

**SISTEM INFORMASI *TRACKING REAL TIME* ANGKUTAN
UMUM PADA HALTE KOTA MEDAN**

SKRIPSI

HABIB ASY MUHYI

72154036



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

**SISTEM INFORMASI *TRACKING REAL TIME* ANGKUTAN
UMUM PADA HALTE KOTA MEDAN**

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

HABIB ASY MUHYI

72154036



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri 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,

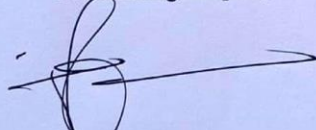
Nama	: Habib Asy Muhyi
Nomor Induk Mahasiswa	: 72154036
Program Studi	: Sistem Informasi
Judul	: Sistem Informasi <i>Tracking Real Time</i> Angkutan Umum Pada Halte Kota Medan

dapat disetujui untuk segera *dimunaqasyahkan*. Atas perhatiannya kami ucapkan terima kasih.

Medan, 09 Januari 2020 M
13 Jumadil Awal 1441 H

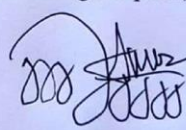
Komisi Pembimbing,

Pembimbing Skripsi I,



Ilka Zufria, M.Kom
NIP. 198506042015031006

Pembimbing Skripsi II,



Muhammad Dedi Irawan, M.Kom
NIP. 199001312019031019

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Habib Asy Muhyi
Nomor Induk Mahasiswa : 72154036
Jurusan : Sistem Informasi
Judul : Sistem Informasi *Tracking Real Time*
Angkutan Umum Pada Halte Kota Medan

Menyatakan dengan ini bahwa sesungguhnya skripsi saya yang berjudul di atas adalah asli karya saya sendiri, kecuali kutipan-kutipan yang disebutkan sumbernya. Saya bersedia menerima segala konsekuensinya apabila pernyataan saya tidak benar.

Demikianlah surat ini saya perbuat dengan sebenar-benarnya untuk dapat dipergunakan sebagaimana mestinya.

Medan, 09 Januari 2020

Penulis,



Habib Asy muhyi

NIM. 72154036



**KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERISUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: B.152/ST/ST.V.2/PP.01.1/09/2021

Judul : Sistem Informasi *Tracking Real Time* Angkutan Umum Pada Halte Kota Medan
Nama : Habib Asy Muhyi
Nomor Induk Mahasiswa : 72154036
Program Studi : Sistem Informasi
Fakultas : Sains dan Teknologi

Telah dipertahankan dihadapan Dewan Penguji Skripsi Program Studi Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan LULUS.

Pada hari/tanggal : Kamis, 30 Januari 2020
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,
Ketua,

Samsudin, ST., M.Kom
NIP. 197612272011011002

Dewan Penguji,

Penguji I,

Ilka Zulfria, M.Kom
NIP. 198506042015031006

Penguji II,

Sriani, M.Kom
NIB. 1100000108

Penguji III,

Triase, ST., M.Kom
NIP. 1100000122

Penguji IV,

M. Irwan Padli Nasution, ST., MM., M.Kom
NIP. 197502132006041003

Mengesahkan,

Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,



Dr. G. M. Jamil, M.A.
NIP. 196609101999031002

ABSTRAK

Dalam perkembangan teknologi informasi yang sangat maju dan *modern* yang berhubungan erat dengan *software*, *hardware* dan *brainware*. Tentu itu semua sangat membantu masyarakat dalam menjalankan kehidupannya sehari-hari. Seperti halnya penggunaan *smartphone* setiap hari makin bertambah penggunaannya. Contoh teknologi *smartphone* adalah penggunaan GPS (*Global Positioning System*) yang bisa mendeteksi keberadaan seseorang hingga menjadi petunjuk arah. Berdasarkan hal tersebut penulis tertarik melakukan penelitian dengan tema *tracking real time* angkutan umum pada halte. Pada saat observasi, sebagian masyarakat dan supir tidak mematuhi peraturan pemerintah yang mengharuskan berhenti pada tempatnya yaitu di halte. Sehingga warga merasa tidak nyaman karena sering terjadi kemacetan disebabkan angkot yang berhenti sembarang tempat. Maka harus ada inovasi baru dalam mewujudkan lalu lintas yang tertib. Maka dari itu dibutuhkan aplikasi *tracking real time* berbasis android dengan memanfaatkan fitur GPS. Aplikasi ini sangat membantu masyarakat, *customer*, dan *driver* dalam hal angkutan umum. Pembuatan aplikasi ini menggunakan Android Studio dan untuk metode pengembangan sistemnya menggunakan *Rapid Application Development* (RAD).

Kata Kunci : *Android Application, GPS, Angkutan Umum, Halte, RAD*

ABSTRACT

In the development of very advanced and modern information technology which is closely related to software, hardware and brainware. Of course, this is very helpful for the community in carrying out their daily lives. As with the use of smartphones every day the number of users is increasing. An example of smartphone technology is the use of GPS (Global Positioning System) which can detect a person's whereabouts to become directions. Based on this, the author is interested in conducting research with the theme of tracking real time public transportation at the bus stop. At the time of observation, some people and drivers did not comply with government regulations requiring them to stop in their place, namely at the bus stop. So that residents feel uncomfortable because traffic jams often occur due to public transportation that stops anywhere. So there must be new innovations in realizing orderly traffic. Therefore, an android-based real time tracking application is needed by utilizing the GPS feature. This application is very helpful for the community, customers, and drivers in terms of public transportation. Making this application using Android Studio and for the system development method using Rapid Application Development (RAD).

Keywords : *Android Application, GPS, Public Transportation, Bus Stop, RAD*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah yang telah melimpahkan rahmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan penyusunan skripsi ini dengan judul “Sistem Informasi *Tracking Real Time* Angkutan Umum Pada Halte Kote Medan”. Dalam menyelesaikan Skripsi ini, tentunya banyak mendapat bantuan dan saran dari pihak-pihak yang telah memberikan kontribusi dan dukungan kepada penulis dalam proses penyusunan skripsi. Maka penulis mengucapkan banyak terima kasih kepada :

1. Rektor UIN Sumatera Utara Medan, Bapak Prof. Dr. Saidurrahman, M.Ag.
2. Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan, Bapak Dr. H. M. Jamil, M.A.
3. Ketua Program Studi Sistem Informasi UIN Sumatera Utara Medan, Bapak Samsudin, ST., M.Kom.
4. Sekretaris Program Studi Sistem Informasi UIN Sumatera Utara Medan, Ibu Triase, ST., M.Kom.
5. Dosen Penasehat Akademik, Bapak Prof. Dr. Al Rasyidin, M.Ag.
6. Bapak Muhammad Irwan Padli Nasution, ST, MM, M.Kom selaku Dosen yang telah memberikan arahan dalam menentukan judul skripsi.
7. Dosen Pembimbing I, Bapak Ilka Zufria, M.Kom yang telah membimbing penulis selama menyelesaikan skripsi ini.
8. Dosen Pembimbing II, Bapak Muhammad Dedi Irwan, M.Kom yang telah membimbing penulis selama menyelesaikan skripsi ini.
9. Bapak dan Ibu Dosen Program Studi Sistem Informasi yang telah membimbing penulis selama mengikuti perkuliahan.
10. Papa saya bernama Asyaruddin dan Mamak saya bernama Lily Herlina serta keluarga yang telah memberikan doa, dukungan, dan semangat dalam mengerjakan skripsi ini.
11. Teman-Teman yang telah memberi semangat dalam menyelesaikan skripsi ini.

Apabila dalam penulisan skripsi ini terdapat bahasa dan penulisan yang tidak sesuai dengan aturan yang telah ditetapkan, oleh karena itu penulis memohon maaf kepada semua pihak karena penulis tidak melakukan hal tersebut dengan sengaja, tetapi kesalahan penulis pada saat mengerjakan skripsi ini. Semoga Allah selalu memberikan kesehatan, rahmat dan hidayah-Nya kepada kita semua.

Medan, 09 Januari 2020

Penulis

Habib Asy Muhyi

DAFTAR ISI

LEMBAR PERSETUJUAN SKRIPSI	i
SURAT PERNYATAAN KEASLIAN SKRIPSI	ii
LEMBAR PENGESAHAN SKRIPSI	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Sistem	5
2.2 Informasi	5
2.3 Sistem Informasi	6
2.4 <i>Tracking</i>	6
2.5 <i>Real Time</i>	6
2.6 <i>Tracking Real Time</i>	6
2.7 Angkutan Umum.....	7
2.8 Halte	7

2.9	Java Mobile	7
2.10	Android	8
2.11	Sejarah Perkembangan Android	9
2.12	Versi Android.....	100
2.13	Android Studio.....	111
2.14	Google API	133
2.15	Android SDK	155
2.16	Xml	155
2.17	<i>Activity</i>	166
2.18	<i>Intent</i>	166
2.19	<i>View dan Widget</i>	166
2.20	GPS	177
2.21	Firestore <i>Realtime Database</i>	177
2.22	<i>Rapid Application Development (RAD)</i>	188
2.23	<i>Unified Modeling Language (UML)</i>	188
2.23.1	<i>Use Case Diagram</i>	19
2.23.2	<i>Activity Diagram</i>	200
2.23.3	<i>Sequence Diagram</i>	222
2.23.4	<i>Class Diagram</i>	244
BAB III METODOLOGI PENELITIAN		277
3.1	Tempat dan Waktu Penelitian	277
3.1.1	Tempat Penelitian	277
3.1.2	Waktu Penelitian.....	277
3.2	Kebutuhan Sistem	300
3.3	Cara Kerja	300

3.4	Kerangka Berfikir	333
BAB IV HASIL DAN PEMBAHASAN		344
4.1	Profil Perusahaan	344
4.1.1	Struktur Organisasi	344
4.1.2	<i>Job Desk</i> Struktur Organisasi	355
4.1.3	Analisis Sistem Berjalan.....	355
4.1.4	Perancangan Sistem Usulan.....	355
4.2	Perancangan Sistem	366
4.2.1	<i>Use Case Diagram</i>	37
4.2.2	<i>Activity Diagram</i>	38
4.2.3	<i>Sequence Diagram</i>	422
4.2.4	<i>Class Diagram</i>	466
4.3	Rancangan <i>Database</i>	477
4.4	Rancangan Tampilan Antar-muka (<i>interface</i>)	4848
4.5	<i>Test</i> (Pengujian)	500
4.5.1	Menentukan Hak Akses.....	555
4.5.2	Implementasi.....	5555
BAB V KESIMPULAN DAN SARAN		Error! Bookmark not defined.7
5.1	Kesimpulan	57
5.2	Saran	57
DAFTAR PUSTAKA		588
LAMPIRAN		

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Tampilan Android Studio	12
2.2	Tampilan untuk mendapatkan API <i>key</i>	14
2.3	Tampilan awal API <i>Library</i>	14
2.4	Contoh <i>Use Case Diagram</i>	20
2.5	Contoh <i>Activity Diagram</i> Kaprodi.....	22
2.6	Contoh <i>Sequence Diagram</i> Sekretaris.....	24
2.7	Contoh <i>Class Diagram</i>	26
3.1	Kerangka Berfikir	33
4.1	Struktur Organisasi PT. Rahayu Medan Ceria	34
4.2	Gambaran Sistem Usulan	36
4.3	<i>Use Case Diagram</i>	37
4.4	<i>Activity Diagram</i> Admin	38
4.5	<i>Activity Diagram</i> Customer	39
4.6	<i>Activity Diagram</i> Driver.....	41
4.7	<i>Sequence Diagram</i> Register Driver.....	42
4.8	<i>Sequence Diagram</i> Register Customer.....	43
4.9	<i>Sequence Diagram</i> Sign In Customer Dan Driver	43
4.10	<i>Sequence Diagram</i> Permintaan Penjemputan Customer	44
4.11	<i>Sequence Diagram</i> Jemput Customer	45
4.12	<i>Sequence Diagram</i> Settings Customer Dan Driver	45
4.13	<i>Sequence Diagram</i> Logout Customer Dan Driver	46
4.14	<i>Class Diagram</i>	46

4.15	Rancangan <i>Database</i>	47
4.16	Rancangan Tampilan Halaman <i>Sign In</i> dan <i>Register Customer</i>	48
4.17	Rancangan Tampilan Halaman <i>Sign In Driver</i>	48
4.18	Rancangan Tampilan Halaman Utama <i>Customer</i>	49
4.19	Rancangan Tampilan Halaman Utama <i>Driver</i>	49
4.20	Tampilan Halaman <i>Sign In Customer</i> Dan <i>Driver</i>	51
4.21	Tampilan Halaman Utama <i>Customer</i>	52
4.22	Tampilan Halaman <i>Settings Customer</i>	53
4.23	Tampilan Halaman Utama <i>Driver</i>	53
4.24	Tampilan <i>Driver</i> Menerima Permintaan <i>Customer</i>	54
4.25	Tampilan Halaman <i>Settings Driver</i>	54

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	Daftar Versi Android.....	10
2.2	Daftar Simbol <i>Use Case Diagram</i>	19
2.3	Daftar Simbol <i>Activity Diagram</i>	21
2.4	Daftar Simbol <i>Sequence Diagram</i>	23
2.5	Daftar Simbol <i>Class Diagram</i>	25
3.1	Jadwal Penelitian	27
4.1	<i>Black Box Testing</i>	50

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	<i>Source Code</i>
2.	Surat Balasan Penelitian

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perubahan lingkungan eksternal memiliki dampak yang sangat besar pada perilaku masyarakat serta dalam bisnis saat ini, salah satu diantaranya adalah perkembangan teknologi informasi yang telah memicu globalisasi (Nasution, 2014). Dalam industri teknologi yang berkembang begitu pesat, serta membutuhkan SDM (Sumber Daya Manusia) yang memadai guna mendukung pembuatan teknologi yang terbaru, maka diperlukan seorang yang ahli dalam pembuatan teknologi, serta mampu menemukan masalah *global* dalam mendukung teknologi yang terbaru. Pada penelitian ini, masalah yang ditemukan khususnya daerah kota Medan adalah tentang angkutan umum dan penggunaan halte yang ada di kota Medan.

Berdasarkan data dari *website* koran SINDO Medan tahun 2015 yang dipublikasikan oleh Syukri Amal, mengatakan bahwa di kota Medan masih sering supir angkutan kota (angkot) berhenti di pinggir jalan yang banyak dilalui pengendara untuk menunggu penumpang terutama di persimpangan jalan yang membuat lalu lintas menjadi macet. Sehingga pengguna jalan menjadi marah. Seperti di persimpangan jalan Iskandar Muda dan Gatot Subroto Medan angkot berhenti disembarang tempat. Maka diperlukan aplikasi untuk supir dan penumpang yang memanfaatkan Halte sebagai tempat pemberhentian Angkot. Berdasarkan data dari *website* Sumut Post yang dipublikasikan oleh Triadi Wibowo, mengatakan bahwa jumlah angkutan umum pada tahun 2018 di kota Medan tidak sampai 5000 unit. Organisasi angkutan darat (Organda) kota Medan mencatat dari sepuluh ribuan angkot yang sebelumnya beroperasi di kota Medan jumlahnya tidak sampai 5.000 unit. Ketua Organda kota Medan Mont Gomery Munthe dalam websitenya Sumut Post mengatakan hampir 5.000 angkutan massal itu terdiri dari berbagai macam perusahaan seperti PT Rahayu Medan Ceria (RMC), KPUM, CV Medan Bus, PT U Morina, PT Nasional dan PT Mars.

Kemudian permasalahan selanjutnya yang ada di kota Medan adalah penggunaan halte di kota Medan yang tidak berfungsi secara maksimal dan tidak terawat. Menurut pantauan dari *website* Medan Bisnis Daily tahun 2017 yang dipublikasikan oleh Sasli Pranoto Simarmata, mengatakan bahwa kondisi fisik halte yang telah rusak dan dikotori oleh poster beberapa perusahaan untuk berpromosi. Serta halte tidak bermanfaat sesuai fungsi, seperti halte di depan Ramayana Teladan di jalan Sisingamangaraja dan di depan Medan Fair Plaza jalan Gatot Subroto beralih fungsi sebagai tempat tongkrongan becak bermotor untuk menunggu penumpang. Dalam permasalahan tersebut, maka peneliti mengajukan judul “SISTEM INFORMASI *TRACKING REAL TIME* ANGKUTAN UMUM PADA HALTE KOTA MEDAN” yang akan menampilkan informasi angkutan umum berdasarkan *Tracking Real Time* dengan menggunakan aplikasi android. Dimana *Tracking Real Time* ini bertujuan untuk mengetahui posisi angkutan umum secara *Real Time* atau secara langsung tanpa melakukan *delay* serta membuat halte berfungsi secara maksimal dengan cara penumpang hanya bisa mengakses aplikasi angkutan umum di Halte tersebut. Saya membuat aplikasi tersebut untuk mempermudah masyarakat dalam hal transportasi umum, karena agama islam mengajarkan kita untuk memudahkan urusan orang lain, maka Insya Allah Urusan kita akan dipermudah oleh Allah.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang di atas, maka rumusan masalahnya adalah sebagai berikut :

1. Bagaimana rancangan sistem informasi *tracking real time* angkutan umum pada perusahaan PT Rahayu Medan Ceria ?
2. Bagaimana rancangan sistem agar halte bisa berfungsi secara maksimal ?
3. Bagaimana pemilik perusahaan angkutan umum PT Rahayu Medan Ceria (RMC) memonitoring armada yang beroperasi ?

