

**PENGUNAAN FITUR *MAPS LOCATION* DALAM PERANCANGAN
APLIKASI *BOOKING ORDER* RENTAL MOBIL PADA
CV.SABAI RENT CAR BERBASIS ANDROID**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER
Pada Jurusan Teknik Informatika
Institut Informatika Dan Bisnis Darmajaya Bandar Lampung



Oleh

RYAN CAHYA NUGRAHA

NPM.1411010051

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2018**



PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan tanggung jawaban sepenuhnya berada dipundak saya.

Bandar Lampung, 10 Desember 2018



Ryan Cahya Nugraha

NPM.1411010051

HALAMAN PERSETUJUAN

**Judul Skripsi : PENGGUNAAN FITUR MAPS LOCATION DALAM
PERANCANGAN APLIKASI BOOKING ORDER RENTAL
MOBIL PADA CV.SABAI RENT CAR BERBASIS ANDROID**

Nama : Ryan Cahya Nugraha

NPM : 1411010051

Jurusan : SI Teknik Informatika



Menyetujui :

Ketua Jurusan

Pembimbing

Teknik Informatika

Harivanto Wibowo, S.Kom., M.T.I

Yuni Arkhiansyah, M.Kom

NIK.00051297

NIK.00480802

HALAMAN PENGESAHAN

Telah di uji dan dipertahankan didepan tim penguji Skripsi Program Studi Teknik Informatika (TI) Institut Informatika & Bisnis Darmajaya Bandar Lampung dan dinyatakan diterima untuk memenuhi syarat guna memperoleh gelar Sarjana Komputer.

MENGESAHKAN

1. **Tim Penguji**

Tanda Tangan

Ketua : Ketut Artaye, S.Kom., M.T.I



Anggota : Apri Triansah, S.Kom., M.T.I



2. **Dekan Fakultas Ilmu Komputer**



Sriyanto, S.Kom., MM

NIK.00210800

Tanggal Lulus Ujian Skripsi : 18 September 2018

RIWAYAT HIDUP

1. Identitas

- a. Nama : Ryan Cahya Nugraha
- b. NPM : 1411010051
- c. Tempat / Tanggal Lahir : Bandar Lampung, 22 Juni 1996
- d. Agama : Islam
- e. Alamat : Jalan Pahlawan No.17 Kel.Surabaya
Kec.Kedaton Kota.Bandar Lampung
- f. Suku : Jawa
- g. Kewarganegaraan : Indonesia
- h. E-Mail : Teknologi55@gmail.com
- i. Nomor Handphone : 0896-4930-3025

2. Riwayat pendidikan yang pernah ditempuh oleh penulis, antara lain:

- a. Pendidikan Sekolah Dasar (SD), SD Negeri 6 Penengahan Bandar Lampung lulus pada tahun 2008.
- b. Pendidikan Sekolah Menengah Pertama (SMP), SMP Negeri 8 Bandar Lampung lulus pada tahun 2011.
- c. Pendidikan Sekolah Menengah Kejuruan (SMK), SMK 2 Mei Bandar Lampung lulus pada tahun 2014.
- d. Pada tahun 2014 diterima di IIB Darmajaya Jurusan S1 Teknik Informatika.

Bandar Lampung, 10 Desember 2018

Ryan Cahya Nugraha

NPM.1411010051

HALAMAN PERSEMBAHAN

Bismillahirrohmanirohim,

Alhamdulillah, segala puji dan rasa syukur hanya kepada Allah SWT. Yang telah memberikan nikmat, waktu dan juga kesempatan saya untuk dapat menyelesaikan laporan skripsi ini.

Kupersembahkan skripsi ini untuk :

- ❖ Kedua Orang Tua saya yang selalu mendukungku, memberi semangat dan do'a-doanya agar anaknya dapat berhasil dan sukses dimasa yang akan datang, semoga semua ini menjadi salah satu karya dari ku yang akan membuat kedua orangtua ku bahagia dan bangga terhadapku.
- ❖ Teman-teman Teknik Informatika angkatan 2014 yang namanya tidak bisa disebutkan satu persatu, yang selalu memberikan semangat dan motivasi dalam proses perkuliahan.
- ❖ Terima kasih banyak kepada semua dosen IIB darmajaya dan juga kepada dosen pembimbingku bapak Hariyanto Wibowo, S.Kom., M.T.I yang telah sabar untuk membimbing untuk menyelesaikan Skripsi ini. Terima kasih bimbingannya.
- ❖ Dan Almamater IIB Darmajaya.

MOTTO

*“Tetaplah fokus kepada tujuanmu,
Walau banyak permasalahan yang menghadang,
Yakinlah suatu saat nanti engkau dapat mencapainya”*

ABSTRAK

PENGGUNAAN FITUR *MAPS LOCATION* DALAM PERANCANGAN APLIKASI *BOOKING ORDER* RENTAL MOBIL PADA CV.SABAI RENT CAR BERBASIS ANDROID

OLEH

RYAN CAHYA NUGRAHA

1411010051

CV.Sabai Rent Car merupakan unit usaha yang bergerak di bidang jasa transportasi yaitu penyewaan kendaraan Mobil atau rental Mobil, yang salah satunya difasilitasi langsung dengan menggunakan *driver* yang berada di kota Bandar Lampung. Dalam penerapan pemesanan kendaraan saat ini CV.Sabai Rent Car masih menggunakan cara pemesanan kendaraan secara konvensional yaitu *customer* yang ingin *booking* kendaraan harus datang langsung ke lokasi rental, dalam hal seperti ini tidak efektif dalam melakukan pelayanan kepada *customer*. Sehingga sangat dibutuhkannya perancangan aplikasi *booking order* rental Mobil. Metode pengembangan perangkat lunak pada penelitian ini menggunakan metode *Prototype* dengan tahapan-tahapan diantaranya komunikasi, rencana cepat, pemodelan desain cepat, konstruksi prototipe dan pengembangan. Perancangan aplikasi ini digunakan untuk mempermudah *customer* dalam melakukan pemesanan kendaraan Mobil secara *realtime* yang berada di Bandar Lampung dengan menggunakan *Smartphone*. Sistem aplikasi *booking order* rental Mobil berbasis Android difasilitasi dengan fitur *maps location*, dimana pada fitur ini dapat menentukan titik lokasi penjemputan *customer*, sehingga *driver* dapat mencari titik lokasi penjemputan dengan mudah.

Kata Kunci: Aplikasi *Booking*, Rental Mobil, *Maps Location*, Android

ABSTRACT

THE USE OF MAP LOCATION FEATURE IN THE ANDROID-BASED CAR BOOKING ORDER APPLICATION IN SABAI RENT CAR LTD

By

RYAN CAHYA NUGRAHA

1411010051

Sabai Rent Car was the business unit engaged in the car rent service. One of the facilities in this business unit was driver in Bandar Lampung. The problem statement of this research was that this business unit still used the conventional-booking method so that the customers who needed to book the car had to come to the office and it became an ineffective service for the business unit. The solution to overcome this problem was that the application with android-based system had to be designed. To develop this software, the prototype was used with several stages i.e., the communication, the direct planning, the quick-designing model, the prototype construction, and the development. This application was used to facilitate customers in booking the car in real time using Smartphone throughout Bandar Lampung. This application was also facilitated with maps location used to find the customer location so that the driver was able to found the customer easily.

Keywords: Booking Application, Car Rent, Maps Location, Android



PRAKATA

Assalamu'alaikum Wr. Wb.

Puji syukur kepada Allah SWT, yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan penelitian skripsi ini. Skripsi ini disusun dengan tujuan untuk melengkapi syarat untuk menyelesaikan dan mendapatkan gelar S1 Jurusan Teknik Informatika, Saya menyadari dalam penulisan ini masih jauh dari sempurna dan masih terdapat banyak kekurangan. Penelitian ini diselesaikan setelah melalui berbagai ujian dan tahapan bimbingan serta atas bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini tidak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Firmansyah Y.A, MBA, M.Sc. Selaku Rektor Institut Informatika dan Bisnis Darmajaya.
2. Bapak Dr. R.Z Abdul Aziz, ST., M.T. Selaku Wakil Rektor 1 Bidang Akademik dan Kemahasiswaan Institut Informatika dan Bisnis Darmajaya.
3. Bapak Yuni Arkhiansyah, M.Kom. Selaku Kepala Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya.
4. Bapak Hariyanto Wibowo, S.Kom., M.T.I. Selaku pembimbing yang telah memberikan bimbingan, saran dan bantuan selama saya menyelesaikan penelitian skripsi ini.
5. Kedua Orang tua saya, sebagai motivator saya seumur hidup yang selalu mensupport saya agar dapat menyelesaikan penelitian skripsi ini tepat waktu.
6. Para Dosen, Staf dan Karyawan Institut Informatika dan Bisnis Darmajaya Bandar Lampung yang telah memberi bantuan baik langsung maupun tidak langsung selama saya menjadi mahasiswa di IIB Darmajaya.

7. Para Teman dan Sahabat mahasiswa Institut Informatika dan Bisnis Darmajaya angkatan 2014, yang saling memotivasi agar bisa lulus tepat waktu.
8. Semua pihak yang telah memberikan bantuan dan petunjuk sehingga saya dapat lebih mudah dalam penyusunan skripsi ini.
9. Almamater tercinta Institut Informatika dan Bisnis Darmajaya yang telah memberikanku banyak bekal ilmu pengetahuan.

Penulis menyadari bahwa penyusunan skripsi ini masih banyak terdapat kekurangan dalam menyelesaikannya, baik dalam pembahasan, materi maupun dalam pengujian aplikasi, oleh karena itu kritik dan saran yang bersifat positif dan membangun merupakan masukan yang sangat berarti bagi penyempurnaan dimasa yang akan datang. Semoga hasil penelitian skripsi ini dapat bermanfaat dan dapat dijadikan bahan pertimbangan informasi bagi pihak yang berkepentingan.

Wassalamu'alaikum Wr. Wb.

Bandar Lampung, 10 Desember 2018

Ryan Cahya Nugraha

NPM.1411010051

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERNYATAAN.....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
RIWAYAT HIDUP	v
HALAMAN PERSEMBAHAN	vi
MOTTO	vii
ABSTRAK	viii
<i>ABSTRACT</i>	ix
PRAKATA	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	
2.1 Sistem Informasi Geografis	6
2.2 Google Map API.....	6
2.3 <i>Location APIs</i>	7

2.4 Aplikasi.....	7
2.5 <i>Booking Order</i>	7
2.6 Rental Mobil.....	8
2.7 <i>Java Script Object Nonation (JSON)</i>	8
2.8 Android Studio	8
2.9 Android.....	10
2.10 Pengembangan Sistem <i>Prototype</i>	18
2.11 UML (<i>Unified Modelling Language</i>)	19
2.12 <i>Use Case Diagram</i>	20
2.13 <i>Sequence Diagram</i>	22
2.14 <i>Activity Diagram</i>	22
2.15 <i>Class Diagram</i>	23
2.16 Pengujian Kotak Hitam (<i>Black-Box Testing</i>)	25
2.17 <i>Firestore Database</i>	26
2.17.1 Kemampuan Utama.....	26
2.17.2 Cara Kerja Sistem <i>Firestore Database</i>	27

BAB III METODELOGI PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak	28
3.1.1 <i>Communication</i>	28
3.1.2 <i>Quick Plan</i>	30
3.1.3 <i>Modeling Quick Design</i>	33
3.1.3.1 Desain UML (<i>Unified Modelling Language</i>) .	33
3.1.3.2 Rancangan <i>Use Case Diagram</i>	33
3.1.3.3 Rancangan <i>Activity Diagram</i>	35

3.1.3.4	Rancangan <i>Sequence Diagram</i>	39
3.1.3.5	Rancangan <i>Class Diagram</i>	44
3.1.3.6	Rancangan Antarmuka (<i>User Interface</i>).....	45
3.1.4	<i>Construction Of Prototype</i>	52
3.1.5	<i>Deployment Delivery And Feedback</i>	54

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1	Hasil Penelitian.....	55
4.2	Tampilan Aplikasi.....	55
4.2.1	Tampilan Menu Aplikasi Login.....	55
4.2.2	Tampilan Aplikasi Menu Daftar Pengguna Baru.....	56
4.2.3	Tampilan Aplikasi Menu Utama Pengguna.....	57
4.2.4	Tampilan Aplikasi Menu Informasi dan <i>Gallery</i>	58
4.2.5	Tampilan Halaman Menu Informasi Kendaraan.....	59
4.2.6	Tampilan Aplikasi Menu Formulir <i>Booking</i>	60
4.2.7	Tampilan Aplikasi Menu Fitur <i>Maps Location</i>	61
4.2.8	Tampilan Aplikasi Menu Status Pemesanan.....	62
4.2.9	Tampilan Aplikasi Menu Informasi Detail Pemesanan	63
4.2.10	Tampilan Aplikasi Menu Profil	64
4.2.11	Tampilan Aplikasi Menu Bantuan dan Tutorial.....	65
4.2.12	Tampilan Aplikasi Menu Utama <i>Admin</i>	66
4.2.13	Tampilan Aplikasi Menu Pendaftaran <i>Driver</i> Baru	67
4.3	Pengujian (<i>Testing</i>).....	68
4.4	Kelebihan dan Kekurangan Aplikasi	71

BAB V SIMPULAN DAN SARAN

5.1	Simpulan.....	73
5.2	Saran	73

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Metode <i>Prototype</i> (Pressman, 2012:51).....	19
Gambar 2.2 Notasi <i>Sequence Diagram</i>	22
Gambar 3.1 Analisa Sistem Yang Sedang Berjalan.....	30
Gambar 3.2 Sistem Aplikasi Yang Akan Dibuat	31
Gambar 3.3 <i>Use Case Diagram</i> Sistem Aplikasi <i>Booking</i> Kendaraan Mobil ..	34
Gambar 3.4 <i>Activity Diagram</i> Login Aplikasi	35
Gambar 3.5 <i>Activity Diagram</i> Menu Utama	36
Gambar 3.6 <i>Activity Diagram</i> Lanjutan Menu Utama.....	37
Gambar 3.7 <i>Activity Diagram</i> Pemesanan Kendaraan.....	38
Gambar 3.8 <i>Sequence Diagram</i> Pendaftaran Pengguna	39
Gambar 3.9 <i>Sequence Diagram</i> Login Aplikasi	40
Gambar 3.10 <i>Sequence Diagram</i> Menu Aplikasi	41
Gambar 3.11 <i>Sequence Diagram</i> Pemesanan Kendaraan	42
Gambar 3.12 <i>Sequence Diagram</i> Pendaftaran <i>Driver</i>	43
Gambar 3.13 <i>Class Diagram</i> Aplikasi <i>Booking Order</i> Rental Mobil.....	44
Gambar 3.14 Tampilan Login Aplikasi	45
Gambar 3.15 Tampilan Menu Utama <i>User</i>	46
Gambar 3.16 Tampilan Menu Informasi Kendaraan	47
Gambar 3.17 Tampilan Menu Formulir Pemesanan <i>Booking</i>	48
Gambar 3.18 Tampilan Menu Pendaftaran Pelanggan Baru.....	49
Gambar 3.19 Tampilan Menu Fitur <i>Maps</i> Penjemputan Pelanggan	50

Gambar 3.20 Tampilan Menu Detail Informasi <i>Booking</i>	51
Gambar 3.21 Tampilan Menu Utama <i>Admin</i>	52
Gambar 4.1 Tampilan Halaman Menu Login <i>Admin</i> , Pengguna dan <i>Driver</i>	55
Gambar 4.2 Tampilan Halaman Menu Daftar Pengguna Baru	56
Gambar 4.3 Tampilan Halaman Menu Utama Pengguna	57
Gambar 4.4 Tampilan Halaman Menu Informasi Rental dan <i>Gallery</i>	58
Gambar 4.5 Tampilan Halaman Menu Informasi Kendaraan Mobil	59
Gambar 4.6 Tampilan Halaman Menu Formulir Pemesanan.....	60
Gambar 4.7 Tampilan Halaman Menu Fitur <i>Maps Location</i>	61
Gambar 4.8 Tampilan Halaman Menu Status Informasi Pemesanan	62
Gambar 4.9 Tampilan Halaman Menu Informasi Detail Pemesanan	63
Gambar 4.10 Tampilan Halaman Menu Profil Pengguna.....	64
Gambar 4.11 Tampilan Halaman Menu Bantuan dan Tutorial Pemesanan.....	65
Gambar 4.12 Tampilan Halaman Menu Utama <i>Admin</i>	66
Gambar 4.13 Tampilan Halaman Menu Pendaftaran <i>Driver</i> Baru	67

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-Simbol Diagram <i>Use Case</i>	20
Tabel 2.2 Komponen <i>Activity Diagram</i>	23
Tabel 2.3 Simbol-Simbol Diagram Kelas	24
Tabel 3.1 Informasi Detail Kendaraan Mobil CV.Sabai Rent Car	29
Tabel 4.1 Uji Pilihan Menu Tombol Aplikasi.....	68
Tabel 4.2 Uji Fungsi Aplikasi	70
Tabel 4.3 Hasil Instalasi Pada Perangkat <i>Smartphone</i>	71

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi pada zaman dimana alat digital sebagai kebutuhan masyarakat Indonesia sudah tidak dapat dipisahkan, baik dibidang kesehatan, pendidikan, telekomunikasi, maupun sebagai media marketing dalam pemasaran sebuah produk atau jasa, sehingga hal tersebut sangat berperan penting pada era zaman digital saat ini.

