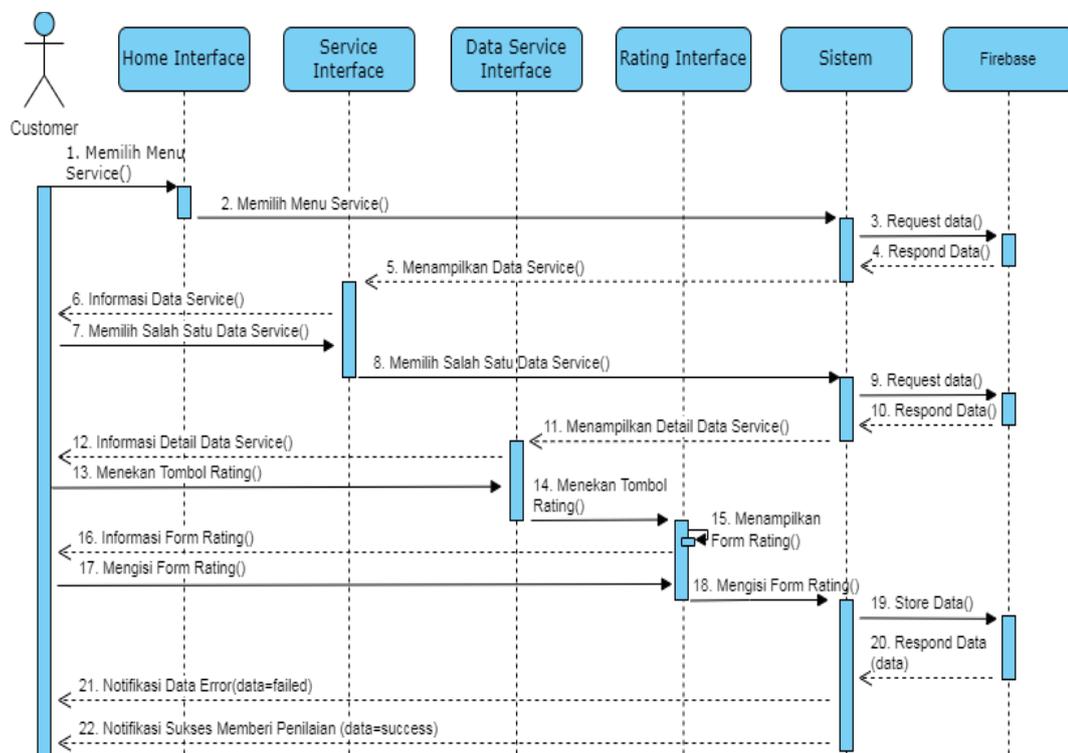
Gambar 4. 22: Sequence Diagram Manajemen Biaya Servis

7. Sequence Diagram Booking Jasa Servis (Customer dan Admin)

Pada diagram dibawah ini menjelaskan skenario tentang bagaimana proses *booking* jasa servis dalam sistem ini terjadi, mulanya *customer* akan memilih kategori jasa servis yang diinginkan, seperti yang telah dibahas sebelumnya bahwa pada penelitian ini akan mencakup dua kategori jasa servis yaitu jasa servis Laptop/PC, dan jasa servis Ponsel. Setelah *customer* memilih kategori yang mereka inginkan maka dibelakang layar sistem akan melakukan permintaan data *Admin* berdasarkan kategori yang dipilih oleh *customer* dan jika permintaan berhasil maka akan menampilkan sesuai permintaan sistem yang kemudian data ini diinfokan kedalam *interface* toko, proses selanjutnya *customer* bisa memilih *Admin* yang disukai nya, dan untuk melakukan hal tersebut *customer* melihat detail profil dari toko yang disukainya tadi, kemudian permintaan oleh sistem kembali dilakukan kepada *server* untuk menampilkan data detail toko ada di database dan menampilkannya kembali kepada *customer* agar bisa mendapatkan informasi toko tersebut. Setelahnya *customer* dapat melakukan servis pada toko

8. Sequence Diagram Rating Toko (Customer)

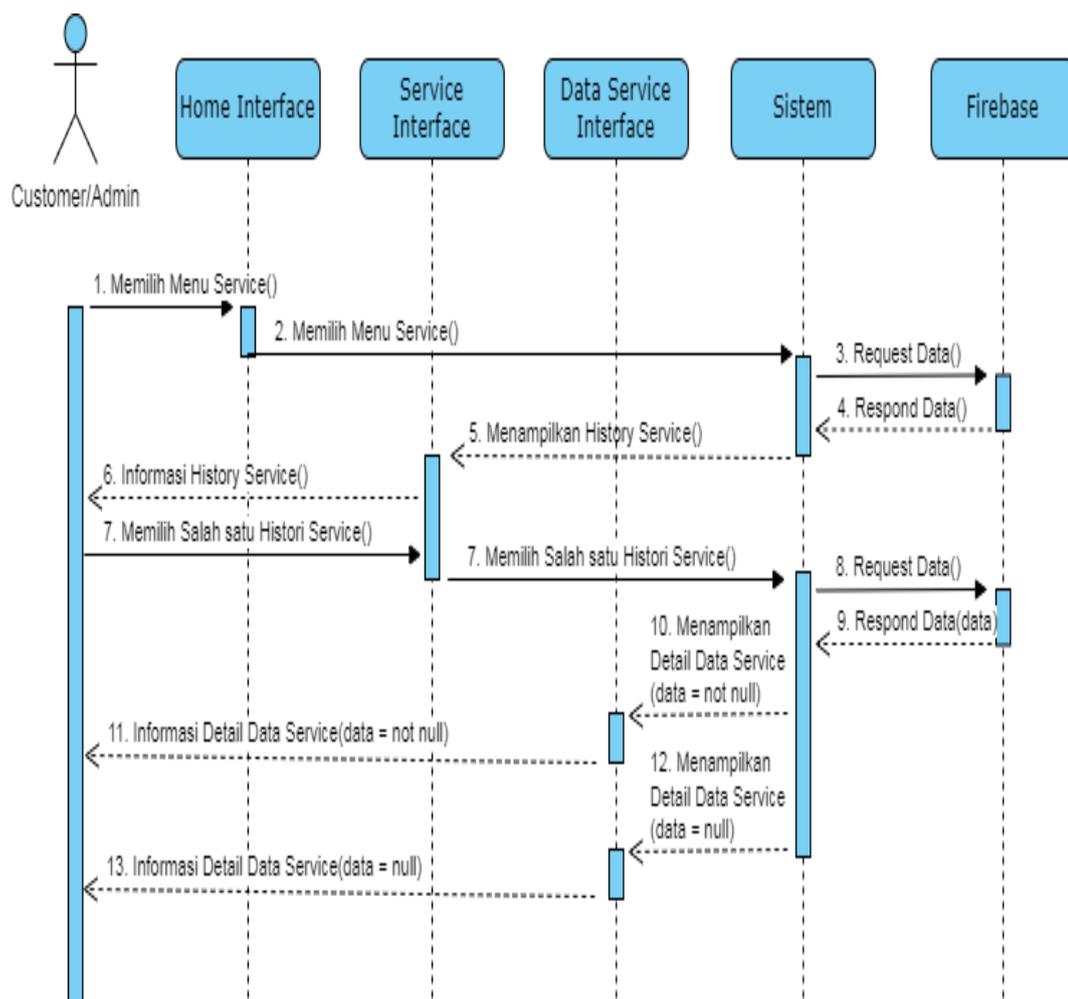
Pada gambar diagram di bawah ini menjelaskan skenario bagaimana proses memberikan *rating* pada *Admin* dalam sistem ini terjadi, dimulai dari *customer* memilih menu *service*, sehingga sistem akan meminta data kepada *server* dan *server* memberikan *respond* data kepada sistem, kemudian sistem menampilkan data servis yang dibutuhkan oleh *customer*, setelah itu *customer* memilih salah satu data servis yang ada pada *service interface*, kemudian sistem kembali meminta data kepada *server* dan *server* kembali merespon data sesuai permintaan sistem, sehingga sistem menampilkan detail data servis kepada *customer*, kemudian *customer* menekan tombol *rating* pada data *service interface* sehingga *customer* diarahkan ke *form rating interface* dan menampilkan *form Rating Interface* kepada *customer*, setelah itu *customer* mengisi *form rating* sesuai penilaian *customer* terhadap pelayanan *Admin* tersebut, kemudian sistem melakukan penyimpanan data kepada *server* serta sistem mengirim notifikasi pemberian *rating* sukses kepada *customer*.



Gambar 4. 24: Sequence Diagram Rating Toko

9. Sequence Diagram History Service (Customer dan Admin)

Pada diagram dibawah ini merupakan skenario bagaimana proses mengambil data riwayat servis pada *History Service*, dimana kedua pihak memilih menu *Service* sehingga sistem akan melakukan *request* data kepada *firebase* sehingga *firebase* memberikan *respond* data yang akan dikelola oleh sistem untuk ditampilkan pada *Service interface*, sehingga kedua pihak mendapatkan informasi *history service*, selanjutnya kedua pihak memilih salah satu histori *service* sehingga sistem melakukan permintaan data kepada *firebase*, selanjutnya sistem menerima data dari *firebase* dan sistem menampilkan data *service* pada data *service interface* dan kedua pihak mendapatkan informasi data *service*.

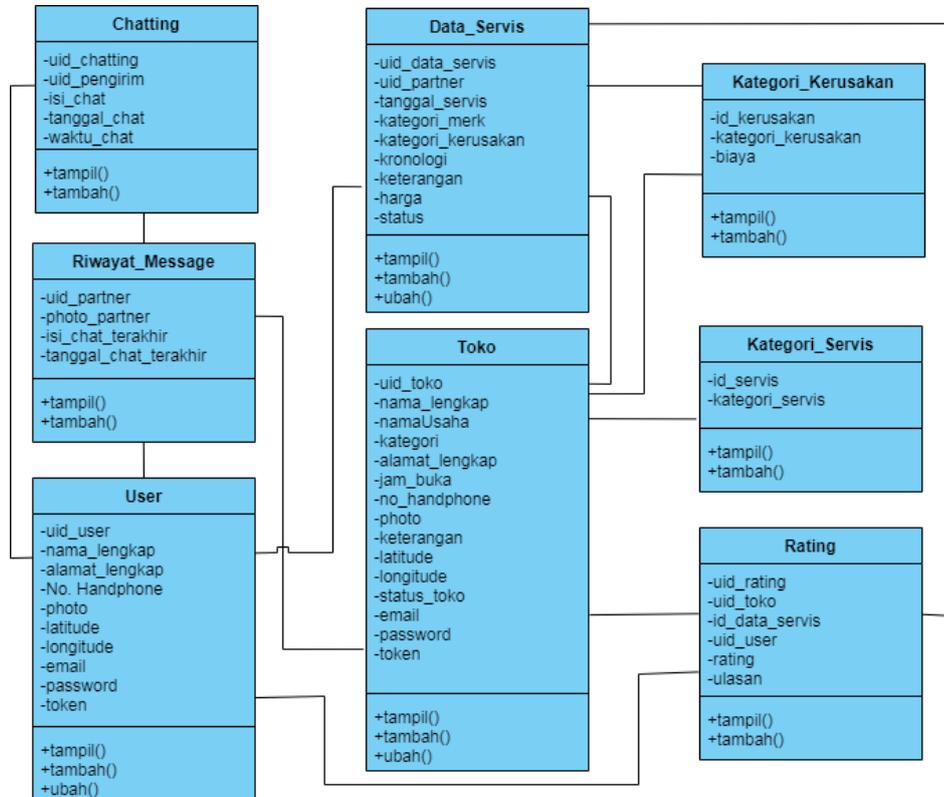


Gambar 4. 25: Sequence Diagram History Service

4.2.1.4. Class Diagram

Class diagram dibawah menjelaskan tentang suatu hubungan antara sebuah objek objek yang akan diimplementasikan ke dalam kode. Menunjuk dari dua aktor dalam usecase yaitu *customer* dan *Admin*, dapat dilihat bahwa *customer* dapat melakukan lebih dari satu aktivitas servis dan lebih dari satu aktivitas melakukan *chatting* kepada setiap toko maupun kepada toko itu kembali, sedangkan *Admin* hanya bisa menunggu data *chatting* dan data servis terlebih dahulu dari *customer*, *Admin* juga mampu menerima lebih dari satu servis dan lebih dari satu *chatting* dari setiap *customer* maupun dengan *customer* yang sama kembali. *Admin* juga dapat melakukan banyak data kategori kerusakan, namun kategori kerusakan hanya bisa diakses berdasarkan toko siapa saja yang melakukan pemanmabahan data.

Berikut ini adalah konsep *class diagram* untuk aplikasi yang dibangun :



Gambar 4. 26: Class Diagram

4.2.2. Design Database

Langkah selanjutnya adalah merancang *database* berdasarkan data dan informasi yang diperoleh. Pada tahap ini nantinya Anda dapat memutuskan entitas mana yang akan menjadi tabel DBMS. Basis data tipe NoSql (*Firestore real-time* dalam kasus ini) dalam format JSON (*Javascript Object Notation*).

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang sangat ringan dan mudah dibaca dan ditulis manusia, serta mudah diubah dan dihasilkan (*generate*) oleh komputer. Kumpulan istilah dan dokumen dikenal sebagai *NoSql*, dan istilah ini sesuai dengan tabel dan baris dalam *database* SQL. Untuk memudahkan dalam membaca dalam penelitian ini, penulis memberikan gambaran tentang *database* yang dirancang dalam format tabel kemudian mengimplementasikan desain tersebut dalam format JSON.