1.3 Batasan masalah

Dalam pembuatan sistemnya, maka batasan masalahnya adalah sebagai berikut :

1. Pada sistem tersebut khusus untuk kota Medan pada perusahaan angkutan umum PT Rahayu Medan Ceria.
2. Pada sistem tersebut menggunakan nomor angkutan umum 103, 104, dan 42. Nomor angkutan tersebut melewati jalan yang memiliki 6 halte mulai dari Jl. Prof. HM.Yamin Sh – Jl. Gatot Subroto. Setiap nomor angkutan umum bisa mengoperasikan 3 armada.
3. Penggunaan aplikasi berbasis android untuk *driver* dan *customer*.
4. Dalam pembuatan aplikasinya dengan menggunakan Android Studio.
5. *Customer* hanya bisa menggunakan aplikasinya di Halte dengan memanfaatkan sistem lokasi.
6. Membuat *tracking real time* angkutan umum dengan memanfaatkan *library* Api Google Maps.

1.4 Tujuan Penelitian

Berdasarkan latar belakang masalah, kenapa sistem ini perlu dibuat. Adapun tujuan penelitiannya sebagai berikut :

1. Mengembangkan aplikasi *tracking real time* untuk digunakan angkutan umum pada halte di kota Medan agar membuat halte di kota Medan berfungsi secara maksimal guna membantu pemerintah dalam penggunaan halte.
2. Dengan adanya sistem tersebut, maka akan membuat *driver* tidak menaikan dan menurunkan penumpang sembarang tempat.
3. Merancang sistem yang memudahkan *customer* dalam mengakses angkutan umum.
4. Merancang sistem yang bisa langsung melakukan *monitoring* oleh pihak PT. Rahayu Medan Ceria.

1.5 Manfaat Penelitian

Dalam penelitian ini, bagaimana sistem yang sudah dibuat bermanfaat bagi *customer*, *driver*, penulis, universitas, masyarakat dan PT. Rahayu Medan Ceria, Adapun manfaat penelitiannya sebagai berikut :

1. *Customer* mengetahui nomor dan lokasi *driver* angkutan umum pada aplikasi android yang sudah terinstal di *smartphone customer*.
2. *Driver* angkutan umum mengetahui lokasi penjemputan *customer* di Halte. Sehingga menjadi teratur dan disiplin dalam menurunkan dan menaikkan penumpang di Halte.
3. Peneliti memahami sistem informasi *tracking real time* guna membuat ahli pada bidang tersebut unutup mendukung karir dimasa depan.
4. Mahasiswa-mahasiswi yang setiap ajaran baru di UINSU Medan mempunyai rujukan penelitian tentang Sistem Informasi Geografis.
5. Masyarakat merasa nyaman dan aman di jalan dengan tidak terjadi kemacetan yang disebabkan angkutan umum berhenti sembarang tempat.
6. Pemilik perusahaan PT Rahayu Medan Ceria dapat *monitoring* armada angkutan umum yang sedang beroperasi.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem adalah seperangkat elemen, komponen, atau variabel yang terorganisir, saling bergantung, terintegrasi, dan berinteraksi satu sama lain. Kenneth Boulding adalah orang pertama yang menjelaskan teori sistem secara umum, yang menekankan pentingnya setiap bagian yang membentuk sistem. Terlalu fokus pada satu komponen sistem organisasi adalah kecenderungan orang untuk menjalankan organisasi. Setiap elemen yang membentuk organisasi dalam teori sistem adalah penting dan harus mendapat perhatian yang cukup agar para manajer dapat bertindak lebih efektif. Bukan hanya bagian fisik yang berwujud yang diwakili oleh elemen-elemen yang membentuk organisasi, tetapi juga hal-hal yang dapat konseptual atau abstrak seperti kegiatan, kelompok informal, misi, pekerjaan, dll (Sutrabri, 2014).

Berdasarkan penjelasan di atas, sistem merupakan kumpulan komponen yang terorganisir dan terintegrasi satu sama lain untuk mewujudkan tujuan dalam sebuah organisasi ataupun perusahaan.

2.2 Informasi

Mengatur pengumpulan data dengan nilai pengetahuan bagi pengguna, hasil pemrosesan, dan manipulasi disebut informasi. Menurut Kadir dan Triwahyuni (2013), informasi digunakan sebagai bahan pengambilan keputusan merupakan sumber daya yang penting dalam sebuah organisasi. Sehubungan dengan hal itu, informasi haruslah berkualitas (Sutrabri, 2014).

Berdasarkan pengertian di atas, informasi merupakan hasil pemrosesan dari suatu data yang banyak untuk menghasilkan informasi yang *valid*.

2.3 Sistem Informasi

Memahami sistem informasi yang dijelaskan oleh Alter, kombinasi prosedur bisnis, orang, informasi, dan teknologi informasi yang diatur untuk mencapai tujuan organisasi, dikenal sebagai sistem informasi. Membentuk sistem informasi dengan saling berinteraksi yang akan memperlihatkan komponen-komponennya (Kadir dan Triwahyuni, 2013).

Berdasarkan pengertian di atas, sistem informasi merupakan suatu pengorganisasian dalam prosedur kerja yang memberikan data berupa informasi dalam mewujudkan tujuan sebuah organisasi ataupun perusahaan.

2.4 Tracking

Proses perekaman dan *monitoring* sepanjang perjalanan menuju suatu tempat disebut *tracking*. Contoh *tracking* antara lain *monitoring* posisi kendaraan pelayanan *public* seperti *ambulans* dan *bus*. Serta pemetaan jalan, sungai, dan lainnya (Marjuki, 2016).

Berdasarkan pengertian tersebut, maka *tracking* dapat disimpulkan merupakan proses *monitoring* dalam melakukan pelacakan suatu *object* bergerak untuk mengetahui posisi *object* tersebut.

2.5 Real Time

Mengetahui dan mengamati kondisi jalan di seluruh dunia secara langsung atau disebut *real time*, yang visualisasi kondisi jalannya di *capture* melalui kamera secara langsung atau dalam bentuk peta (Irwansyah, 2013).

Bedasarkan pengertian tersebut, maka *real time* dapat disimpulkan merupakan proses visualiasi bagi pengguna untuk mengetahui kondisi pada *object* tertentu secara langsung tanpa *delay*.

2.6 Tracking Real Time

Berdasarkan pengertian *Tracking* dan *Real Time* di atas, maka dapat disimpulkan *tracking real time* merupakan proses *monitoring* dalam melakukan pelacakan suatu *object* yang sudah terpasang smart alat pelacakannya untuk mengetahui pergerakan dan posisi *object* tersebut secara langsung tanpa *delay*.

2.7 Angkutan Umum

Dalam undang-undang No. 22 Tahun 2009 tentang pengertian angkutan umum perkotaan disebutkan bahwa orang atau barang dari suatu tempat ke tempat yang lain dengan menggunakan kendaraan dalam kawasan perkotaan yang terikat dalam trayek tetap dan teratur (Mutiawati, 2019). Berdasarkan pengertian tersebut, maka angkutan umum dapat disimpulkan merupakan alat transportasi *public* yang dalam penggunaannya tertib berlalu lintas dengan tidak berhenti sembarang tempat, tetapi berhenti di halte yang sudah disediakan oleh pemerintah agar pengguna jalan lainnya merasa aman dan nyaman dalam berkendara.

2.8 Halte

Halte adalah suatu tempat pemberhentian angkutan umum ataupun bus untuk melakukan kegiatan menaikkan dan menurunkan penumpang. Dalam penggunaannya, khususnya halte daerah kota Medan masih banyak yang belum berfungsi secara maksimal. Berdasarkan penjelasan latar belakang masalah diatas, terlihat dari kondisi fisik halte yang telah rusak dan dikotori oleh poster beberapa perusahaan untuk berpromosi. Serta halte tidak bermanfaat sesuai fungsi, seperti halte di depan Ramayana Teladan di jalan Sisingamangaraja beralih fungsi sebagai tempat tongkrongan becak bermotor untuk menunggu penumpang. Berdasarkan penjelasan tersebut, maka pemanfaatan halte haruslah maksimal dengan merawat dan menjaga kebersihannya agar masyarakat yang menunggu transportasi umum merasa nyaman berada di halte.

2.9 Java Mobile

Java merupakan dasar pemrograman Android karena aplikasi Android ditulis dalam bahasa Java. Lingkungan runtime Java yang disediakan oleh Android disebut Mesin Virtual Dalvik, yang telah dioptimalkan untuk perangkat dengan memori sistem kecil (EMS, 2013).

Java diciptakan dalam proyek Sun Microsystems oleh tim yang dipimpin oleh Patrick Naughton dan James Gosling, yang mengkodekan Green untuk membuat bahasa komputer sederhana yang dapat berjalan pada perangkat sederhana tanpa terikat pada arsitektur tertentu. Terakhir teknologi Java pada

tahun-tahun berikutnya mulai dari tahun (1998-1999) lahir J2ME (Java 2 Micro Edition) yang diadopsi oleh Nokia, Siemens, Sony Ericsson, Motorola, Samsung untuk memproduksi aplikasi *mobile* seperti *game*, *software* bisnis dan jenis *software* lainnya yang dapat dijalankan pada perangkat *mobile* seperti ponsel (Rickyanto, 2003).

Program yang biasa digunakan pada pager, PDA dan smartphone hingga saat ini adalah J2ME (Java 2 Micro Edition). J2ME adalah program yang menggunakan bahasa pemrograman Java yang mencakup Java Virtual Machine (JVM) yang digunakan untuk menjalankan program Java pada emulator atau perangkat seluler dan Java API (Application Programming Interface) dan alat lain untuk mengembangkan aplikasi Java seperti Motorola Emulator, Java Emulator dan J2ME Wireless Toolkit. J2ME adalah sekumpulan spesifikasi yang berfokus pada perangkat konsumen dengan kapasitas memori terbatas, layar kecil, konsumsi baterai rendah, dan jaringan lemah (Wahana, 2014). Berdasarkan penjelasan tersebut, maka Java *Mobile* dapat disimpulkan merupakan bahasa pemrograman khusus dalam pembuatan aplikasi android.

2.10 Android

Salah satu sistem operasi smartphone yang paling baru muncul adalah Android. Dengan jumlah pengguna sistem operasi Android saat ini telah meningkat banyak. Alasannya, antara lain, dukungan vendor smartphone yang mengimplementasikan sistem operasi luar biasa ini di lini produknya. Jadi ketika membeli ponsel atau tablet, pemain secara otomatis menggunakan sistem operasi Android. Android adalah nama perangkat lunak yang digunakan pada perangkat *mobile* yang terdiri dari berbagai komponen yaitu *middleware*, sistem operasi dan aplikasi utama yang dirilis oleh Google. Android mencakup semua aplikasi, mulai dari sistem operasi hingga pengembangan aplikasi Android itu sendiri. Menggunakan bahasa pemrograman Java dasar dalam mengembangkan aplikasi di platform Android. Dalam perkembangannya, aplikasi Android memiliki kode sumber terbuka atau lisensi terbuka, memungkinkan mereka untuk membuat aplikasi yang kaya dan inovatif (EMS, 2013).

Android sistem operasi yang mendukung dalam pembuatan aplikasi berbasis Android. Sehingga sangat membantu masyarakat dalam melakukan aktifitasnya sehari-hari dengan mudah dan lebih efisien dengan menggunakan *smartphone*.

2.11 Sejarah Perkembangan Android

Pada tahun 2000, Google mengakuisisi sebuah perusahaan bernama Android. Berfokus pada pengembangan sistem operasi dan perangkat lunak tertanam untuk perangkat seluler, perusahaan ini adalah perusahaan yang relatif baru. Pendiri Android Andy Rubin, Rich Miner, Nick Sears dan Chris White pindah ke Google setelah Android Inc. adalah Perusahaan Google. Sistem operasi Android pada saat itu belum memiliki lisensi open source dan dikembangkan hanya untuk kebutuhan internal Google. Evolusi Google membentuk Open Handset Alliance (OHA) untuk meningkatkan adopsi teknologi Android.

OHA adalah perusahaan perangkat keras, perangkat lunak dan telekomunikasi dengan lebih dari 34 anggota. Di antara perusahaan tersebut adalah HTC, Intel, Qualcomm, T-Mobile, Motorola dan Nvidia, Asus. Pada tanggal 5 November 2007, dengan OHA, Android pertama kali dirilis dengan standar open source, alasan Google merilis kode open source Android adalah untuk menembus pasar produk berbasis Android yang digunakan oleh Google. digunakan sebagai media baru dengan penayangan iklan dapat meningkatkan pendapatan Google. , sebagian untuk mengembangkan sistem operasi dan aplikasi Android, dan untuk memfasilitasi akses pengguna ke layanan Google (Istiyanto, 2013).

Sistem operasi Android yang pertama kali di gunakan hanya untuk internal google saja. Setelah perkembangan zaman, google membuat sistem operasi Android dalam bentuk *open source*.

2.12 Versi Android

Android masih terus berkembang hingga saat ini dan ditandai dengan versinya. Yang digunakan untuk mengidentifikasi application programming interface (API) pada versi Android adalah dengan memberikan API dengan mendeklarasikan nomor unik. Setiap versi Android ditandai dengan API level (Kadir, 2013).

Tabel 2.1 Daftar Versi Android

(Sumber : [Https://Mainthebest.Com/Smartphones/Tingkatan-Versi-Android/](https://Mainthebest.Com/Smartphones/Tingkatan-Versi-Android/))

Nama	Versi	Tahun Peluncuran
Cupcake	1.5	27 April 2009
Donut	1.6	15 September 2009
Eclair	2.0 – 2.1	26 Oktober 2009
Froyo	2.2 – 2.2.3	20 Mei 2010
Gingerbread	2.3 – 2.3.7	6 Desember 2010
Honeycomb	3.0–3.2.6	22 Pebruari 2011
Ice Cream Sandwich	4.0 – 4.0.4	18 Oktober 2011
Jelly Bean	4.1 – 4.3.1	9 Juli 2012
KitKat	4.4 – 4.4.4	31 Oktobe 2013
Lollipop	5.0 – 5.1.1	12 November 2014
Marshmallow	6.0 – 6.0.1	5 Oktober 2015
Nougat	7.0	Agustus / September 2016
Oreo	8.0	Agustus 2017
Pie	9.0	Agustus 2018

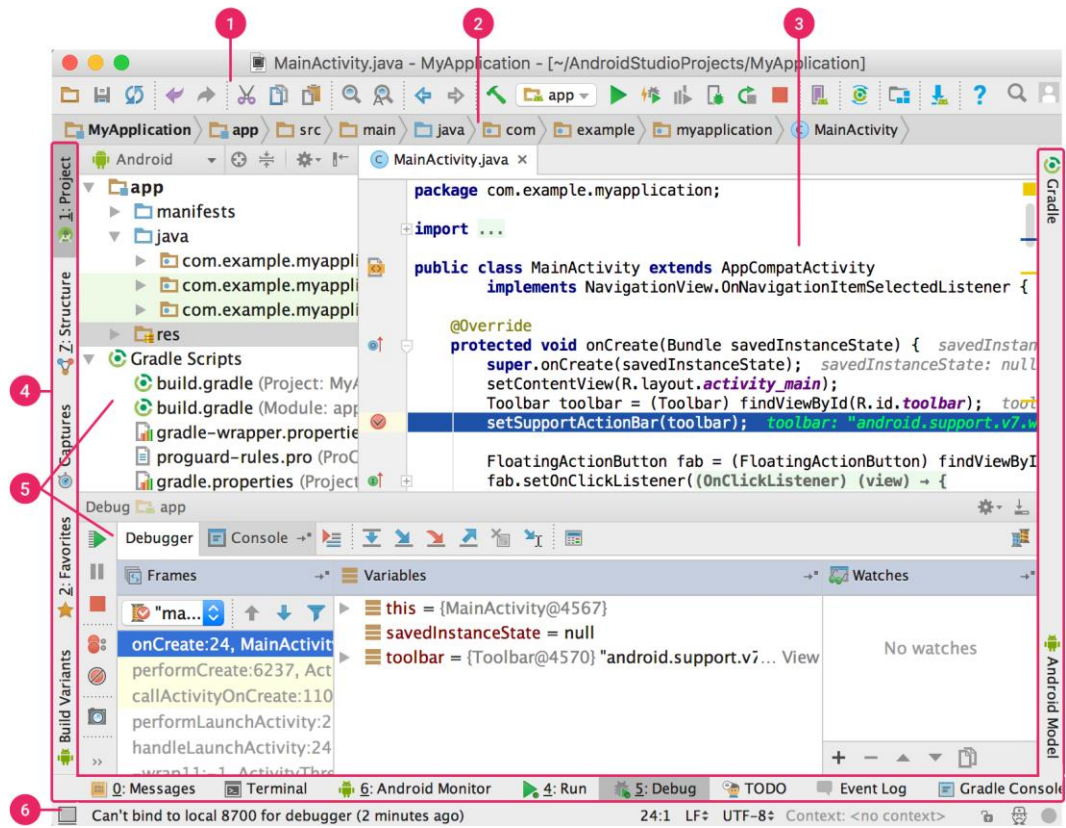
Sistem operasi android mempunyai versi yang berbeda-beda seiring berjalannya waktu dan ada perubahan dan penambahan fitur yang membuatnya lebih baik dari versi sebelumnya, seperti sistem operasi Android terbaru yang telah menggunakan kecerdasan buatan (AI) yaitu pie.