Dengan meningkatnya teknologi digital saat ini sehingga banyak usaha-usaha baru yang muncul, untuk mengikuti perkembangan zaman dengan menggunakan sentuhan teknologi pada usahanya, sehingga usaha tersebut tidak kehilangan peluang dalam menarik *customer*. Salah satu bidang usaha yang harus melakukan sentuhan teknologi digital yaitu jasa penyewaan kendaraan Mobil atau rental Mobil.

CV.Sabai Rent Car merupakan unit usaha yang bergerak di bidang jasa transportasi yaitu penyewaan kendaraan Mobil atau rental Mobil, yang salah satunya difasilitasi langsung dengan menggunakan *driver* atau supir yang berada di kota Bandar Lampung. Sistem yang diterapkan pada CV.Sabai Rent Car saat ini sebelum menggunakannya fasilitas aplikasi yaitu, *customer* yang ingin melakukan pemesanan armada Mobil harus konfirmasi terlebih dahulu menggunakan Telepon maupun harus datang secara langsung ke lokasi rental Mobil tersebut untuk melakukan pemesanan Mobil. Dari sistem proses bisnis tersebut memiliki kekurangan, diantaranya lebih membutuhkan banyak waktu untuk harus datang ke lokasi rental Mobil CV.Sabai Rent Car. Permasalahan lain jika menggunakan sistem pemesanan secara konvensional, banyaknya *customer* yang melakukan pemesanan armada Mobil pada saat *weekend* maupun libur panjang sehingga pemilik rental mengalami permasalahan terhadap proses data antara armada kendaraan yang masih tersedia dengan armada kendaraan yang sudah dipesan sebelumnya.

Selain itu dalam melakukan usaha rental Mobil, peran dari *driver* rental merupakan salah satu peranan yang sangat penting dalam pelayanan terhadap *customer*, ketepatan waktu dalam penjemputan terhadap *customer* sangat berperan sebagai totalitas profesional pemilik usaha rental untuk menjaga nama baik usahannya kepada *customer*, namun sering kali *driver* mengalami kendala yaitu sulitnya menemukan lokasi penjemputan *customer* dikarenakan alamat yang diinfokan dari *customer* kurang lengkap sehingga *driver* harus mencari alamat yang sesuai dengan lokasi penjemputan, hal tersebut mengakibatkan *driver* harus membutuhkan waktu yang lebih lama untuk mencari alamat lokasi penjemputan yang sesuai dengan informasi yang diberikan dari *customer*.

Dalam permasalahan tersebut sehingga penulis ingin melakukan sebuah penelitian dengan merancang sebuah aplikasi *booking* rental Mobil yang difasilitasi dengan fitur *maps location* sehingga dapat menentukan lokasi penjemputan *customer* dengan tepat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijabarkan di atas, dapat dirumuskan beberapa permasalahan antara lain:

1. Bagaimana penerapan fitur *maps location* pada proses *booking* rental Mobil dapat membantu mempermudah *driver* dalam mengetahui lokasi penjemputan ke *customer*?
2. Bagaimana merancang dan membuat sebuah aplikasi *booking* rental Mobil ini dapat membantu perusahaan dalam pelayanan jasa yang ditawarkan, serta dapat menginformasikan fasilitas-fasilitas yang diberikan kepada *customer*?
3. Bagaimana *customer* dapat mudah mencari armada Mobil rental yang sesuai dengan yang diinginkan?
4. Bagaimana perancangan aplikasi *booking* rental ini dapat mempermudah penjadwalan pada status ketersediaan armada Mobil?

1.3 Batasan Masalah

Batasan masalah yang penulis ambil dalam penulisan ilmiah perancangan aplikasi *booking order* rental Mobil antara lain:

1. Aplikasi ini hanya akan berjalan pada pengembangan perangkat lunak berbasis Android.
2. Aplikasi yang dirancang hanya membahas proses pemesanan armada Mobil Sabai Rent Car.
3. Aplikasi yang dirancang tidak mencakup dalam proses pembayaran *booking*.
4. Penerapan fitur *maps location* hanya pada titik lokasi penjemputan *customer* / pengguna.

5. Perancangan aplikasi *driver* hanya menampilkan keseluruhan pesanan yang ada pada pesanan aplikasi *admin* tidak menspesifikasikan pesanan ke masing-masing *driver*.
6. Pemesanan kendaraan tidak melayani untuk tujuan ke luar provinsi Lampung.

1.4 Tujuan Penelitian

Adapun tujuan penulisan penelitian ini antara lain:

1. *Customer* dapat melakukan pemesanan kendaraan Mobil dengan mudah dan cepat.
2. Dapat menerapkan fitur *maps location* lokasi penjemputan.
3. Dapat merancang aplikasi untuk mengetahui secara *real time* ketersediaan armada Mobil pada CV.Sabai Rent Car.
4. Mempermudah driver rental dalam mengetahui titik lokasi penjemputan *customer*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan *customer*, baik *customer* yang berada didalam kota Bandar Lampung maupun dari luar provinsi Lampung dapat melakukan pemesanan armada Mobil hanya dengan menggunakan aplikasi, tanpa harus datang langsung ke lokasi rental.

1.6 Sistematika Penulisan

Adapun sistematika dalam penelitian proposal ini antara lain:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang konsep – konsep dan teori – teori penunjang yang berhubungan dengan masalah yang di teliti dan yang diambil dari beberapa referensi.

BAB III METODE PENELITIAN

Berisi mengenai gambaran umum metode yang digunakan, sumber data, data yang diperlukan, teknik pengumpulan data dan teknik analisis data .

BAB IV HASIL DAN PEMBAHASAN

Bab ini menguraikan pembahasan tentang implementasi dan pengujian serta membahas masalah yang dikemukakan dengan membandingkan teori yang didapat dengan penelitian dilapangan.

BAB V KESIMPULAN DAN SARAN

Bab ini mengemukakan kesimpulan dari pembahasan yang telah dibahas dan saran untuk perusahaan yang merupakan tindak lanjut dari kesimpulan yang penulis ambil.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

LANDASAN TEORI

2.1 Sistem Informasi Geografis

Menurut Riyanto (2009:35) Sistem Informasi Geografis (SIG) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah database.

Sedangkan menurut Budiyo (2005, 2) Sistem Informasi Geografis adalah kumpulan yang terorganisir dari perangkat keras, perangkat lunak, data geografi, dan personel yang didesain untuk memperoleh, menyimpan, memperbaiki, memanipulasi, menganalisis, dan menampilkan semua bentuk informasi yang bereferensi geografi. Data sistem informasi geografis adalah salah satu komponen krusial dan penting. Di dalam SIG terdapat dua jenis data yaitu data spasial dan data atribut/non spasial.

2.2 Google Map API

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat di gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Selain itu dapat menambahkan fitur *Google Maps* dalam web yang telah dibuat atau pada blog yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*. (Kindarto, 2008).

2.3 Location APIs

Location APIs memudahkan dalam membuat aplikasi berbasis lokasi tanpa membutuhkan fokus pada detail yang mendasari teknologi lokasi. Juga meminimalkan penggunaan daya dengan menggunakan kemampuan yang ada dalam perangkat keras yang digunakan. *Location APIs* merupakan API yang simple sehingga memungkinkan developer yang menentukan kebutuhan seperti akurasi atau daya rendah daripada layanan penyedia lokasi itu sendiri. *Location APIs* memberikan aplikasi akses yang terbaik dalam mendeteksi lokasi terbaru user.

2.4 Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer eksekutif*, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

2.5 Booking Order

Perngertian *Booking* atau *reservasi* adalah sebuah proses perjanjian yang berupa pemesanan produk barang ataupun jasa, namun belum ditutup dengan sebuah transaksi jual beli. Proses *reservasi* ditandai dengan adanya proses tukar menukar informasi antara produsen dan konsumen mengenai produk ataupun jaya yang dipesan (Pamudji).

Menurut Bodnar dan Hopwood (2001:323), yaitu: “bahwa pengadaan barang adalah proses bisnis dalam memilih sumber daya-sumber daya, pemesanan dan perolehan barang atau jasa”.

2.6 Rental Mobil

Rental Mobil atau penyewaan kendaraan Mobil menurut Tesaurus Bahasa Indonesia (2008:408), rental adalah penyewaan.

Pengertian Mobil menurut Heru Budianto, Mobil adalah sarana berkendara. Selain motor, Mobil adalah pilihan lain dalam suatu keluarga sebagai alat transportasi.

2.7 *Java Script Object Nonation (JSON)*

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, *JavaScript*, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data [JSON].

2.8 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) 3 Edisi. 1 Volume. 1 Agustus 2015 ISSN : 2089-9033 pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event *Google I/O Conference* untuk tahun 2013. Sejak

saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*). Android studio memiliki fitur:

- a. Projek berbasis pada *Gradle Build*.
- b. *Refactory* dan pembenahan bug yang cepat.
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard And App-signing* untuk keamanan.
- e. Memiliki GUI (*Graphical User Interface*) aplikasi Android lebih mudah.
- f. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

1) Kelebihan Android Studio

- a. Menurut pengembang Android Studio, software ini dilengkapi dengan intellegent code editor yang mampu mengolah dan menganalisis kode secara lengkap yang menjadikan developer semakin produktif. Selain itu pula, Android Studio dilengkapi dengan Code Templates dan Github integration yang memudahkan developer Android dalam mengembangkan aplikasi mereka dari sampel-sampel kode yang disediakan ataupun mengimpornya dari Github.
- b. Android Studio dilengkapi dengan emulator yang mencakup semua devices, baik ukuran maupun bentuk. Keunggulan tersebut jelas memudahkan developer untuk melihat hasil project mereka dari berbagai devices yang ada.
- c. Android Studio dilengkapi dengan Flexible Gradle-based build system, yang berfungsi untuk membuat banyak Apk untuk aplikasi Android

dengan fitur yang berbeda menggunakan project code yang sama. Fitur ini memudahkan developer untuk mengembangkan varian dari aplikasi yang mereka buat.

2) Kelemahan Android Studio

- a. Tidak banyak kelemahan yang bisa ditemukan untuk produk-produk Google, termasuk software Android Studio. Namun ukuran file Android Studio untuk Windows yang direkomendasikan sangat besar (hampir 1 GB) yang seringkali menjadi masalah saat mendownload software ini. Sedangkan kalau ingin mendownload software ini tanpa SDK tools hanya sekitar 240 MB untuk Windows.
- b. Jika mengalami error pada program tidak mudah untuk menemukan permasalahannya.

2.9 Android

Menurut Arifianto (2011, 1), Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Hermawan (2011, 1), Android merupakan OS (*Operating System*) mobile yang tumbuh ditengah OS lainnya yang berkembang.

Menurut Nasruddin Safaat h (Pemrograman aplikasi *mobile smartphone* dan tablet PC berbasis Android 2012:1) Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel,

Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007.

Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Dalam perjalanannya Android telah merilis banyak versi tentunya dengan berbagai macam fitur dan perbaikan ditiap versi yang baru. Berikut versi-versi Android yang telah dirilis hingga saat ini:

1. Android 1.1

Pertama kali platform Android diluncurkan pada tahun 2009, setahun sebelumnya Google telah merilis versi beta yang diperkenalkan kepada khalayak ramai. Android versi pertama dikenal dengan Android 1.1.

Berikut adalah fitur-fitur Android 1.1:

1. Mampu menyimpan attachment dalam pesan.
2. Waktu default screen in-call lebih lama bila menggunakan speakerphone plus kemampuan untuk menampilkan atau menyembunyikan dialpad.

2. Android 1.5 (*Cupcake*)

Pada tanggal 27 April 2009 Google juga merilis Android versi *Cupcake*. *Cupcake* adalah versi Android yang memulai tradisi penamaan kue untuk rilis Android. Ini dapat menambahkan beberapa fitur dan peningkatan baru dibanding versi sebelumnya.

Berikut adalah fitur yang dimilikinya:

1. Dukungan untuk tampilan aplikasi widget miniatur yang bisa disematkan di aplikasi lain (seperti home screen) dan menerima update berkala.

2. Rekaman video ditambahkan ke kamera bersamaan dengan kemampuan untuk langsung mengupload video ke YouTube.

3. Android 1.6 (*Donut*)

Google merilis Android 1.6 Donut pada bulan September tahun 2009. Penambahan fitur terbesar adalah dukungan untuk CDMA yang memperluas pasar pengguna baru ke Android. CDMA adalah teknologi yang digunakan oleh jaringan mobile Amerika pada saat itu.

Berikut adalah fitur yang dimilikinya:

1. Navigasi Google Maps ditambahkan bersamaan dengan dukungan navigasi satelit.
2. Pengenalan *Search Box*.
3. Donut menyertakan fitur gallery untuk memperlancara penangkapan media.
4. Toggling cepat antara kamera.

4. Android 2.0-2.1 (*Eclair*)

Pada bulan Oktober 2009, sekitar setahun setelah peluncuran Android 1.0, Google merilis versi 2.0 dari OS, dengan nama Android *Eclair*. Versi ini adalah pertama yang menambahkan dukungan Text-to-Speech.

Berikut adalah fitur yang menyertainya:

1. Dukungan multi-touch ditambahkan ke Android.
2. Dukungan untuk mencari di dalam pesan teks.
3. Eclair membawa kotak masuk terpadu ke Android. Dukungan untuk beberapa akun Google telah ditambahkan.

5. Android 2.2.3 (*Froyo*)

Android versi terbaru kembali dirilis pada 20 Mei 2010. Google menamainya dengan Froyo. Nama *Froyo* ini diambil dari singkatan frozen

yogurt. *Smartphone* pertama yang membawa merek Google Nexus, Nexus One, dirilis dengan Android 2.1 dari kotak pada awal tahun 2010, namun dengan cepat menerima update over-the-air ke Froyo akhir tahun . Pada Android versi ini mulai dilengkapi dengan fitur *friendly user* seperti opsi untuk mematikan akses data pada jaringan seluler.

Berikut adalah fitur yang menyertainya:

1. Hotspot WiFi portabel untuk berbagi koneksi 3G perangkat dengan gadget lainnya.
2. flash telepon juga bisa digunakan dalam video.
3. Setelan bergabung dengan kontak dan email untuk memback up ke server Google yang memungkinkan pembaca mengembalikan segalanya secara otomatis ke perangkat baru.

6. Android 2.3 - 2.3.7 (*Gingerbread*)

Gingerbread dirilis pada tahun 2010 . Pada 13 September 2010, Google menunjukkan bahwa hanya 0,6 persen dari semua perangkat Android yang saat ini menjalankan beberapa versi *Gingerbread*. Fitur utama termasuk dukungan NFC, SIP untuk panggilan Internet.

Berikut adalah fitur yang menyertainya:

1. UI overhaul untuk menghindari screen burn-in dan meningkatkan daya tahan baterai.
2. Dukungan kamera menghadap depan untuk panggilan video.
3. Download manager untuk mengawasi download Pembaca.
4. Peningkatan pada keyboard layar dengan cara pintas dan kursor untuk membantu copy paste.

7. Android 3.0 - 3.2.6 (*Honeycomb*)

Honeycomb diluncurkan pada bulan Mei 2011 pada dasarnya untuk memperluas Android untuk mendukung layar tablet. Versi Android ini

paling diabaikan dari semua. Karena dirilis khusus untuk tablet dan tidak pernah sampai ke ponsel.

Berikut adalah fitur yang menyertainya:

1. Beberapa perbaikan UI memanfaatkan layar besar.
2. Tombol perangkat keras diatuhkan untuk mendukung tombol di layar.
3. Browser web mengenalkan tabbed browsing.
4. Widget yang lebih besar.
5. Aplikasi seperti Gmail dan YouTube dirancang ulang untuk menggunakan layar besar.

8. Android 4.0 - 4.0.4 (*Ice Cream Sandwich*)

Android *Ice Cream Sandwich* dirilis pada bulan Oktober 2011, versi Android *Ice Cream Sandwich* menghadirkan sejumlah fitur baru bagi pengguna. Ini menggabungkan banyak fitur versi *Honeycomb* tablet saja dengan *Smartphone* yang berorientasi pada *Gingerbread*.

Berikut adalah fitur yang menyertainya:

1. Desain ulang terbesar ke Android dengan tema Holo.
2. Pengalaman browsing lebih cepat.
3. Ruang penyimpanan multi untuk aplikasi.
4. Pengenalan wajah untuk membuka kunci telepon.

9. Android 4.1 - 4.3.1 (*Jelly Bean*)

Jellybean dimulai pada bulan Juni 2012 dengan merilis Android 4.1. Google dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di bawah label *Jelly Bean*, masing-masing pada bulan Oktober 2012 dan Juli 2013.

Berikut adalah fitur yang menyertainya:

1. *Google Now*, tool bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian.
2. Project Butter untuk mendukung frame rate yang lebih tinggi saat menggesek menu dan layar rumah.
3. Mampu melihat foto dengan menggesek dari kamera untuk menuju ke filmstrip.
4. Widget menyetel ulang diri mereka untuk menambahkan yang baru.
5. Pemberitahuan fitur yang lebih banyak.
6. Fitur gerak dan aksesibilitas baru.