4.2.2.1. Tabel User

Table 4. 5: Tabel User

<i>Field</i>	<i>Type Data</i>	Deskripsi
Uid_user	String	<i>Unique ID</i> serta dan <i>Primary Key</i>
Nama_lengkap	String	Nama lengkap <i>user</i>
Alamat_lengkap	String	Alamat lengkap <i>user</i>
No_Handphone	String	Nomor ponsel <i>user</i>
Photo	Image	Foto profil <i>user</i> yang di convert ke base64
Latitude	Long	Garis lintang <i>user</i> di maps
Longitude	Long	Garis bujur <i>user</i> di maps
Email	String	Email <i>user</i>
password	String	Password <i>user</i>
Token	String	<i>Id token</i> untuk <i>cloud Messaging</i>

4.2.2.2. Tabel Toko

Table 4. 6: Tabel Toko

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Uid_toko	String	<i>Unique ID dan Primary Key</i>
Nama_lengkap	String	Nama lengkap pemilik toko
Nama_Usaha	String	Nama Toko
Kategori	String	Kategori jasa servis toko
Alamat_lengkap	String	Alamat lengkap
Jam_buka	String	Jam buka dan tutup toko
No_Handphone	String	Nomor hanphone toko
Photo	Image	Foto profil toko yang di convert ke base64
Keterangan	String	Keterangan tentang toko
Latitude	String	Garis lintang toko di maps
Longitude	String	Garis bujur toko di maps
Status_Toko	String	Status toko sudah buka atau belum
Email	String	Email toko
password	String	Password toko
Token	String	<i>ID token untuk Cloud Messaging</i>

4.2.2.3. Tabel Kategori Servis

Table 4. 7: Tabel Kategori Servis

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Kategori_Servis	String	Kategori servis
Id_servis	String	<i>Primay Key</i>

4.2.2.4. Tabel Kategori Kerusakan

Table 4. 8: Tabel Kategori Kerusakan

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Id_kerusakan	String	<i>Primary Key</i>
Gejala_kerusakan	String	Gejala yang biasa terjadi berdasarkan kerusakan
Biaya	Long	Biaya perbaikan kerusakan

4.2.2.5. Tabel Rating

Table 4. 9: Tabel Rating

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Uid_rating	String	<i>Unique ID dan Primary Key</i>
Uid_toko	String	Nama partner
Uid_user	Date	Tanggal terakhir pengiriman chat
Id_data_servis	String	<i>Id data servis</i>
rating	Long	Data proses yang sedang terjadi
Ulasan	String	Ulasan <i>user</i> terhadap pelayanan toko

4.2.2.6. Tabel Riwayat Message

Table 4. 10: Table Riwayat Message

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Isi_chat_terakhir	String	Chat terakhir yang dikirim
Tanggal_chat_terakhir	Date	Tanggal terakhir pengiriman chat
Uid_partner	String	UID dari admin
Photo_partner	Image	Foto profil <i>Partner</i>

4.2.2.7. Tabel Chatting

Table 4. 11: Tabel Chatting

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Isi_chat	String	Isi <i>chat</i> dari pengirim
Tanggal_chat	Date	Tanggal terakhir pengiriman <i>chat</i>
Waktu_chat	Time	Pukul terakhir pengiriman <i>chat</i>
Uid_pengirim	String	UID dari pengirim pesan
Uid_chatting	String	<i>Unique ID</i> dan <i>Primary Key</i>

4.2.2.8. Tabel Data Servis

Table 4. 12: Tabel Data Servis

<i>Field</i>	<i>Type Data</i>	<i>Deskripsi</i>
Uid_data_servis	String	<i>Unique ID dan Primary Key</i>
Tanggal_servis	String	Tanggal melakukan servis
Kategori_merk	Date	Merk barang elektronik
Kategori_kerusakan	String	Perkiraan kerusakan elektronik
Kronologi	String	Kronologi sebelum barang rusak
Keterangan	String	Keterangan tambahan
Harga	Long	Total biaya servis
Status	String	Proses berjalannya servis elektronik
Uid_partner	String	UID yang didapat dari mitra

4.2.3. Design Interface

Subbab ini menjelaskan bagaimana antarmuka diimplementasikan dalam sistem. Desain antarmuka (*application interfaces*) dibuat menggunakan *wireframes*. *Wireframes* adalah garis besar atau coretan kasar yang digunakan untuk menyusun elemen situs web atau halaman seluler sebelum proses desain yang sebenarnya dimulai.

Saat mengembangkan aplikasi berbasis Android, ada beberapa komponen kunci yang disebut view atau widget, beberapa di antaranya adalah:

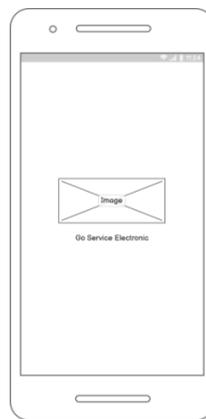
Table 4. 13: Macam-macam Widget Android

Nama Widget	Fungsi
<i>TextView</i>	Tampilan atau <i>widget</i> untuk menampilkan pengujian seperti kata, frasa, dan paragraf yang digunakan dalam 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 <i>list</i> yang dapat di ubah sesuai kebutuhan
<i>FrameLayout</i>	Merupakan suatu komponen yang saling menimpah misalkan memasukkan komponen tombol ke dalam komponen gambar

1. Design Wireframe Customer

a. Rancangan Halaman *Splash Screen*

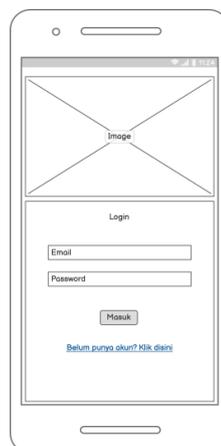
Pada gambar diatas adalah rancangan halaman pertama kali saat aplikasi dibuka atau biasa disebut *Splash Screen*, dimana pada rancangan halaman ini nantinya sistem akan melakukan pengecekan apakah sudah pernah *login* sebelumnya pada aplikasi, apabila sudah pernah, maka sistem akan mengalihkan ke rancangan halaman *Home*, jika belum *login* maka sistem akan mengalihkannya ke halaman *login*.



Gambar 4. 27: Rancangan Halaman *Splash Screen*

b. Rancangan Halaman *Login*

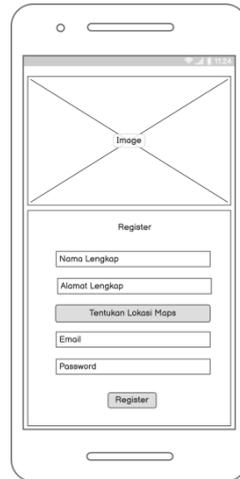
Pada gambar dibawah adalah rancangan halaman *Login* pada bagian *customer*, dimana *customer* nantinya akan melakukan input *email* dan *password* yang sudah terdaftar sebelumnya, kemudian menekan tombol masuk.



Gambar 4. 28: Rancangan Halaman *Login*

c. Rancangan Halaman *Register*

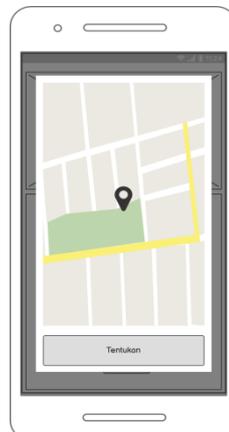
Pada gambar dibawah adalah rancangan halaman *Register* pada *customer*, dimana *customer* akan mengisi *form* nama lengkap, alamat lengkap, *email*, *password*, serta menentukan titik lokasi rumah *customer* untuk mempermudah pencarian toko servis terdekat.

A wireframe of a smartphone screen showing a registration form. At the top, there is a placeholder for a profile picture labeled 'Image'. Below it, the form is titled 'Register' and contains several input fields: 'Nama Lengkap', 'Alamat Lengkap', 'Email', and 'Password'. There is a button labeled 'Tentukan Lokasi Maps' and a final 'Register' button at the bottom.

Gambar 4. 29: Rancangan Halaman *Register*

d. Rancangan Halaman *Location Point*

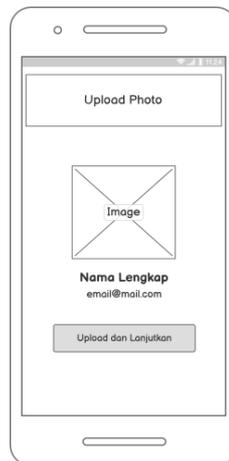
Pada gambar dibawah adalah rancangan halaman *Location Point* pada kedua pihak, dimana rancangan halaman ini untuk menentukan titik lokasi kedua pihak pada *google maps*, serta untuk menghitung jarak lokasi antara *customer* dengan *Admin*, untuk mendapatkan titik kordinat *latitude* dan *longitude*, kedua pihak diwajibkan menyalakan fitur lokasi yang ada pada *smartphone*.



Gambar 4. 30: Rancangan Halaman *Location Point*

e. Rancangan Halaman *Upload Photo*

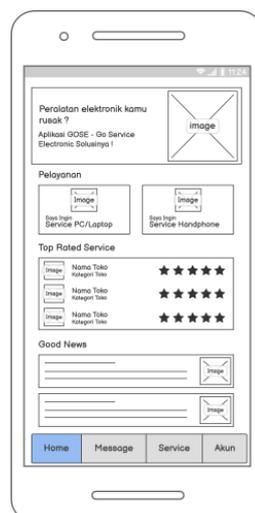
Pada gambar dibawah adalah rancangan halaman *Upload photo* pada *customer* dan *Admin* dimana pada bagian ini kedua pihak melakukan penginputan gambar kedalam aplikasi untuk menjadi foto profil akun.



Gambar 4. 31: Rancangan Halaman *Upload Photo*

f. Rancangan Halaman Menu *Home*

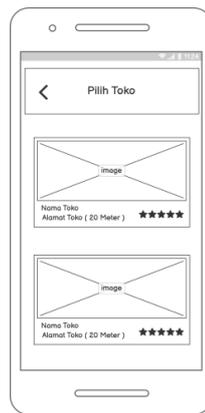
Pada gambar dibawah adalah rancangan halaman Menu *Home* pada aplikasi *customer* dimana pada bagian ini sistem akan menampilkan kategori pelayanan servis yang tersedia yaitu servis *PC/Laptop* dan servis *Handphone*, kemudian sistem juga akan menampilkan tiga nilai *rating* tertinggi pada toko jasa servis tersebut.



Gambar 4. 32: Rancangan Halaman Menu *Home*

g. Rancangan Halaman Pilih Toko Servis

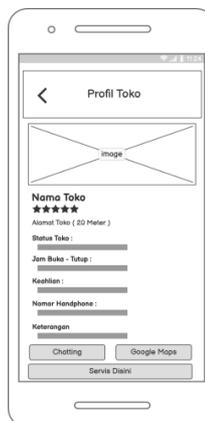
Pada gambar dibawah adalah rancangan halaman pilih toko servis pada aplikasi *customer*, pada bagian ini sistem akan menampilkan beberapa toko jasa servis berdasarkan kategori servis yang telah dipilih sebelumnya pada halaman *home*, serta sistem akan mengurutkan toko berdasarkan jarak terdekat toko jasa servis dengan *customer*



Gambar 4. 33: Rancangan Halaman Pilih Toko Servis

h. Rancangan Halaman Profil Toko Jasa Servis

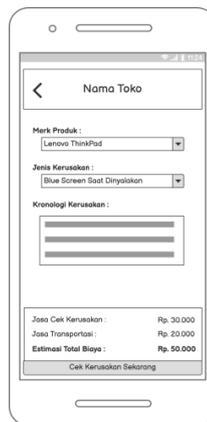
Pada gambar dibawah adalah rancangan halaman profil toko jasa servis, dimana pada bagian ini sistem akan menampilkan profil toko secara detail mulai dari gambar toko, *rating* pada toko, status toko, serta informasi lainnya. Pada bagian ini juga menampilkan tombol *chatting* untuk bisa melakukan konsultasi dengan pihak toko, serta tombol *google Maps* untuk melihat lokasi toko.



Gambar 4. 34: Rancangan Halaman Profil Toko Jasa Servis

i. Rancangan Halaman Cek Estimasi Biaya

Pada gambar dibawah adalah rancangan halaman cek estimasi biaya pada aplikasi *customer*, dimana pada bagian ini *customer* akan melakukan input data perangkat elektronik yang ingin di servis, serta kronologi kerusakan yang terjadi pada perangkat elektronik tersebut, sistem juga akan menampilkan estimasi biaya pengecekan perangkat elektronik serta biaya jasa transportasi antar jemput perangkat.



Gambar 4. 35: Rancangan Halaman Cek Estimasi Biaya

j. Rancangan Halaman Konfirmasi Servis

Pada gambar dibawah adalah rancangan halaman Konfirmasi Servis pada aplikasi *customer*, dimana rancangan halaman ini akan muncul setelah pihak toko jsasa servis selesai melakukan pengecekan. Pada bagian ini nantinya sistem akan menampilkan data perangkat elektronik serta total keseluruhan biaya perbaikan perangkat.



Gambar 4. 36: Rancangan Halaman Konfirmasi Servis

k. Rancangan Halaman Perbaikan Telah Selesai

Pada gambar dibawah adalah rancangan halaman perbaikan telah selesai pada aplikasi *customer*, dimana sistem akan memberikan informasi perangkat elektronik, total biaya keseluruhan, serta informasi bahwa perbaikan telah selesai dilakukan. Pada bagian ini juga memiliki tombol Beri Penilaian untuk masuk ke rancangan halaman Penilaian *rating* toko.



Gambar 4. 37: Rancangan Halaman Perbaikan Telah Selesai

l. Rancangan Halaman *Input Rating*

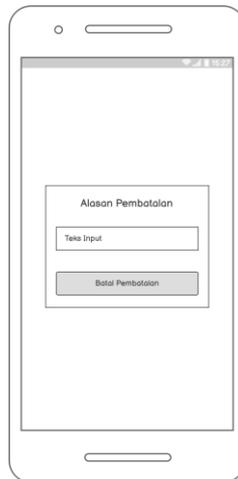
Pada gambar dibawah adalah rancangan halaman *input rating* pada aplikasi *customer*, dimana pada bagian ini *customer* akan memberikan penilaian terhadap toko jasa servis atas pelayanan yang telah diberikan dari pihak toko tersebut. Penilaian ini berdasarkan bintang yang telah disediakan sebanyak lima, dimana nilai tertinggi penilain yaitu lima bintang.



Gambar 4. 38: Rancangan Halaman *Input Rating*

m. Rancangan Halaman Pembatalan Servis

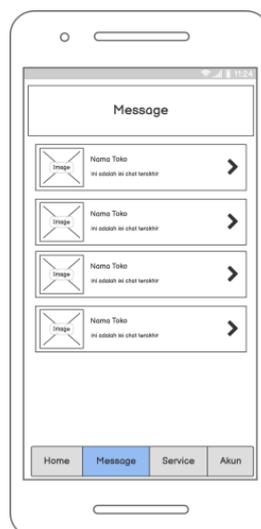
Pada gambar dibawah adalah rancangan halaman Pembatalan Servis pada aplikasi *customer*, dimana pada bagian ini system akan meminta alasan pembatalan servis, sehingga *customer* harus melakukan *input* data alasan pembatalan dan menekan tombol pembatalan servis.