2.13 Android Studio

Berdasarkan IntelliJ IDEA, Android Studio adalah lingkungan pengembangan terintegrasi (IDE) untuk mengembangkan aplikasi Android terintegrasi. Selain pengembang dan editor kode IntelliJ yang efektif, di Android Studio menyediakan lebih banyak fitur untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

1. Berbasis Gradle yang sistem versinya fleksibel
2. Emulator cepat dan kaya fitur
3. Lingkungan menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run mendorong untuk perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi dengan GitHub yang bertujuan untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja ekstensif
7. Alat Lint yang bertujuan untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan dengan C+ dan NDK
9. Dukungan dengan bawaan untuk Google *Cloud Platform*, yang bertujuan mempermudah pengintegrasian Google *Cloud Messaging* dan *App Engine* (Suryana, 2018).

Berdasarkan penjelasan tersebut, maka Android Studio dapat disimpulkan merupakan *tools* pembuatan aplikasi android yang terbaik hingga saat ini, karena disupport oleh Google yang terdapat banyak *library* bisa di ambil dari Google *cloud platform* guna mendukung aplikasi andorid yang elegan dan bermanfaat bagi masyarakat.



Gambar 2.1 Tampilan Android Studio

(Sumber : <https://developer.android.com/studio/intro?hl=ID>)

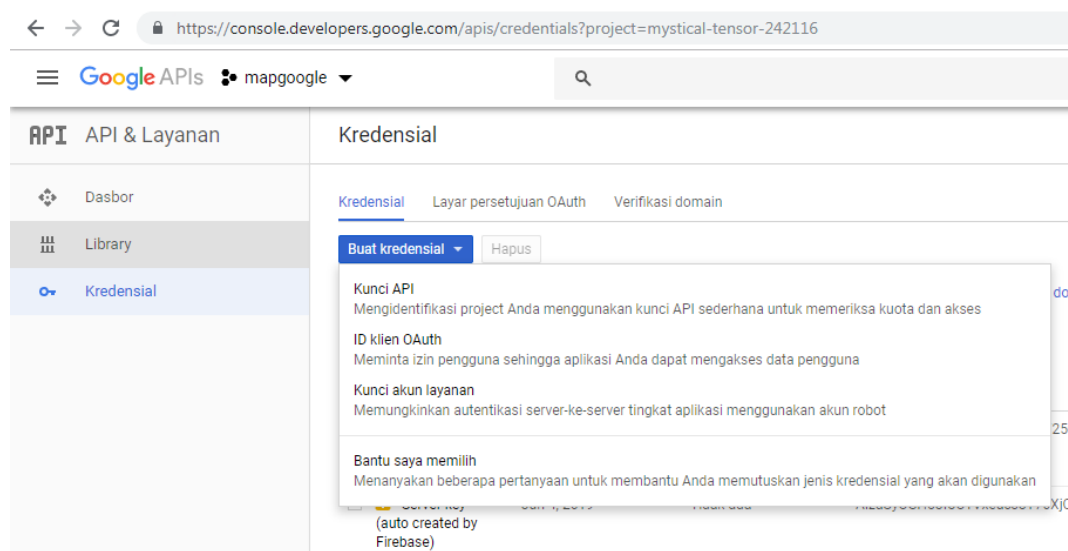
Berdasarkan gambar di atas, ada beberapa hal penting dalam membangun aplikasi Android. Berikut penjelasannya:

1. Toolbar digunakan untuk melakukan berbagai tindakan, seperti menjalankan aplikasi dan meluncurkan fitur Android.
2. Menu Browse digunakan untuk menelusuri proyek dan membuka file untuk diedit. Menu ini memberikan tampilan yang lebih ringkas dari struktur yang ditemukan di jendela Proyek.
3. Jendela editor adalah tempat Anda dapat membuat dan mengedit kode. Editor ini memiliki kemampuan untuk berkembang sesuai dengan jenis file yang ada. Misalnya, saat melihat file tata letak, editor akan menampilkan editor tata letak.

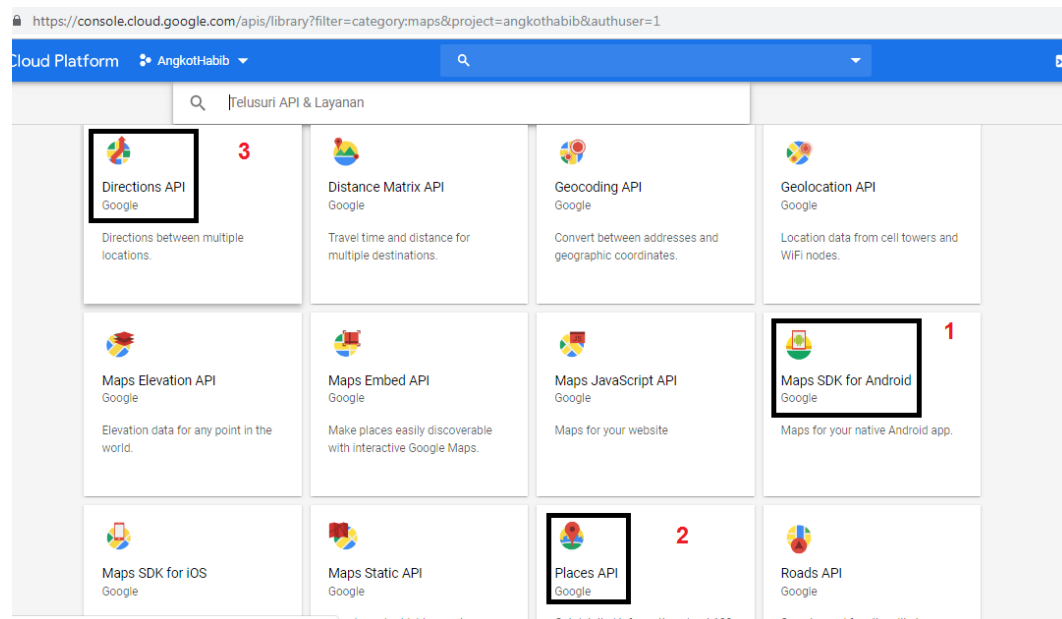
4. Panel jendela fitur terletak di luar jendela IDE dan berisi tombol yang memungkinkan Anda untuk memaksimalkan atau meminimalkan setiap jendela fitur.
5. Jendela alat menyediakan akses ke tugas-tugas tertentu seperti manajemen proyek, navigasi, kontrol versi, dan banyak lagi. Anda juga dapat memperluas dan menciutkan jendela ini.
6. Bilah status menunjukkan status proyek dan IDE itu sendiri, serta peringatan atau pesan apa pun.

2.14 Google API

Aplikasi Android dapat diintegrasikan dengan Google API, seperti membangun aplikasi untuk mendeteksi lokasi pengguna menggunakan peta. Ini akan memakan waktu berjam-jam atau berhari-hari dengan ratusan/ribuan kode untuk membuat sistem pemetaan. Namun, Google telah menyediakan API peta Android yang dapat digunakan langsung di aplikasi. Fitur ini pasti akan menghemat waktu dan pikiran Anda ketika harus membuat ulang dari awal. Dan banyak Google API lainnya yang bisa langsung digunakan (EMS, 2013).



Gambar 2.2 Tampilan untuk mendapatkan API key
(Sumber : [Https://console.developers.google.com/apis/](https://console.developers.google.com/apis/))



Gambar 2.3 Tampilan awal API Library

(Sumber : <https://console.developers.google.com/apis/library/>)

Ada beberapa API Library Google Maps yang akan peneliti gunakan dalam pembuat aplikasinya, yaitu :

1. Maps SDK for Android, digunakan khusus aplikasi android yang dalam penggunaannya memanggil api *key* untuk memanggil mapsnya.
2. Places API, digunakan dalam hal melakukan pencarian lokasi atau tempat yang diinginkan dengan data dari google maps.
3. Direction API, digunakan untuk melakukan *polyline* dari posisi awal ke posisi tujuan dengan menampilkan estimasi waktu dan jarak tempuh.

Berdasarkan tampilan gambar tersebut, maka Google Maps API merupakan libray yang diciptakan untuk mempermudah dalam pembuatan dan pengembangan peta yang tidak perlu lagi buat dari awal.

2.15 Android SDK

Kumpulan perangkat lunak yang berisi pustaka, debugger, emulator, dokumentasi, kode sampel, dan instruksi disebut kit pengembangan perangkat lunak (SDK). Android SDK dapat diunduh secara gratis di <http://developer.android.com/sdk/>. Adanya emulator memungkinkan untuk membuat dan menguji aplikasi Android tanpa harus memiliki hardware berbasis Android, seperti halnya pada komputer yang tidak menjalankan Android. Sebenarnya, ini tidak hanya dapat diuji pada Windows tetapi juga pada platform lain seperti Linux dan Mac (Kadir, 2013). Berdasarkan penjelasan tersebut, maka android SDK dapat disimpulkan merupakan *platform* yang mendukung dalam pembuatan aplikasi android.

2.16 Xml

Tidak semua murni menggunakan bahasa pemrograman Java dalam pembuatan aplikasi android, akan tetapi menggunakan bahasa XML dan platform ApacheAnt untuk mengembangkan aplikasi Android. Yang termasuk bahasa web turunan dari SGML (Standard Generalized Markup Language) yang digunakan di Android adalah XML (eXtensible Markup Language). Bahasa XML hampir sama dengan bahasa pemrograman HTML, keduanya berasal dari SGML. Sederhananya, XML adalah bahasa yang digunakan untuk menggambarkan dan memanipulasi dokumen secara terstruktur. XML, secara teknis bahasa markup meta, menyediakan format khusus untuk dokumen yang berisi data terstruktur. Bahasa markup adalah mekanisme untuk mengenali struktur dalam dokumen. Sedangkan ApacheAnt adalah perangkat lunak berbasis Java yang digunakan untuk keperluan membangun alat. ApacheAnt, yang menyediakan sumber daya untuk menjalankan proses yang memungkinkan perangkat lunak dibangun dari kode sumber hingga aplikasi yang siap didistribusikan (EMS, 2013). Berdasarkan penjelasan tersebut, maka xml dapat disimpulkan merupakan bagian dari bahasa pemrograman android selain bahasa pemrograman java.

2.17 *Activity*

Activity adalah *Class* yang bertanggung jawab terhadap isi dari antar muka sehingga pengguna dapat berinteraksi (Junindar, 2017). Berdasarkan penjelasan tersebut, maka *activity* dapat disimpulkan merupakan kumpulan *activity* dengan kode program yang berbeda-beda yang saling berkaitan dalam menampilkan antar muka aplikasi android.

2.18 *Intent*

Intent adalah biasa yang digunakan untuk menjalankan sebuah *activity* dan *service* didalam android (Junindar, 2017). Berdasarkan pengertian tersebut, maka *intent* dapat disimpulkan merupakan pelayanan yang menghubungkan antara satu *activity* dengan yang lainnya agar bisa saling berinteraksi.

2.19 *View dan Widget*

Elemen dasar UI adalah *View*, yang merupakan area tertentu dari layar, dimungkinkan untuk menempatkan komponen dan melakukan penanganan peristiwa pada mereka. Beberapa pandangan tersebut antara lain:

1. *Context Menu*
2. *Menu*
3. *View*

Elemen *userinterface* adalah *Widget*, yaitu bertindak sebagai antarmuka untuk kemungkinan interaksi antara pengguna dan sistem Android. Beberapa utilitas di Android meliputi:

1. *Check box*
2. *Button*
3. *Image view*
4. *Relative layout* (EMS, 2013).

View dan *widget* merupakan tampilan interface pada aplikasi android baik berupa menu, *navigation*, dan *button*.

2.20 GPS

Dapat dikombinasikan sistem operasi Android dengan radio GPS memudahkan pengembang untuk membuat aplikasi yang dapat mendeteksi lokasi pengguna pada suatu titik waktu. Selain itu, juga dapat mendeteksi pergerakan atau perubahan lokasi pengguna, saat mereka berpindah dari posisi semula. Contoh aplikasinya adalah jejaring sosial foursquare yang menggunakan GPS untuk menentukan lokasi ponsel dan akses web untuk mengidentifikasi alamat publik di dekat lokasi pengguna ponsel. Atau aplikasi lain yaitu Peta atau peta dapat menentukan lokasi lokasi yang diinginkan dan kemudian menghasilkan rute yang menunjukkan bagaimana solusi jalan akan mencapai lokasi tersebut (EMS, 2013). Berdasarkan penjelasan tersebut, maka GPS dapat disimpulkan merupakan sistem lokasi pengguna yang sudah terintegrasi di dalam smartphone.

2.21 *Firestore Realtime Database*

Berdasarkan website dari firebase.google.com. adalah database yang di-host di cloud disebut Firestore Realtime Database. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis. Simpan dan sinkronkan data dengan database cloud NoSQL. Data disinkronkan pada semua klien secara realtime dan tetap tersedia meski aplikasi dalam keadaan offline.

2.22 *Rapid Application Development (RAD)*

Model proses untuk pengembangan perangkat lunak sekuensial linier yang menekankan pada siklus pengembangan yang sangat singkat dikenal sebagai model *Rapid Application Development (RAD)*. Model RAD ini merupakan adaptasi "kecepatan tinggi" dari model sekuensial linier di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika persyaratan dipahami dengan baik, proses RAD memungkinkan tim pengembangan untuk membuat "sistem yang berfungsi penuh" dalam waktu yang sangat singkat (sekitar 60-90 hari). Karena terutama digunakan dalam membangun aplikasi sistem, pendekatan RAD mencakup fase-fase berikut: pemodelan bisnis, pemodelan data, pemodelan proses, pembuatan, dan pengujian, aplikasi, dan pendapatan (Yurindra, 2017). Berdasarkan penjelasan tersebut, maka *Rapid Application Development (RAD)* dapat disimpulkan merupakan pengembangan sistem yang mempunyai waktu sedikit yang memungkinkan penggunaan biaya pun juga tidak begitu mahal.