10. Android 4.4 (*KitKat*)

Android *KitKat* dirilis pada November 2013, Android 4.4 adalah satu-satunya versi OS yang benar-benar menggunakan nama sepotong permen dan menjadi salah satu versi Android yang paling disukai oleh pengguna *Smartphone* di dunia. *KitKat* memiliki fitur yang istimewa dari OS Android sebelumnya.

Berikut adalah fitur yang menyertainya:

1. *Immersive mode* untuk konsumsi konten yang lebih baik.
2. Bar navigasi yang lebih baik untuk masuk dan keluar dari mode *Immersive*.
3. Dukungan widget layar kunci.
4. Dialer baru dengan fitur Caller ID.
5. Wallpaper layar penuh.
6. Emoji keyboard untuk emoticon.
7. Aplikasi *Hangouts* dan perpesanan terpadu.

8. Dukungan cloud print yang lebih baik.
9. Integrasi Google Now yang lebih cerdas dan *handsfree*.

11. Android 5.0 (*Lollipop*)

Android 5.0 *Lollipop* pertama kali diperkenalkan pada Mei 2014., Android *lollipop* merupakan perancangan ulang terbesar untuk Android. *Smartphone* Google Nexus 6, bersama dengan tablet Nexus 9-nya, merupakan perangkat pertama yang memiliki *Lollipop* yang telah terpasang sebelumnya.

Peningkatan terbesar yang dilakukan oleh *Lollipop* adalah pengenalan *Material Design* yang dengan cepat menjadi bahasa desain terpadu yang diterapkan di seluruh produk Google.

Berikut adalah fitur yang dimilikinya:

1. Dukungan pengaturan cepat yang lebih baik.
2. Masa pakai baterai yang disempurnakan dengan mode *Battery*.
3. *Saver* yang baru.
4. Layar kunci baru.
5. Fitur *Smart Lock* melalui Layanan Google Play.
6. Mode tamu untuk berbagi perangkat.
7. Pemasangan tombol.

12. Android 6.0 (*Marshmallow*)

Android 6.0 (*Marshmallow*) di rilis pada tahun 2015. Ini Perangkat pertama yang dikirim bersama *Marshmallow* yang telah terpasang sebelumnya adalah *Smartphone* Google Nexus 6P dan Nexus 5X, dengan tablet Pixel C-nya. Tujuan *marshmallow* memoles sudut kasar dan membuat versi *Lollipop* lebih baik lagi.

Berikut adalah fitur yang dimilikinya:

1. Dukungan sidik jari resmi untuk perangkat.
2. Dukungan untuk pembayaran seluler melalui Android Pay.
3. Model perizinan yang lebih baik untuk aplikasi.
4. Google Now di Tap.
5. Deep menghubungkan Apps.

13. Android 7.0 (*Nougat*)

Android 7.0 (*Nougat*) Dirilis pada Tahun ,2016. Sebelum *Nougat* terungkap "Android N" dirujuk secara internal oleh Google sebagai "*New York Cheesecake*".

Berikut adalah fitur yang dimilikinya:

1. *Doze on the Go* untuk waktu siaga yang lebih baik lagi.
2. Multi Window untuk penggunaan dua aplikasi secara bersamaan.
3. Aplikasi Setelan yang Lebih Baik.
4. Hapus semua di layar aplikasi baru-baru ini.
5. Balas langsung ke pemberitahuan.
6. Notifikasi dibundel.
7. Pengaturan Cepat akan mengubah kustomisasi.

14. Android 8.0 (*Oreo*)

Pada bulan Maret 2017, Google Rilis Android 8.0 *Oreo*,bulan Agustus, Google mengkonfirmasi *Oreo* akan menjadi nama publik untuk Android 8.0. Seperti yang diketahui Ini adalah kedua kalinya Google memilih nama merek dagan untuk Android (*Oreo* dimiliki oleh *Nabisco*).

Adapaun sekarang versi ini adalah semua yang baru yang dimilikinya, diantaranya adalah sebagai berikut:

1. Pemberitahuan untuk prioritas dan kategorisasi yang lebih baik.
2. Pengelolaan warna lebih baik.
3. Android O memiliki koleksi emoji baru yang telah didesain ulang.
4. Waktu boot lebih cepat. Pada perangkat Pixel, sekarang bisa mengalami waktu boot dua kali lebih cepat dibandingkan dengan Nougat.
5. Mengisi otomatis dan mengingat kata sandi dalam aplikasi.

2.10 Pengembangan Sistem *Prototype*

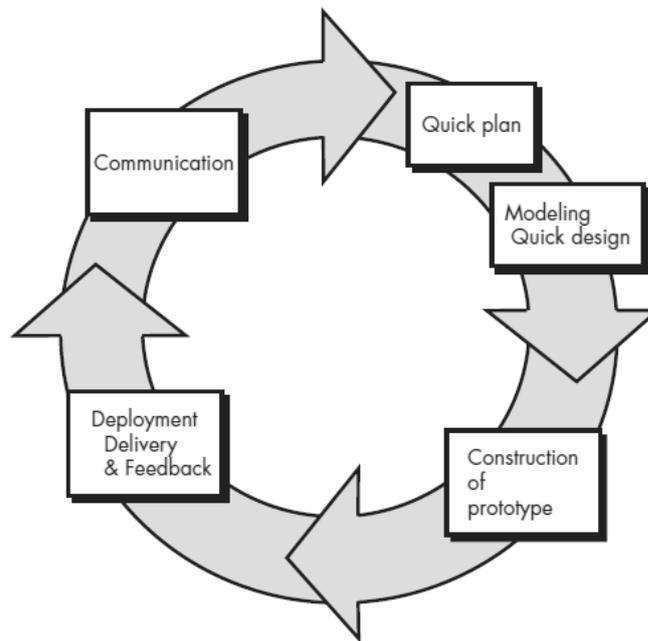
Menurut Pressman (2012:50), dalam melakukan perancangan sstem yang akan dikembangkan dapat menggunakan metde *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum di produksi secara benar.

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik.

Berikut adalah tahapan dalam metode *prootype*:

1. Komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna.
2. *Quick design* (desain cepat), yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.

3. Pembentukan *prototype*, yaitu memproduksi perangkat *prototype* termasuk pengujian dan penyempurnaan.
4. Evaluasi terhadap *prototype*, yaitu mengevaluasi *prototype* dan memperhalus analisis terhadap kebutuhan pengguna.
5. Perbaiki *prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi *prototype*.
6. Produksi akhir, yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Metode *Prototype* (Pressman, 2012:51)

2.11 UML (*Unified Modelling Language*)

Sukanto dan Shalahuddin (2013:133), “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an. Pada 1996, *Object Management Group* (OMG)

mengajukan proposal agar adanya standardisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language* (UML) diakomodasi oleh *Object Management Group* (OMG) sehingga sampai saat ini *Unified Modeling Language* (UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

2.12 Use Case Diagram

Sukanto dan Shalahuddin (2013:155), “*Use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada *diagram use case* menurut Rosa dan Shalahuddin (2013:156):

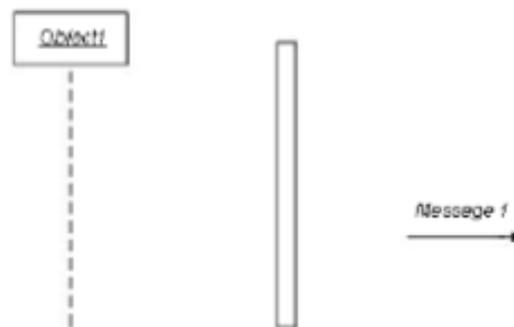
Tabel 2.1 Simbol-Simbol Diagram Use Case.

No.	Simbol	Keterangan
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .

2.	<p>Aktor / Actor</p>  <p>Nama Aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama <i>actor</i>.</p>
3.	<p>Asosiasi / association</p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>
4.	<p>Ekstensi / extend</p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi extend-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	<p>Generalisasi/generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6.	<p>Menggunakan / uses</p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>.</p>

2.13 Sequence Diagram

Menurut Booch (2005), suatu *sequence diagram* adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. *Sequence diagram* menangkap objek dan class yang terlibat dalam skenario dan urutan pesan yang ditukar antara objek diperlukan untuk melaksanakan fungsionalitas skenario. *Sequence diagram* berasosiasi dengan use case selama proses pengembangan. Dalam *Unified Model Language* (UML), objek dalam *sequence diagram* digambar dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara (nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous object*)). Berikut notasi *sequence diagram* seperti terlihat pada gambar dibawah ini:

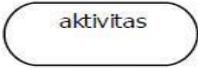
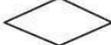
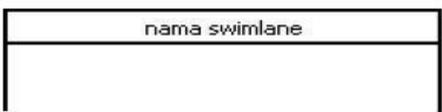
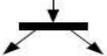


Gambar 2.2 Notasi *Sequence Diagram*

2.14 Activity Diagram

Activity Diagram adalah representasi grafis dari *workflow* dari kegiatan dan tindakan bertahap dengan dukungan untuk pilihan, iterasi dan *concurrency*. Dalam *Unified Modeling Language*, diagram aktivitas dimaksudkan untuk model kedua proses komputasi dan organisasi (yaitu *workflow*). *Activity diagram* menunjukkan aliran keseluruhan kontrol. Komponen-komponen *Activity diagram* ditunjukkan pada tabel dibawah ini:

Tabel 2.2 Komponen *Activity Diagram*

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / <i>decision</i> 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / <i>join</i> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
<i>fork</i> , 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
<i>join</i> , 	digunakan utk menunjukkan kegiatan yg digabungkan

2.15 Class Diagram

Widodo dan Herlawati (2011:3) menjelaskan, “*Class diagram* adalah penggambaran satu set objek yang memiliki atribut dan behavior yang sama”.

Sukamto dan Shalahuddin (2013:141), “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

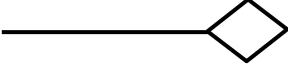
4. Kelas yang diambil dari pendefinisian data (model)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Rosa dan Shalahuddin (2013:146):

Tabel 2.3 Simbol-Simbol Diagram Kelas

No	Simbol	Keterangan
1.	<p style="text-align: center;">Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p style="text-align: center;">nama_kelas</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p style="text-align: center;">+atribut</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p style="text-align: center;">+operasi()</p> </div>	Kelas pada struktur sistem
2.	<p style="text-align: center;">Antarmuka / <i>Interface</i></p> <div style="text-align: center;">  </div> <p style="text-align: center;">nama_ <i>interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p style="text-align: center;">Asosiasi / Association</p> <div style="text-align: center;">  </div>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	<p style="text-align: center;">Asosiasi berarah / <i>Directed association</i></p> <div style="text-align: center;">  </div>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

5.	<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi spesialisasi (umum khusus)
6.	<p>Kebergantungan / <i>Dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.	<p>Agrgasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber: Buku Karangan Rosa A.S dan M. Shalahuddin (2014:146).

2.16 Pengujian Kotak Hitam (*Black-Box Testing*)

Menurut Roger S. Pressman (2012: 79) *Black Box Testing* atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black-Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program.

Black-Box Testing bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*.

Black Box Testing untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi tidak benar atau hilang.
2. Kesalahan interface atau antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja atau perilaku.
5. Kesalahan inisialisasi dan terminasi.

2.17 *Firestore Database*

Firestore adalah *Cloud Service Provider* dan *Backend as a Service* yang dimiliki oleh Google. *Firestore* merupakan solusi yang ditawarkan oleh Google untuk mempermudah dalam pengembangan aplikasi mobile maupun web. Selain itu *Firestore database* dapat menyimpan dan sinkronkan data dengan *database cloud NoSQL*. Data disinkronkan pada semua klien secara realtime dan tetap tersedia meski aplikasi *offline*.

2.17.1 Kemampuan Utama

1. Realtime

Sebagai ganti permintaan HTTP biasa, *Firestore Realtime Database* menggunakan sinkronisasi data setiap kali data berubah, semua perangkat yang terhubung akan menerima *update* dalam waktu milidetik. Memberikan pengalaman yang kolaboratif dan imersif tanpa perlu memikirkan kode jaringan.

2. Offline

Aplikasi *Firestore* tetap responsif bahkan saat offline karena *SDK Firestore Realtime Database* menyimpan data ke disk. Setelah konektivitas pulih, perangkat klien akan menerima setiap perubahan yang terlewat dan melakukan sinkronisasi dengan status server saat ini.

3. Dapat Diakses dari Perangkat Klien

Firestore Realtime Database dapat diakses secara langsung dari perangkat seluler atau browser web, server aplikasi tidak diperlukan. Keamanan dan validasi data dapat diakses melalui Aturan Keamanan *Firestore Realtime Database* yang merupakan kumpulan aturan berbasis ekspresi dan dijalankan ketika data dibaca atau ditulis.

4. Menskalakan di beberapa database

Dengan *Firestore Realtime Database* pada paket harga *Blaze*, dapat mendukung kebutuhan data aplikasi pada skala tertentu dengan membagi data di beberapa *instance database* di *project Firestore* yang sama. Menyederhanakan *autentikasi* dengan *Firestore Authentication* pada *project* dan mengautentikasi pengguna di *instance database*. Mengontrol akses ke data di tiap *database* dengan Aturan *Firestore Realtime Database* khusus untuk tiap *instance database*.

2.17.2 Cara Kerja Sistem Firestore Database

Firestore Realtime Database memungkinkan untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke *database*, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat *offline* sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, *Realtime Database* akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien *offline*, sehingga setiap perbedaan akan otomatis digabungkan.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang *fleksibel*, atau disebut juga Aturan Keamanan *Firestore Realtime Database*, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan *Firestore Authentication*, *developer* dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya. *Realtime Database* adalah *database NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan *database* terkait. *API Realtime Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan adalah dengan menggunakan metode model *Prototype*. Karena dengan melakukan pengembangan perangkat lunak menggunakan metode *Prototype* sangat cocok dalam perancangan aplikasi *booking order* rental Mobil, selain itu dalam penggunaan model *Prototype* digunakan dengan tujuan untuk menghemat waktu dalam pengembangan sistem. Selain itu juga dapat mengatasi ketidakserasian antara pengguna dan pengembangan sistem. Adapun tahapan dalam metode model *Prototype* yang digunakan antara lain:

3.1.1 Communication

Dalam proses penelitian perancangan aplikasi *booking order* rental Mobil dengan tujuan untuk mengumpulkan kebutuhan pengguna, maka perlu dilakukannya pengumpulan data. Pengumpulan data bertujuan untuk mendapatkan informasi secara benar dengan langsung ke pelanggan atau narasumber sesuai dengan referensi data yang diperlukan. Berikut ini tahapan rincian dalam proses pengumpulan data antara lain:

1) Studi Pustaka

Dalam metode studi pustaka dilakukan dengan cara mendapatkan informasi dan mengumpulkan data dengan melihat sumber kepustakaan seperti jurnal ilmiah, buku-buku dan sebagainya yang tentunya dapat di pertanggung jawabkan yang nantinya akan jadi bahan referensi dalam pembuatan aplikasi. Pelaksanaan dimulai dari tahap pengumpulan data, perancangan sistem (desain), pembuatan program, uji program (*testing*), hingga penyusunan laporan penelitian.

2) Observasi

Dalam metode observasi ini penulis diberikan kesempatan untuk melakukan pengumpulan data dengan cara pengamatan secara langsung ke lokasi rental yaitu pada CV.Sabai Rent Car, Jalan Pulau Belitung No.5 Kec.Sukabumi, Kota. Bandar Lampung. Provinsi Lampung.

3) Wawancara

Dalam metode wawancara ini penulis langsung menemui narasumber informasi dan mengajukan beberapa pertanyaan terkait tentang bagaimana dengan proses mekanisme dan prosedur dalam pemesanan kendaraan Mobil dan harga untuk setiap kendaraan Mobil.

Dari hasil metode wawancara yang dilakukan pada proses pengumpulan data penelitian pada CV.Sabai Rent Car, penulis mendapatkan informasi-informasi detail mengenai jumlah armada yang dimiliki oleh pemilik atau *owner* dari rental Mobil yang dicantumkan kedalam Tabel 3.1 Informasi Detail Kendaraan Mobil CV.Sabai Rent Car, diantaranya:

Tabel 3.1. Informasi Detail Kendaraan Mobil CV.Sabai Rent Car.