Gambar 4. 39: Rancangan Halaman Pembatalan Servis

n. Rancangan Halaman Menu Message

Pada gambar dibawah adalah rancangan halaman menu *message* pada aplikasi *customer*, dimana pada bagian ini sistem akan menampilkan seluruh data riwayat pesan terakhir yang pernah dilakukan oleh *customer* kepada setiap toko.



Gambar 4. 40: Rancangan Halaman Menu Message

o. Rancangan Halaman *Chatting*

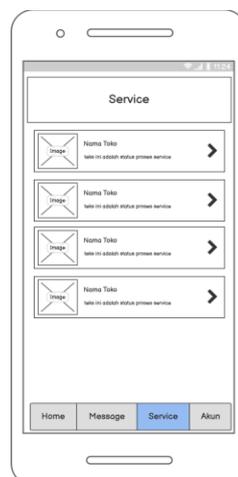
Pada gambar dibawah adalah rancangan halaman *chatting* pada aplikasi *customer*, dimana pada bagian ini *customer* dan toko jasa servis saling berinteraksi melalui pengiriman pesan antara kedua pihak. Pada bagian ini *customer* wajib melakukan pengiriman pesan terlebih dahulu kepada pihak toko jasa servis agar pihak toko jasa servis dapat melihat pesan masuk dari *customer*.



Gambar 4. 41: Rancangan Halaman *Chatting*

p. Rancangan Halaman Menu *Service*

Pada gambar dibawah adalah rancangan halaman menu *service* pada aplikasi *customer*, dimana pada bagian ini sistem akan menampilkan seluruh informasi data riwayat servis yang sudah pernah dilakukan oleh *customer* kepada beberapa toko maupun kepada toko yang sama kembali.



Gambar 4. 42: Rancangan Halaman Menu *Service*

q. Rancangan Halaman Menu Akun

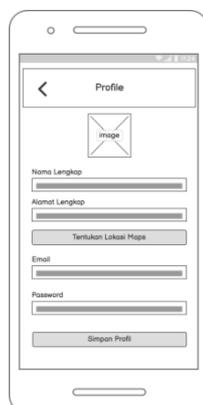
Pada gambar dibawah adalah rancangan halaman menu akun pada aplikasi *customer*, dimana pada bagian ini sistem akan menampilkan foto profil *customer*, nama lengkap, dan *email customer*, pada bagian ini sistem juga menyediakan tombol *update* profil, melihat titik lokasi terakhir yang di inputkan *customer* pada *google maps*, serta tombol *logout* untuk menghapus seluruh data *login* yang sudah dilakukan oleh *customer* sebelumnya.



Gambar 4. 43: Rancangan Halaman Menu Akun

r. Rancangan Halaman Update Profil

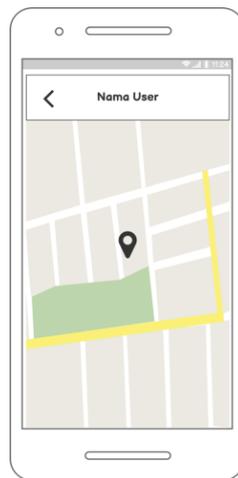
Pada gambar dibawah adalah rancangan halaman *update* profil pada aplikasi *customer*, dimana sistem akan menampilkan data *customer* yang terdahulu kedalam *form* input, kemudian *customer* melakukan perubahan data yang ingin diubah, seperti foto profil, nama lengkap, titik lokasi pada *google Maps*, serta dapat melakukan perubahan *password login*.



Gambar 4. 44: Rancangan Halaman Update Profil

s. Rancangan Halaman Lihat Titik Lokasi Pada *Google Maps*

Pada gambar dibawah adalah rancangan halaman lihat titik lokasi pada *google maps*, dimana sistem akan menampilkan *maps* yang telah disediakan oleh pihak *google*, kemudian sistem akan menentukan titik lokasi *customer* berdasarkan *latitude* dan *longitude* yang telah di *input* oleh *customer* saat melakukan *registrasi* maupun saat *update* data profil.

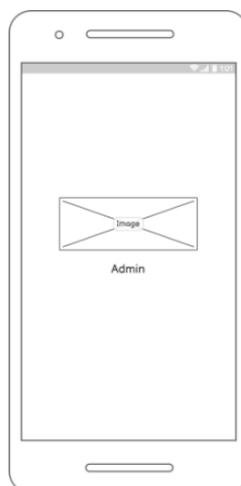


Gambar 4. 45: Rancangan Halaman Lihat Titik Lokasi Pada *Google Maps*

2. Design Wireframe Admin

a. Rancangan Halaman *Splash Screen*

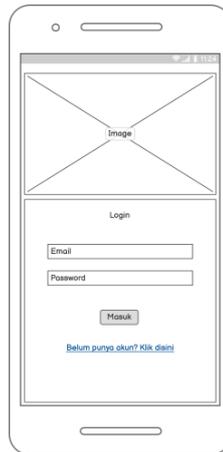
Pada gambar dibawah adalah rancangan halaman *splash screen* pada bagian aplikasi *admin*.



Gambar 4. 46: Rancangan Halaman *Splash Screen*

b. Rancangan Halaman *Login*

Pada gambar dibawah adalah rancangan halaman *login* pada aplikasi *admin*, dimana *admin* nantinya akan melakukan *input email* dan *password* yang sudah terdaftar di aplikasi ini sebelumnya, kemudian menekan tombol masuk.



Gambar 4. 47: Rancangan Halaman *Login*

c. Rancangan Halaman *Registrasi*

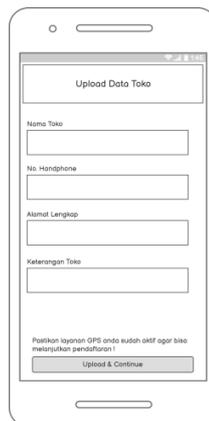
Pada gambar dibawah adalah rancangan halaman *registrasi* pada aplikasi *admin*, dimana *admin* nantinya akan memilih kategori servis yang akan digunakan, kemudian melakukan *input email* dan *password* untuk akun *login* nantinya. Setelah melakukan *input data* kemudian *admin* menekan tombol *registrasi*



Gambar 4. 48: Rancangan Halaman *Registrasi*

d. Rancangan Halaman *Upload Data Toko*

Pada gambar dibawah adalah rancangan halaman upload data toko pada aplikasi admin, dimana *admin* akan melakukan *input* data nama toko, nomor ponsel toko, alamat lengkap toko, serta data toko lainnya guna untuk memberikan informasi tentang toko kepada para *customer*. Setelah melakukan *input* data, maka *admin* menekan tombol *upload* dan *continue*.

A mobile application screen titled "Upload Data Toko". It features four input fields: "Nama Toko", "No. Handphone", "Alamat Lengkap", and "Keterangan Toko". Below the fields is a small text prompt: "Pastikan layanan GPS anda sudah aktif agar bisa melanjutkan pendaftaran!". At the bottom, there is a button labeled "Upload & Continue".

Gambar 4. 49: Rancangan Halaman *Upload Data Toko*

e. Rancangan Halaman *Maps Location Coordinate*

Pada gambar dibawah adalah rancangan halaman *maps location coordinate* pada aplikasi *admin*, dimana pada bagian ini *admin* menentukan titik kordinasi lokasi toko berada untuk mempermudah *costumer* saat ingin dating ke lokasi toko *admin* tersebut.



Gambar 4. 50: Rancangan Halaman *Maps Location Coordinate*

f. Rancangan Halaman *Upload Logo Toko*

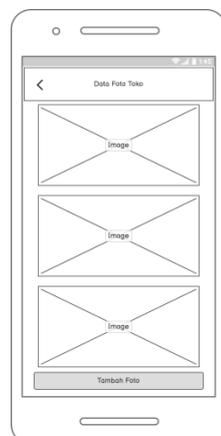
Pada gambar dibawah ini adalah rancangan halaman *upload* logo toko pada aplikasi *admin*, dimana nantinya *admin* akan melakukan *input* data foto toko maupun logo toko. Setelah melakukan *upload* foto, maka *admin* menekan tombol *upload* dan *continue*.



Gambar 4. 51: Rancangan Halaman Upload Logo Toko

g. Rancangan Halaman Data Foto Toko

Pada gambar dibawah ini adalah rancangan halaman data foto toko pada aplikasi *admin*, dimana nantinya *admin* akan melakukan *input* data foto toko untuk ditampilkan pada profil toko pada aplikasi *customer*. Data foto toko ini bias menjadi solusi untuk menampilkan dokumentasi perbaikan pada peralatan elektronik.



Gambar 4. 52: Rancangan Halaman Data Foto Toko

h. Rancangan Halaman Jadwal Toko Buka

Pada gambar dibawah adalah rancangan halaman jadwal toko buka pada aplikasi *admin*, dimana *admin* akan menentukan jadwal buka pada toko dalam 7 hari. Data ini nantinya akan ditampilkan pada profil toko aplikasi *customer*.



Gambar 4. 53: Rancangan Halaman Jadwal Toko Buka

i. Rancangan Halaman Data Estimasi Jasa Servis

Pada gambar dibawah adalah rancangan halaman data estimasi jasa servis pada aplikasi *admin*, dimana *admin* akan melakukan *input* data estimasi jasa servis untuk ditampilkan pada aplikasi *customer* guna agar *customer* mengetahui estimasi biaya perbaikan pada kerusakan perangkat elektronik



Gambar 4. 54: Rancangan Halaman Data Estimasi Jasa Servis

j. Rancangan Halaman Konfirmasi Pengambilan Barang

Pada gambar dibawah adalah rancangan halaman konfirmasi pengambilan barang pada aplikasi *admin*, dimana *admin* akan menekan konfirmasi pengambilan barang agar *customer* mengetahui bahwa perangkat elektronik akan diproses, pada rancangan halaman ini *admin* dapat menekan informasi *customer* untuk melihat lokasi *maps customer*.



Gambar 4. 55: Rancangan Halaman Konfirmasi Pengambilan Barang

k. Rancangan Halaman Konfirmasi Barang Telah Tiba

Pada gambar dibawah adalah rancangan halaman konfirmasi barang telah tiba pada aplikasi *admin*, dimana nantinya saat perangkat elektronik rusak milik *customer* telah tiba di lokasi toko *admin*, maka *admin* akan menekan tombol konfirmasi barang telah tiba.



Gambar 4. 56: Rancangan Halaman Konfirmasi Barang Telah Tiba

l. Rancangan Halaman Konfirmasi Kerusakan Barang

Pada gambar dibawah adalah rancangan halaman konfirmasi kerusakan barang pada aplikasi *admin*, dimana pada bagian ini nantinya *admin* akan melakukan pengecekan barang dan akan melakukan *input* data kerusakan barang berdasarkan kerusakan pada perangkat elektronik *customer*.



Gambar 4. 57: Rancangan Halaman Konfirmasi Kerusakan Barang

m. Rancangan Halaman Konfirmasi Perbaikan Selesai

Pada gambar dibawah adalah rancangan halaman konfirmasi perbaikan selesai pada aplikasi *admin*, dimana rancangan halaman ini akan muncul setelah *customer* melakukan konfirmasi lanjut perbaikan perangkat elektronik dan admin akan melakukan konfirmasi perbaikan telah selesai apabila perangkat telah selesai diperbaiki.



Gambar 4. 58: Rancangan Halaman Konfirmasi Perbaikan Selesai

n. Rancangan Halaman Pembatalan Servis

Pada gambar dibawah adalah rancangan halaman pembatalan servis pada aplikasi *admin*, dimana pada bagian ini sistem hanya akan menampilkan data biaya jasa cek kerusakan barang serta biaya transportasi pengambilan dan pengembalian perangkat elektronik *customer*.



Gambar 4. 59: Rancangan Halaman Pembatalan Servis

o. Rancangan Halaman Melakukan Pengembalian Barang

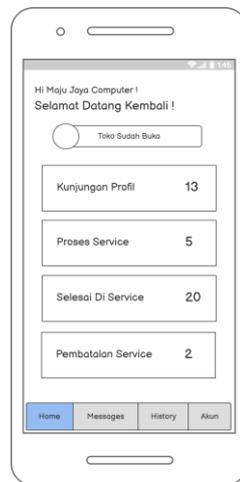
Pada gambar dibawah adalah rancangan halaman melakukan pengembalian barang pada aplikasi *admin*, dimana sistem akan menampilkan data biaya kerusakan barang dan biaya jasa transportasi pengambilan maupun pengembalian perangkat elektronik *customer*.



Gambar 4. 60: Rancangan Halaman Melakukan Pengembalian Barang

p. Rancangan Halaman Utama

Pada gambar dibawah adalah rancangan halaman utama pada aplikasi *admin*, dimana pada bagian ini sistem akan menampilkan data jumlah kunjungan profil pada toko, jumlah proses perbaikan, jumlah perbaikan selesai, jumlah perbaikan yang dibatalkan, rating toko tertinggi serta memiliki tombol status toko sedang buka atau sudah tutup.



Gambar 4. 61: Rancangan Halaman Utama

q. Rancangan Halaman Detail Ulasan

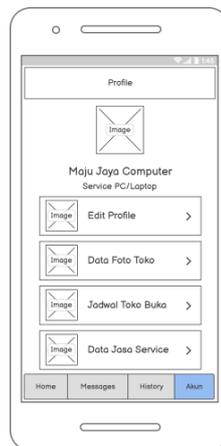
Pada gambar dibawah adalah rancangan halaman detail ulasan pada aplikasi *admin*, dimana *admin* dapat melihat seluruh ulasan yang diberikan oleh *customer* terhadap pelayanan pada toko *admin* tersebut secara detail.



Gambar 4. 62: Rancangan Halaman Detail Ulasan

r. Rancangan Halaman Halaman Akun

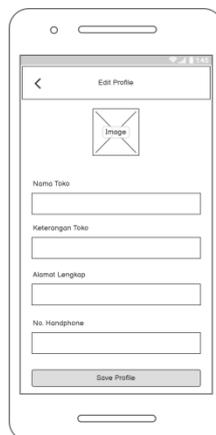
Pada gambar dibawah adalah rancangan halaman halaman akun pada aplikasi *admin*, dimana pada bagian ini akan menampilkan data nama toko, kategori toko, serta foto profil toko, adapun tombol yang tersedia pada gambar ini salah satunya yaitu *edit* profil, data foto toko, data jadwal toko serta tombol lainnya.