2.23 *Unified Modeling Language (UML)*

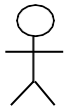


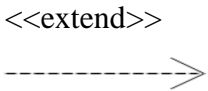

Pengertian UML (*Unified Modeling Language*) yang diuraikan oleh Dharwiyanti adalah bahasa berbasis grafik/gambar untuk membangun, memvisualisasikan, menentukan, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis OO (Object Oriented). UML juga menyediakan standar untuk menulis sistem cetak biru, termasuk konsep proses bisnis, menulis kelas dalam bahasa pemrograman tertentu, skema database, dan komponen yang diperlukan dalam sistem perangkat lunak (Zufria, 2016). Berdasarkan penjelasan tersebut, maka *Unified Modeling Language (UML)* dapat disimpulkan merupakan pola atau alur dari sebuah sistem dalam bentuk diagram hingga menghasilkan *output*. Desain UML yang digunakan adalah *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

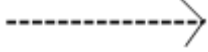
2.23.1 Use Case Diagram

Pengertian *use case diagram* yang diuraikan oleh Rosa dan Shalahuddin adalah merupakan sebuah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case diagram* yang menggambarkan interaksi antara satu atau lebih aktor dengan sistem (Paramanindo, 2014). Simbol yang digunakan dalam *use case diagram* adalah sebagai berikut:

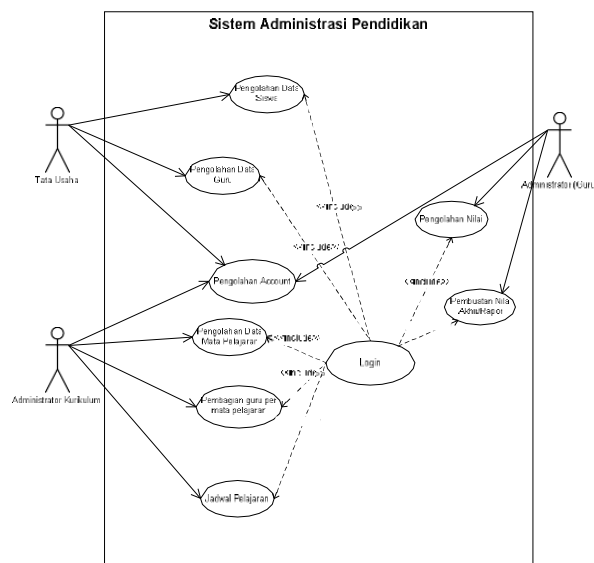
Tabel 2.2 Daftar Simbol *Use Case Diagram*

(Sumber : Paramanindo, 2014)

Simbol	Nama	Deskripsi
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
	<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
	<i>Generalization</i>	Hubungan generalisas dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

<p><<include>></p> 	<p><i>Include</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
------------------------------------------------------------------------------------------------------------------	-----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Berikut ini merupakan contoh *use case diagram* :



Gambar 2.4 Contoh *Use Case Diagram*

(Sumber : Zufria, 2016)

Berdasarkan tampilan gambar tersebut, maka *use case diagram* dapat disimpulkan merupakan interaksi antara satu *user* dengan *user* lainnya di dalam satu sistem yang mempunyai tugasnya masing-masing.






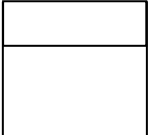
2.23.2 *Activity Diagram*

Pengertian *activity diagram* yang diuraikan oleh Rosa dan Shalahuddin adalah menggambarkan workflow (aliran kerja) atau kegiatan suatu sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Apa yang harus dipertimbangkan di sini adalah bahwa *diagram* operasi yang menggambarkan pengoperasian sistem, bukan apa yang dilakukan aktor, sehingga operasi dapat

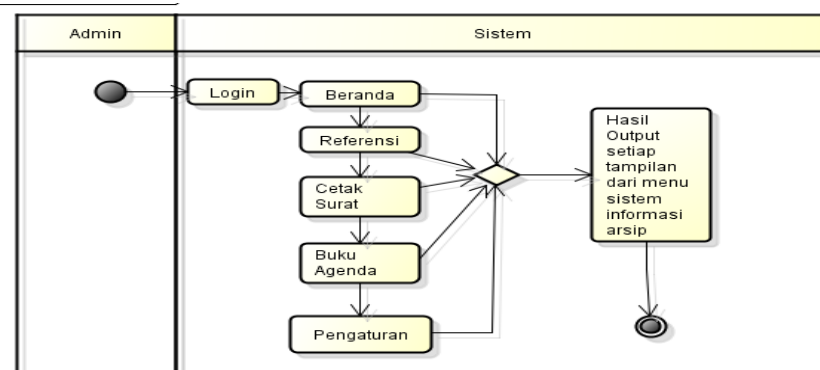
diimplementasikan oleh sistem (Paramanindo, 2014). Simbol yang digunakan dalam *activity diagram* sebagai berikut:

Tabel 2.3 Daftar Simbol *Activity Diagram*

(Sumber : Paramanindo, 2014)

Simbol	Nama	Deskripsi
	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Berikut ini merupakan contoh *activity diagram* :



Gambar 2.5 Contoh *Activity Diagram* Kaprodi

(Sumber : Irawan dan Simargolang, 2018)

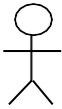

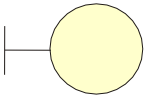



Berdasarkan tampilan gambar tersebut, maka *activity diagram* dapat disimpulkan merupakan proses alur kerja sistem yang dimulai dari aktivitas awal dari sebuah sistem sampai selesai.

2.23.3 *Sequence Diagram*

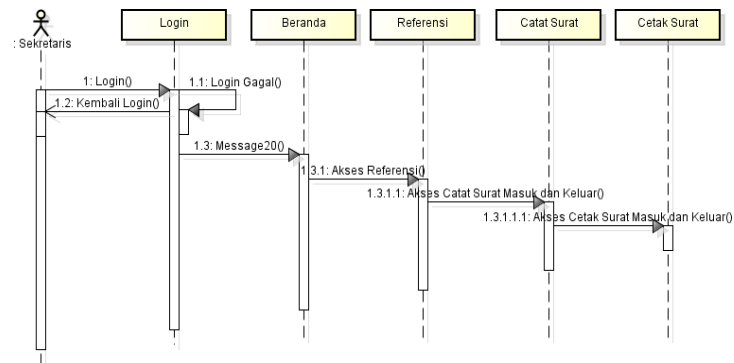
Pengertian *sequence diagram* yang diuraikan oleh Rosa dan Shalahuddin adalah menggambarkan perilaku objek dalam *use case* dengan menggambarkan masa hidup objek dan pesan yang dikirim dan diterima antara objek. *Sequence diagram* menunjukkan urutan peristiwa yang terjadi pada waktu tertentu. Elemen *sequence diagram* terdiri dari objek yang ditulis dalam kotak segi empat bernama. Pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan oleh perkembangan vertikal (Paramanindo, 2014). Simbol- simbol yang digunakan dalam *sequence diagram* adalah:

Tabel 2.4 Daftar Simbol *Sequence Diagram*

(Sumber : Paramanindo, 2014)

Simbol	Nama	Deskripsi
	<i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari <i>form</i>
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
	<i>A focus of control</i>	Menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> (pesan)
	<i>A line of life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Berikut ini merupakan contoh *sequence diagram* :



Gambar 2.6 Contoh *Sequence Diagram* Sekretaris

(Sumber : Irawan dan Simargolang, 2018)

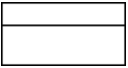


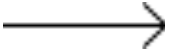

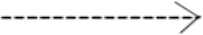
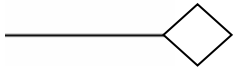
Berdasarkan tampilan gambar tersebut, maka *sequence diagram* dapat disimpulkan merupakan penggambaran kelakuan *object* pada pengguna.

2.23.4 *Class Diagram*

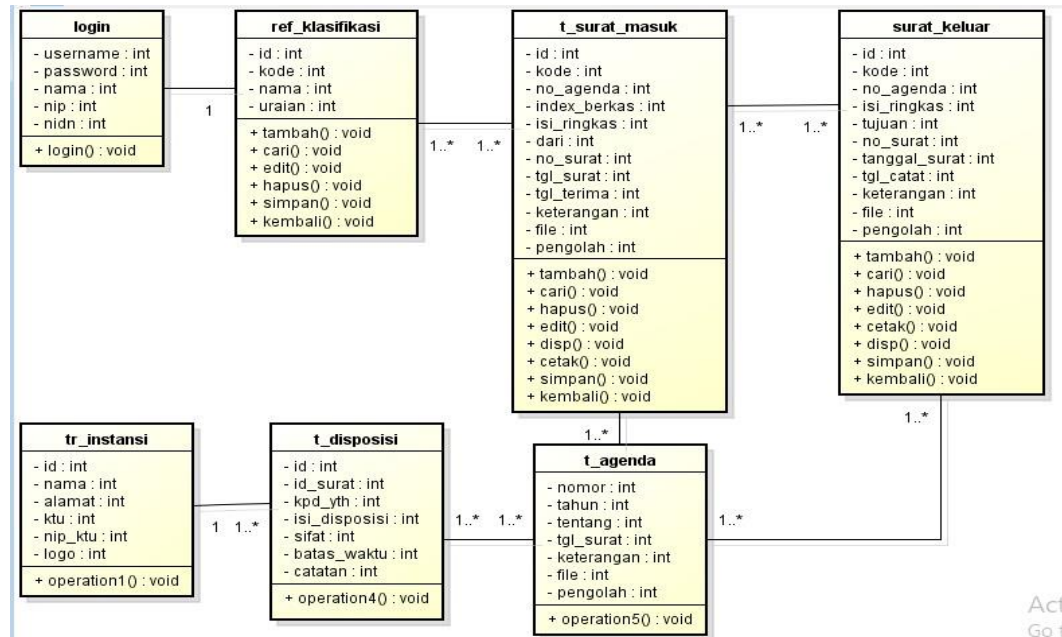
Pengertian *class diagram* yang diuraikan oleh Rosa dan Shalahuddin adalah menggambarkan struktur sistem dalam mendefinisikan kelas-kelas yang akan dihasilkan untuk membangun sistem. *Class diagram* dibuat untuk pembuat program atau programmer untuk membuat kelas-kelas sesuai rancangan dalam *class diagram* sehingga dokumentasi desain dan *software* menjadi sinkron (Paramanindo, 2014). Simbol *class diagram* sebagai berikut :

Tabel 2.5 Daftar Simbol *Class Diagram*

(Sumber : Paramanindo, 2014)

Simbol	Nama	Deskripsi
	<i>Class</i>	Kelas pada stuktur sistem.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
	<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Berikut ini merupakan contoh *class diagram* :



Gambar 2.7 Contoh *Class Diagram*

(Sumber : Irawan dan Simargolang, 2018)

Berdasarkan tampilan gambar tersebut, maka *class diagram* dapat disimpulkan merupakan rancangan *database* yang terdapat kelas-kelas dalam sebuah sistem yang saling berkaitan atau berelasi.

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Dalam pelaksanaan skripsi ini, penulis penelitian di perusahaan PT. Rahayu Medan Ceria berlokasi di Komplek Setia Budi Center blok. B No. 12, Jl. Setia budi, Medan Perjuangan, Tunggal, Tj. Rejo, Kec. Medan Sunggal, Kota Medan, Sumatera Utara.

3.1.2 Waktu Penelitian

Penelitian ini membutuhkan jadwal dan langkah-langkah penelitian untuk mencapai batas waktu realisasi sistem. Periode penelitian dari Juli hingga Oktober 2019 sebagai berikut :

Tabel 3.1 Jadwal Penelitian

Jadwal Kegiatan	Agustus			September			Oktober				
	M I	M III	M IV	M I	M II	M III	M IV	M I	M II	M III	M IV
Persiapan											
Bimbingan Skripsi											
Riset Kebutuhan Institusi/perusahaan											
Pengajuan Berkas Seminar Proposal											
Seminar Proposal											
Pengumpulan Data											
Studi Pustaka											
Revisi Proposal Skripsi											
Perancangan Sistem											

Desain Interface										
Pembuatan Coding										
Testing										

Adapun tahap yang dilakukan dalam penelitian ini sebagai berikut :

1. Persiapan

Pada tahap persiapan, peneliti melakukan penelitian untuk mencari objek yang menjadi masalah pada masyarakat serta solusi yang akan dijadikan penelitian, setelah itu pengajuan judul proposal skripsi ke prodi sistem informasi hingga disetujui sampai keluar surat tugas untuk pembimbing skripsi dan minggu ke 4 awal melakukan bimbingan skripsi.

2. Bimbingan Skripsi

Pada tahap ini, peneliti melakukan bimbingan skripsi kepada bapak Muhammad Dedi Irawan, M. Kom dan Bapak Ilka Zufria, M.Kom.

3. Riset Kebutuhan Institusi/Perusahaan

Pada tahap ini, peneliti melakukan observasi dan wawancara langsung ke PT. Medan Raharu Ceri. Pada bulan Juli minggu ke 3 untuk mendapatkan Izin, serta data yang diambil berupa struktur organisasi perusahaan tersebut dan pada pada bulan Mei minggu ke 1 untuk mendapatkan sejarah perusahaan, tugas pimpinan dan pegawai, jumlah pegawai, dan jumlah armada angkutan umum guna melaksanakan instruksi pembimbing 1.

4. Pengajuan Berkas Seminar Proposal

Pada tahap ini, peneliti melakukan pengajuan berkas sebagai syarat sempro kepada bagian akademik.

5. Seminar Proposal

Pada tahap ini, peneliti seminar proposal untuk menentukan apakah objek penelitian yang diangkat oleh peneliti layak dan memenuhi syarat untuk dilanjutkan ke skripsi.

6. Pengumpulan Data

Pada tahap ini, peneliti akan mengumpulkan data serta sumber-sumber yang diperlukan secara Observasi, Studi literatur dan lain-lain.

7. Analisis Data

Pada tahap ini, peneliti melakukan analisa terhadap semua data yang telah didapat dengan cara memilih data yang akurat serta keaslian data untuk dijadikan bahan skripsi.

8. Studi Pustaka

Pada tahap ini, peneliti melakukan studi pustaka yang di bimbing oleh dosen pembimbing 1 dan dosen pembimbing 2 untuk menghindari kecurangan dan plagiat selama penelitian berlangsung.

9. Desain Sistem

Pada tahap ini, peneliti merancang sistem dengan menggunakan pemodelan UML (*Unified Modelling Language*) untuk mendapatkan hasil yang sistematis terkait dengan alur sistem yang akan dirancang.

10. Desain *Interface*

Pada tahap ini, peneliti mulai melakukan perancangan pada desain *interface* sistem yang akan digunakan.

11. Pembuatan *Coding*

Pada tahap ini, peneliti mulai melakukan perancangan dengan menggunakan kode pemrograman pada android untuk memberikan fungsi terhadap subsistem yang digunakan,.

12. *Testing*

Pada tahap ini, peneliti melakukan tes terhadap keseluruhan sistem yang telah dibuat agar mengetahui keberhasilan dari sistem tersebut dan melakukan evaluasi sistem jika ada yang tidak sesuai.

3.2 Kebutuhan Sistem

Kebutuhan sistem yang diperlukan dalam pengerjaan skripsi ini mulai dari tahap perancangan sistem hingga tahap pemrograman menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) yaitu :

1. Perangkat Keras (*Hardware*) spesifikasi perangkat dalam pembuatan aplikasi android, yang terdiri dari:
 - b. Sistem Operasi Microsoft Windows 7.
 - c. *Processor Intel* ® *Core*TM *i3*.
 - d. *Memory* RAM 4.00 GB
 - e. *Harddisk* 465,76GB HDD
 - f. *Smartphone* Samsung J7
2. Perangkat Lunak (*software*), yang terdiri dari :
 - a. Android Studio Versi 3.5
 - b. Emulator Android Nox 6.2.8.3
 - c. Draw.io
 - d. MS Word 2007
 - e. Mendeley Dekstop Versi 1803
 - f. Browser

3.3 Cara Kerja

Tahapan metode penelitian yang digunakan pada pembuatan aplikasinya mulai dari teknik pengumpulan data, teknik penelitian kepustakaan, dan metode pengembangan sistemnya adalah sebagai berikut :

1. Teknik Pengumpulan Data

a. Observasi

Pada tahap penelitian ini, peneliti melakukan observasi langsung ke perusahaan dengan melakukan pengumpulan data mengenai ruang lingkup pada perusahaan PT. Rahayu Medan Ceria serta mengamati proses kinerja perusahaan tersebut.

b. Wawancara

Pada tahap penelitian ini, peneliti melakukan wawancara kepada Bapak Eko Wahyudi selaku Kepala Personalia PT.Rahayu Medan Ceria terkait. Informasi perusahaan, pak Wahyudi menyampaikan bahwasanya jumlah armada perusahaan sebanyak 3000, sedangkan yang aktif setengahnya.

2. Teknik Penelitian Kepustakaan

a. Studi Literatur

Pada tahap ini, peneliti mencari data dan informasi yang bersumber dari berbagai jurnal yang bersangkutan dengan materi penelitian dan juga dari buku, seperti pengertian *tracking* oleh Marjuki (2016), *real time* oleh Irwansyah (2013), UML oleh Zufria (2016), dan contoh *sequence diagram* oleh Irawan dan Simargolang (2018).