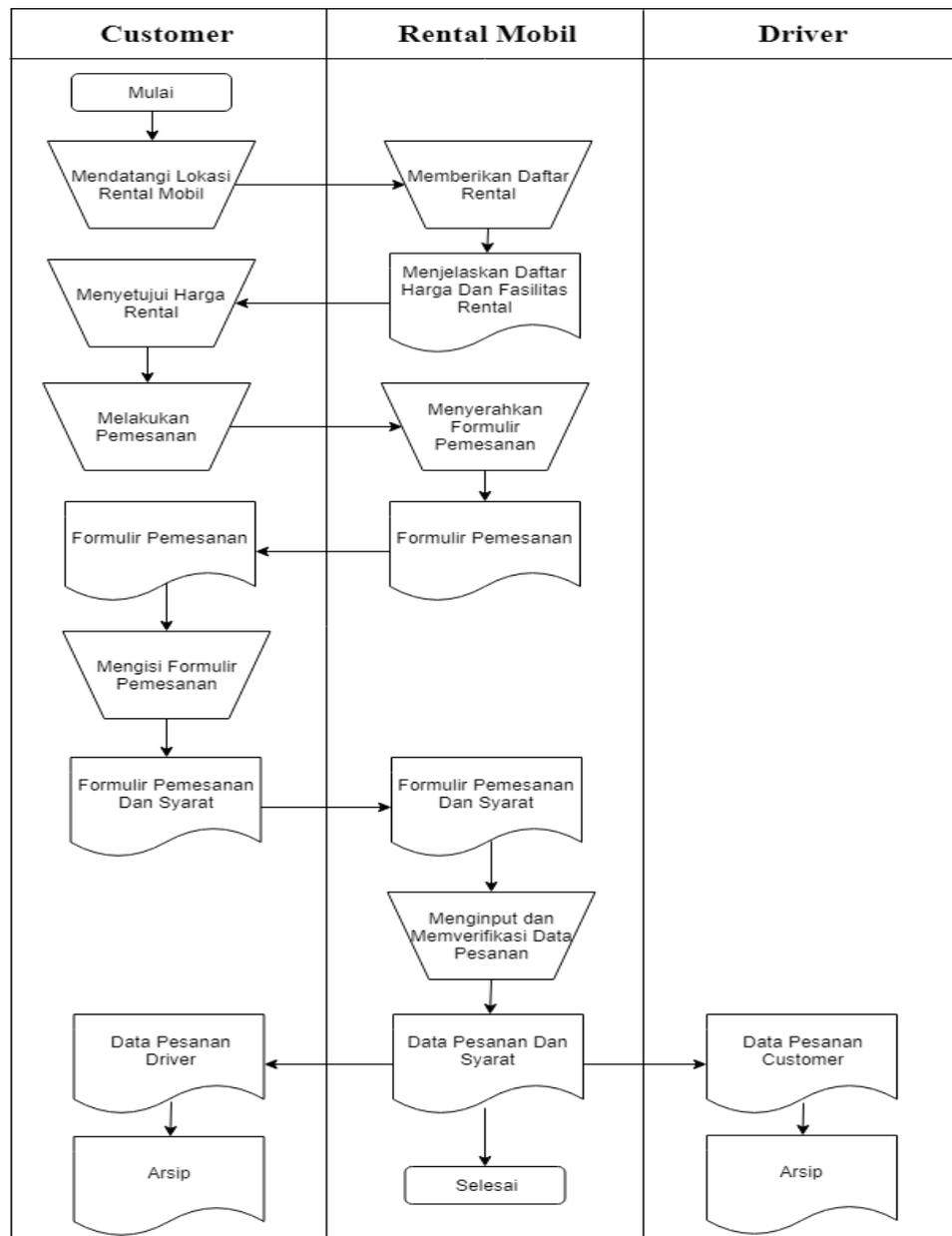
No.	Jenis Kendaraan	Jumlah
1.	Toyota Avanza	6 Unit
2.	Toyota Kijang Innova	3 Unit
3.	Honda Mobilio	2 Unit
4.	Toyota Rush	1 Unit
5.	Honda BR-V	1 Unit
6.	Toyota Calya	1 Unit

3.1.2 Quick Plan

Quick Plan (rencana cepat) merupakan tahapan dengan melakukan analisa dan perancangan setelah mendapatkan data-data dari tahapan mendengarkan pelanggan.

1) Analisis sistem yang berjalan

Proses pemesanan kendaraan Mobil rental yang digunakan pada CV.Sabai Rent Car sebelum melakukannya perancangan aplikasi adalah:



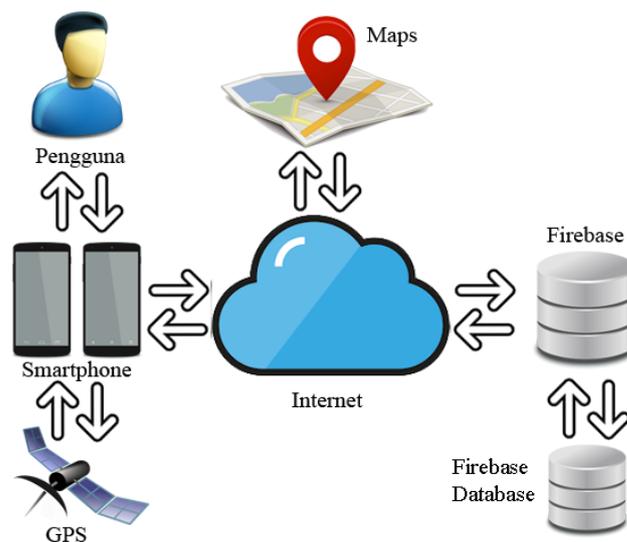
Gambar 3.1. Analisa Sistem Yang Sedang Berjalan.

Berdasarkan analisa terhadap sistem yang sedang berjalan, masih terdapat beberapa kelemahan diantaranya:

- a. Untuk melakukan pemesanan kendaraan *customer* harus datang langsung ke lokasi rental.
- b. Terbatasnya waktu dan jarak untuk melakukan pemesanan kendaraan rental berdampak pada jumlah pemesanan.

2) Analisis sistem yang diajukan

Menganalisa dari sebuah permasalahan yang dialami, maka kebutuhan dalam perancangan pembuatan aplikasi sangat dibutuhkan, untuk dapat melakukan pemesanan rental Mobil secara *realtime*, sekaligus aplikasi yang dirancang ditambahkan dengan fitur *maps location* dalam menentukan lokasi penjemputan *customer* yang dapat digunakan pada perangkat *Smartphone* berbasis Android. Perancangan aplikasi ini harus dilakukan secara benar, hal-hal yang harus diperhatikan dalam perancangan aplikasi ini bagaimana aplikasi yang dirancang dapat berjalan dengan baik pada pengembangan perangkat lunak berbasis Android, bagaimana sistem yang dirancang dapat mengetahui ketersediaan kendaraan secara *realtime* dan bagaimana fitur *maps location* dapat bekerja pada aplikasi sesuai yang diinginkan.



Gambar 3.2. Sistem Aplikasi Yang Akan Dibuat.

3) Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi *booking order* rental Mobil berbasis Android adalah sebagai berikut:

- a. Perangkat lunak sistem operasi pada PC yang digunakan adalah *Microsoft Windows 10 64 bit*.
- b. Perangkat lunak sistem operasi pada Android minimal adalah Android versi 4.4 (*KitKat*).
- c. Perangkat lunak untuk pembuatan program aplikasi yang digunakan adalah *Android Studio 3.0*.
- d. Perangkat lunak yang digunakan untuk pembuatan gambar desain yang ada pada aplikasi, software yang digunakan adalah *CorelDrawX7* dan *Adobe Photoshop CC 2017*.
- e. Perangkat lunak yang digunakan untuk melakukan testing terhadap aplikasi yang dibuat dengan menggunakan *Emulator Simulator*.

4) Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yang digunakan untuk membangun sebuah sistem aplikasi *booking order* rental Mobil berbasis Android adalah sebagai berikut :

- a. Spesifikasi *Personal Computer* yang digunakan :
 - 1) Monitor Asus LED MX279H.
 - 2) Processor Intel Core i5-4210U.
 - 3) Ram 16 Gb.
 - 4) Harddisk 750 Gb.
 - 5) Keyboard dan Mouse.

b. Spesifikasi *Smartphone* yang digunakan pada Android :

- 1) Qualcomm Snapdragon 835 Octa-core 2,35 GHz Quad + 1,9 GHz.
- 2) Ram 4 Gb.
- 3) Storage 64 Gb.

Analisa tersebut bukanlah hal yang mutlak, namun merupakan pendapat peneliti terhadap spesifikasi penggunaan perangkat keras yang dipakai dalam pengembangan aplikasi *booking order* rental Mobil berbasis Android.

3.1.3 Modeling Quick Design

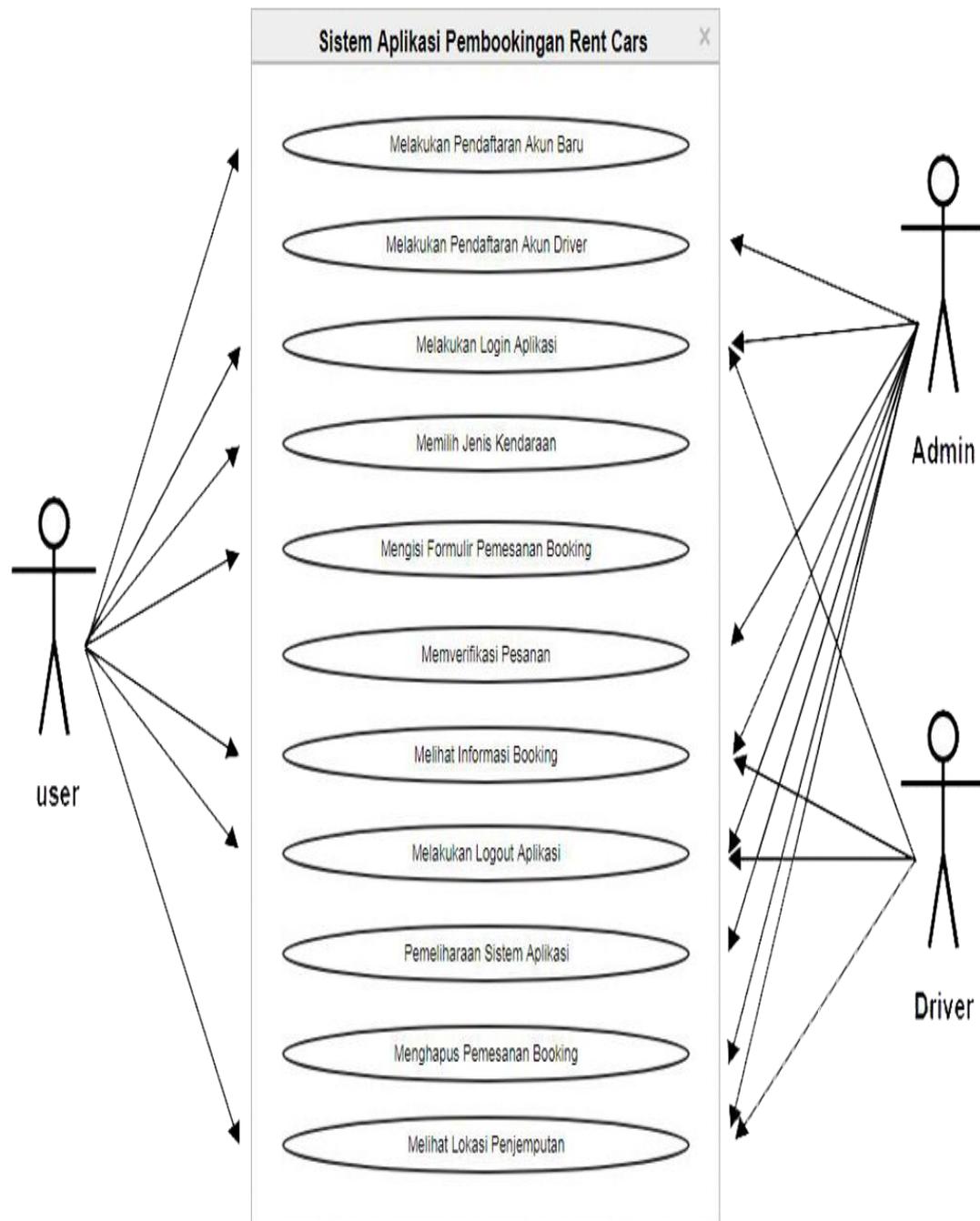
Analisis kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini, dan dilakukannya gambaran desain sistem yang akan dirancang. Desain sistem dapat membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3.1.3.1 Desain UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Pada Penelitian kali ini UML yang dipakai adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan Perancangan Antarmuka (*interface*).

3.1.3.2 Rancangan Use Case Diagram

Pemodelan yang akan digunakan dalam menggambarkan kebutuhan fungsional pada aplikasi yang dibangun yaitu dengan pemodelan *Use Case Diagram*. *Use case Diagram* dalam perancangan aplikasi ini yang dibagi menjadi 3 *actor* diantaranya bagian *admin*, *user* dan *driver* yang digambarkan pada gambar 3.3 adalah sebagai berikut:



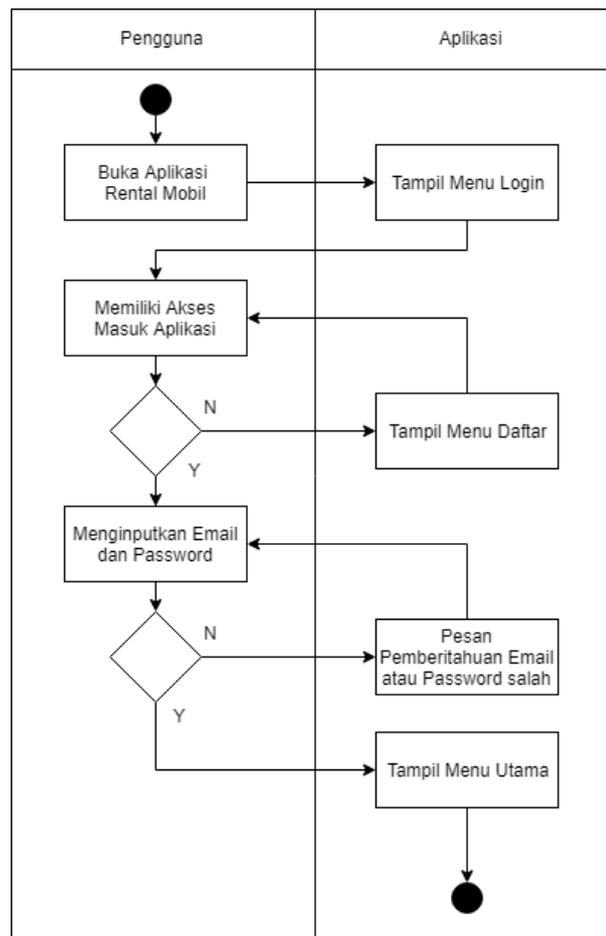
Gambar 3.3. Use Case Diagram Sistem Aplikasi Booking Kendaraan Mobil.

3.1.3.3 Rancangan *Activity Diagram*

Activity Diagram merupakan bagian dari penggambaran sistem secara fungsional yang menjelaskan proses-proses logika atau fungsi yang diimplementasikan oleh kode program. *Activity Diagram* memodelkan event-event yang terjadi didalam suatu *use case* dan digunakan untuk pemodelan aspek dinamis dari sistem. Berikut ini rancangan *Activity Diagram* pada aplikasi *booking order* rental Mobil berbasis Android:

a) *Activity Diagram* Login Aplikasi

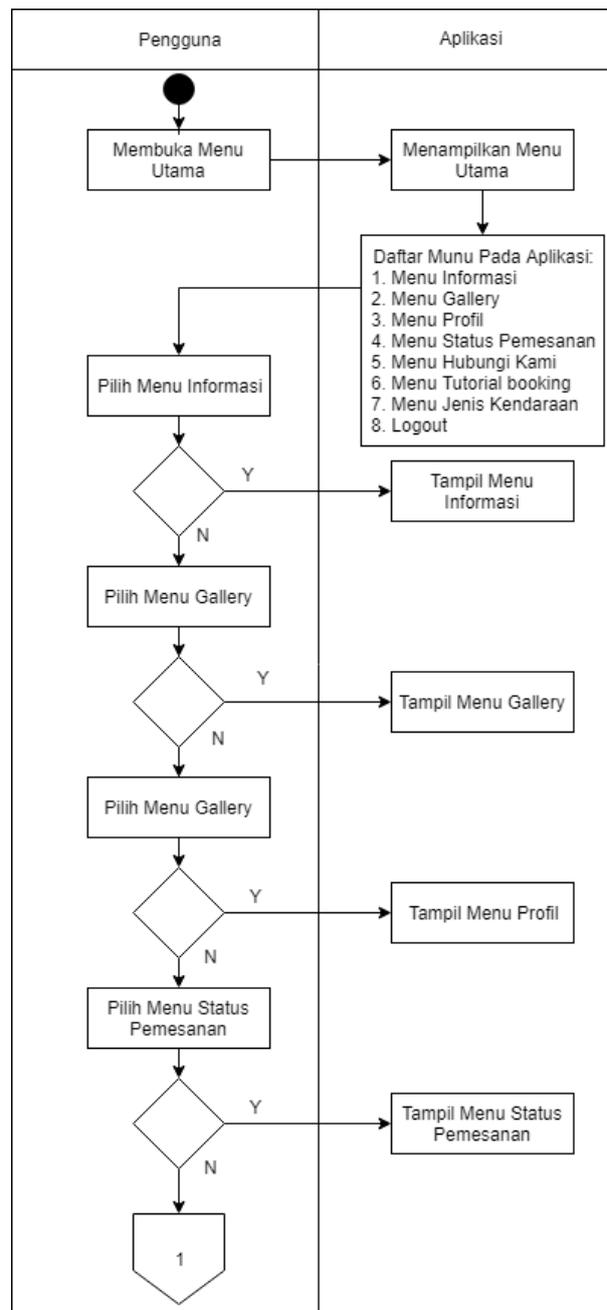
Activity Diagram login aplikasi menjelaskan urutan proses didalam *actor (user)* dalam melakukan login kedalam aplikasi. Tampilan *Activity Diagram* login aplikasi dapat dilihat pada gambar 3.4.