Gambar 4. 63: Rancangan Halaman Halaman Akun

s. Rancangan Halaman *Edit Profile*

Pada gambar dibawah adalah rancangan halaman *edit profile* data toko pada aplikasi *admin*, dimana bagian ini untuk melakukan perubahan data apabila ada data yang sudah tidak sesuai, adapaun data yang bias diubah adalah nama toko, alamat toko, nomor ponsel serta *input* data lainnya.



Gambar 4. 64: Rancangan Halaman *Edit Profile*

4.3. Implementasi dan Pengujian Sistem

Setelah peneliti melakukan analisis dan perancangan sistem sebelumnya, maka peneliti melakukan tahap implementasi dari perancangan menjadi sebuah sistem android menggunakan *framework react native* pada aplikasi *go service electronic*.

4.3.1. Implementasi *React Native*

Pada sub-bab ini akan dibahas tentang bagaimana membangun aplikasi android menggunakan *framework react native*. *React native* menyediakan dua metode lingkungan pengembangan, Jika Anda baru mengenal pengembangan seluler, cara termudah untuk memulai adalah dengan *Expo CLI*.

Expo adalah seperangkat alat yang dibangun di sekitar *React Native* dan, meskipun memiliki banyak fitur, fitur yang paling relevan bagi kami saat ini adalah dapat membuat Anda menulis aplikasi *React Native* dalam hitungan menit. Anda hanya memerlukan *Node.js* versi terbaru dan ponsel atau *emulator*. Jika Anda ingin mencoba *React Native* langsung di browser web Anda sebelum menginstal alat apa pun, Anda dapat mencoba *Snack*.

Jika Anda sudah terbiasa dengan pengembangan seluler. Anda mungkin ingin menggunakan *React Native CLI*. Ini membutuhkan *Xcode* atau *Android Studio* untuk memulai. Jika Anda sudah menginstal salah satu dari alat ini, Anda akan dapat memulai dan menjalankannya dalam beberapa menit. Jika tidak diinstal, Anda akan menghabiskan waktu sekitar satu jam untuk menginstal dan mengonfigurasinya.

Pada kali ini penulis menggunakan metode pengembangan aplikasi *React Native CLI* untuk membangun aplikasi berbasis *android*. Anda memerlukan *Node*, antarmuka baris perintah *React Native*, *JDK*, dan *Android Studio*. Meskipun Anda dapat menggunakan *editor* pilihan anda untuk mengembangkan aplikasi. Anda perlu menginstal *Android Studio* untuk menyiapkan alat yang diperlukan untuk membangun *aplikasi React Native* Anda untuk *Android*.

Kami merekomendasikan untuk menginstal *Node* via *Chocolatey*, manajer paket populer untuk *Windows*. Kemudian buka *Command Prompt Administrator*

(klik kanan *Command Prompt* dan pilih "*Run as Administrator*"), lalu jalankan perintah berikut:

```
choco install -y nodejs-lts openjdk11
```

1. *Install Android Studio*

Unduh dan *install Android Studio*. Saat berada di *wizard* penginstalan *Android Studio*, pastikan kotak di samping semua item berikut dicentang yaitu *Android SDK*, *Android SDK Platform*, *Android Virtual Device*. Kemudian, klik "*Next*" untuk menginstal semua komponen ini.

2. *Install Android SDK*

Android Studio menginstal *Android SDK* terbaru secara *default*. Membangun aplikasi *React Native* dengan *native code*, bagaimanapun, membutuhkan *Android 10 (Q) SDK* secara khusus. *SDK Android* tambahan dapat diinstal melalui *SDK Manager* di *Android Studio*. Untuk melakukannya, buka *Android Studio*, klik tombol "Konfigurasi" dan pilih "*SDK Manager*".



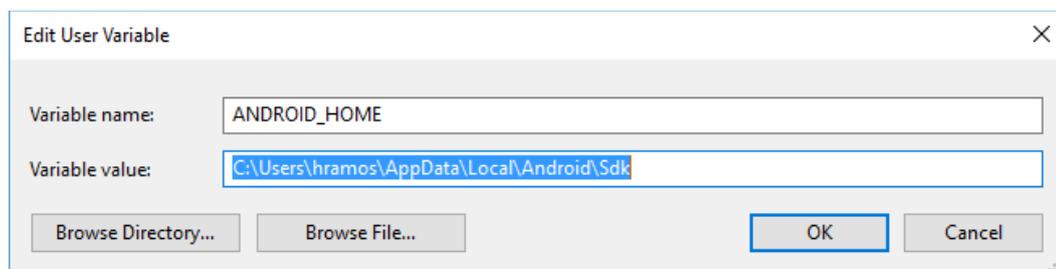
Gambar 4. 65: Tampilan Android Studio

Pilih tab "*SDK Platforms*" dari dalam *SDK Manager*, lalu centang kotak di samping "*Show Package Details*" di sudut kanan bawah. Cari dan perluas *Android 10 (Q)* entri, lalu pastikan item berikut dicentang yaitu *Android SDK Platform 29*, *Intel x86 Atom_64 System Image* atau *Google APIs Intel x86 Atom System Image*. Selanjutnya, pilih tab "*SDK Tools*" dan centang kotak di samping "*Show Package Details*" di sini juga. Cari dan perluas entri "*Android SDK Build-Tools*", lalu pastikan itu 29.0.2 dipilih. Terakhir, klik "Terapkan" untuk mengunduh dan menginstal *SDK Android* dan alat pembangunan terkait.

3. Konfigurasi Variabel Lingkungan *ANDROID_HOME*

React Native memerlukan beberapa variabel lingkungan yang harus disiapkan untuk membangun aplikasi dengan *native code*.

- a. Buka Panel Kontrol *Windows*.
- b. Klik Akun Pengguna, lalu klik Akun Pengguna lagi
- c. Klik Ubah variabel lingkungan saya
- d. Klik Baru... untuk membuat *ANDROID_HOME* variabel pengguna baru yang mengarah ke jalur ke *Android SDK* Anda:



Gambar 4. 66: Tampilan Edit User Variable

4. Tambahkan Alat *Platform* ke *Path*

- a. Buka Panel Kontrol *Windows*.
- b. Klik Akun Pengguna, lalu klik Akun Pengguna lagi
- c. Klik Ubah variabel lingkungan saya
- d. Pilih variabel *Path* .
- e. Klik Sunting.
- f. Klik Baru dan tambahkan jalur ke alat *platform* ke daftar.

g. Lokasi default untuk folder ini adalah :

```
%LOCALAPPDATA%\Android\Sdk\platform-tools
```

Setelah semua proses instalasi sudah dilaksanakan sesuai dengan instruksi sebelumnya, maka kita bias melakukn *project* aplikasi *react native*. *React Native* memiliki antarmuka baris perintah bawaan. Daripada menginstal dan mengelola versi tertentu dari *CLI* secara global, sebaiknya Anda mengakses versi saat ini saat *runtime* menggunakan *npx*, yang dikirimkan bersama *Node.js*. Dengan *npx react-native <command>*, versi stabil *CLI* saat ini akan diunduh dan dijalankan pada saat perintah dijalankan.

React Native memiliki antarmuka baris perintah bawaan, yang dapat Anda gunakan untuk menghasilkan proyek baru. Anda dapat mengaksesnya tanpa menginstal apa pun secara global menggunakan *npx*, yang dikirimkan bersama *Node.js*. Mari buat proyek *React Native* baru bernama "GOSE".

```
npx react-native init GOSE
```

Setelah kita membuat proyek aplikasi GOSE berbasis *android*, selanjutnya kita mengetik *source code* dibawah ini untuk menjalankan aplikasi kita tersebut.

```
npx react-native run-android
```

Jika semuanya sudah diatur dengan benar, Anda akan segera melihat aplikasi baru Anda berjalan di *emulator Android* Anda

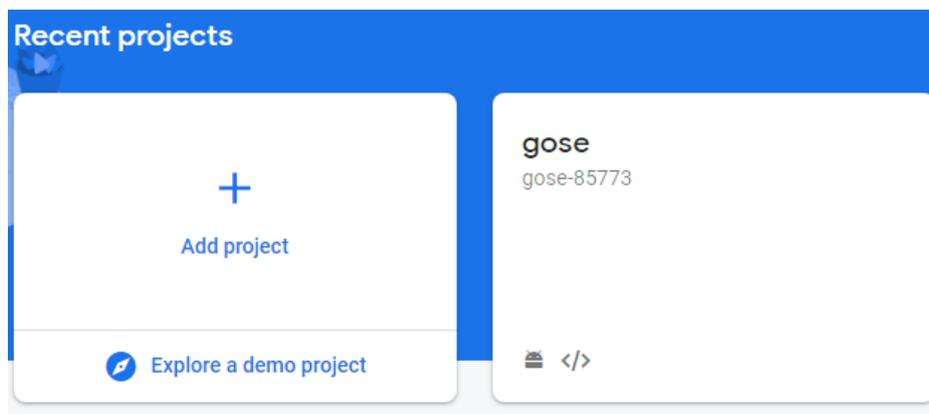


Gambar 4. 67: Tampilan Home React Native

4.3.2. Implementasi *Firebase*

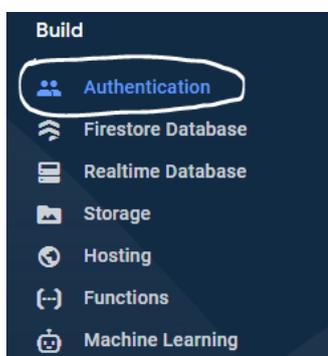
1. *Firebase Auth*

Firebase Authentication Service menyederhanakan proses login tanpa harus memikirkan arsitektur sistem yang rumit. Ini dicapai dengan mengintegrasikan aplikasi *Android* asli dengan layanan otentikasi *Firebase*. Di bawah ini adalah deskripsi lengkap tentang bagaimana proses *Firebase Authentication Service* dibuat. Mendaftarkan akun *Google* dengan layanan *Firebase*. Setelah akun Anda terdaftar, *Firebase* berarti memberikan akses kepada pengembang (insinyur perangkat lunak) yang dapat membuat lingkungan proyek aplikasi Anda. Dalam hal ini, proyek *Firebase* Anda terintegrasi ke dalam aplikasi *Android* Anda.



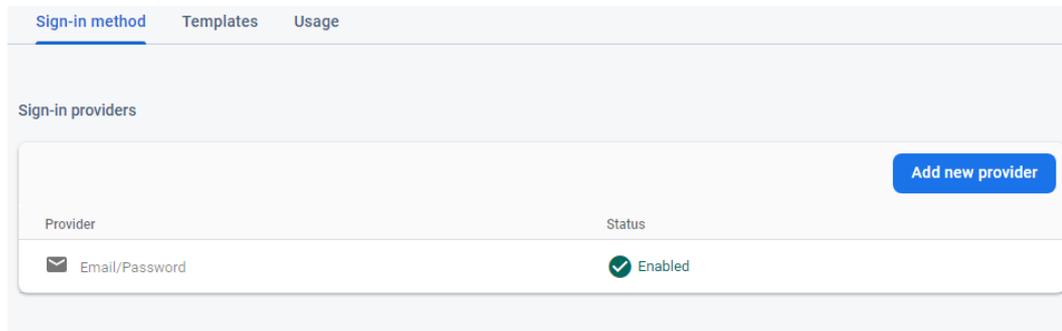
Gambar 4. 68: Halaman Project Environment dari *Firebase*

Kemudian dapat diklik dari *project* yang telah dibuat tadi dan akan menampilkan menu seperti gambar berikut :



Gambar 4. 69: Menu dari Project Layanan *Firebase*

Mengaktifkan layanan *autentikasi* yang akan dipilih, dalam penelitian ini *autentikasi* yang akan dipakai adalah *Email* atau *Sandi*.



Gambar 4. 70: Layanan Penyedia Authentication Firebase

2. *Firebase Realtime Database*

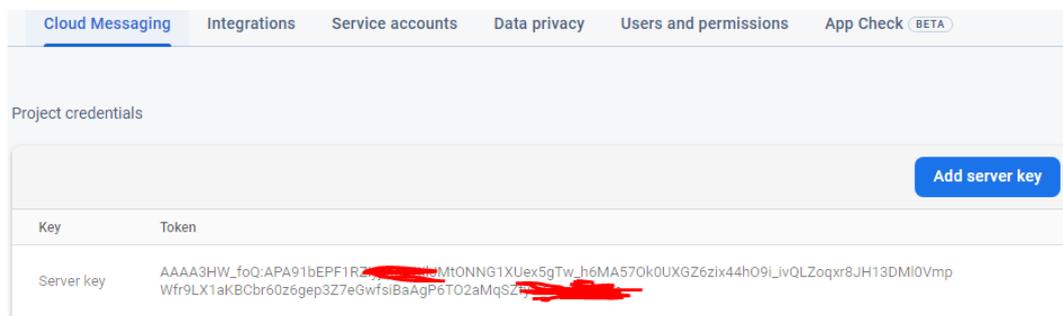
Berikut ini menjelaskan cara menerapkan *database real-time Firebase* yang membantu Anda menangani penyimpanan dan operasi data dalam bentuk *database*. Hal pertama yang perlu Anda lakukan untuk mengimplementasikan *database real-time* ini adalah mengonfigurasi aturan dalam *database real-time*. Aturan itu sendiri merupakan aturan keamanan saat menggunakan *database real-time* dan terkait dengan hak akses semua pengguna yang dapat membaca dan menulis data. Dalam investigasi, aturan yang digunakan adalah membenarkan akses baca, melalui proses *login*, dan menulis ke *database real-time Firebase* seperti yang ditunjukkan di bawah ini.



Gambar 4. 71: Aturan Realtime Database

3. *Firestore Cloud Messaging*

Dalam mengembangkan *platform* GOSE, Anda menggunakan *Firestore Cloud Messaging* untuk menangani notifikasi di aplikasi manajemen Anda, yang memicu data transaksi masuk. Oleh karena itu, ketika pelanggan melakukan layanan atau obrolan, *administrator* diberitahu bahwa ada pesan baru. Untuk mengakses *Firestore Cloud Messaging*, Anda memerlukan kunci API agar perangkat atau perangkat Anda dapat berkomunikasi dengan server. Anda bisa mendapatkan kunci API dari dasbor *Firestore* seperti yang ditunjukkan di bawah ini.