3. Metode Pengembangan sistem

Metode pengembangan sistem dengan menggunakan RAD (*Rapid Application Development*) dalam pembuatan Sistem Informasi *Tracking Real Time* Angkutan Umum Pada Halte Kota Medan. Adapun tahapan yang dilakukan sebagai berikut :

a. *Bussiness Modeling*

Dalam Tahap ini , penulis menganalisis suatu bisnis di dalam perusahaan PT. Rahayu Medan Ceria dengan melakukan perencanaan membuat aplikasi android untuk pelayanan angkutan umum pada konsumen. Agar nantinya aliran informasi diberikan kepada konsumen melalui *smartphone*. Informasi tersebut berupa posisi angkot dan nomor angkot.

b. *Data Modeling*

Pada tahap ini, penulis megumpulkan data dari perusahaan tersebut untuk mendapatkan jumlah angkot yang diperlukan dalam penelitian ini, serta melakukan pendataan halte yang dilalui angkutan umum pada perusahaan tersebut.

c. *Process Modeling*

Pada tahap ini, penulis melakukan rancangan aplikasi yang akan dibangun sesuai dengan kebutuhan sistem.

d. *Application Generation*

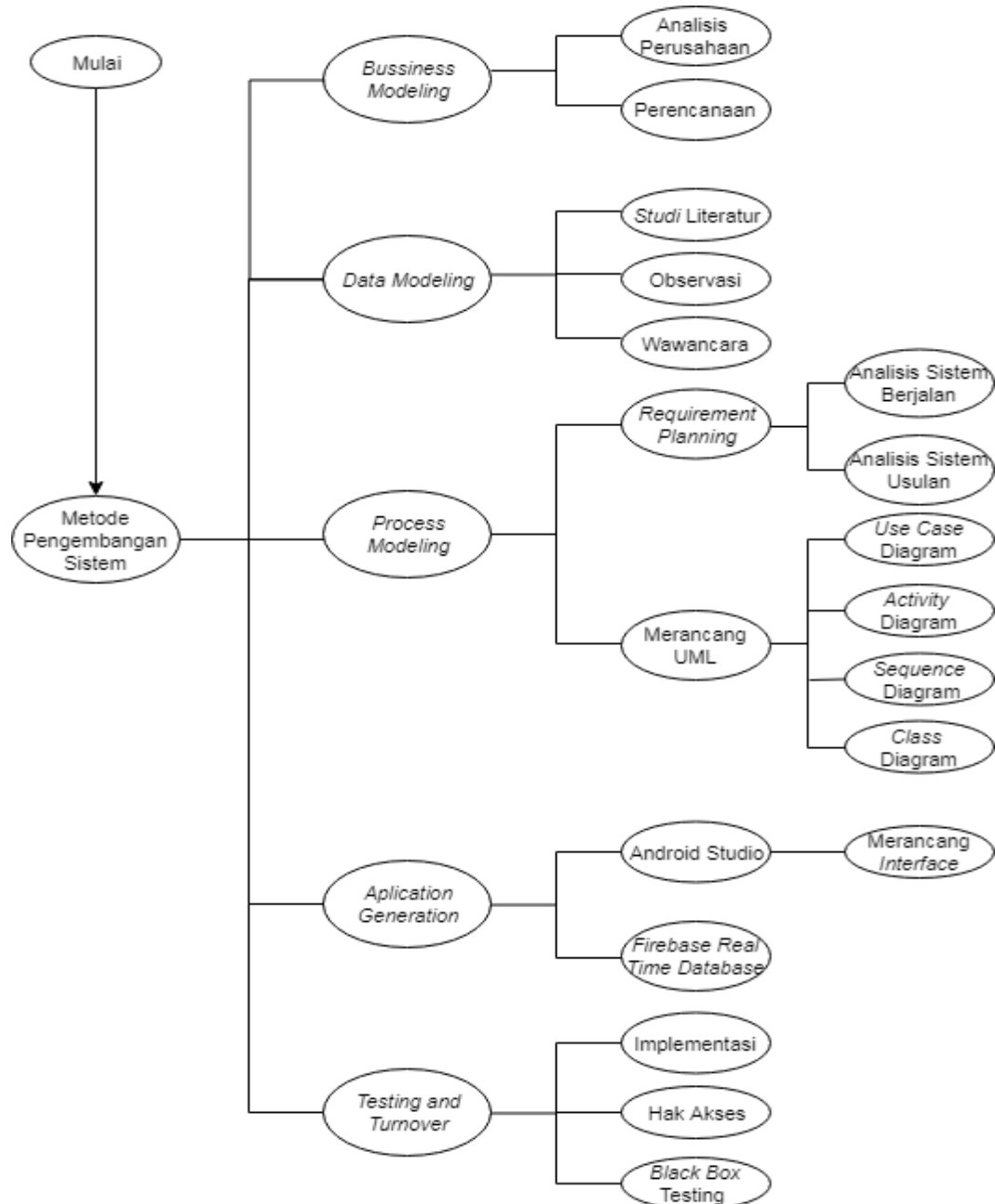
Pada tahap ini, penulis melakukan pengcodangan bahasa pemrograman java dengan menggunakan *tools* aplikasi android studio dengan menggunakan *real time database* Firebase.

e. *Testing and Turnover*

Pada tahap ini, penulis melakukan pengujian seluruh sistem secara berulang apabila ada penambahan coding atau *interface* guna memastikan aplikasi tersebut berjalan dengan baik tidak adanya *error*.

3.4 Kerangka Berfikir

Dalam mewujudkan sistem yang terukur dan terencana, maka harus ada alur kerja bagaimana membuat sistem serta apa saja kegiatan setiap metode pengembangan sistem, adapun Kerangka berfikir yang dilakukan dalam pengembangan sistem adalah sebagai berikut :



Gambar 3.1 Kerangka Berfikir

BAB IV

HASIL DAN PEMBAHASAN

4.1 Profil Perusahaan

Perusahaan PT. Rahayu Medan Ceria yang beralamat di Komplek Setiabudi Center blok. B No. 12, Jl. Setia budi, Medan Perjuangan, Tunggol, Tj. Rejo, Kec. Medan Sunggal, Kota Medan, Sumatera Utara. Perusahaan tersebut bergerak di bidang pengangkutan orang dan taksi online. Jumlah armada angkutan kota 3.000 sedangkan yang aktif kurang lebih 1.500 dan jumlah armada taksi online kurang lebih 200. Jumlah karyawan di kantor 10, 100 freelancer bagian lapangan, dan 50 mandor lapangan. Perusahaan tersebut pertama kali didirikan pada tahun 1985 oleh Bapak S.J. Munthe hingga pergantian yang diteruskan sampai sekarang oleh Bapak Drs. Mont Gomery Munthe. Pada awal berdirinya, armada angkutan mempunyai pintu masuk penumpang dari belakang mobil dan ditahun 2015 seluruh armada angkutan pintu masuk penumpang dari samping mobil dengan menggunakan kendar produk Daihatsu Espas dan Grand Max.

4.1.1 Struktur Organisasi

Adapun struktur organisasi pada perusahaan PT. Rahayu Medan Ceria adalah sebagai berikut :



Gambar 4.1 Struktur Organisasi PT. Rahayu Medan Ceria

(Sumber : PT. Rahayu Medan Ceria)

4.1.2 *Job Desk* Struktur Organisasi

Berdasarkan gambar struktur organisasi pada perusahaan PT. Rahayu Medan Ceria di atas, maka berikut ini tugas pada bagian tersebut :

1. Direktur, bertugas mengurus perizinan, memonitoring dan memberikan arahan kepada karyawan.
2. Wakil Direktur, bertugas mengurus segala keuangan perusahaan.
3. Manager Administrasi, bertugas mengelola data perusahaan.
4. Tenaga Admin, bertugas menerima karcis iuran dari pemilik armada angkutan.
5. Koordinator Lapangan, bertugas mengawasi kinerja mandor.
6. Mandor Lapangan, bertugas mengatur, menertibkan, mengawasi, angkutan kota pada perusahaan.

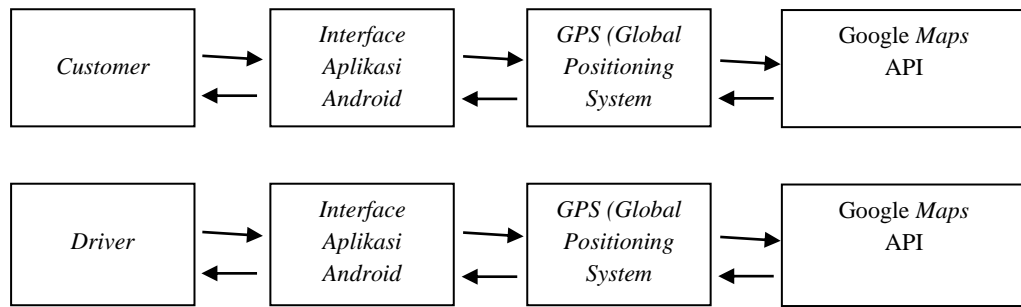
4.1.3 Analisis Sistem Berjalan

Analisis sistem berjalan akan menjadi tolak ukur dalam pengembangan sistem yang selanjutnya. Adapun prosesnya yaitu masyarakat memberhentikan angkutan umum di sembarang tempat kemudian supir menurunkan penumpang tidak di Halte yang sudah disediakan pemerintah. Setelah dianalisis sistem berjalan pada angkutan umum, maka ditemukan kelemahannya yaitu :

1. Lalu lintas tidak teratur karena penumpang dimanapun bisa memberhentikan angkutan umum dan supir juga menurunkan tidak di Halte.
2. Belum efisien, dikarenakan bisa menyebabkan kemacetan.

4.1.4 Perancangan Sistem Usulan

Sistem usulan angkutan umum yang akan dibangun adalah sebuah aplikasi yang menggunakan *smartphone* berbasis android sebagai media interface nya. *Customer* dan *Driver* dapat menjalankan sistem ini di *smartphone* android yang sudah terinstal aplikasi angkot berbasis android. Berikut gambaran dari sistem usulan yang dimaksud :



Gambar 4.2 Gambaran Sistem Usulan

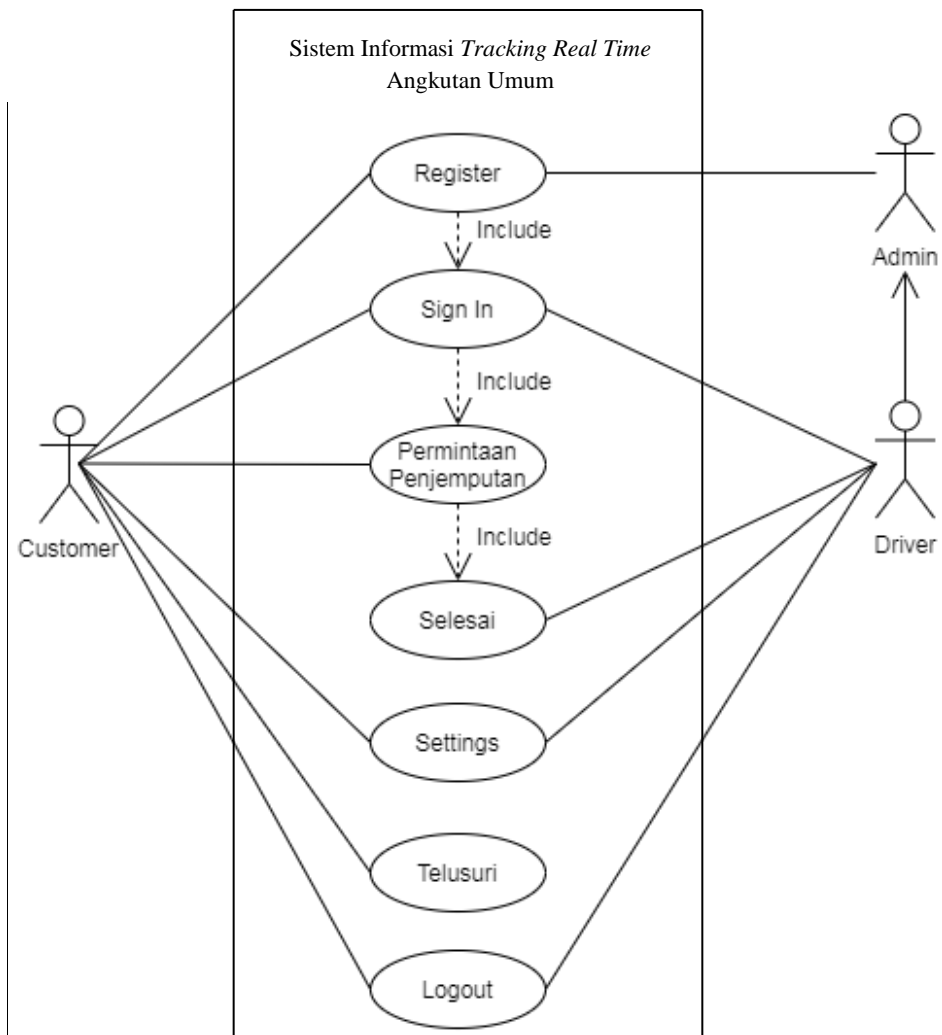
Berdasarkan gambaran sistem usulan di atas, maka dapat dijelaskan proses awalnya yaitu user melakukan masuk ke sistem dan mendapatkan GPS dari pengguna tersebut yang sudah terdeteksi di google *maps* api nya, begitu juga sama dengan *driver* nya.

4.2 Perancangan Sistem

Dalam penelitian ini, dibutuhkan rancangan sistem agar memudahkan peneliti dalam membangun sistem informasi *tracking real time* angkutan umum pada Halte Kota Medan, perlu dilakukan perancangan *use case* diagram, *activity* diagram, *sequence* diagram, dan *class* diagram.

4.2.1 Use Case Diagram

Secara umum, *use case diagram* berfungsi untuk melakukan beberapa pekerjaan yang menggambarkan proses bisnis dalam sistem, berikut ini rancangan *use case diagram* pada studi kasus sistem informasi *tracking real time* angkutan umum pada Halte Kota Medan :



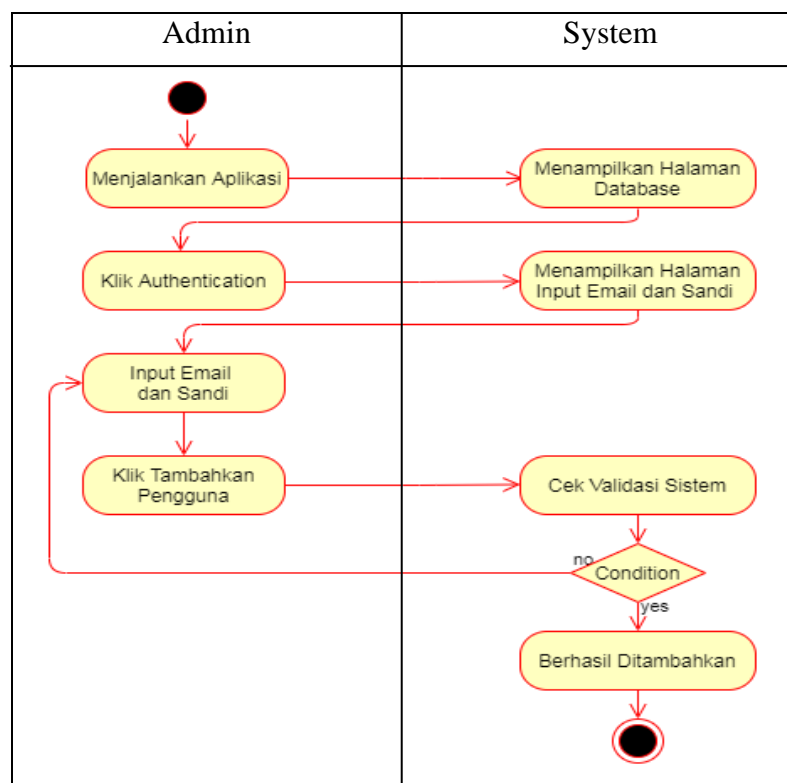
Gambar 4.3 Use Case Diagram

Dalam pembuatan rancangan *use case diagram*, peneliti membuat 3 aktor yaitu *driver*, *admin* dan *customer*. *Driver* mengirim data diri ke *admin*, kemudian *admin* memproses untuk *register* data *driver* tersebut. Selanjutnya *driver* bisa *sign in*. Kemudian klik tombol *settings*, Setelah itu *driver* menerima permintaan penjemputan dari *customer*, *driver* mengklik tombol *selesai* dan *logout* jika ingin

mengakhiri. Sedangkan untuk *customer* bisa langsung *register*. Selanjutnya melakukan *sign in*, *settings*, mengklik tombol permintaan penjemputan, mengklik telusuri lokasi tujuan, jika *customer* ingin keluar klik tombol *logout*.