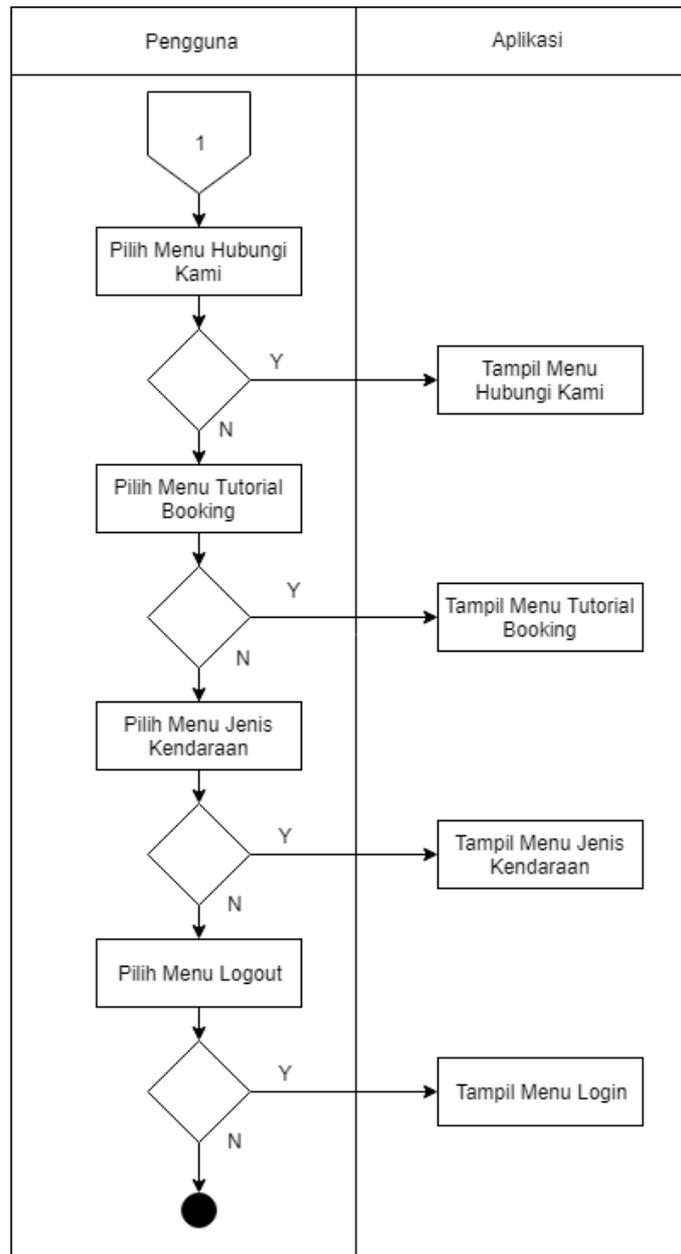
Gambar 3.4. *Activity Diagram* Login Aplikasi.

b) *Activity Diagram* Menu Utama

Activity Diagram menu utama menjelaskan urutan proses didalam *actor* (*user*) dalam melakukan pemilihan menu pada aplikasi. Tampilan *Activity Diagram* login aplikasi dapat dilihat pada gambar 3.5 dan gambar 3.6.



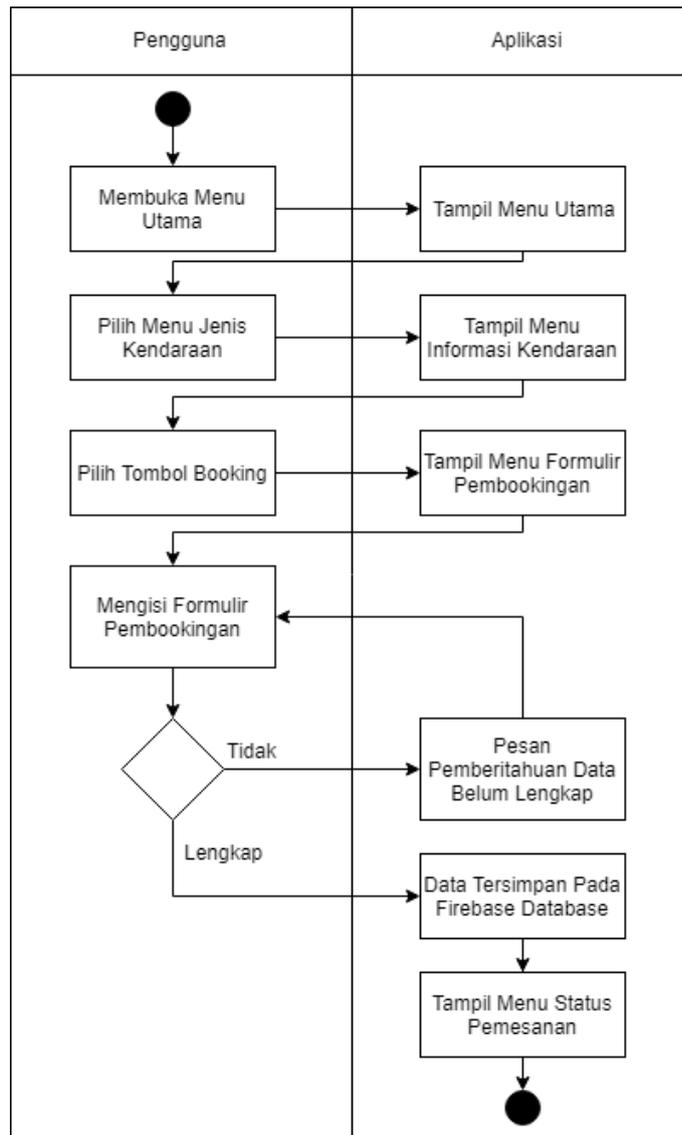
Gambar 3.5. *Activity Diagram* Menu Utama.



Gambar 3.6. *Activity Diagram* Lanjutan Menu Utama.

c) *Activity Diagram* Pemesanan Kendaraan

Activity Diagram pemesanan kendaraan menjelaskan urutan proses didalam *actor* (*user*) dalam melakukan proses pemesanan kendaraan pada aplikasi. Tampilan *Activity Diagram* pemesanan kendaraan dapat dilihat pada gambar 3.7.



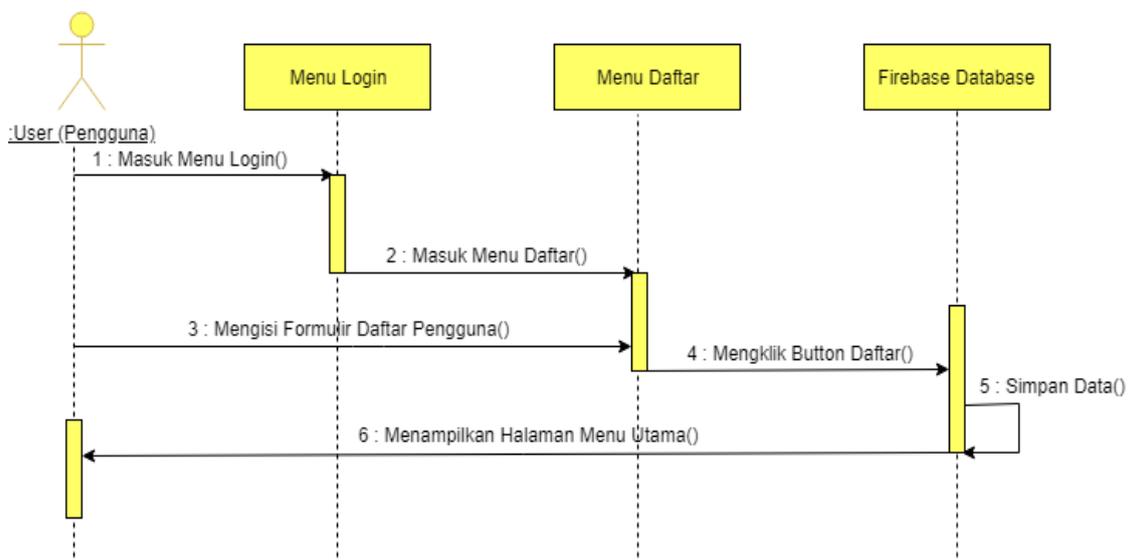
Gambar 3.7. *Activity Diagram* Pemesanan Kendaraan.

3.1.3.4 Rancangan *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dan menjelaskan bagaimana alur yang akan dijalankan tersebut pada aplikasi. Berikut rancangan *Sequence Diagram* pada aplikasi *booking order* rental Mobil berbasis Android:

a) *Sequence Diagram* Pendaftaran Pengguna

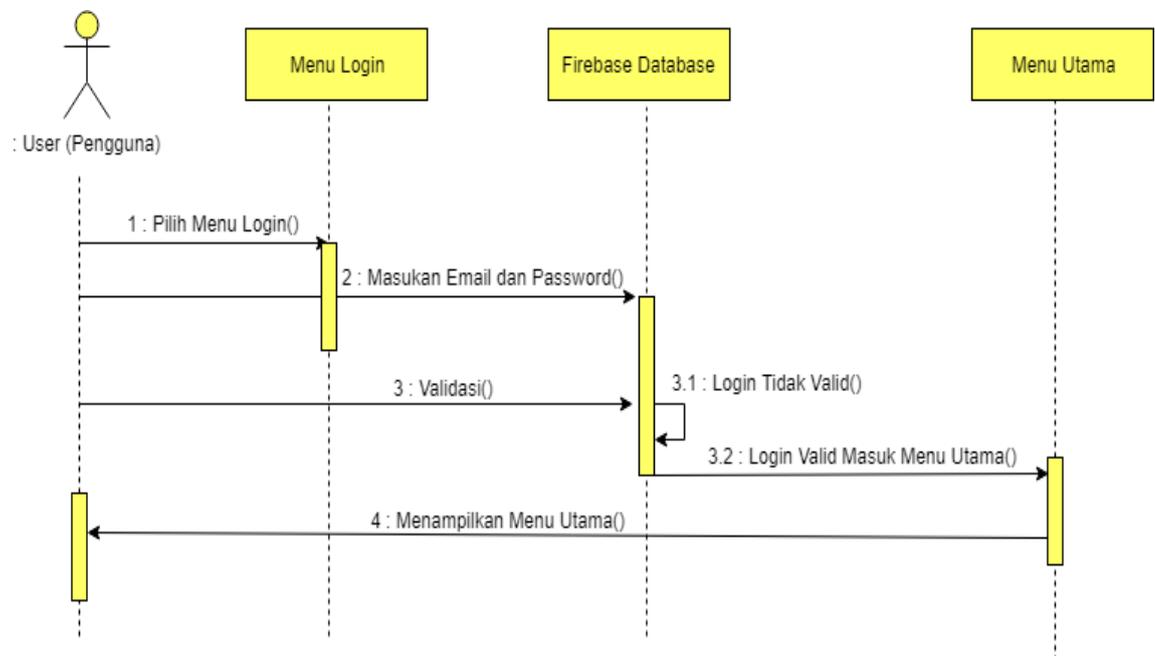
Pada *User* (Pengguna) akan masuk kehalaman menu login aplikasi dimana jika pengguna belum memiliki akses masuk kedalam menu utama aplikasi kemudian pengguna dapat memilih menu daftar dan mengisi formulir pendaftaran, jika data pendaftaran sudah terisi pengguna dapat mengklik tombol daftar kemudian data akan terkirim langsung otomatis data tersebut tersimpan ke *firebase database* dan selanjutnya akan masuk ke menu utama aplikasi. Tampilan *Sequence Diagram* pendaftaran pengguna dapat dilihat pada gambar 3.8.



Gambar 3.8. *Sequence Diagram* Pendaftaran Pengguna.

b) *Sequence Diagram Login Aplikasi*

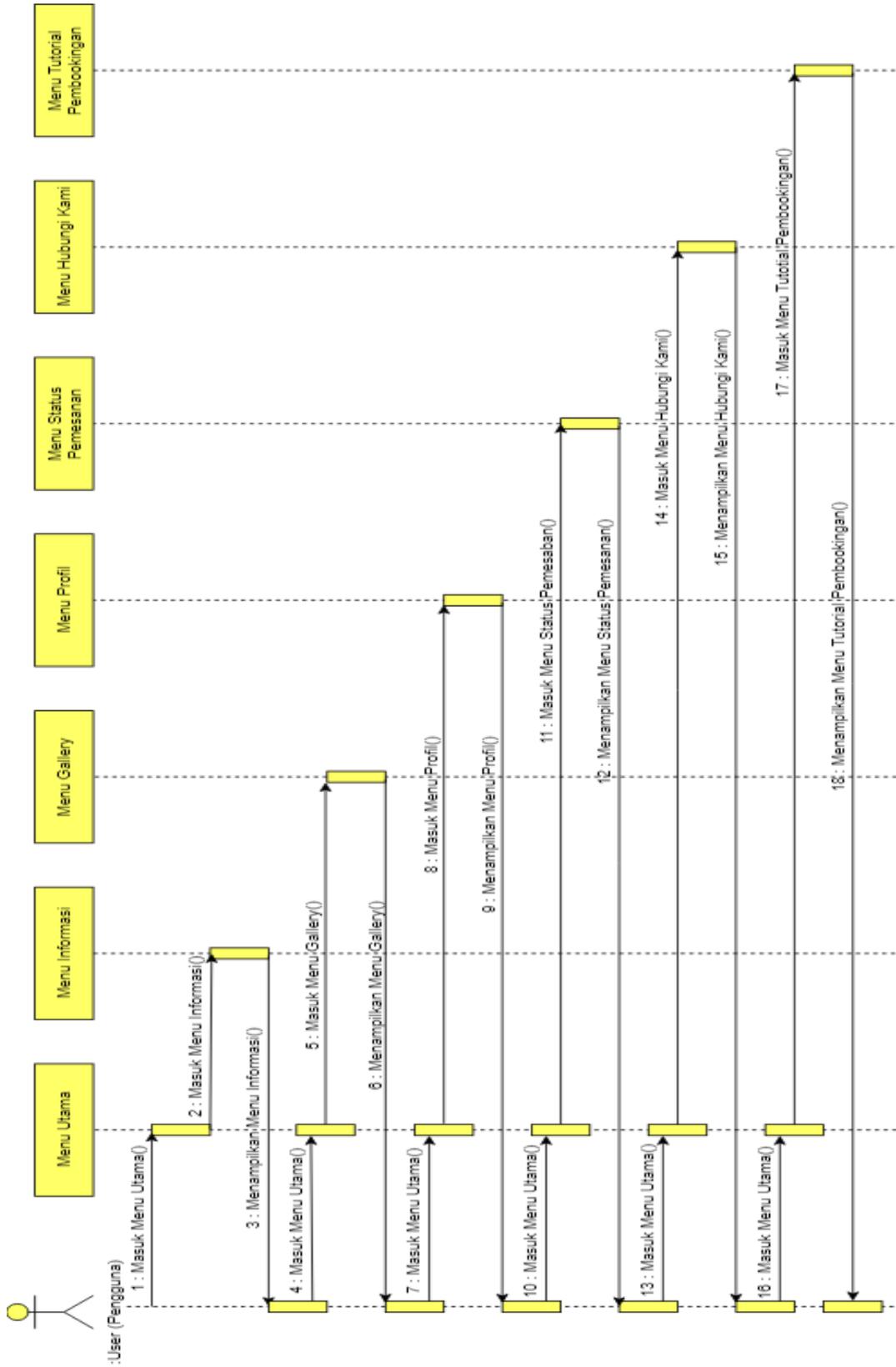
Pada *User* (Pengguna) akan masuk kehalaman menu login aplikasi dimana jika pengguna sudah memiliki akses masuk kedalam menu utama aplikasi, pengguna dapat menginputkan *email* dan *password* ke menu login kemudian data yang diinputkan akan dibaca oleh *firebase database* apakah *valid* atau tidak, jika tidak maka muncul peringatan pesan login gagal, jika data yang diinputkan *valid* maka otomatis akan langsung masuk ke menu utama aplikasi. Tampilan *Sequence Diagram* login aplikasi dapat dilihat pada gambar 3.9.