Gambar 4. 72: Kunci Server Pada Cloud Messaging

4.3.3. Implementasi Antarmuka *Customer*

1. Tampilan *Splash Screen*

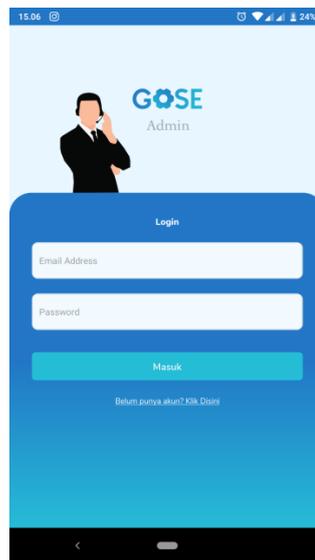
Berikut ini adalah tampilan *splash screen* pada aplikasi *go service electronic* pada sisi *customer* :



Gambar 4. 73: Tampilan *Splash Screen*

2. Tampilan *Login*

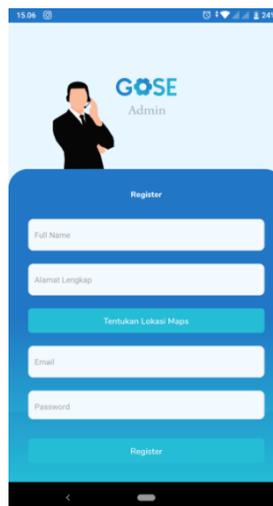
Berikut ini adalah tampilan *login* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 74: Tampilan login

3. Tampilan *Register*

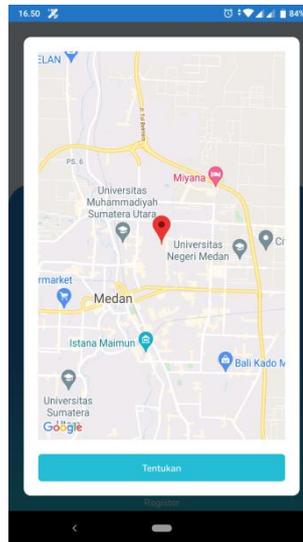
Pada gambar dibawah ini adalah tampilan *register* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 75: Tampilan Registrasi

4. Tampilan *Location Point*

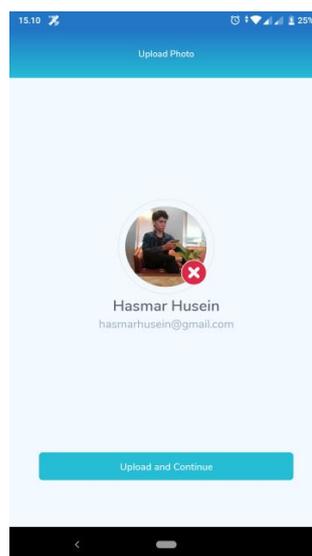
Pada gambar dibawah ini adalah tampilan *location point* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 76: Tampilan *Location Point*

5. Tampilan *Upload Photo*

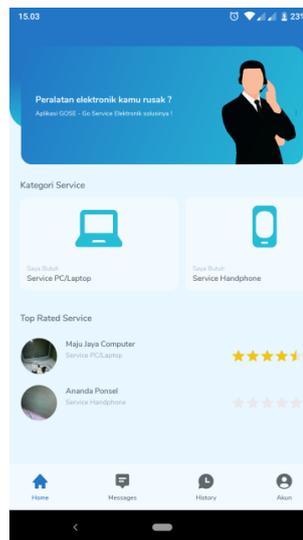
Pada gambar dibawah ini adalah tampilan *upload photo* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 77: Tampilan *Upload Photo*

6. Tampilan *Home*

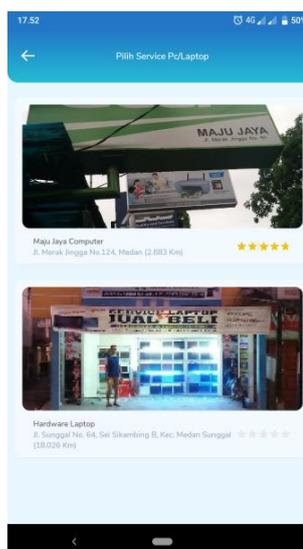
Pada gambar dibawah ini adalah tampilan *home* pada aplikasi *go service electronic* pada sisi *customer*..



Gambar 4. 78: Tampilan Home

7. Tampilan Pilih Toko Servis

Pada gambar dibawah ini adalah tampilan kategori servis pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 79: Tampilan Pilih Toko Servis

8. Tampilan Profil Toko Jasa Servis

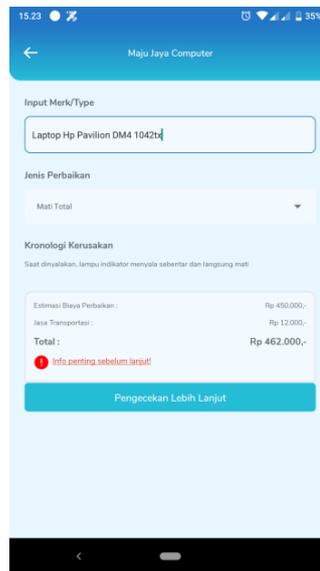
Pada gambar dibawah ini adalah tampilan profil toko jasa servis pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 80: Tampilan Profil Toko Jasa Servis

9. Tampilan Cek Estimasi Biaya

Pada gambar dibawah ini adalah tampilan cek estimasi biaya servis pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 81: Tampilan Cek Estimasi Biaya

10. Tampilan Konfirmasi Servis

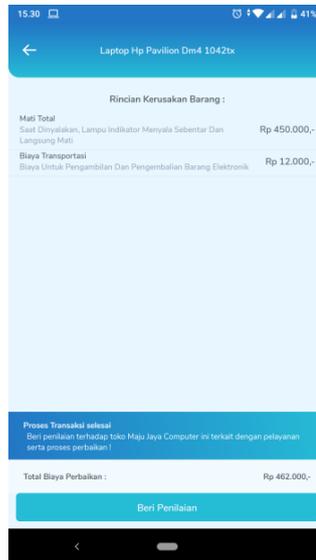
Pada gambar dibawah ini adalah tampilan konfirmasi servis pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 82: Tampilan Konfirmasi Servis

11. Tampilan Perbaikan Telah Selesai

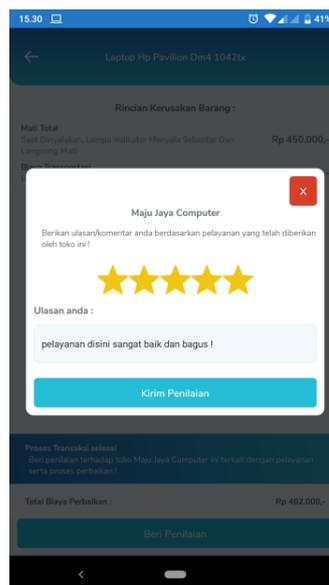
Pada gambar dibawah ini adalah tampilan perbaikan telah selesai pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 83: Tampilan Perbaikan Telah Selesai

12. Tampilan *Input Rating*

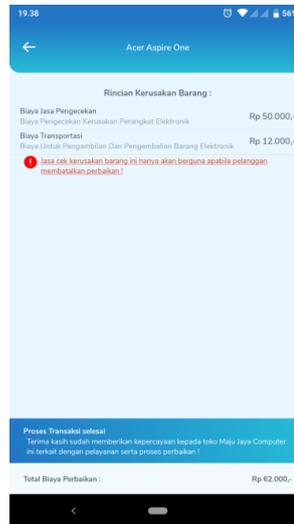
Pada gambar dibawah ini adalah tampilan *input rating* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 84: Tampilan *Input Rating*

13. Tampilan Pembatalan Servis

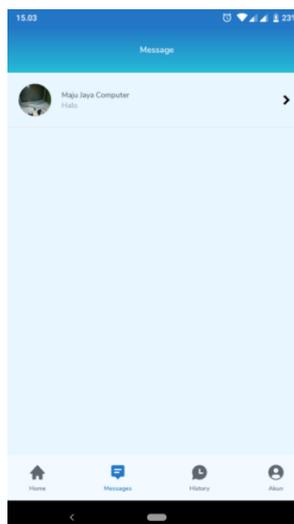
Pada gambar dibawah ini adalah tampilan pembatalan servis pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 85: Tampilan Pembatalan Servis

14. Tampilan Menu Message

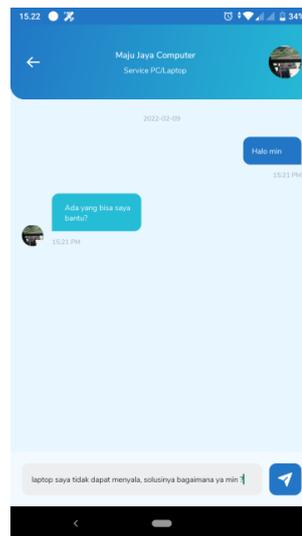
Pada gambar dibawah ini adalah tampilan menu *message* pada aplikasi *go service electronic* pada sisi *customer*, dimana sistem akan menampilkan data histori *chatting*.



Gambar 4. 86: Tampilan Menu Message

15. Tampilan *Chatting*

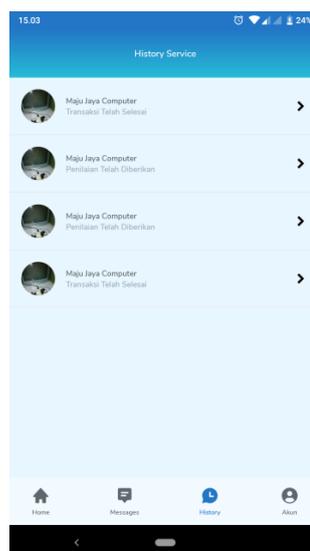
Pada gambar ini adalah tampilan *chatting* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 87: Tampilan Chatting

16. Tampilan Menu *Service*

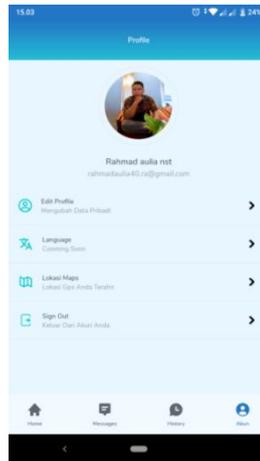
Pada gambar di bawah ini adalah tampilan menu *service* di aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 88: Tampilan Menu Service

17. Tampilan Menu Akun

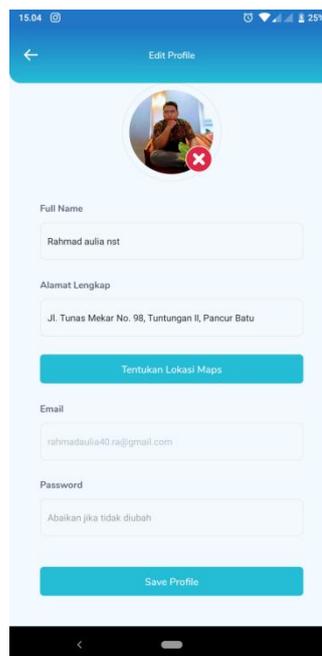
Pada gambar di bawah ini adalah tampilan menu akun pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 89: Tampilan Menu Akun

18. Tampilan Update Profile

Pada gambar dibawah ini adalah tampilan *update profile* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 90: Tampilan Update Profile

19. Tampilan Lihat Titik Lokasi Pada *Google Maps*

Pada gambar di bawah ini adalah tampilan lihat titik lokasi pada *google maps* pada aplikasi *go service electronic* pada sisi *customer*.



Gambar 4. 91: Tampilan Lihat Titik Lokasi Pada Google Maps

4.3.4. Implementasi Antarmuka *Admin*

1. Tampilan *Splash Screen*

Pada gambar dibawah ini adalah tampilan *splash screen* pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 92: Tampilan Splash Screen

2. Tampilan *Login*

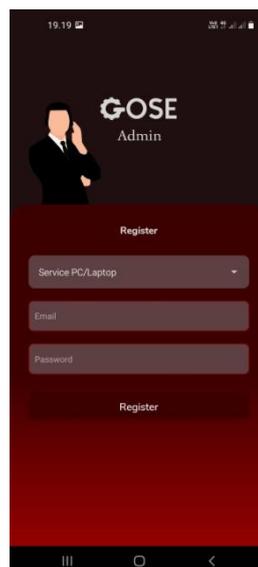
Pada gambar dibawah ini adalah tampilan *login* pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 93: Tampilan Login

3. Tampilan *Registrasi*

Pada gambar dibawah ini adalah tampilan *registrasi* pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 94: Tampilan Registrasi

4. Tampilan *Upload Data Toko*

Pada gambar dibawah ini adalah tampilan *upload* data toko pada aplikasi *go service electronic* pada sisi *admin*.



The screenshot shows a mobile application interface for uploading store data. The title is "Upload Data Toko". The form contains the following fields:

- Nama Toko:** Hardware Laptop
- No. Handphone:** 081277678865
- Alamat Lengkap:** Jl. Sunggal No. 64, Sei Sikambang B, Kec. Medan Sunggal
- Jam Buka Toko:** 09.00 - 20.00
- Keterangan Toko:** Menerima segala servis laptop

At the bottom of the form is a red button labeled "Upload & Continue".