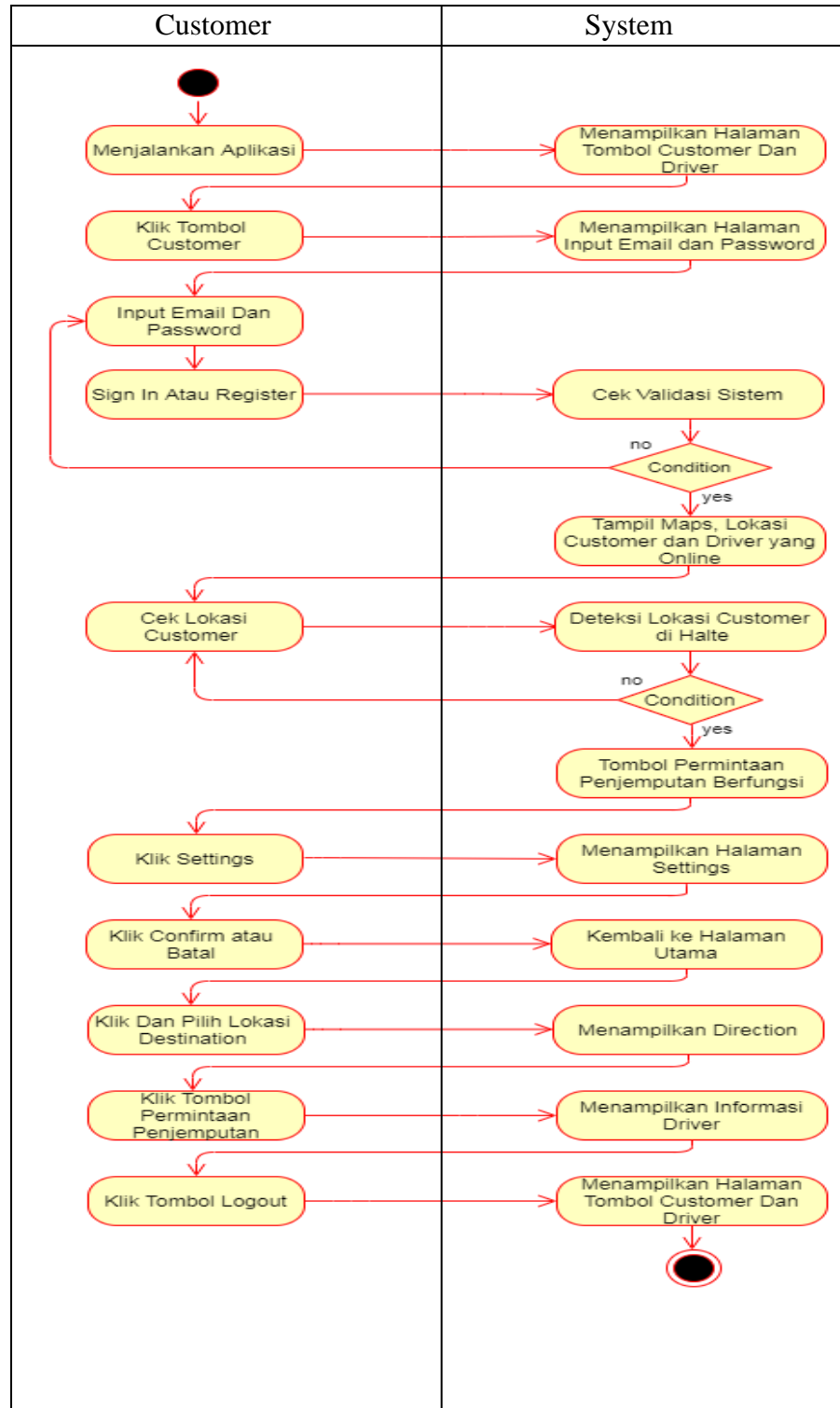
4.2.2 Activity Diagram

- Berikut ini rancangan *activity diagram* admin :

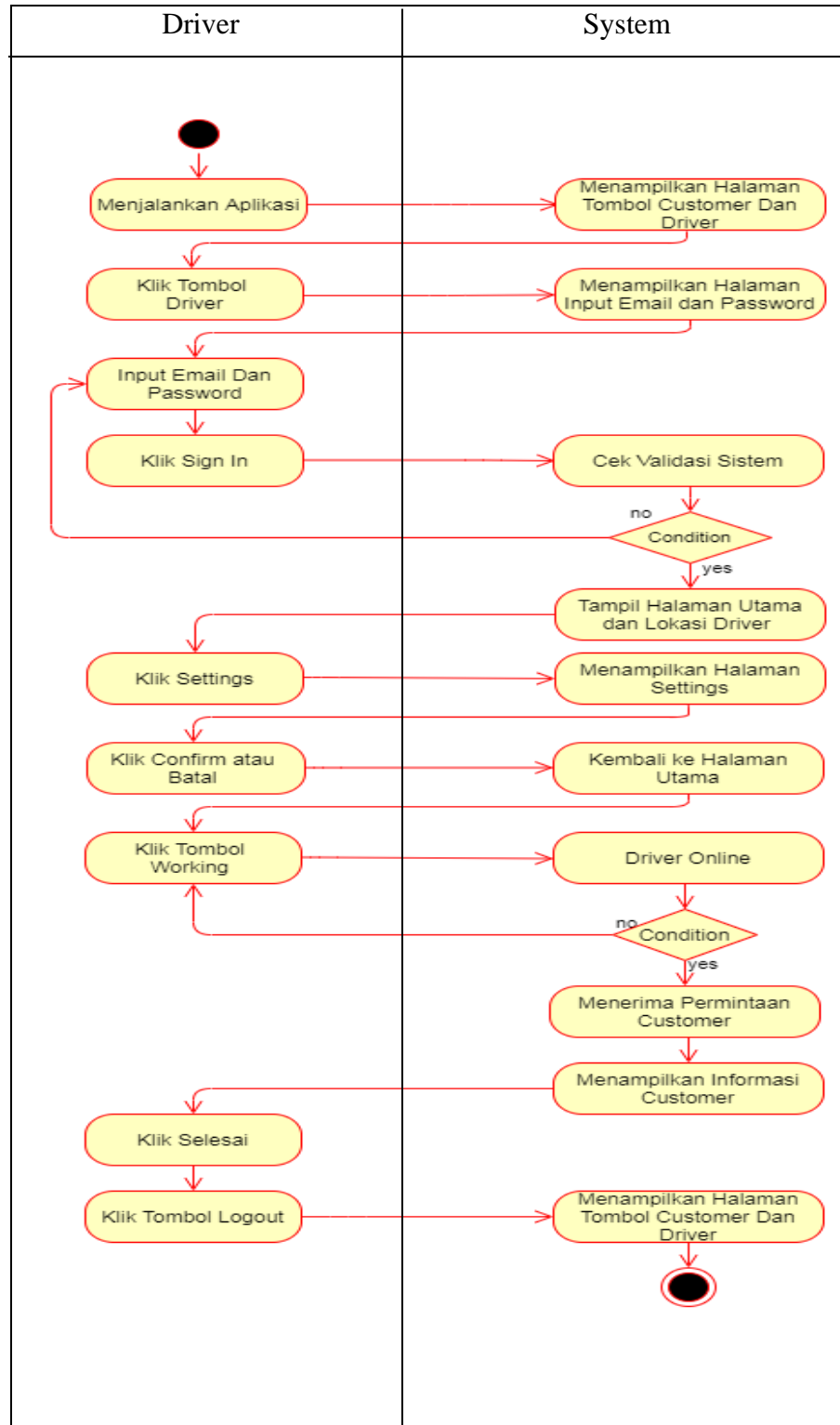


Gambar 4.4 Activity Diagram Admin

Pada gambar di atas, admin menjalankan aplikasi berbasis web. Lalu menampilkan halaman *database*. Kemudian mengklik *authentication* yang akan menampilkan halaman input email dan sandi. Setelah itu admin menginputkan email dan sandi untuk *driver*, admin mengklik tambahkan pengguna, maka sistem mengecek validasi, apabila *valid* berhasil ditambahkan. Namun jika tidak *valid*, admin kembali menginputkan email dan sandi *driver* tersebut.

2. *Activity Diagram Customer*Gambar 4.5 *Activity Diagram Customer*

Berdasarkan gambar diatas, maka dapat dijelaskan proses awalnya mulai dari menjalankan aplikasi, dengan mengklik tombol *customer* setelah itu sistem menampilkan *interface* tombol *Sign In* dan Register, maka *customer* melakukan register atau *sign in* dengan memasukkan *email* dan *password*. Setelah dinyatakan *valid* maka akan masuk Halaman utama. Kemudian jika *customer* ingin mengubah data pribadi maka klik tombol *settings*. Selanjutnya *customer* terlebih dahulu harus mengecek lokasinya agar terdeteksi di halte, jika tidak tombol sistem utama tidak akan berfungsi. Kemudian jika benar, tombol permintaan penjemputan akan berfungsi. Sebelum mengeklik tombol permintaan penjemputan, *customer* menginputkan lokasi *destination* maka menampilkan *direction* berupa navigasi yang akan mengarahkan lokasi tujuannya. Setelah itu diklik tombol permintaan penjemputan maka akan menampilkan informasi *driver* angkot. Jika *Customer* ingin keluar dari aplikasi maka pilih tombol *logout*. Setelah itu sistem akan menampilkan halaman tombol *Customer* dan *Driver*.

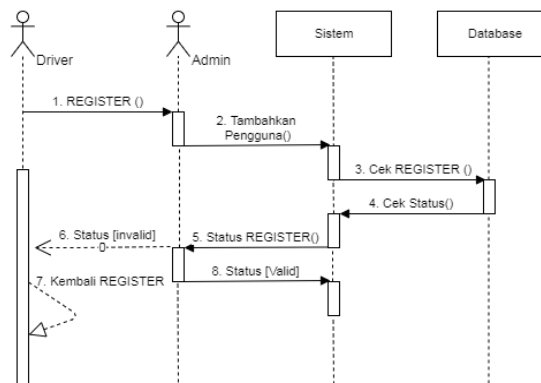
1. *Activity Diagram Driver*Gambar 4.6 *Activity Diagram Driver*

Berdasarkan gambar tersebut, maka dapat dijelaskan proses awalnya mulai dari menjalankan aplikasi, dengan mengklik tombol *driver* setelah itu sistem menampilkan *interface* tombol *sign in* dan register, maka *driver* melakukan register atau *sign in* dengan memasukkan *email*, *password*.. Setelah *valid*, selanjutnya masuk ke sistem halaman utama. Setelah itu *driver* terlebih dahulu harus mengecek tombol onlinenya, jika tidak diaktifkan tombol tersebut sistem tidak akan bisa menerima permintaan penjemputan dari *customer*. Kemudian setelah online, *driver* bisa menerima permintaan masuk dari *customer* yang akan menampilkan informasi *customer* untuk melakukan penjemputan. Jika *driver* ingin keluar dari aplikasi maka pilih tombol *logout*. Setelah itu sistem akan menampilkan halaman tombol *customer* dan *driver*.

4.2.3 Sequence Diagram

Sequence Diagram menggambarkan *driver* interaksi dengan admin pada aplikasi angkot. Berikut ini adalah gambaran *sequence* untuk sistem yang di rancang.

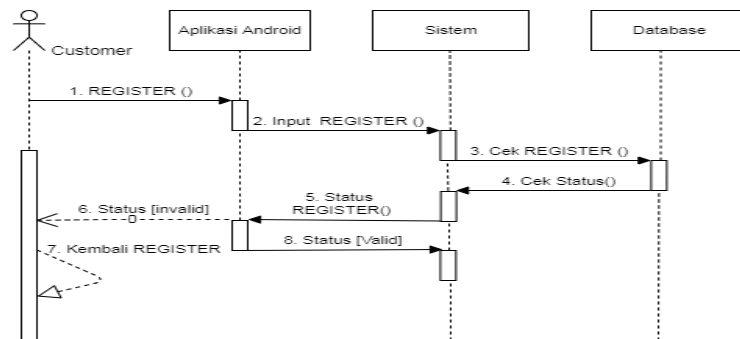
- Berikut ini *sequence diagram register driver* :



Gambar 4.7 *Sequence Diagram Register Driver*

Pada gambar di atas terdapat 2 aktor yaitu *driver* dan admin. *Driver* melakukan *register* dengan mengirimkan email dan *password* ke admin, kemudian admin menambahkan pengguna dengan menginputkan data *driver* ke dalam sistem yang akan tersimpan di *database* . Jika statusnya *invalid*, maka kembali *register* dan jika *valid* data tersebut masuk ke sistem.

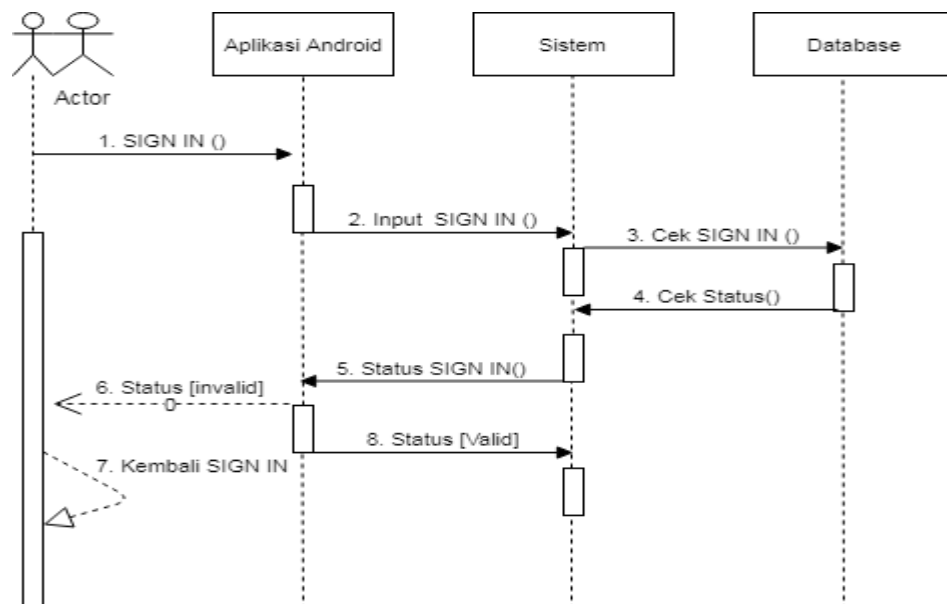
2. Berikut ini *sequence diagram register customer* :



Gambar 4.8 *Sequence Diagram Register Customer*

Berdasarkan gambar tersebut, *customer* melakukan *register* di aplikasi android, kemudian menginputkan data ke dalam sistem yang akan tersimpan di *database*. Kemudian mengecek statusnya, apabila *invalid* kembali *register* dan jika *valid* data tersebut masuk ke sistem.

3. *Sequence Diagram Sign In Customer dan Driver Pada Aplikasi Android*

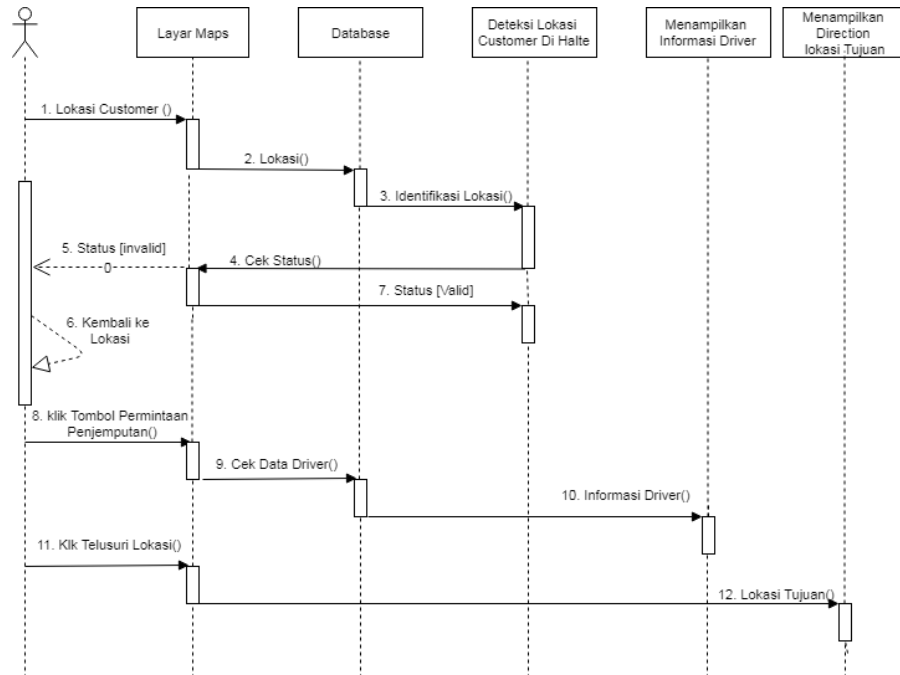


Gambar 4.9 *Sequence Diagram Sign In Customer Dan Driver*

Berdasarkan gambar di atas, menjelaskan bagaimana proses *customer* dan *driver* melakukan *sign in*. Yang dilakukan adalah input data, setelah itu dicek

dalam *database* apakah data input *valid* atau tidak, jika *valid* sistem dimulai dari awal lagi. Apabila benar proses *sign in* berhasil.