Gambar 3.9. *Sequence Diagram Login Aplikasi.*

c) *Sequence Diagram Menu Aplikasi*

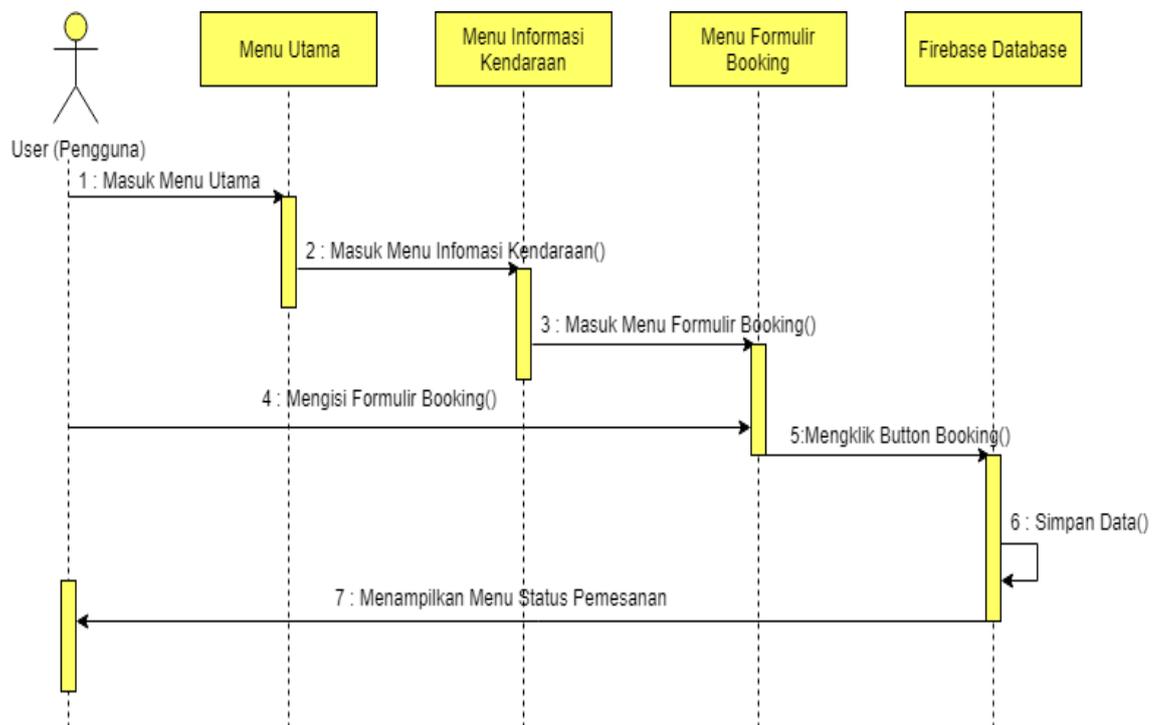
Pada *User* (Pengguna) akan masuk kehalaman menu utama aplikasi dimana pengguna pada halaman ini dapat memiliki pilihan menu yang ada pada aplikasi diantaranya menu informasi, *gallery*, profil, status pemesanan, hubungi kami, dan menu tutorial pemesanan. Tampilan *Sequence Diagram* menu aplikasi dapat dilihat pada gambar 3.10.



Gambar 3.10. *Sequence Diagram* Menu Aplikasi.

d) *Sequence Diagram* Pemesanan Kendaraan

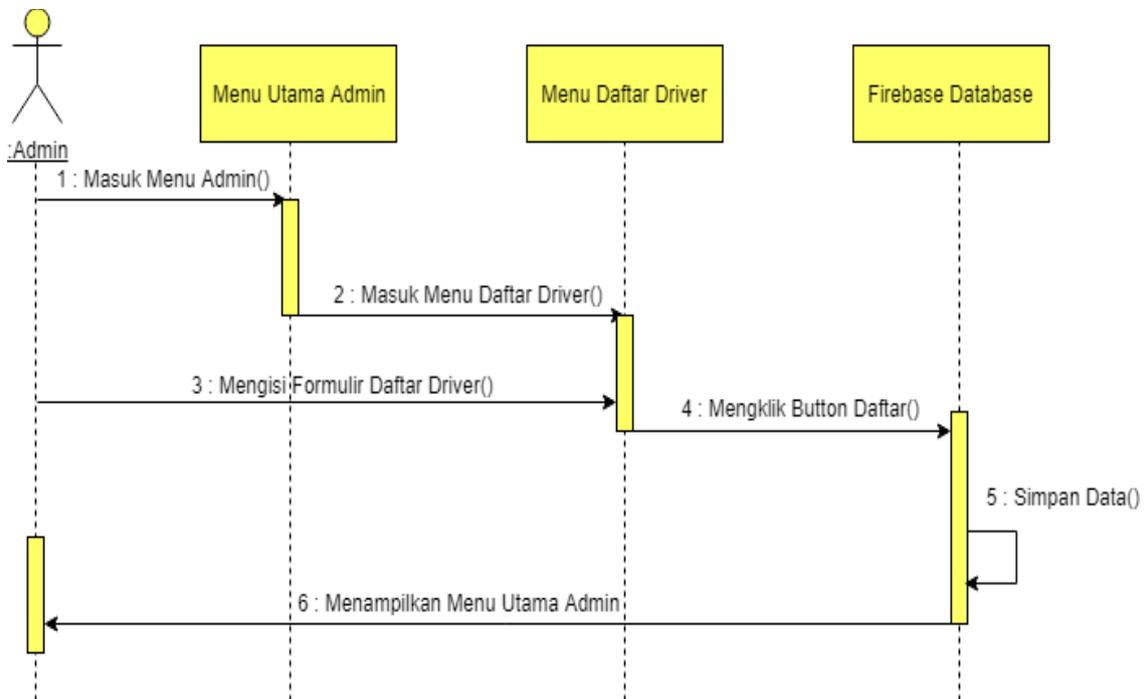
Pada *User* (Pengguna) akan masuk kehalaman menu utama aplikasi dimana pada tahapan pemesanan kendaraan pengguna harus memilih jenis kendaraan yang ingin dilakukan pemesanan, selanjutnya pengguna akan masuk ke menu informasi kendaraan, kemudian pengguna akan masuk ke menu formulir pemesanan, pada tahap ini pengguna dapat mengisi biodata pemesanan dengan benar pada formulir yang sudah sediakan, selanjutnya jika data yang diinputkan dianggap sudah benar maka pengguna dapat mengklik tombol *booking* kemudian data yang sudah diinputkan sebelumnya akan otomatis dikirimkan ke *firebase database* untuk selanjutnya data tersebut disimpan dan menampilkan hasil pemesanan pada halaman menu status pemesanan. Tampilan *Sequence Diagram* pemesanan kendaraan dapat dilihat pada gambar 3.11.



Gambar 3.11. *Sequence Diagram* Pemesanan Kendaraan.

e) *Sequence Diagram* Pendaftaran Driver

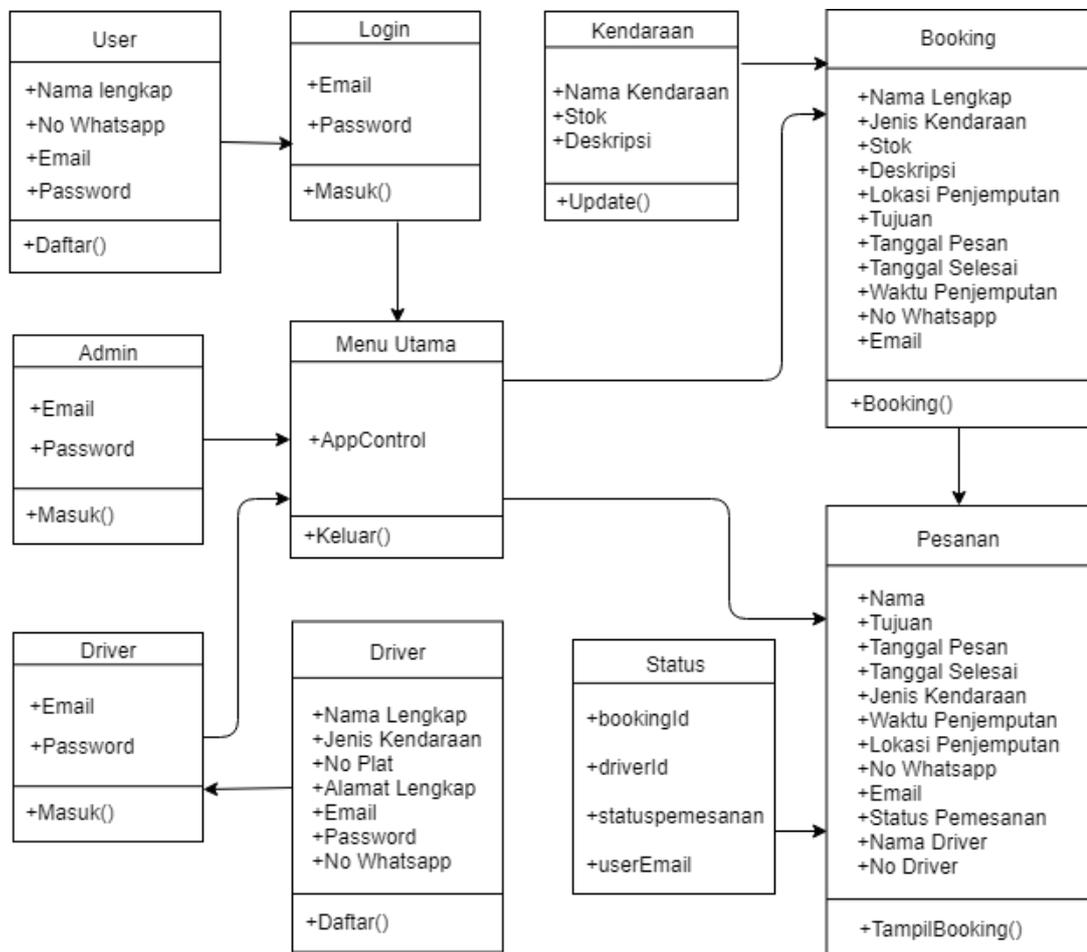
Pada *Admin* akan masuk kehalaman menu utama *admin* aplikasi dimana pada tahapan ini, *admin* dapat mendaftarkan *driver* baru pada aplikasi dengan cara masuk ke menu daftar *driver*, kemudian mengisi biodata *driver* pada formulir pendaftaran dengan benar dan selanjutnya *admin* dapat mengklik tombol daftar untuk selanjutnya data akan disimpan ke *firebase database*, dan jika proses pendaftaran *driver* baru berhasil, maka akan masuk kembali ke menu utama *admin* aplikasi. Tampilan *Sequence Diagram* pendaftaran driver dapat dilihat pada gambar 3.12.



Gambar 3.12. *Sequence Diagram* Pendaftaran Driver.

3.1.3.5 Rancangan Class Diagram

Class diagram merupakan diagram yang menggambarkan struktur dan komponen-komponen yang terkait pada sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Selain itu *class diagram* berfungsi menjelaskan tipe dari objek sistem dan hubungannya dengan objek lain. Objek merupakan nilai tertentu yang memiliki atribut dan metode atau operasi. *Class diagram* pada perancangan aplikasi *booking order* rental Mobil dapat dilihat pada Gambar 3.13.



Gambar 3.13. *Class Diagram* Aplikasi *Booking Order* Rental Mobil.

3.1.3.6 Rancangan Antarmuka (*User Interface*)

Rancangan desain *user interface* merupakan desain antarmuka pengguna yang diterapkan ke dalam aplikasi. Rancangan tampilan digunakan untuk mempermudah pengguna dalam menjalankan atau menggunakan aplikasi yang dibuat. Berikut ini akan dijelaskan konsep rancangan *interface* dari masing-masing layar yang akan ditampilkan dalam perancangan aplikasi ini.

a) Rancangan Tampilan Menu Login Aplikasi

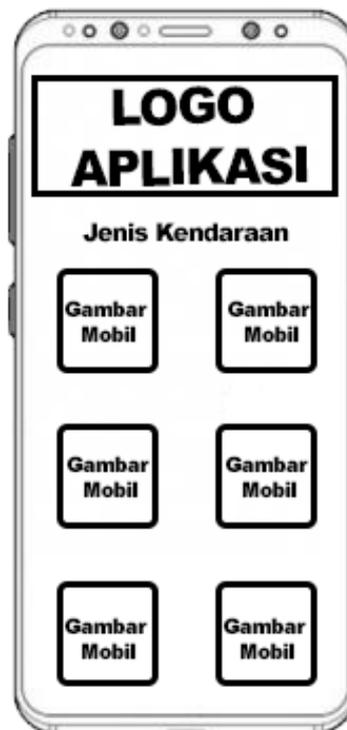
Pada rancangan tampilan pertama yaitu rancangan tampilan halaman login dimana terdapat dua buah *TextEdit* diantaranya email atau *username* dan *password* yang berfungsi untuk menginputkan email atau *username* dan *password* dan juga terdapat dua tombol yaitu tombol untuk masuk ke menu utama dan tombol untuk daftar pengguna baru, dan pada posisi atas interfece ditambahkan logo dari aplikasi rental. Tampilan rancangan halaman menu tampilan login dapat dilihat pada gambar 3.14.



Gambar 3.14. Tampilan Login Aplikasi.

b) Rancangan Tampilan Menu Utama User

Pada rancangan tampilan menu utama pengguna atau user ini, dirancang agar *interface* yang dibuat dapat mudah dipahami dan digunakan oleh pengguna. Pada rancangan layar *interface* ini terdapat logo aplikasi yang terletak diatas layout, dan terdapat enam buah *ButtonImage*, pada masing-masing *ButtonImage* berisikan gambar jenis kendaraan Mobil yang berfungsi sebagai pilihan jenis-jenis kendaraan pada aplikasi rental ini, sehingga pengguna dapat memilih jenis kendaraan yang diinginkan. Tampilan rancangan menu utama user dapat dilihat pada gambar 3.15.

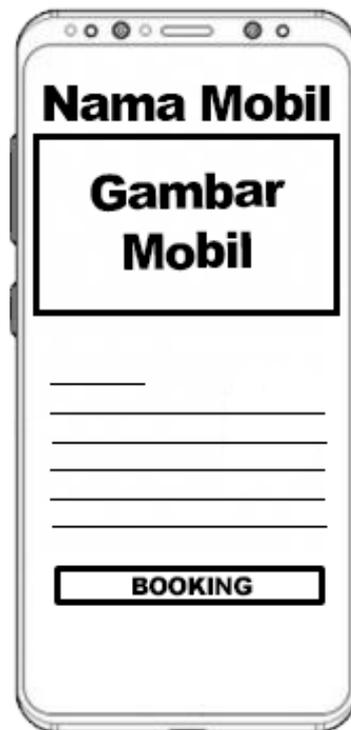


Gambar 3.15. Tampilan Menu Utama User.

c) Rancangan Tampilan Menu Informasi Kendaraan

Pada menu Tampilan ini, terdapat satu buah *ImageView* terletak pada atas *layout*, dan terdapat *TextView* yang berfungsi untuk menginformasikan nama jenis dari kendaraan yang dipilih dan berisikan informasi-informasi kendaraan baik dari informasi harga, durasi, maupun fasilitas-fasilitas yang diberikan dalam aplikasi

pemesanan rental Mobil ini, dan juga terdapat satu tombol *Booking* yang jika tombol tersebut diklik akan otomatis masuk ke menu formulir untuk melakukan pemesanan armada kendaraan rental. Tampilan rancangan menu informasi kendaraan dapat dilihat pada gambar 3.16.

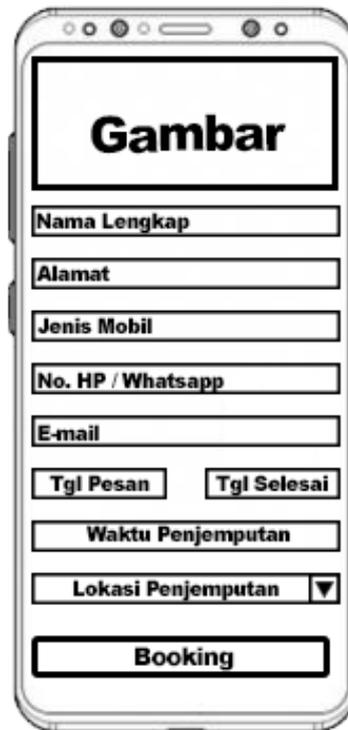


Gambar 3.16. Tampilan Menu Informasi Kendaraan.

d) Rancangan Tampilan Menu Formulir Pemesanan *Booking*

Rancangan tampilan selanjutnya adalah tampilan menu formulir pemesanan dimana jika sebelumnya pada tampilan informasi kendaraan di klik tombol *booking* maka akan masuk ke menu formulir pemesanan. Dalam rancangan tampilan ini terdapat satu buah *SlideShow* gambar pada bagian atas layout, selanjutnya pada rancangan layout ini berisikan *TextView* dan *Spinner* yang berfungsi untuk menginputkan informasi data pemesanan yang diantaranya terdiri dari nama lengkap pengguna, alamat, jenis kendaraan, nomor whatsapp, email, tanggal pemesanan, tanggal selesai, waktu penjemputan dan lokasi penjemputan.

Dan satu tombol *booking* untuk memverifikasi pesanan yang sudah di input sebelumnya agar dapat langsung masuk kedalam menu status informasi pesanan. Tampilan rancangan menu formulir pemesanan *booking* dapat dilihat pada gambar 3.17.



The image shows a mobile application interface for a booking form. At the top, the word "Gambar" is displayed in a large, bold font. Below this, there are several input fields: "Nama Lengkap", "Alamat", "Jenis Mobil", "No. HP / Whatsapp", and "E-mail". There are also two date pickers labeled "Tgl Pesan" and "Tgl Selesai", a field for "Waktu Penjemputan", and a dropdown menu for "Lokasi Penjemputan". At the bottom of the form is a large button labeled "Booking".

Gambar 3.17. Tampilan Menu Formulir Pemesanan *Booking*.

e) Rancangan Tampilan Menu Pendaftaran Pelanggan Baru

Rancangan tampilan pada menu pendaftaran terdiri dari satu buah *ImageView* atau *SlideShow*, dan terdapat empat buah *TextEdit*, diantaranya untuk menginputkan nama lengkap, nomor whatsapp, email dan *password*, dan memiliki satu buah tombol daftar untuk memverifikasi data yang diinputkan pada *TextEdit*. Tampilan rancangan menu pendaftaran pelanggan baru dapat dilihat pada gambar 3.18.