Gambar 4. 95: Tampilan *Upload Data Toko*

5. Tampilan *Upload Maps Location Coordinate*

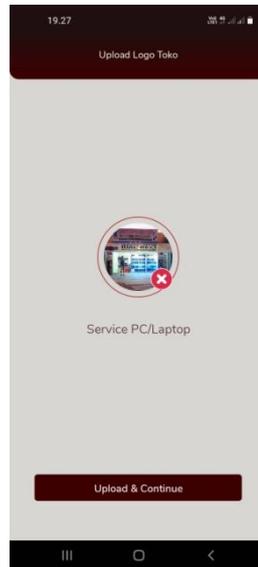
Pada gambar dibawah ini adalah tampilan *upload maps location coordinate* pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 96: Tampilan *Upload Maps Location Coordinate*

6. Tampilan *Upload Logo Toko*

Pada gambar dibawah ini adalah tampilan *upload logo toko* pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 97: Tampilan *Upload Logo Toko*

7. Tampilan Data Foto Toko

Pada gambar dibawah ini adalah tampilan data foto toko pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 98: Tampilan *Data Foto Toko*

8. Tampilan Jadwal Toko Buka

Pada gambar dibawah ini adalah tampilan jadwal toko buka pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 99: Tampilan Jadwal Toko Buka

9. Tampilan Data Estimasi Jasa Servis

Pada gambar dibawah ini adalah tampilan data estimasi jasa servis pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 100: Tampilan Data Estimasi Jasa Servis

10. Tampilan Konfirmasi Pengambilan Barang

Pada gambar dibawah ini adalah tampilan konfirmasi pengambilan barang pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 101: Tampilan Konfirmasi Pengambilan Barang

11. Tampilan Konfirmasi Barang Telah Tiba

Pada gambar dibawah ini adalah tampilan konfirmasi barang telah tiba pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 102: Tampilan Konfirmasi Barang Telah Tiba

12. Tampilan Konfirmasi Kerusakan Barang

Pada gambar dibawah ini adalah tampilan konfirmasi kerusakan barang pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 103: Tampilan Konfirmasi Kerusakan Barang

13. Tampilan Konfirmasi Perbaikan Selesai

Pada gambar dibawah ini adalah tampilan konfirmasi perbaikan selesai pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 104: Tampilan konfirmasi Perbaikan Selesai

14. Tampilan Pembatalan Servis

Pada gambar dibawah ini adalah tampilan pembatalan servis pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 105: Tampilan Pembatalan Servis

15. Tampilan Melakukan Pengembalian Barang

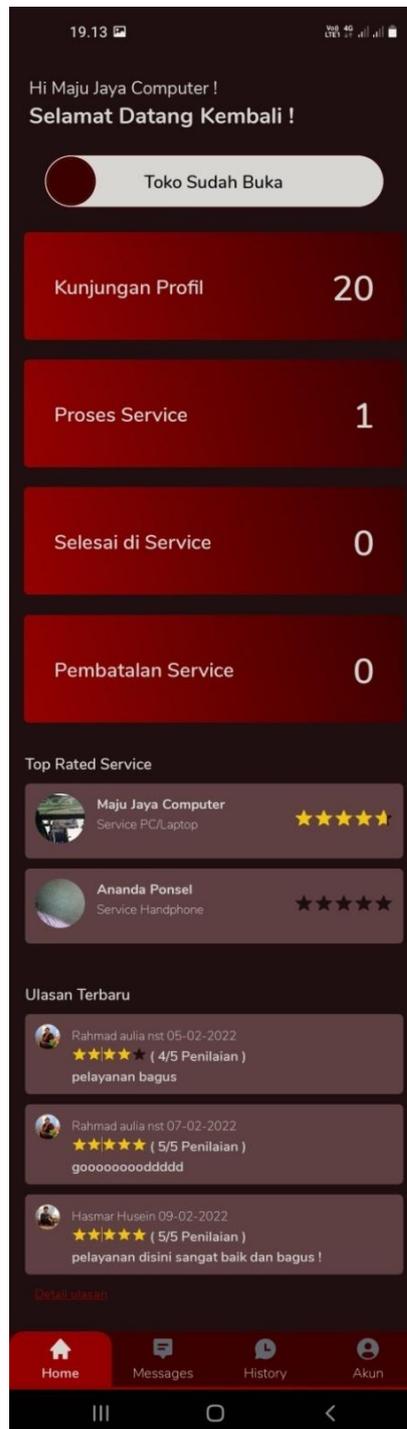
Pada gambar dibawah ini adalah tampilan melakukan pengembalian barang pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 106: Tampilan Melakukan Pengembalian Barang

16. Tampilan Halaman Utama

Pada gambar dibawah ini adalah tampilan halaman utama pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 107: Tampilan Halaman Utama

17. Tampilan Detail Ulasan

Pada gambar dibawah ini adalah tampilan detail ulasan pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 108: Tampilan Detail Ulasan

18. Tampilan Akun

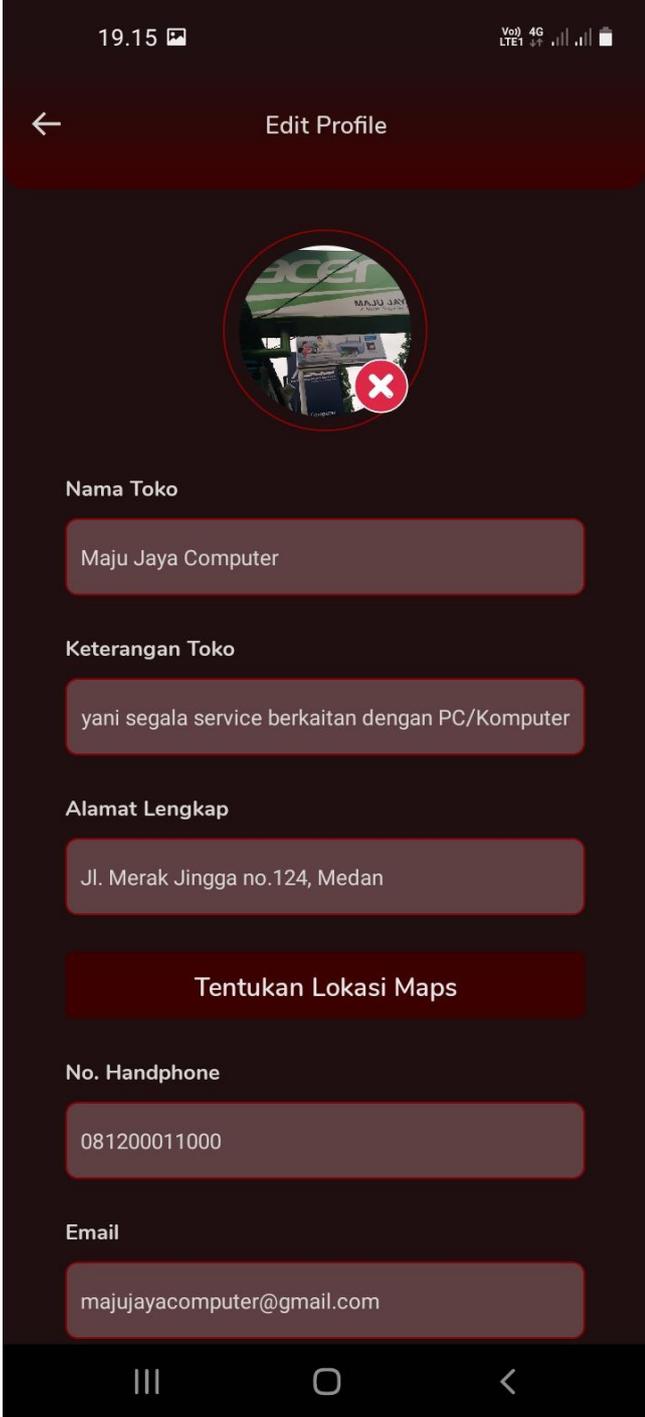
Pada gambar dibawah ini adalah tampilan akun pada aplikasi *go service electronic* pada sisi *admin*.



Gambar 4. 109: Tampilan Akun

19. Tampilan *Edit Profile*

Pada gambar dibawah ini adalah tampilan *edit profile* pada aplikasi *go service electronic* pada sisi *admin*.



19.15 VoLTE 4G LTE1

← Edit Profile



Nama Toko

Maju Jaya Computer

Keterangan Toko

yani segala service berkaitan dengan PC/Komputer

Alamat Lengkap

Jl. Merak Jingga no.124, Medan

Tentukan Lokasi Maps

No. Handphone

081200011000

Email

majujayacomputer@gmail.com

Gambar 4. 110: Tampilan *Edit Profile*

4.3.5. Pengujian Sistem

Uji sistem yang digunakan dengan metode pengujian black box. Metode ini merupakan metode pengujian yang menitikberatkan pada kebutuhan fungsional sistem. Oleh karena itu, pengujian kotak hitam memungkinkan pengembangan perangkat lunak untuk membuat serangkaian batasan input yang melatih semua persyaratan fungsional suatu program.

Table 4. 14: Data Diri Validator

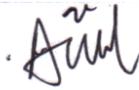
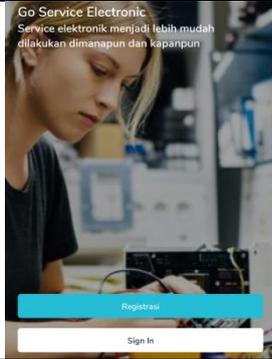
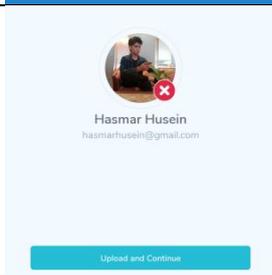
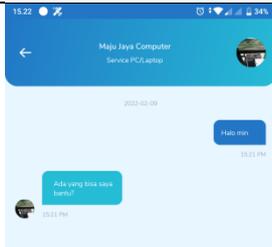
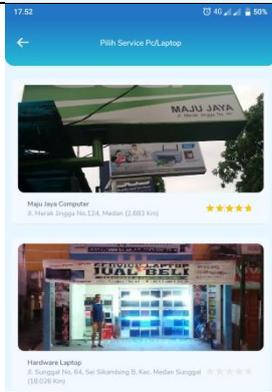
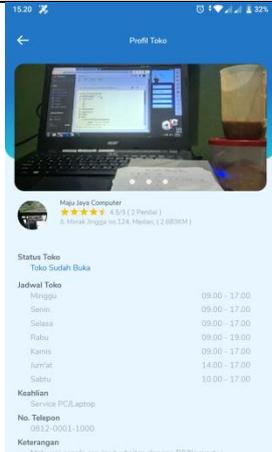
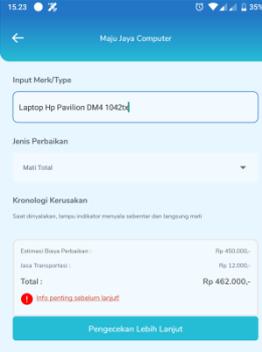
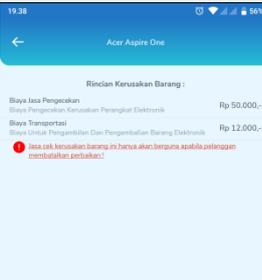
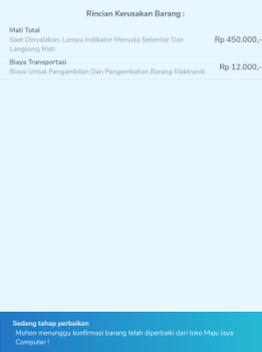
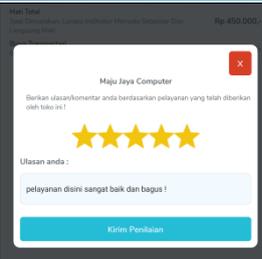
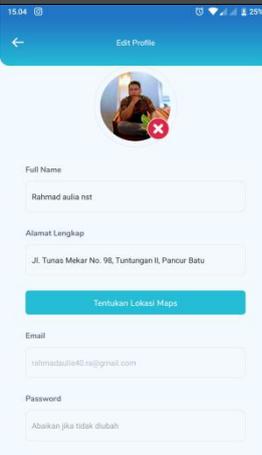
Validator	Nama Validator	Tanda Tangan
Validator 1	Nama : Ahwat Jabatan : Pemilik Toko Maju Jaya Computer	
Validator 2	Nama : Arjoko Siahaan, S.Kom Jabatan : Pemilik Toko Hardware Laptop	

Table 4. 15: Pengujian Sistem Pada Aplikasi Customer

No	Rancangan Input/Output	Hasil Yang Diharapkan	Tampilan	Validator 1	Validator 2
1	Membuka sistem	Masuk ke halaman <i>Started</i>		✓	✓
2	Melakukan Login	Login Sukses, Menampilkan Halaman <i>Home</i>		✓	✓

3	Melakukan Login	Login Gagal, Memunculkan Notifikasi		✓	✓
4	Melakukan Register	Masuk ke halaman Upload Photo		✓	✓
5	Melakukan Chatting	Menampilkan Data Chat		✓	✓
6	Melihat Toko Tersedia	Menampilkan Data Toko		✓	✓
7	Memilih Toko	Menampilkan Rincian Toko		✓	✓

8	Melakukan Cek Estimasi Biaya	Menampilkan Biaya Estimasi		✓	✓
9	Melakukan Konfirmasi Pembatalan Servis	Menampilkan Data Pembatalan Servis		✓	✓
10	Melakukan Konfirmasi Lanjut Servis	Menampilkan Data Proses Perbaikan		✓	✓
11	Memberikan Ulasan Kepada Toko	Menampilkan Data Pengisian Ulasan		✓	✓
12	Mengubah Data Customer	Menampilkan Data Customer		✓	✓

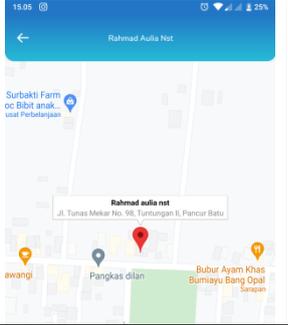
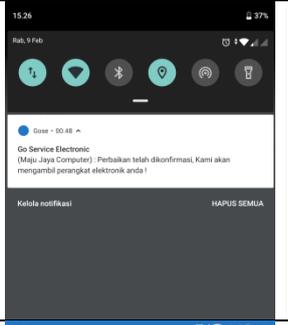
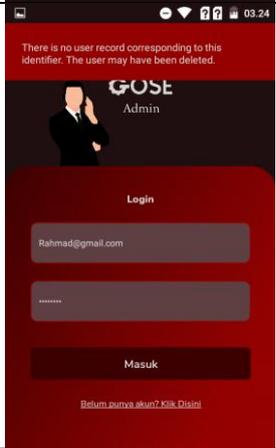
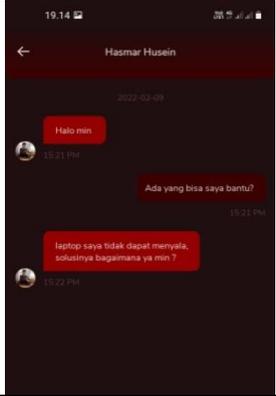
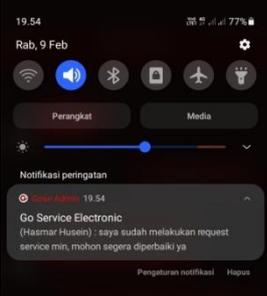
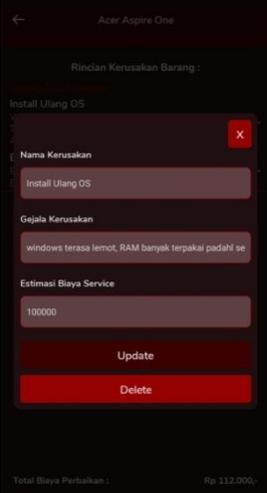
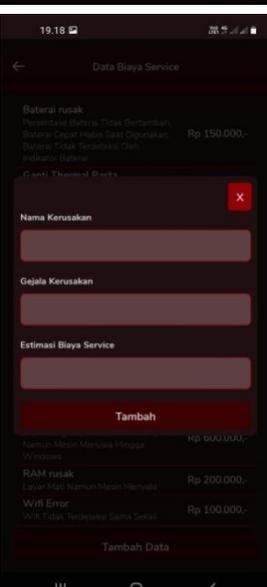
13	Melihat Lokasi <i>Maps</i>	Menampilkan Lokasi <i>Maps</i>		✓	✓
14	Menerima Informasi Perbaikan Perangkat Elektronik	Menampilkan Notifikasi Konfirmasi Perbaikan Terbaru		✓	✓
15	Melakukan <i>Logout</i> Akun	Menampilkan Notifikasi <i>Logout</i>		✓	✓