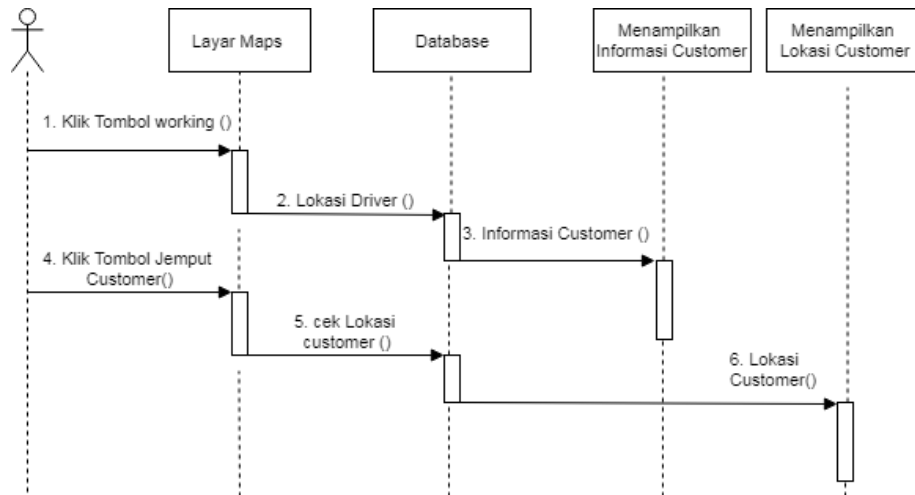
4. *Sequence Diagram* Permintaan Penjemputan



Gambar 4.10 *Sequence Diagram* Permintaan Penjemputan

Berdasarkan gambar di atas, menjelaskan bagaimana proses *customer* ingin melihat lokasinya dan menampilkan informasi *driver*. Posisi *customer* harus pada jangkauan Halte agar terdeteksi. Jika sudah berada posisi tersebut maka tombol Permintaan Penjemputan akan berfungsi dengan menampilkan informasi *driver*. Jika ingin mencari lokasi tujuan, harus melakukan input Telusuri yang akan menampilkan *direction* berdasarkan tujuannya.

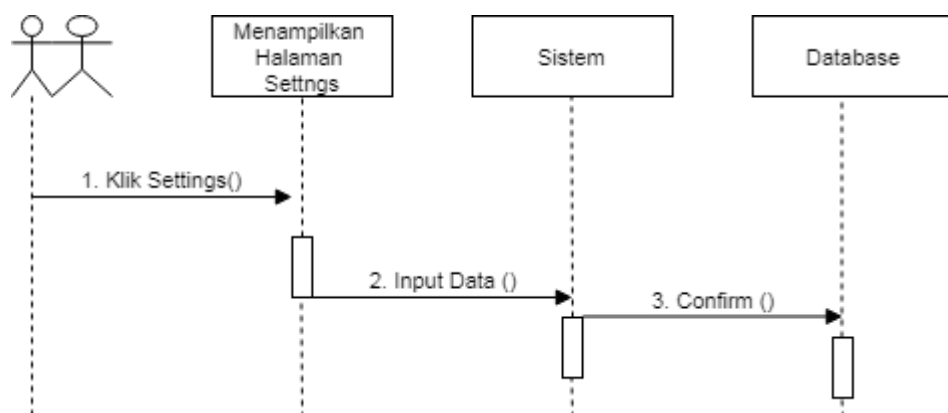
5. *Sequence Diagram Jemput Customer*



Gambar 4.11 *Sequence Diagram Jemput Customer*

Berdasarkan gambar di atas, menjelaskan bagaimana proses *driver* ingin melihat lokasi *customer*. Saat masuk ke tampilan layar *maps*, maka *driver* harus online dengan mengaktifkan tombol *working*. Jika sudah aktif, maka apabila *customer* ingin minta jemput maka tampil informasi *customer*. Setelah itu klik jemput *customer*.

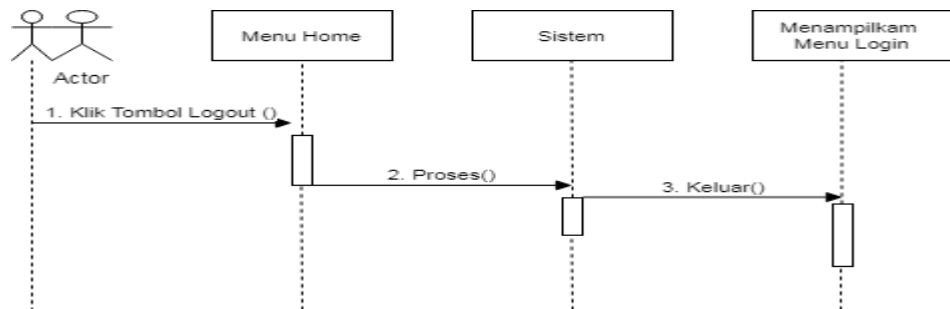
6. *Sequence Diagram Settings Driver dan Customer Pada Aplikasi Android*



Gambar 4.12 *Sequence Diagram Settings Customer Dan Driver*

Berdasarkan gambar di atas, menjelaskan bagaimana proses *customer* dan *driver* melakukan *Settings* agar mengetahui datanya.

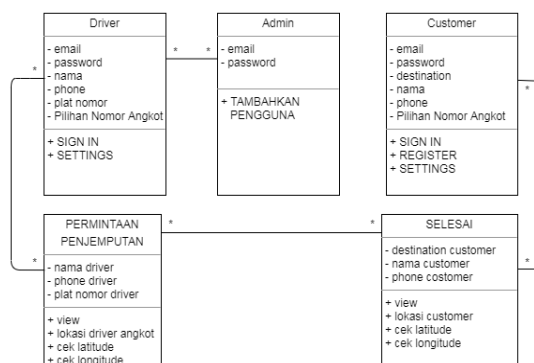
7. Sequence Diagram Logout Driver dan Customer Pada Aplikasi Android



Gambar 4.13 Sequence Diagram Logout Customer Dan Driver

Berdasarkan gambar di atas, menjelaskan bagaimana proses *customer* dan *driver* melakukan *logout*. Yang pertama dilakukan adalah saat aplikasi android sedang berjalan, pilih *logout* pada menu *home*, setelah itu diproses oleh sistem. Maka akan tampil halaman awal untuk *Sign In* dan *Register*.

4.2.4 Class Diagram



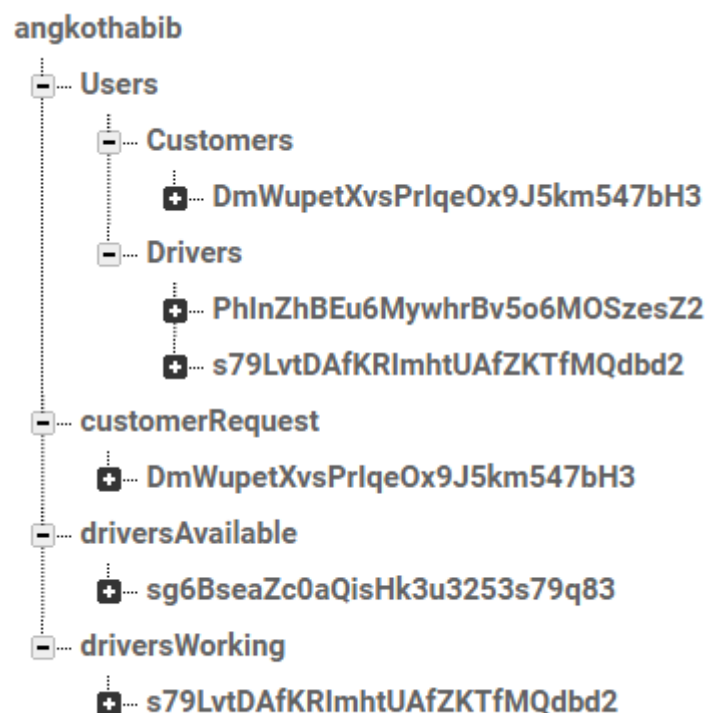
Gambar 4.14 Class Diagram

Pada gambar di atas *driver* dan *customer* mempunyai 6 atribut di mana fungsinya untuk melakukan proses validasi data, kemudian *driver* berelasi dengan Jemput *Customer* berdasarkan latitude dan longitude *driver* yang akan menampilkan nama, *phone* dan plat nomor *driver* pada aplikasi *customer*, begitu

juga dengan *customer* yang berelasi dengan Permintaan Penjemputan berdasarkan latitude dan longitude *customer* yang akan menampilkan *destination*, nama dan *phone customer* pada aplikasi *driver*.

4.3 Rancangan Database

Rancangan *database* dalam membangun aplikasi angkutan umum terdiri dari tabel *users*, *customers*, *drivers*, *driver available* dan *driver working*. Berikut rancangan *database* dengan menggunakan *firebase realtime database* :



Gambar 4.15 Rancangan Database

Pada gambar di atas dijelaskan bahwa tabel *User* yang di dalamnya terdapat *customers* dan *drivers* berfungsi sebagai mengupdate data pribadi. *CustomerRequest* dan *driversAvailable* adalah mengupdate lokasi secara *real time*. Sedangkan pada *driversWorking* adalah *driver* yang menerima permintaan penjemputan dari *customer*.

4.4 Rancangan Tampilan Antar-muka (*interface*)

Dalam membuat suatu aplikasi maka yang di perlukan adalah rancangan *interface*, karena itu akan mempermudah gambaran sistem yang dibuat nantinya. Berikut rancangan *interface* nya :

1. Rancangan Tampilan halaman *Sign In* dan *Register Customer*

The wireframe shows a page titled 'ANGKOT RMC'. At the top, it says 'Customer' and 'Tolong Gunakan Email Untuk Masuk'. Below this are two input fields: 'Username' and 'Password'. There are two buttons: 'Register' and 'Sign In'. At the bottom, there is a button labeled 'CUSTOMER' and the text 'Aplikasi Angkutan Kota Medan'.

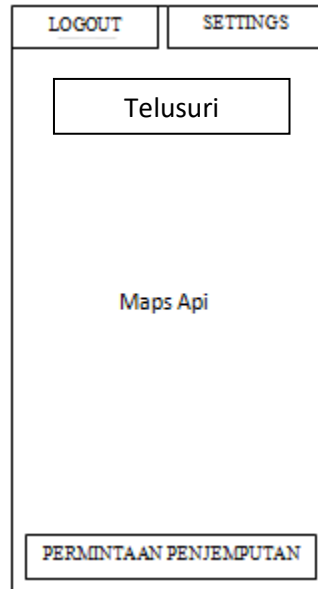
Gambar 4.16 Rancangan Tampilan halaman *Sign In* dan *Register Customer*

2. Rancangan Tampilan Halaman *Sign In Driver*

The wireframe shows a page titled 'ANGKOT RMC'. At the top, it says 'Driver' and 'Tolong Gunakan Email Untuk Masuk'. Below this are two input fields: 'Username' and 'Password'. There is one button labeled 'Sign In'. At the bottom, there is a button labeled 'DRIVER' and the text 'Aplikasi Angkutan Kota Medan'.

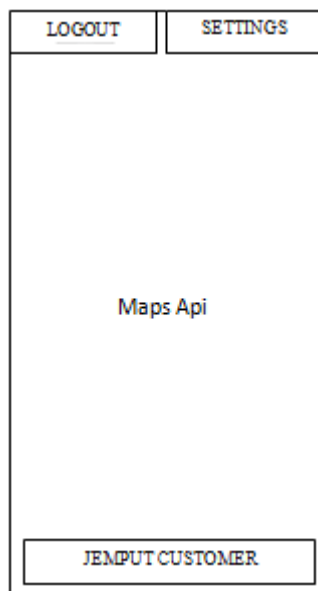
Gambar 4.17 Rancangan Tampilan Halaman *Sign In Driver*

3. Rancangan Tampilan Halaman Permintaan Penjemputan pada aplikasi *Customer*



Gambar 4.18 Rancangan Tampilan Halaman Utama *Customer*

4. Rancangan Tampilan Halaman Jemput *Customer* pada aplikasi *Driver*



Gambar 4.19 Rancangan Tampilan Halaman Utama *Driver*

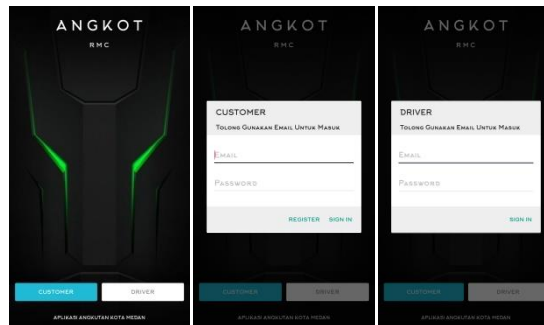
4.5 Test (Pengujian)

Pengujian aplikasi dilakukan dengan uji coba langsung pada *smartphone* android apakah berjalan sesuai dengan kebutuhan.

Tabel 4.1 *Black Box Testing*

No	Aktivitas Pengujian	Tampilan	Pengujian	Fungsi	Tidak Fungsi	Penilaian
1.	Masuk Halaman <i>Sign In</i> dan <i>Register</i>	Tampilan <i>Sign In</i> dan <i>Register</i>	<i>Database</i>	✓	-	100%
2.	Klik <i>Settings</i>	Masuk Halaman <i>Settings</i>	Input Data ke <i>Database</i>	✓	-	100%
3.	Klik Permintaan Penjemputan <i>Customer</i>	Menampilkan informasi <i>Driver</i>	GPS (<i>Global Positioning System</i>) dan <i>Database</i>	✓	-	100%
4.	<i>Driver</i> Menerima Permintaan <i>Customer</i>	Menampilkan informasi <i>Customer</i>	GPS (<i>Global Positioning System</i>) dan <i>Database</i>	✓	-	100%
5.	Klik <i>Logout</i>	Kembali ke halaman awal	Perpindahan <i>Activity</i>	✓	-	100%

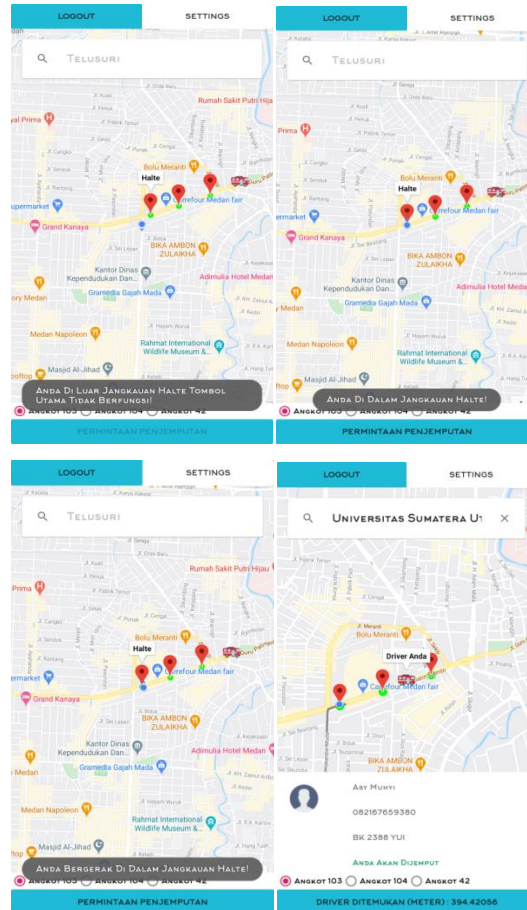
1. Tampilan Awal Aplikasi Angkot



Gambar 4.20 Tampilan Halaman *Sign In Customer Dan Driver*

Penjelasan dari gambar diatas adalah pada aplikasi angkot terdapat 2 user yaitu masuk sebagai *customer* atau *driver*. Tampilan diatas menampilkan background, tombol customer dan driver, jika ingin masuk sebagai *customer* maka klik tombol *customer* maka tampil form *login* yang akan diisi email dan password. Jika belum regis maka klik tombol *register* setelah mengisi email dan passwordnya maka akan masuk ke halaman utama *customer*. Setelah terdaftar maka bisa langsung klik tombol *sign in*. Begitu juga jika ingin masuk sebagai *driver* akan melakukan hal yang sama.

2. Tampilan Halaman Utama *Customer*

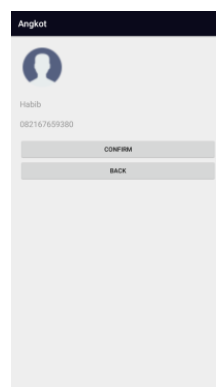


Gambar 4.21 Tampilan Halaman Utama *Customer*

Gambar halaman utama *customer* yang menampilkan *maps*, lokasi *customer* dengan tanda titik biru, lokasi halte dengan penanda warna merah didalam lingkaran warna hijau, lokasi *driver* angkot yang tersedia, tombol permintaan penjemputan, pilihan nomor angkot, pencarian lokasi tujuan yang akan menampilkan *direction*, tombol *logout* dan *settings*. Jika posisi *customer* di luar jangkauan halte pada lingkaran hijau, maka tombol utama tidak berfungsi atau warnanya menjadi redup. Apabila sudah berada di dalam halte atau bergerak dijangkau halte, maka tombol utama berfungsi dan warnanya menjadi jelas. pada aplikasi angkot terdapat proses ketika permintaan penjemputan diklik pada halaman utama *customer*. Sebelum mengklik pilih lokasi tujuan dan nomer angkot. Setelah itu, klik tombol permintaan penjemputan maka proses pencarian

driver yang tersedia akan terdeteksi dengan radius 5 km yang akan menampilkan *form* informasi nama *driver*, nomor *handphone*, plat nomer kendaraan, dan pernyataan bahwa anda akan dijemput. Aplikasi tersebut juga menampilkan jarak antara lokasi *smartphone customer* dengan lokasi *smartphone driver* hingga mengetahui *driver* sudah sampai di lokasi customer dengan jarak 10 meter.