Gambar 3.18. Tampilan Menu Pendaftaran Pelanggan Baru.

f) Tampilan Menu Fitur *Maps Location* Penjemputan Pelanggan

Pada tahap tampilan rancangan ini adalah tampilan menu dari fitur aplikasi yang akan dibuat yaitu fitur *maps location* penjemputan pelanggan dimana pada rancangan tampilan ini terdapat *maps google* yang akan mengarahkan langsung dengan titik lokasi pengguna, selanjutnya terdapat satu buah *TextEdit* yang berfungsi untuk mencari lokasi-lokasi sebagai titik penjemputan pengguna aplikasi, dan terdapat satu tombol lokasi penjemputan berguna sebagai memverifikasi alamat pada *TextEdit* seach kemudian diproses langsung kembali ke manu formulir pemesanan. Tampilan rancangan menu fitur *maps location* penjemputan dapat dilihat pada gambar 3.19.



Gambar 3.19. Tampilan Menu Fitur *Maps* Penjemputan Pelanggan.

g) Rancangan Tampilan Menu Detail Informasi *Booking*

Rancangan tampilan selanjutnya yaitu perancangan tampilan menu detail informasi *booking* pada *driver*. Dimana pada perancangan tampilan informasi *booking* ini terdapat satu buah *ImageView* dan satu buah *ListView* yang berfungsi untuk melihat informasi detail pemesanan yang masuk pada aplikasi. Tampilan rancangan menu detail informasi *booking* dapat dilihat pada gambar 3.20.



Gambar 3.20. Tampilan Menu Detail Informasi *Booking*.

h) Rancangan Tampilan Menu Utama *Admin*

Tampilan rancangan yang terakhir adalah rancangan tampilan pada menu utama *admin*, pada rancangan tampilan ini terdiri dari satu buah *ImageView* ataupun *SlideView*, dua buah *Spinner* diantaranya jenis kendaraan dan bulan, satu buah tombol, dan satu buah *ListView* yang berfungsi untuk melihat informasi detail pesanan yang masuk kedalam aplikasi *booking* yang akan dibuat. Tampilan rancangan menu utama *admin* dapat dilihat pada gambar 3.21.



Gambar 3.21. Tampilan Menu Utama *Admin*.

3.1.4 *Construction Of Prototype*

Pada tahap ini aplikasi yang dibuat dengan menggunakan bahasa pemrograman tertentu dan dapat diuji bagaimana program berjalan.

1) Pembuatan aplikasi

Dalam penelitian ini aplikasi dibuat pada dasarnya dengan menggunakan *Android Studio* versi 3.0 dan juga perangkat lunak penunjang yang telah dijelaskan pada sub bab analisis kebutuhan perangkat lunak. Perangkat lunak tersebut digunakan karena cocok dalam pembuatan aplikasi ini. Sedangkan untuk bahasa pemrograman pada *Android Studio* mendukung bahasa pemrograman Java dan Kotlin.

2) Pengujian aplikasi

Pengujian aplikasi dilakukan dengan menggunakan metode *black-box testing*. Pengujian yang akan dilakukan yaitu dengan menguji fungsi tombol pada aplikasi dan uji fungsi program aplikasi dari masing-masing halaman yang terdapat pada aplikasi. Pengujian dilakukan dengan menggunakan tiga buah perangkat yang berbeda, dan yang memiliki spesifikasi yang berbeda, dengan kriteria spesifikasi rendah, sedang, dan tinggi.

Berikut adalah ketiga perangkat tersebut dan spesifikasinya:

1. Perangkat Kategori Rendah

- a. Nama Perangkat : Xiaomi Redmi Note 3
- b. Sistem Operasi : Android Versi 5.1 (Lollipop)
- c. Kecepatan Processor : 1,4 Ghz
- d. Kapasitas RAM : 3 Gb

2. Perangkat Kategori Sedang

- a. Nama Perangkat : Samsung Galaxy A5
- b. Versi Sistem Operasi : Android Versi 8.0 (Oreo)
- c. Kecepatan Processor : 1,6 Ghz
- d. Kapasitas RAM : 3 Gb

3. Perangkat Kategori Tinggi

- a. Nama Perangkat : Samsung Galaxy S8
- b. Versi Sistem Operasi : Android versi 8.0 (Oreo)
- c. Kecepatan Processor : 1,9 Ghz
- d. Kapasitas RAM : 4 Gb

3.1.5 *Deployment Delivery and Feedback*

Deployment Delivery and Feedback atau penerapan program merupakan tahapan ketika aplikasi telah selesai dibuat. Aplikasi yang telah dibuat akan distribusikan melalui *Google Play Store* sehingga semua orang bisa dapat mengunduh dan menggunakan aplikasinya

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

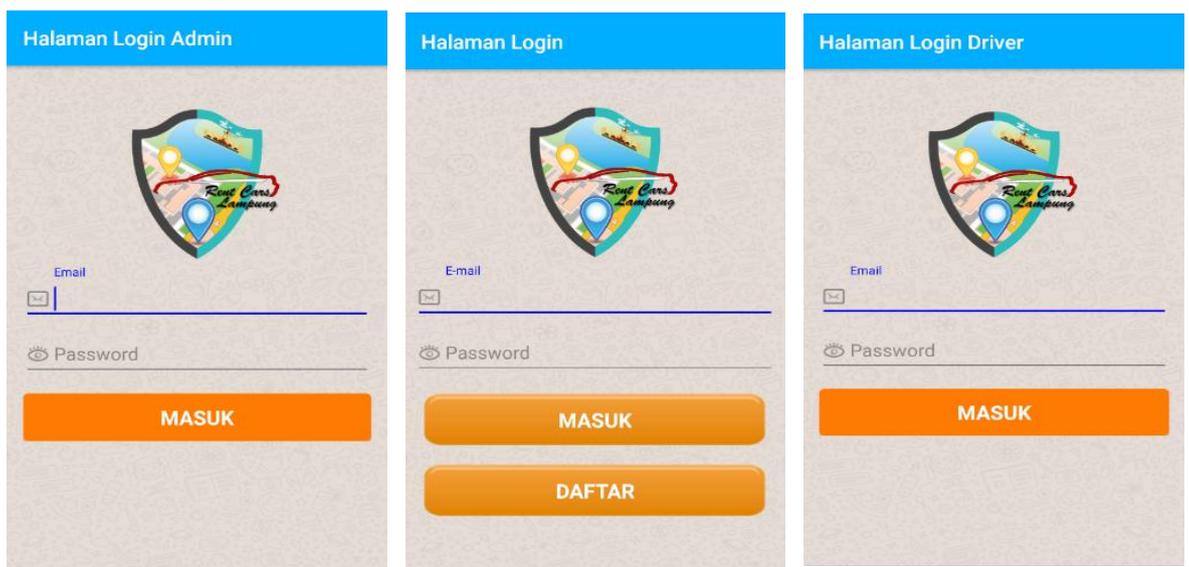
4.1 Hasil Penelitian

Hasil penelitian merupakan lanjutan dari tahap perancangan dan tahap proses pemrograman. Pada tahap ini, aplikasi yang telah dirancang akan di implementasikan ke dalam perangkat *Smartphone* Android. Hasil uji coba aplikasi dan tampilan desain *interface* aplikasi pemesanan rental Mobil adalah sebagai berikut:

4.2 Tampilan Aplikasi

4.2.1 Tampilan Aplikasi Menu Login

Tampilan ini merupakan tampilan awal yaitu layout halaman login *admin*, pengguna maupun *driver*, dalam memulai aplikasi pemesanan rental Mobil. Menu ini merupakan akses masuk kedalam halaman utama aplikasi, jika pengguna belum memiliki akses masuk kedalam aplikasi maka pengguna bisa mengklik tombol button daftar untuk mendaftarkan akun. Tampilan menu login dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Halaman Menu Login *Admin*, Pengguna dan *Driver*.

4.2.2 Tampilan Aplikasi Menu Daftar Pengguna Baru

Tampilan dibawah ini merupakan tampilan aplikasi menu daftar pengguna baru, pengguna baru yang belum memiliki akses email dan *password* untuk masuk kedalam menu utama aplikasi, pengguna dapat mendaftarkan akun ke halaman pendaftaran yang sudah tersedia pada aplikasi, dengan cara mengisi formulir pendaftaran pengguna baru dengan mengisi nama lengkap, nomor handphone atau whatsapp, email dan *password*. Sehingga jika pengguna sudah mengisi formulir pendaftaran dengan benar dan lengkap maka biodata yang diinputkan akan otomatis masuk ke sistem *firebase database* dan *authentication* untuk disimpan sebagai akses masuk kedalam menu utama aplikasi. Tampilan menu daftar dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Halaman Menu Daftar Pengguna Baru.

4.2.3 Tampilan Aplikasi Menu Utama Pengguna

Tampilan selanjutnya merupakan tampilan aplikasi menu utama pengguna dimana pada tampilan ini terdiri dari enam jenis pilihan Mobil yang tersedia diantaranya Toyota Avanza, Toyota Kijang Innova, Toyota Rush, Honda BR-V, Honda Mobilio dan Toyota Calya. Dalam tampilan ini, pengguna dapat memilih jenis kendaraan Mobil sesuai dengan yang diinginkan, selain pilihan jenis kendaraan pada halaman utama ini terdapat menu-menu pendukung lainnya diantaranya: menu informasi rental, menu *gallery* dan menu *setting* untuk mengaktifkan lokasi pada *Smartphone*, dan terdapat SlideShow yang berfungsi sebagai gambar promosi yang sedang berlangsung pada rental Mobil, dan memiliki satu buah navigation drawer sebagai menu tambahan pada tampilan menu utama ini, diantaranya: menu profil, menu status pemesanan, menu hubungi kami, menu tutorial pemesanan dan menu logout aplikasi. Tampilan menu utama pengguna dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Halaman Menu Utama Pengguna.

4.2.4 Tampilan Aplikasi Menu Informasi dan *Gallery*

Tampilan dibawah ini merupakan tampilan aplikasi menu informasi rental dan menu gallery, pada menu informasi rental menjelaskan mengenai informasi-informasi mengenai mekanisme proses rental yang ada di Lampung, fasilitas-fasilitas yang diberikan, spesifikasi kendaraan yang tersedia, dan informasi-informasi lainnya. Dan pada menu gallery terdapat kumpulan foto-foto yang ada di Lampung, baik berupa foto tempat yang ada di Bandar Lampung, baik dari foto Bandara, tempat-tempat destinasi wisata yang populer yang ada di provinsi Lampung. Tampilan menu informasi dan *gallery* dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Halaman Menu Informasi Rental dan Gallery.

4.2.5 Tampilan Halaman Menu Informasi Kendaraan

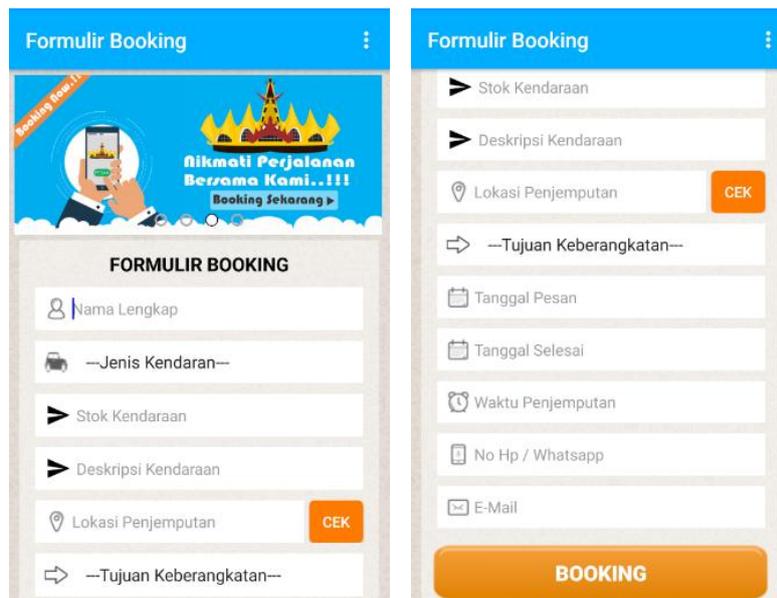
Tampilan halaman selanjutnya adalah tampilan halaman informasi kendaraan, dimana pada tampilan ini berisikan informasi masing-masing dari jenis kendaraan yang tersedia di aplikasi rental Mobil ini. Pada menu utama sebelumnya terdapat jenis-jenis pilihan kendaraan jika disentuh maka akan masuk kedalam menu informasi kendaraan ini, yang berfungsi untuk menginformasikan secara detail mengenai informasi dari masing-masing kendaraan baik berupa nama kendaraan, harga untuk dalam atau luar kota, durasi untuk setiap harinya, dan fasilitas yang diberikan untuk masing-masing jenis kendaraan. Tampilan menu informasi kendaraan dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan Halaman Menu Informasi Kendaraan Mobil.

4.2.6 Tampilan Aplikasi Menu Formulir *Booking*

Tampilan dibawah ini merupakan tampilan aplikasi pada menu formulir pemesanan, dimana pada halaman formulir ini setelah memilih jenis kendaraan yang cocok untuk dilakukan pemesanan, pengguna dapat mengisi formulir biodata pemesanan kendaraan dengan benar, baik nama lengkap, tujuan keberangkatan, tanggal pemesanan kendaraan, tanggal selesai kendaraan, jenis kendaraan yang dipilih, status kendaraan apakah tersedia ataupun tidak, waktu penjemputan, lokasi penjemputan, email user atau pengguna dan nomor handphone atau whatsapp pengguna yang aktif. Sehingga jika data yang diinputkan dirasa sudah benar maka pengguna dapat menyentuh tombol *booking* untuk selanjutnya disimpan ke database dan akan diproses langsung ke menu status pemesanan. Tampilan menu formulir pemesanan dapat dilihat pada gambar 4.6.



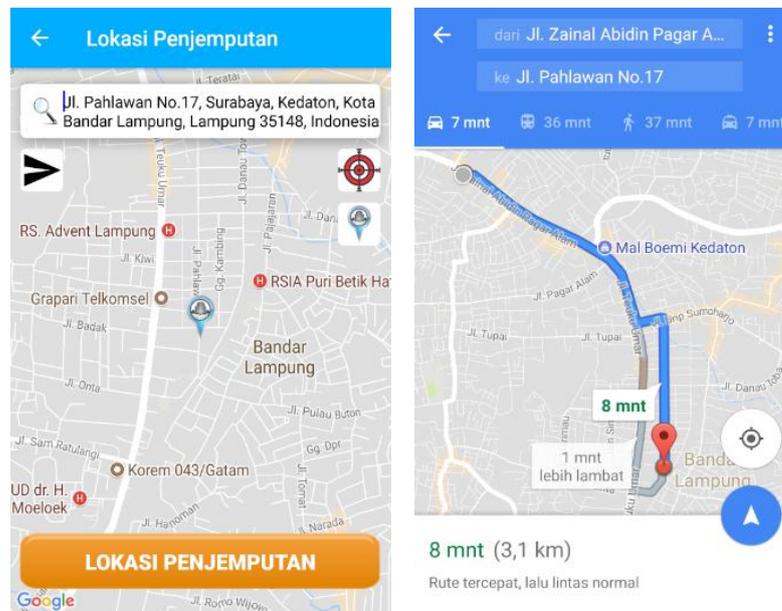
The image shows two side-by-side screenshots of a mobile application interface for booking a vehicle. The left screenshot displays the main form titled 'FORMULIR BOOKING' with a blue header. The form includes fields for: 'Nama Lengkap' (Full Name), 'Jenis Kendaraan' (Vehicle Type), 'Stok Kendaraan' (Vehicle Stock), 'Deskripsi Kendaraan' (Vehicle Description), 'Lokasi Penjemputan' (Pickup Location) with a 'CEK' button, and 'Tujuan Keberangkatan' (Destination). A promotional banner at the top reads 'Nikmati Perjalanan Bersama Kami...!!! Booking Sekarang >'. The right screenshot shows a detailed view of the form fields, including: 'Stok Kendaraan', 'Deskripsi Kendaraan', 'Lokasi Penjemputan' with a 'CEK' button, 'Tujuan Keberangkatan', 'Tanggal Pesan' (Booking Date), 'Tanggal Selesai' (Completion Date), 'Waktu Penjemputan' (Pickup Time), 'No Hp / Whatsapp', 'E-Mail', and a large orange 'BOOKING' button at the bottom.

Gambar 4.6 Tampilan Halaman Menu Formulir Pemesanan.

4.2.7 Tampilan Aplikasi Menu Fitur *Maps Location*

Tampilan dibawah ini merupakan tampilan aplikasi pada menu fitur *maps location*, dimana pada tampilan ini merupakan penerapan fitur *maps location* pada aplikasi pemesanan rental Mobil, pada saat pengguna mengisi formulir lokasi penjemputan, pengguna dapat menyentuh tombol cek untuk masuk ke menu halaman ini, setelah masuk ke dalam halaman lokasi penjemputan pengguna dapat memilih dua pilihan penjemputan diantaranya: yang pertama pengguna dapat mengisi text search untuk menentukan lokasi penjemputan maupun menentukan titik penjemputan secara manual dengan mengklik tombol dengan simbol marker. Selanjutnya jika lokasi penjemputan sudah ditentukan pengguna bisa mengklik tombol lokasi penjemputnya.