Table 4. 16: Pengujian Sistem Pada Aplikasi Admin

No	Rancangan Input/Output	Hasil Yang Diharapkan	Tampilan	Validator 1	Validator 2
1	Membuka sistem	Masuk ke halaman <i>login</i>		✓	✓

2	Melakukan Login	Login Sukses, Menampilkan Halaman Home		✓	✓
3	Melakukan Login	Login Gagal, Memunculkan Notifikasi		✓	✓
4	Melakukan Register	Masuk ke halaman Upload Data Toko		✓	✓
5	Melakukan Chatting	Menampilkan Data Chat		✓	✓

6	Menerima <i>Chat Customer</i>	Menampilkan Notifikasi <i>Chat</i>		✓	✓
7	Menerima <i>Servis Terbaru</i>	Menampilkan Notifikasi <i>Servis Terbaru</i>		✓	✓
8	Melakukan <i>Cek Data Servis Terbaru</i>	Menampilkan <i>Data Servis</i>		✓	✓
9	Melakukan <i>Cek Lokasi Customer</i>	Menampilkan <i>Data Lokasi Maps Costumer</i>		✓	✓

<p>10</p>	<p>Melakukan <i>Update Data</i> Kerusakan</p>	<p>Menampilkan <i>Data Update</i> Kerusakan</p>		<p>✓</p>	<p>✓</p>
<p>11</p>	<p>Melakukan <i>Tambah Data</i> Kerusakan</p>	<p>Menampilkan <i>Form</i> <i>Tambah Data</i> Kerusakan</p>		<p>✓</p>	<p>✓</p>
<p>12</p>	<p>Menambah <i>Data</i> <i>Estimasi</i> <i>Biaya</i> <i>Perbaikan</i></p>	<p>Menampilkan <i>Data</i> <i>Penambahan</i> <i>Estimasi</i> <i>Biaya</i> <i>Perbaikan</i></p>		<p>✓</p>	<p>✓</p>

13	Menambah Jadwal Toko Buka	Menampilkan Data Penambahan Jadwal Toko Buka		✓	✓
14	Melakukan Update Data Toko	Kembali Ke Halaman Home		✓	✓
15	Melakukan Logout Akun	Menampilkan Notifikasi Logout		✓	✓

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, maka penulis dapat menarik kesimpulan, sebagai berikut :

1. Dengan menggunakan *Framework React Native* dan *Firebase Realtime Database* dapat diterapkan pada aplikasi *Go Service Electronic* untuk mempermudah *customer* dan penyedia jasa servis elektronik mengetahui lokasi jasa servis, estimasi biaya servis, serta melakukan servis perangkat elektronik, serta mampu memperluas pemasaran jasa servis.
2. Pada sistem ini dibuat dua aplikasi yaitu untuk *customer* dan *admin*, untuk menghindari terjadinya *crash* karena saat permintaan data antara *admin* dan *customer* serta menghindari pembobolan data penting.
3. Dengan adanya aplikasi ini dapat menambah pengetahuan bagi penulis dan pembaca dalam hal membangun aplikasi *android* menggunakan *framework react native* dan *firebase database*.

5.2. Saran

Berdasarkan kesimpulan dan tahapan analisis yang telah dilakukan sebelumnya, maka penulis memberikan saran untuk pengembangan selanjutnya, yaitu :

1. Penggunaan aplikasi diwajibkan untuk menghidupkan GPS terlebih dahulu untuk mengetahui lokasi titik koordinat saat ingin menentukan lokasi.
2. Untuk mengoptimalkan kesetabilan sistem, diwajibkan menggunakan *android* minimal versi 6.0 dan maksimal versi 10.0 agar tidak terjadinya *crash* dalam menjalankan sistem.
3. Aplikasi *Go Service Electronic* ini masih bisa dijalankan di sistem operasi android, diharapkan untuk pengembangan selanjutnya dapat dijalankan di sistem operasi lainnya.

DAFTAR PUSTAKA

- Abdulloh, R. 2020. *Menguasai React JS Untuk Pemula: Panduan Belajar JavaScript Dari Dasar Hingga Membuat Aplikasi Web Modern*. Rohi Abdulloh.
- Adikara, Fransiskus. 2013. "Analisis Dan Perancangan Sistem Absensi Berbasis Global Positioning System (GPS) Pada Android 4.X." *Seminar Nasional Teknologi Informasi* 4–9.
- Ahmad, Ashari, and Herri Setiawan. 2011. "Cloud Computing : Solusi ICT?" *Sistem Informasi* 3(6):1–5.
- Akil, Ibnu, and M. Kom. 2018. "Referensi Dan Panduan UML 2.4 Singkat Tepat Jelas." *Jakarta: Andi*.
- Almuttaqin, Givo. 2016. "Sistem Informasi Pendaftaran Pernikahan Berbasis Online Menggunakan Metode Waterfall (Study Kasus : Kantor Urusan Agama Kecamatan." *Jurnal Rekayasa Dan Manajemen Sistem Informasi* 2(2):52–55.
- Cancer, Yunie, and Zikrul Alim. 2016. *PLATFORM AS A SERVICE (PaaS) SEBAGAI LAYANAN SISTEM OPERASI CLOUD COMPUTING*.
- Cholifah, Wahyu Nur, Yulianingsih Yulianingsih, and Sri Melati Sagita. 2018. "Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap." *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)* 3(2):206. doi: 10.30998/string.v3i2.3048.
- Eisenman, B. 2017. *Learning React Native: Building Native Mobile Apps with JavaScript*. O'Reilly Media.
- Eri Mardiani, N. R. H. K. D. I. S. J. 2016. *Kumpulan Latihan SQL*. Elex Media Komputindo.
- Gunawan, G., M. Damanik, F. B. Larasati, A. F. Zuhri, S. Solikhun, R. Watrianthos, and J. Simarmata. 2021. *Dasar-Dasar Pemrograman Android*. Yayasan Kita Menulis.
- Habibi, R., and R. Aprilian. 2020. *Tutorial Dan Penjelasan Aplikasi E-Office Berbasis Web Menggunakan Metode RAD*. Kreatif.
- Habibi, R., and K. Sandi. 2020. *Aplikasi Bank Sampah Istimewa Menggunakan*

Framework PHP Codeigniter Dan DBMS MySQL. Kreatif.

- Hendini, Ade. 2016. "Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)." *Jurnal Khatulistiwa Informatika* IV(2):201–5. doi: 10.2135/cropsci1983.0011183x002300020002x.
- Herwanto, R., O. W. Purbo, and R. Z. A. Aziz. 2021. *CLOUD COMPUTING: Manajemen Dan Perencanaan Kapasitas*. Penerbit Andi.
- Husain, Muhammad Zarlis, Zulkifli Nasution, Hengki Tamado Sihotang, and Sri Wahyuni. 2018. "Filsafat Ilmu Komputer Dan Cloud Computing Secara Etimologis." *Jurnal Mantik Penusa* 2(2):15–21.
- Ilham, S. S. M. S. 2021. *E-Governance*. Deepublish.
- Ilhami, Mirza. 2017. "Pengenalan Google Firebase Untuk Hybrid Mobile Apps Berbasis Cordova." *Jurnal IT CIDA* 3(124):16–29.
- Irawan, Muhammad Dedi, and Herviana Herviana. 2019. "Implementasi Logika Fuzzy Dalam Menentukan Jurusan Bagi Siswa Baru Sekolah Menengah Kejuruan (Smk) Negeri 1 Air Putih." *Jurnal Teknologi Informasi* 2(2):129. doi: 10.36294/jurti.v2i2.427.
- Kusniyati, Harni. 2016. "Culture Is a Way of Life That Developed and Shared by a Group of People , and Inherited from One Technology as a Competitive Sector That Can Added Value to the Business Processes That Run . The Development of Information and Communication Technology Make." *APLIKASI EDUKASI BUDAYA TOBA SAMOSIR BERBASIS ANDROID Harni* 9(1):9–18.
- Muhamad Alda, S. K. M. S. I. 2020. *Aplikasi CRUD Berbasis Android Dengan Kodular Dan Database Airtable*. Media Sains Indonesia.
- Muhamad Alda, S. K. M. S. I. 2021. *Rekayasa Perangkat Lunak Berorientasi Objek*. Media Sains Indonesia.
- Mutia, Intan. 2016a. "Pemanfaatan Komputasi Awan (Cloud Computing) Bagi Pembelajaran Mahasiswa Perguruan Tinggi." *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)* 1(1). doi: 10.30998/string.v1i1.963.
- Mutia, Intan. 2016b. "Pemanfaatan Komputasi Awan (Cloud Computing) Bagi

- Pembelajaran Mahasiswa Perguruan Tinggi.” *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)* 1(1):1–9. doi: 10.30998/string.v1i1.963.
- Nurhidayah, S., M. N. Fauzan, and W. I. Rahayu. 2020. *Implementasi Metode Analytic Hierarchy Process (AHP) Dengan PHP*. Kreatif.
- Pane, S. F., M. Zamzam, and M. D. Fadillah. 2020. *Membangun Aplikasi Peminjaman Jurnal Menggunakan Aplikasi Oracle Apex Online*. Kreatif.
- Pratama, Ndaru Adi, and Catur Hermawan. 2016. “Aplikasi Pembelajaran Tes Potensi Akademik Berbasis Android.” *Jurnal Penelitian Dosen FIKOM (UNDA)* 6(1):1–6.
- Samsudin, Samsudin. 2019. “Optimalisasi Penerimaan Remunerasi Dosen Menggunakan Metode Rule Base Reasoning.” *Klik - Kumpulan Jurnal Ilmu Komputer* 6(3):224. doi: 10.20527/klik.v6i3.185.
- Silalahi, Mesri. 2018. “Perbandingan Performansi Database MongoDB Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web.” *Computer Based Information System Journal* 6(1):63. doi: 10.33884/cbis.v6i1.574.
- Sonita, Anisya, and Rizki Fitrah Fardianitama. 2018. “Aplikasi E-Order Menggunakan Firebase Dan Algoritme Knuth Morris Pratt Berbasis Android.” *Pseudocode* 5(2):38–45. doi: 10.33369/PSEUDOCODE.5.2.38-45.
- Sudiarta, I. Ketut Gede, I. Nyoman Eddy Indrayana, and I. Wayan Suasnawa. 2018. “Membangun Struktur Realtime Database Firebase Untuk Aplikasi Monitoring Pergerakan Group Wisatawan.” *Jurnal Ilmu Komputer* 11(2):96. doi: 10.24843/jik.2018.v11.i02.p04.
- 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* 3(1):1–9.
- Supardi, I. Y. 2017. *Koleksi Program Tugas Akhir Dan Skripsi Dengan Android*. Elex Media Komputindo.
- Supardi, I. Y. 2021. *Semua Bisa Menjadi Programmer JavaScript & Node.js*. Elex Media Komputindo.

LAMPIRAN

Maju Jaya Computer

Medan, 10 Januari 2022

Jl. Putri Merak Jingga No.176,
Kec. Medan Barat,
Kota Medan

Kepada Yth :
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Sehubungan dengan surat masuk No : B.027/ST.I/ST.V.2/TL.00/01/2022 perihal permohonan izin penelitian pada Maju Jaya Computer sebagai syarat untuk menyelesaikan studi atas nama :

Nama : Rahmad Aulia
NIM : 0702163047
Program Studi : Sistem Informasi

Kami Mengizinkan mahasiswa tersebut melakukan penelitian dengan judul penelitian "**Membangun Aplikasi GOSE (Go Service Electronic) Berbasis Android Menggunakan Framework React Native dan Firebase Realtime Database**" dengan ketentuan mahasiswa tersebut menaati peraturan dan tata tertib yang berlaku di Maju Jaya Computer serta mematuhi protocol COVID-19.

Demikian surat ini kami buat agar dapat digunakan sebagaimana mestinya dan atas perhatiannya kami ucapkan terima kasih.



**Horas Computer****Central Spesialis Service Laptop**

Jl. Sisingamangaraja No.154, Teladan Barat, Kec. Medan Kota

Hp : 0859-5387-2444

Medan, 08 Januari 2022

Nomor : 67/HC/MDN/2022
Hal : Balasan Surat Izin Riset

Kepada Yth :
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Dengan Hormat,

Sehubungan dengan surat permohonan izin riset yang diajukan oleh mahasiswa dengan No : B.029/ST.I/ST.V.2/TL.00/01/2022 pada Horas Computer sebagai syarat untuk menyelesaikan studi atas nama :

Nama : Rahmad Aulia
NIM : 0702163047
Program Studi : Sistem Informasi

Dengan Judul **“Membangun Aplikasi GOSE (Go Service Electronic) Berbasis Android Menggunakan Framework React Native dan Firebase Realtime Database”**.

Dengan ini kami menyatakan persetujuan dan member izin kepada mahasiswa tersebut untuk melakukan kegiatan riset tugas akhir dengan ketentuan mahasiswa tersebut menaati peraturan dan tata tertib yang berlaku di Horas Computer serta mematuhi protocol COVID-19.

Demikian, atas perhatiannya kami ucapkan terima kasih.