3. Tampilan Halaman *Settings Customer*



Gambar 4.22 Tampilan Halaman *Settings Customer*

Penjelasan dari gambar diatas adalah halaman *settings customer* untuk mengubah nama dan nomor *handhone*.

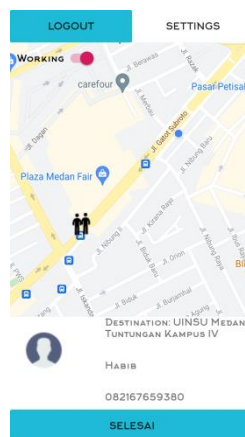
4. Tampilan Halaman Utama *Driver*



Gambar 4.23 Tampilan Halaman Utama *Driver*

Penjelasan dari gambar tersebut adalah pada halaman utama *driver* tampilan halaman utama menampilkan *maps*, lokasi *driver* dengan tanda titik biru, *working on/off*, tombol *logout* dan *settings*. Sebelum *driver* menampilkan lokasinya maka *driver* harus mengaktifkan tombol *working* atau menjadi *online* agar terdeteksi lokasinya dengan halaman utama *customer*.

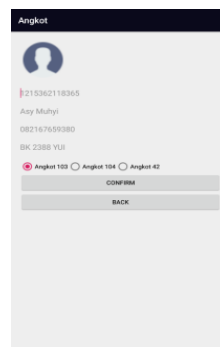
5. Tampilan *Driver* Menerima Permintaan *Customer*



Gambar 4.24 Tampilan *Driver* Menerima Permintaan *Customer*

Penjelasan dari gambar tersebut adalah *driver* menerima permintaan penjemputan dari *customer* lebih dari satu orang yang menampilkan informasi *customer* berupa lokasi penjemputan dengan tanda 2 orang berdiri, *destination*, nama, dan nomer *handphone*.

6. Tampilan Halaman *Settings Driver*



Gambar 4.25 Tampilan Halaman *Settings Driver*

Penjelasan dari gambar tersebut adalah setelah di klik tombol *settings* pada halaman utama *driver* maka akan tampil halaman *settings driver* untuk mengubah nama, nomor *handhone*, nomor plat kendaraan dan pilihan jenis nomor angkot. Setelah itu melakukan *confirm*, agar ketika permintaan penjemputan di halaman *customer* kepada *driver* maka nama *driver*, nomor *handhone*, nomor plat kendaraan tersebut akan tampil berupa *from* informasi di halaman *customer* pada aplikasi angkot.

4.5.1 Menentukan Hak Akses

Hak akses ini di gunakan oleh siapa saja yang memakai aplikasi berbasis android yang ada di Medan, karena aplikasi ini hanya digunakan di Medan. Untuk mendapatkan aplikasinya dengan cara melakukan instalasi pada *smartphone* android dengan nama aplikasinya adalah Angkot.

4.5.2 Implementasi

Implementasi aplikasi angkutan umum beserta penjelasan fungsi-fungsi yang terletak pada aplikasi tersebut. Berikut penjelasan simulasinya:

1. *Customer* melakukan *register* pada aplikasi angkot berbasis android.
2. *Driver* melakukan *register* pada Admin.
3. Admin melakukan input data *driver* pada halaman *database*.
4. *Driver* dan *Customer* melakukan *sign in* pada aplikasi tersebut.
5. *Driver* dan *Customer* masuk ke halaman utama berupa *maps* yang langsung menunjukkan posisinya.
6. *Driver* dan *Customer* mengklik tombol *settings* untuk mengupdate data pribadi.
7. *Driver* mengaktifkan tombol *working* untuk *online* agar bisa menerima permintaan penjemputan dari *customer*.
8. Pada aplikasi *customer*, tombol permintaan penjemputan hanya bisa berfungsi di Halte. Jika di luar jangkauan halte, tombol permintaan penjemputan tidak berfungsi. Setelah di Halte maka *customer* mengklik tombol permintaan penjemputan yang akan menampilkan informasi dan lokasi *driver* tersebut.

9. Setelah *customer* mengklik tombol permintaan penjemputan, informasi dan lokasi *customer* akan tampil di aplikasi *driver*, setelah itu *driver* mengklik tombol selesai.
10. Setelah itu, *driver* dan *customer* mengklik tombol *logout* yang akan kembali ke halaman awal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

1. Aplikasi angkutan umum dapat membuat *customer* mengetahui informasi dan lokasi *driver* angkot secara *real time* dengan menggunakan sistem GPS dan *driver* juga mengetahui lokasi *customer* yang akan dijemput.
2. Aplikasi ini menyediakan fitur deteksi lokasi *customer* di Halte yang membuat halte berfungsi secara maksimal.
3. Dengan menggunakan aplikasi ini, pihak PT. Medan Rahayu Ceria dapat memonitoring armada yang beroperasi apabila *driver* mengaktifkan tombol *working*.

5.2 Saran

Berdasarkan kesimpulan dan analisis yang telah dilakukan, maka terdapat beberapa saran sebagai berikut:

1. Aplikasi ini perlu ditambahkan fitur estimasi waktu tibanya di Halte.
2. Aplikasi ini perlu ditambahkan fitur kapasitas angkot apabila sudah penuh.
3. Aplikasi ini berbasis android, maka perlu adanya pengembangan aplikasi berbasis IOS sehingga aplikasi ini multi platform.
4. Aplikasi ini perlu dimasukkan ke Playstore.

DAFTAR PUSTAKA

- Amal, S., *Https://daerah.sindonews.com/read/1039281/151/angkot-ngetem-bikin-macet-1441078867*, Diakses pada tanggal 1 Agustus 2019.
- EMS, T., 2013, *Android All In One*, PT Elex Media Komputindo, Jakarta.
- Fantastic, *Https://Mainthebest.Com/Smartphones/Tingkatan-Versi-Android/*, Diakses pada tanggal 20 Juli 2019.
- Https://console.developers.google.com/apis/library?filter=category:maps&project=mystical-tensor-242116*, Diakses pada tanggal 1 Agustus 2019.
- Https://developer.android.com/studio/intro?hl=ID*, Diakses pada tanggal 9 September 2019.
- Irawan, M.D.dan Simargolang, S.A., 2018, Implementasi E-Arsip Pada Program Studi Teknik Informatika, *Jurnal Teknologi Informasi*, **2** , 67–68.
- Irwansyah, E., 2013,*SISTEM INFORMASI GEOGRAFIS: Prinsip Dasar Dan Pengembangan Aplikasi*, Digibooks, Yogyakarta.
- Istiyanto, J.E., 2013,*Pemrograman Smartphone Menggunakan SDK Android Dan Hacking Android*, Graha Ilmu, Yogyakarta.
- Junindar,2017, *Xamarin Android: Mudah Membangun Aplikasi Mobile*, Ebookuid.
- Kadir, A., 2013,*Pemrograman Aplikasi Android*, Penerbit Andi, Yogyakarta.
- Kadir, A. dan Triwahyuni, T.C., 2013,*Pengantar Teknologi Informasi Edisi Revisi*, Penerbit Andi, Yogyakarta.
- Marjuki, B., 2016,*Survei Dan Pemetaan Menggunakan GPS Dan GIS*, Penerbit Bramantiyo Marjuki.
- Mutiawati,Dkk., 2019,*Kinerja Pelayanan Angkutan Umum Jalan Raya*, Deepublish, Yogyakarta.
- Nasution, M.I.P., 2014, Keunggulan Kompetitif dengan Teknologi Informasi, *Jurnal Elektronik*.
- Paramanindo, F., 2014, *APLIKASI SIMULASI UJIAN NASIONAL PADA SMA NEGERI 1 RAMBANG DANGKU BERBASIS DESKTOP*, *Eprint*, Politeknik Negeri Sriwijaya.

- Rickyanto, I., 2003, *Pemrograman Berorientasi Objek Dengan Java 2 (JDK 1.4)*, Penerbit Andi, Yogyakarta.
- Simarmata,S,P.,[Http://www.medanbisnisdaily.com/news/online/read/2017/07/30/1126/halte_di_kota_medan_tak_terawat_dan_berubah_fungsi/](http://www.medanbisnisdaily.com/news/online/read/2017/07/30/1126/halte_di_kota_medan_tak_terawat_dan_berubah_fungsi/), Diakses pada tanggal 1 Agustus 2019.
- Sitohang, O. dan Situmorang, A.E., 2019, Analisis Efektifitas Halte di Kota Medan, *Jurnal Rekayasa Konstruksi Mekanika Sipil*,**Vol. 2, No.1**, 68.
- Suryana, D., 2018,*Android Studio: Belajar Android Studio*, Dayat Suryana Independent, Bandung.
- Sutrabri, T., 2014,*Pengantar Teknologi Informasi*, Penerbit Andi, Yogyakarta.
- Wahana, K., 2014,*Membangun Aplikasi Mobile Cross Platform Dengan Phonegap*, PT Elex Media Komputindo, Jakarta.
- Wibowo, T.,[Https://sumutpos.co/2018/01/12/jumlah-angkot-menyusut-tinggal-5-000-unit/](https://sumutpos.co/2018/01/12/jumlah-angkot-menyusut-tinggal-5-000-unit/), Diakses pada tanggal 1 Agustus 2019.
- Yurindra, 2017, *Software Engineering*, Deepublish, Yogyakarta.
- Zufria, I., 2013, *Pemodelan Berbasis UML (Unified Modeling Language) Dengan Strategi Teknik Orientasi Objek User Centered Design (UCD) Dalam Sistem Administrasi Pendidikan*.*Researchgate*.

LAMPIRAN

1. Source Code

MainActivity.java

```
private void showLoginDialog() {
    final AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("DRIVER");
    dialog.setMessage("Tolong Gunakan Email Untuk Masuk");

    LayoutInflater inflater = LayoutInflater.from(this);
    View login_layout = inflater.inflate(R.layout.activity_driver_login, null);

    final MaterialEditText mEmail = login_layout.findViewById(R.id.editEmail);
    final MaterialEditText mPass = login_layout.findViewById(R.id.editPass);

    dialog.setView(login_layout);

    //set button
    dialog.setPositiveButton("SIGN IN", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
            //set disable button jika sedang processing
            //cek validasi
            if (TextUtils.isEmpty(mEmail.getText().toString())) {
                Snackbar.make(rootLayout, "Tolong Masukkan Alamat Email",
Snackbar.LENGTH_SHORT)
                    .show();
                return;
            }
            if (TextUtils.isEmpty(mPass.getText().toString())) {
                Snackbar.make(rootLayout, "Tolong Masukkan Kata Sandi",
Snackbar.LENGTH_SHORT)
                    .show();
                return;
            }
            if (mPass.getText().toString().length() < 6) {
```

```

        Snackbar.make(rootLayout, "Kata Sandi Terlalu Pendek !!!",
Snackbar.LENGTH_SHORT)
            .show();
        return;
    }

    final AlertDialog waitingDialog = new SpotsDialog(MainActivity.this);
    waitingDialog.show();
    //login
    mAuth.signInWithEmailAndPassword(mEmail.getText().toString(),
mPass.getText().toString())
        .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {
                waitingDialog.dismiss();
                startActivity(new Intent(MainActivity.this, DriverMapActivity.class));
                finish();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                waitingDialog.dismiss();
                Snackbar.make(rootLayout, "Gagal Login " + e.getMessage(),
Snackbar.LENGTH_SHORT)
                    .show();
                //Aktif Button
                mDriver.setEnabled(true);
            }
        });
    }

});
dialog.setNegativeButton("REGISTER", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
        //set disable button jika sedang prosessing
        //cek validasi
        if (TextUtils.isEmpty(mEmail.getText().toString())) {
            Snackbar.make(rootLayout, "Tolong Masukkan Alamat Email",

```

```

Snackbar.LENGTH_SHORT)
        .show();
        return;    }
        if (TextUtils.isEmpty(mPass.getText().toString())) {
            Snackbar.make(rootLayout, "Tolong Masukkan Kata Sandi",
Snackbar.LENGTH_SHORT)
                .show();
            return;
        }
        if (mPass.getText().toString().length() < 6) {
            Snackbar.make(rootLayout, "Kata Sandi Terlalu Pendek !!!",
Snackbar.LENGTH_SHORT)
                .show();
            return;
        }

        final AlertDialog waitingDialog = new SpotsDialog(MainActivity.this);
        waitingDialog.show();
        //login
        mAuth.createUserWithEmailAndPassword(mEmail.getText().toString(),
mPass.getText().toString())
            .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                @Override
                public void onSuccess(AuthResult authResult) {
                    waitingDialog.dismiss();
                    startActivity(new Intent(MainActivity.this, DriverMapActivity.class));
                    finish();
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    waitingDialog.dismiss();
                    Snackbar.make(rootLayout, "Gagal Registrasi " + e.getMessage(),
Snackbar.LENGTH_SHORT)
                        .show();
                    //Aktif Button
                    mDriver.setEnabled(true);
                }
            }); } });

```

CustomerMapActivity.java

```
boolean getDriversAroundStarted = false;
List<Marker> markers = new ArrayList<Marker>();
private void getDriversAround(){
    getDriversAroundStarted = true;
    DatabaseReference driverLocation =
    FirebaseDatabase.getInstance().getReference().child("driversAvailable");

    GeoFire geoFire = new GeoFire(driverLocation);
    GeoQuery geoQuery = geoFire.queryAtLocation(new
    GeoLocation(mLastLocation.getLatitude(),mLastLocation.getLongitude()), radius);

    geoQuery.addGeoQueryEventListener(new GeoQueryEventListener() {
        @Override
        public void onKeyEntered(String key, GeoLocation location) {

            for(Marker markerIt : markers){
                if(markerIt.getTag().equals(key))
                    return;
            }

            LatLng driverLocation = new LatLng(location.latitude, location.longitude);

            Marker mDriverMarker = mMap.addMarker(new MarkerOptions()
                .position(driverLocation)
                .title(key)
                .icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_car)));
            mDriverMarker.setTag(key);
            markers.add(mDriverMarker);


        }
    });
}
```

DriverMapActivity.java

```
Marker pickupMarker;
private DatabaseReference assignedCustomerPickupLocationRef;
private ValueEventListener assignedCustomerPickupLocationRefListener;
private void getAssignedCustomerPickupLocation(){
    assignedCustomerPickupLocationRef =
    FirebaseDatabase.getInstance().getReference().child("customerRequest").child(customerId).child("l");
    assignedCustomerPickupLocationRefListener =
    assignedCustomerPickupLocationRef.addValueEventListener(new
    ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(dataSnapshot.exists() && !customerId.equals("")){
                List<Object> map = (List<Object>) dataSnapshot.getValue();
                double locationLat = 0;
                double locationLng = 0;
                if(map.get(0) != null){
                    locationLat = Double.parseDouble(map.get(0).toString());
                }
                if(map.get(1) != null){
                    locationLng = Double.parseDouble(map.get(1).toString());
                }
                pickupLatLng = new LatLng(locationLat,locationLng);
                pickupMarker = mMap.addMarker(new
                MarkerOptions().position(pickupLatLng).title("pickup
                location").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_pickup)));
                getRouteToMarker(pickupLatLng);
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
        }
    });
}
```


2. Surat Balasan Penelitian



P.T. RAHAYU MEDAN CERIA

Jl. Setia Budi Komplek Pertokoan Setia Budi Centre B - 12 Telp. 8215838 - 8226251 Medan - 20122

SURAT KETERANGAN
No.169/SK/VII/19

Yang bertanda-tangan dibawah ini :

Nama	: Eko Wahyudi
Jabatan	: Ka. Personalia
Alamat	: Jl. Setia budi Koop. Pertokoan Setiabudi Centre B-12 Medan

PT. Rahayu Medan Ceria dengan ini menerangkan bahwa :

Nama	: Habib Asy Muhyi
NIM	: 72154036
Jurusan	: Sistem Informasi

Dengan ini kami beri izin untuk melakukan riset di perusahaan kami sesuai dengan waktu yang ditentukan oleh perusahaan.

Demikianlah surat keterangan ini diperbuat dengan sebenarnya agar dipergunakan seperlunya.

Medan, 26 Juli 2019



cc. - File