Hormat Kami,

HORAS COMPUTER
Zepi Lingga
Tgl. 08 / 01 / 2022



Hardware Laptop
Jual – Beli Laptop, Service & Accessories
Jl. Sunggal No. 64, Sei Sikambing B, Kec. Medan Sunggal, Medan
No. Handphone :0823 6661 2314

Medan, 09 Januari 2022

Nomor : 035/HL/MDN/B/2022
Hal : Izin Riset

Kepada Yth :
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Sehubungan dengan surat masuk No : B.028/ST.I/ST.V.2/TL.00/01/2022 perihal permohonan izin penelitian pada Hardware Laptop sebagai syarat untuk menyelesaikan studi atas nama :

Nama : Rahmad Aulia
NIM : 0702163047
Program Studi : Sistem Informasi

Kami Mengizinkan mahasiswa tersebut melakukan penelitian dengan judul penelitian "**Membangun Aplikasi GOSE (Go Service Electronic) Berbasis Android Menggunakan Framework React Native dan Firebase Realtime Database**" dengan ketentuan mahasiswa tersebut menaati peraturan dan tata tertib yang berlaku di Hardware Laptop serta mematuhi protocol COVID-19.

Demikian surat ini kami buat agar dapat digunakan sebagaimana mestinya dan atas perhatiannya kami ucapkan terima kasih.

Hormat Kami,

Arioke Sahaan, S.Kom
HARDWARE LAPTOP
Tgl. 9 / 1 / 2022

Hasil Wawancara Penelitian

Tanggal : 5 Januari 2022

Peneliti : Rahmad Aulia

Informan : Bapak Ahwat

Jabatan : Pemilik Maju Jaya *Computer*.

NO	Peneliti	Informan
1.	Sudah berapa lama toko jasa servis ini berjalan ?	Saya membuka jasa servis ini sudah sekitar enam tahun
2.	Dalam menjalankan bisnis ini, apakah ada kendala untuk menarik pelanggan?	Sampai saat ini yang menjadi permasalahan yaitu kurangnya informasi tentang jasa servis saya
3.	Apakah penduduk sekitar sudah mengetahui lokasi toko servis ini ?	Untuk saat ini warga disekitar sudah mulai mengetahui bahwa saya membuka jasa servis laptop disini
4.	Saat pelanggan tiba dan ingin melakukan servis, berapa persen kemungkinan pelanggan membatalkan perbaikan setelah mengetahui harga jasa servis yang ditawarkan ?	Selama ini sejak mulai saya buka jasa laptop ini, mungkin sekitar empat puluh persen yang melakukan pembatalan setelah pelanggan mengetahui biaya jasa servis yang saya berikan.
5.	Apa saja yang menjadi alasan para pelanggan saat melakukan pembatalan perbaikan ?	Salah satu faktor yaitu dikarenakan pelanggan ingin mencari biaya servis lebih murah, serta melakukan perbandingan harga dengan jasa servis lain.
6.	Apakah bapak pernah menerima pelanggan yang ingin servis ditempat pelanggan tersebut ?	Saya pernah mendapat pelanggan yang meminta untuk datang kerumahnya, berkebetulan memang sudah pelanggan tetap, ia meminta melakukan pengecekan perangkat komputer dirumahnya.
7.	Pelayanan apa saja yang diberikan terhadap pelanggan?	Saat ini pelayanan yang dapat saya berikan pastinya garansi terhadap barang yang sudah diperbaiki berkisar dua minggu.
8.	Apakah pernah menawarkan jasa servis ini melalui dunia maya ?	Untuk saat ini saya belum menawarkan di dunia maya.

9.	Dengan adanya wabah <i>covid19</i> ini, apakah berpengaruh terhadap pendapatan toko ?	Ya, sangat berpengaruh, terutama dalam hal jasa servis ini, dikarenakan adanya <i>social distencing</i> , sehingga berimbas kepada pebisnis UMKM seperti saya ini.
10.	Bagaimana pendapat anda jika ada aplikasi yang mendukung untuk meningkatkan peluang bisnis di bidang jasa servis ?	Saya sangat setuju, itu bisa menjadi peluang terbaik untuk jasa servis saya ini, dan toko saya ini bisa lebih mudah diketahui pelanggan lainnya.



Disetujui Oleh,

Pemilik Maju Jaya Computer

Hasil Wawancara Penelitian

Tanggal : 6 Januari 2022

Peneliti : Rahmad Aulia

Informan : Bapak Arjoko Siahaan, S.Kom

Jabatan : Pemilik *Hardware* Laptop.

NO	Pertanyaan	Deskripsi
1.	Sudah berapa lama bekerja di toko jasa servis ini ?	Saya mulai bekerja pada toko jasa servis ini sudah sekitar tiga tahun
2.	Dalam menjalankan pekerjaan ini selaku bidang teknisi, apakah ada kendala dalam hal servis perangkat elektronik ?	Sampai saat ini untuk permasalahan memperbaiki perangkat elektronik terutama pada perangkat laptop masih bisa diatasi, namun yang menjadi kendala yaitu saat sedang banyak perbaikan perangkat, sehingga proses pengerjaannya menjadi lama, sehingga mengurangi kepercayaan dari pelanggan karena mereka menganggap perangkat mereka tidak diperbaiki.
3.	Selama anda bekerja disini, apakah pelanggan banyak melakukan servis pada toko jasa servis ini?	Menurut saya tergantung situasi, terkadang banyak pelanggan yang sedang memperbaiki, ada kalanya juga pelanggan ragu melakukan servis pada toko ini dikarenakan tidak adanya hasil penilaian yang bisa diperlihatkan dari pelanggan sebelumnya
4.	Saat pelanggan tiba dan ingin melakukan servis, berapa persen kemungkinan pelanggan membatalkan perbaikan setelah mengetahui harga jasa servis yang ditawarkan ?	Selama ini sejak saya bekerja pada toko jasa servis ini, mungkin sekitar tiga puluh persen yang melakukan pembatalan setelah pelanggan mengetahui biaya jasa servis perangkat elektronik pelanggan tersebut.
5.	Apa saja yang menjadi alasan para pelanggan saat melakukan pembatalan perbaikan ?	Penyebab utamanya yaitu dikarenakan pelanggan belum yakin dengan jasa servis kami, pelanggan juga ingin mencari biaya servis lebih murah.
6.	Apakah anda pernah menerima pelanggan yang	Saya pernah mendapat pelanggan berkebetulan teman dekat juga, ia meminta

	ingin servis dirumah pelanggan tersebut ?	melakukan instalasi laptop dirumahnya.
7.	Pelayanan apa saja yang diberikan terhadap pelanggan?	Saat ini pelayanan yang dapat kami berikan pastinya garansi terhadap barang yang sudah diperbaiki berkisar satu bulan.
8.	Dengan adanya wabah covid19 ini, apakah berpengaruh terhadap pelanggan yang ingin melakukan perbaikan perangkat elektronik ?	Ya, sangat berpengaruh, terutama dalam hal jasa servis ini, dikarenakan adanya <i>social distencing</i> , sehingga berimbas kepada pekerja seperti saya ini dikarenakan berkurangnya pendapatan toko sehingga berpengaruh juga terhadap gaji saya selaku teknisi pada toko ini.
9.	Bagaimana pendapat anda jika ada aplikasi yang mendukung untuk meningkatkan kembali pendapatan pada toko jasa servis ini?	Saya sangat setuju, itu bisa menjadi peluang terbaik untuk toko jasa servis ini, sehingga nantinya kami bisa mendapatkan pelanggan lebih banyak lagi agar dapat memulihkan pendapatan pada toko ini .

Disetujui Oleh,
 Pemilik *Hardware* Laptop

 Arjoko Sahaan, S.Kom
 Tgl. 6 / 1 / 2022

Hasil Wawancara Penelitian

Tanggal : 7 Januari 2022

Peneliti : Rahmad Aulia

Informan : Bapak Zepi Lingga

Jabatan : Pemilik Horas *Computer*.

NO	Pertanyaan	Deskripsi
1.	Sudah berapa lama toko jasa servis ini berjalan ?	Saya membuka jasa servis ini sekitar tiga tahun
2.	Dalam menjalankan bisnis ini, apakah ada kendala untuk menarik pelanggan?	Banyak sekali kendala yang saya hadapi terutama apabila terdapat sedikit miskomunikasi antara customer dan teknisi saya
3.	Apakah penduduk sekitar sudah mengetahui lokasi toko servis ini ?	Alhamdulillahnya sebagian besar orang di daerah saya sudah mengetahui bahwasanya toko Hardware Laptop itu adalah toko yang dapat dipercaya.
4.	Saat pelanggan tiba dan ingin melakukan servis, berapa persen kemungkinan pelanggan membatalkan perbaikan setelah mengetahui harga jasa servis yang ditawarkan ?	Kemungkinan pembatalan tergantung dengan kerusakan, apabila kerusakannya bisa kami atasi mungkin hanya sepuluh persen, Sembilan puluh persen <i>customer</i> membatalkan servis biasanya di harga servis yang kerusakannya parah.
5.	Apa saja yang menjadi alasan para pelanggan saat melakukan pembatalan perbaikan ?	Alasannya ya ketidakcocokan harga tadilah harga servis sama ataupun harga jasanya.
6.	Apakah bapak pernah menerima pelanggan yang ingin servis ditempat pelanggan tersebut ?	Pernah lah terima pelanggan yang kayak gitu sekali
7.	Pelayanan apa saja yang diberikan terhadap pelanggan?	Pelayanan yang kami berikan tergantung kebutuhan <i>customer</i> , seperti garansi, terkadang <i>customer</i> juga ingin barangnya cepat selesai.
8.	Apakah pernah menawarkan jasa servis ini melalui dunia maya ?	Kami pernah memasang iklan di Facebook juga di WhatsApp itu aja.

9.	Dengan adanya wabah <i>covid19</i> ini, apakah berpengaruh terhadap pendapatan toko ?	Semenjak adanya <i>covid</i> omset kami menurun sangat drastis hingga di angka enam puluh persen. Mengapa demikian ,salah satunya yaitu rasa takut untuk keluar rumah diakibatkan wabah <i>covid</i> masih berlanjut, kemudian para pelanggan juga ada yang sedang bermasalah perihal keuangan, sehingga mereka ragu untuk melakukan servis dikarenakan takut akan biaya yang terlalu besar.
10.	Bagaimana pendapat bapak jika ada aplikasi yang mendukung untuk meningkatkan peluang bisnis di bidang jasa servis ?	Saya kira aplikasi yang sesuai pada masa kondisi seperti ini, mengapa demikian, apabila kita bisa melakukan pemasaran melalui aplikasi, maka toko saya ini bisa bangkit kembali dengan menarik pelanggan melalui aplikasi, saya kira itu sangat dibutuhkan

Disetujui Oleh,

Pemilik Horas Computer

HORAS COMPUTER

Zepi Lingga
Tgl. 7/1/2022

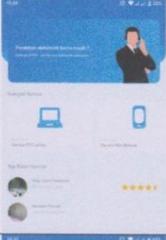
VALIDATOR SISTEM

20 Januari 2022

Validator I
 Nama Validator : Ahwat
 Jabatan : Pemilik Maju Jaya Computer

Validator II
 Nama Validator : Arjoko Siahaan, S.Kom
 Jabatan : Pemilik Hardware Laptop

Pengujian Sistem Pada Aplikasi Customer

No	Rancangan Input/Output	Hasil Yang Diharapkan	Tampilan	Validator 1	Validator 2
1	Membuka sistem	Masuk ke halaman Started		✓	✓
2	Melakukan Login	Login Sukses, Menampilkan Halaman Home		✓	✓
3	Melakukan Login	Login Gagal, Memunculkan Notifikasi		✓	✓

9	Melakukan Konfirmasi Pembatalan Servis	Menampilkan Data Pembatalan Servis		✓	✓
10	Melakukan Konfirmasi Lanjut Servis	Menampilkan Data Proses Perbaikan		✓	✓
11	Memberikan Ulasan Kepada Toko	Menampilkan Data Pengisian Ulasan		✓	✓
12	Mengubah Data Customer	Menampilkan Data Customer		✓	✓
13	Melihat Lokasi Maps	Menampilkan Lokasi Maps		✓	✓

14	Menerima Informasi Perbaikan Perangkat Elektronik	Menampilkan Notifikasi Konfirmasi Perbaikan Terbaru		✓	✓
15	Melakukan Logout Akun	Menampilkan Notifikasi Logout		✓	✓

Disetujui Oleh :



VALIDATOR SISTEM

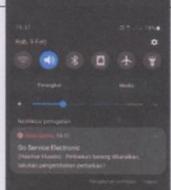
20 Januari 2022

Validator I
 Nama Validator : Ahwat
 Jabatan : Pemilik Maju Jaya Computer

Validator II
 Nama Validator : Arjoko Siahaan, S.Kom
 Jabatan : Pemilik Hardware Laptop

Pengujian Sistem Pada Aplikasi Admin

No	Rancangan Input/Output	Hasil Yang Diharapkan	Tampilan	Validator 1	Validator 2
1	Membuka sistem	Masuk ke halaman login		✓	✓
2	Melakukan Login	Login Sukses, Menampilkan Halaman Home		✓	✓

3	Melakukan Login	Login Gagal, Memunculkan Notifikasi		✓	✓
4	Melakukan Register	Masuk ke halaman Upload Data Toko		✓	✓
5	Melakukan Chatting	Menampilkan Data Chat		✓	✓
6	Menerima Chat Customer	Menampilkan Notifikasi Chat		✓	✓
7	Menerima Servis Terbaru	Menampilkan Notifikasi Servis Terbaru		✓	✓

8	Melakukan Cek Data Servis Terbaru	Menampilkan Data Servis		✓	✓
9	Melakukan Cek Lokasi Customer	Menampilkan Data Lokasi Maps Customer		✓	✓
10	Melakukan Update Data Kerusakan	Menampilkan Data Update Kerusakan		✓	✓
11	Melakukan Tambah Data Kerusakan	Menampilkan Form Tambah Data Kerusakan		✓	✓

12	Menambah Data Estimasi Biaya Perbaikan	Menampilkan Data Penambahan Estimasi Biaya Perbaikan		✓	✓
13	Menambah Jadwal Toko Buka	Menampilkan Data Penambahan Jadwal Toko Buka		✓	✓
14	Melakukan Update Data Toko	Kembali Ke Halaman Home		✓	✓

15	Melakukan Logout Akun	Menampilkan Notifikasi Logout		✓	✓
----	-----------------------	-------------------------------	---	---	---

Disetujui Oleh :