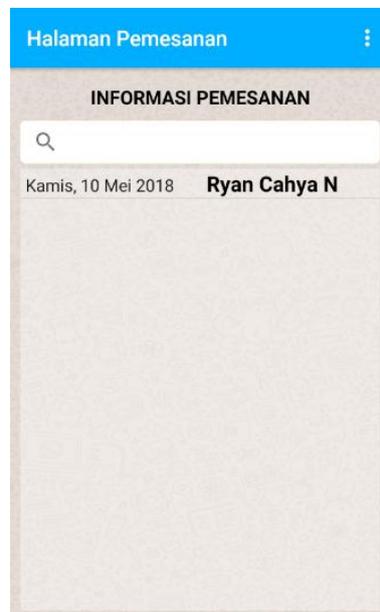
Setelah data pesanan tersimpan pada status pemesanan, *driver* dapat melihat lokasi penjemputan *customer*, *driver* dapat mengklik tombol cek lokasi pejemputan dan membutuhkan fasilitas tambahan berupa aplikasi maps google sebagai arahan dari titik lokasi *driver* menuju ke lokasi penjemputan *customer*.



Gambar 4.7 Tampilan Halaman Menu Fitur *Maps Location*.

4.2.8 Tampilan Aplikasi Menu Status Informasi Pemesanan

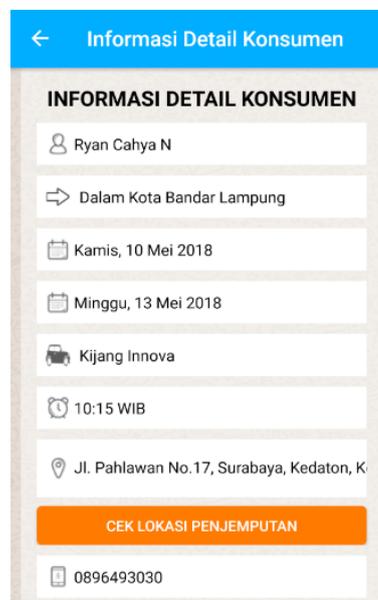
Tampilan selanjutnya adalah tampilan aplikasi pada menu status informasi pemesanan, dimana pada tampilan ini berisikan informasi tanggal pemesanan dan nama pemesanan aplikasi, setelah pengguna melakukan pemesanan pada halaman formulir pemesanan maka akan otomatis masuk ke halaman status informasi pemesanan. Untuk tampilan antara informasi pemesanan pengguna dengan informasi pemesanan antara *admin* dan *driver* memiliki tampilan yang berbeda yaitu penambahan fitur *search* yang difungsikan untuk mempermudah pencarian data dari informasi pemesanan. Tampilan menu status informasi kendaraan dapat dilihat pada gambar 4.8.



Gambar 4.8 Tampilan Halaman Menu Status Informasi Pemesanan.

4.2.9 Tampilan Aplikasi Menu Informasi Detail Pemesanan

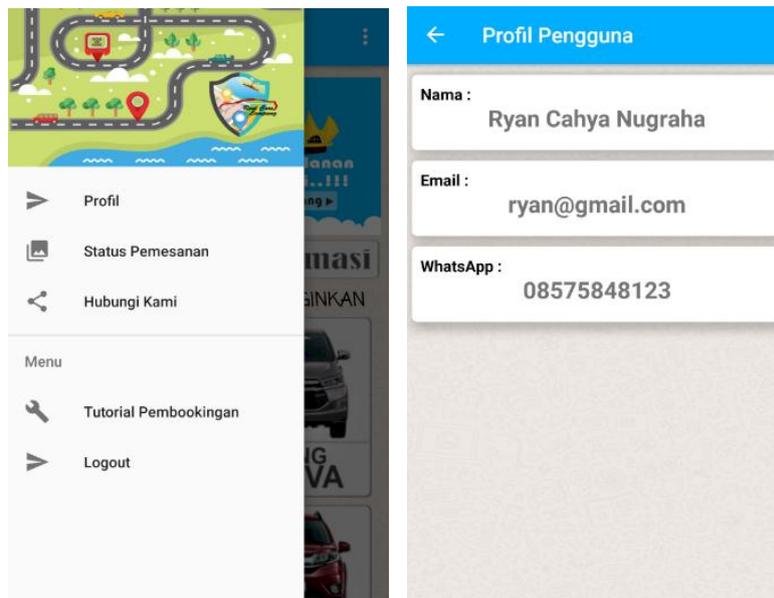
Tampilan selanjutnya adalah tampilan aplikasi pada menu informasi detail pemesanan, dimana pada tampilan ini berisikan informasi detail mengenai data-data pengguna yang sudah melakukan pemesanan sebelumnya, pada tampilan ini melihatkan informasi data pengguna baik nama lengkap, tujuan keberangkatan, tanggal pemesanan kendaraan, tanggal selesai, jenis kendaraan yang dipesan, waktu penjemputan, lokasi penjemputan, email user atau pengguna dan nomor handphone atau whatsapp pengguna. Tampilan menu informasi detail pemesanan dapat dilihat pada gambar 4.9.



Gambar 4.9 Tampilan Halaman Menu Informasi Detail Pemesanan.

4.2.10 Tampilan Aplikasi Menu Profil

Tampilan selanjutnya adalah tampilan aplikasi pada menu profil pengguna, pada tampilan ini merupakan informasi-informasi data dari pengguna diantaranya nama lengkap email dan nomor whatsapp. Data profil ini diambil dari data formulir pendaftaran pengguna baru pada halaman pendaftaran. Tampilan menu profil pengguna dapat dilihat pada gambar 4.10.



Gambar 4.10 Tampilan Halaman Menu Profil Pengguna.

4.2.11 Tampilan Aplikasi Menu Bantuan Dan Tutorial

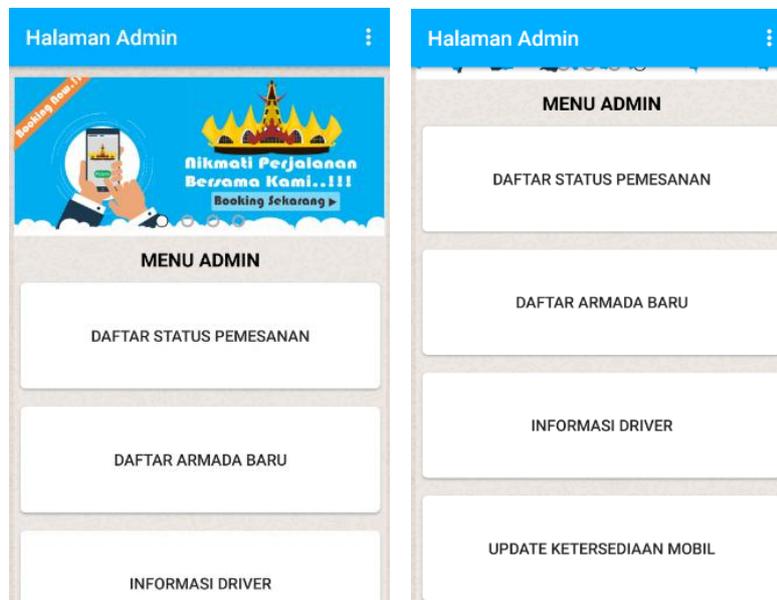
Tampilan selanjutnya adalah tampilan aplikasi pada menu bantuan dan tutorial pemesanan, pada tampilan ini berisikan informasi mengenai cara menghubungi *admin* atau pemilik dari rental Mobil. Dalam halaman bantuan terdapat empat jenis pilihan bantuan diantaranya tombol telepon, tombol pesan, tombol whatsapp, dan tombol email, dimana pada masing-masing tombol sudah di settingkan nomor dan email langsung menuju ke *admin*. selain itu terdapat juga tampilan tutorial pemesanan dan pembatalan. Sehingga pengguna yang masih bingung cara untuk melakukan pemesanan ataupun cara untuk membatalkan pesanan, pengguna dapat melihat pada halaman ini. Tampilan menu bantuan dan tutorial pemesanan dapat dilihat pada gambar 4.11.



Gambar 4.11 Tampilan Halaman Menu Bantuan dan Tutotrial Pemesanan.

4.2.12 Tampilan Aplikasi Menu Utama *Admin*

Tampilan selanjutnya adalah tampilan aplikasi pada menu utama *admin*, pada tampilan ini berisikan empat buah pilihan menu diantaranya menu daftar status pemesanan yaitu berfungsi untuk melihat daftar pemesanan yang melakukan pemesanan, menu daftar armada baru yaitu berfungsi untuk menambahkan *driver* baru pada aplikasi, menu informasi *driver* yaitu melihat informasi detail mengenai *driver*, dan terakhir menu update ketersediaan mobil yaitu mengubah stok ketersediaan mobil yang masih tersedia. Tampilan menu utama *admin* dapat dilihat pada gambar 4.12.



Gambar 4.12 Tampilan Halaman Menu Utama *Admin*.

4.2.13 Tampilan Aplikasi Menu Pendaftaran *Driver* Baru

Tampilan terakhir adalah tampilan aplikasi pada menu pendaftaran *driver* baru, pada tampilan ini berisikan formulir pendaftaran *driver* baru ke dalam sistem aplikasi, data-data pada formulir pendaftaran *driver* ini diantaranya nama lengkap, jenis kendaraan, nomor plat kendaraan, alamat lengkap *driver*, email, *password*, dan nomor whatsapp. Pada saat formulir pendaftaran sudah disikan data dengan benar maka dapat menyentuh tombol daftar sehingga data dari *driver* baru akan otomatis tersimpan di *firebase database*. Tampilan menu pendaftaran *driver* baru dapat dilihat pada gambar 4.13.



The image shows a mobile application interface for driver registration. At the top, there is a blue header with a back arrow and the text 'Daftar Driver'. Below the header is a decorative banner with a car icon and the text 'Booking Sekarang >'. The main content area is titled 'DAFTAR DRIVER BARU' and contains a vertical stack of input fields, each with a corresponding icon: a person icon for 'Nama Lengkap', a car icon for 'Jenis Kendaraan', a license plate icon for 'No. Plat Kendaraan', a house icon for 'Alamat Lengkap', an envelope icon for 'E-Mail', an eye icon for 'Password', and a mobile phone icon for 'No Hp / Whatsapp'. At the bottom of the form is a large orange button labeled 'DAFTAR'.

Gambar 4.13 Tampilan Halaman Menu Pendaftaran *Driver* Baru.

4.3 Pengujian (*Testing*)

Pengujian atau *testing* dilakukan untuk mengetahui kualitas dari perangkat lunak yang dirancang ataupun yang sedang dikembangkan apakah sesuai dengan perancangan yang sebelumnya sudah dibuat ataupun tidak. Pengujian program merupakan pengujian terhadap program yang dijalankan, sehingga dapat diketahui kesalahan-kesalahan maupun kekurangan dari program itu sendiri sehingga jika dalam proses pengujian mengalami masalah, *error* maupun tidak sesuai dengan proses yang sebelumnya sudah ditentukan, maka bisa dilakukan proses perbaikan rancangan aplikasi. Berikut ini uji fungsi aplikasi ditampilkan pada Tabel 4.1.

Tabel 4.1 Uji Pilihan Menu Tombol Aplikasi

Tombol	Uji Fungsi	Hasil Uji
Halaman Login, tombol daftar	Menyentuh tombol daftar	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman pendaftaran pengguna baru.
Halaman utama, tombol jenis kendaraan	Menyentuh Image Button jenis-jenis kendaraan	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman informasi kendaraan.
Halaman utama, tombol informasi	Menyentuh tombol informasi	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman informasi rental Mobil.
Halaman utama, tombol <i>gallery</i>	Menyentuh tombol <i>gallery</i>	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman <i>gallery</i> .
Halaman utama, tombol profil	Menyentuh tombol profil	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman profil user atau pengguna.
Halaman utama, tombol hubungi kami	Menyentuh tombol hubungi kami	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman hubungi kami dan langsung mengakses ke <i>admin</i> .
Halaman utama, tombol status pemesanan	Menyentuh tombol status pemesanan	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman status pemesanan pelanggan.

Halaman utama, tombol tutorial pemesanan	Menyentuh tombol tutorial pemesanan	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman tutorial pemesanan.
Halaman utama, tombol setting	Menyentuh tombol setting	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman settingan mengaktifkan lokasi.
Halaman formulir, tombol cek lokasi	Menyentuh tombol cek lokasi	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman <i>Maps Location</i> .
Halaman formulir, tombol <i>booking</i>	Menyentuh tombol <i>booking</i>	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman status pemesanan pelanggan.
Halaman status pemesanan, <i>listview</i> pesan	Menyentuh list status pemesanan	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman informasi detail konsumen.
Halaman informasi detail konsumen,	Menyentuh tombol cek lokasi penjemputan	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman <i>maps location</i> peta penjemputan pelanggan.
Halaman utama <i>admin</i> , tombol daftar <i>driver</i> baru	Menyentuh tombol daftar armada baru	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan ke halaman pendaftaran <i>driver</i> baru.
Halaman daftar <i>driver</i> , tombol daftar	Menyentuh tombol daftar <i>driver</i> baru	Sukses, <i>output</i> sesuai dengan yang diinginkan, menampilkan halaman pendaftaran <i>driver</i> baru berhasil.

Selain pengujian tombol pada aplikasi, dilakukan juga pengujian terhadap fungsi-fungsi yang terdapat pada aplikasi pemesanan rental Mobil. Berikut ini proses pengujian fungsi aplikasi ditampilkan pada Tabel 4.2.

Tabel 4.2 Uji Fungsi Aplikasi

Tombol	Uji Fungsi	Hasil Uji
Fungsi daftar pengguna baru	Mengisi biodata dan menyentuh tombol daftar	Sukses, langsung otomatis masuk ke halaman utama.
Fungsi login	Menginputkan <i>email</i> dan <i>password</i> dan menyentuh tombol masuk	Sukses, langsung otomatis dapat masuk ke halaman utama.
Fungsi pencarian lokasi penjemputan	Mengetikan alamat penjemputan	Sukses, dapat mencari otomatis titik lokasi penjemputan.
Fungsi pemesanan	Mengisi biodata pemesanan dengan lengkap dan menyentuh tombol <i>booking</i>	Sukses, sesuai dengan yang diinginkan formulir pesananan yang sudah dipesan langsung otomatis masuk ke menu status pemesanan.
Fungsi <i>maps location</i>	Menyentuh tombol cek lokasi penjemputan	Sukses, sesuai dengan yang diinginkan yaitu dapat melihat titik lokasi penjemputan <i>customer</i> .
Fungsi daftar <i>driver</i>	Mengisi biodata <i>driver</i> dan menyentuh tombol daftar	Sukses mendaftarkan <i>driver</i> baru dan data otomatis masuk ke <i>firebase database</i> .

Tahapan pengujian selanjutnya yaitu mengimplementasikan aplikasi yang sudah dirancang ke pada beberapa *device* atau perangkat *Smartphone* berbasis android, data hasil pengujian aplikasi ke perangkat Android dapat dilihat dan ditampilkan pada Tabel 4.3.

Tabel 4.3 Hasil Instalasi pada Perangkat *Smartphone*

No	Type <i>Smartphone</i>	Sistem Operasi	Instalasi
1.	Xiaomi 4X Prime	Android 7.1 (<i>Nougat</i>)	Berhasil
2.	Samsung Galaxy S8	Android 8.0 (<i>Oreo</i>)	Berhasil
3.	Xiaomi Redmi Note 3	Android 5.1 (<i>Lolipop</i>)	Berhasil
4.	Samsung Galaxy j7 prime	Android 5.0 (<i>Lolipop</i>)	Berhasil
5.	Samsung Galaxy A5	Android 8.0 (<i>Oreo</i>)	Berhasil
6.	Asus Zenfone Max Pro M1	Android 8.0 (<i>Oreo</i>)	Berhasil

Pada tabel 4.3 diatas dapat diketahui bahwa aplikasi pemesanan rental Mobil yang dirancang dapat terinstalasi dengan baik pada beberapa *Smartphone* berbasis Android yang berbeda, dan juga pada sistem operasi Android yang berbeda juga berhasil terinstal, dan aplikasi yang diinstal dapat berjalan dengan baik.

4.4 Kelebihan dan Kelemahan Aplikasi

- 1) Kelebihan Aplikasi:
 - a. Tampilan aplikasi yang dirancang dapat *responsive* ke berbagai jenis ukuran layar *Smartphone*.
 - b. Aplikasi yang dirancang dapat menentukan persediaan kendaraan dan menyimpan data secara *realtime*.
 - c. Tampilan aplikasi dirancang semenarik mungkin sehingga mudah dipahami oleh pengguna baru.
 - d. Didalam aplikasi proses penjemputan *customer* menggunakan fitur *maps location*.

2) Kelemahan Aplikasi:

- a. Aplikasi pemesanan rental Mobil ini hanya dapat dijalankan pada *Smartphone* berbasis Android.
- b. Harus membutuhkannya koneksi internet untuk melihat halaman status pemesanan *customer*.
- c. Pada aplikasi *driver* hanya dapat menampilkan keseluruhan pesanan pada aplikasi *admin*, tidak menspesifikasikan status pemesanan ke masing-masing *driver* saja.
- d. Pada aplikasi ini belum tersedianya fasilitas halaman menu pembayaran.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil kesimpulan bahwa:

- 1) Aplikasi *booking order* rental Mobil dapat menerapkan dan menampilkan fitur *maps location* titik penjemputan *customer*.
- 2) Dengan melakukannya penelitian ini dalam penerapan perancangan aplikasi *booking order* rental Mobil, sehingga dapat membantu customer untuk melakukan pemesanan kendaraan dengan menggunakan *Smartphone*.
- 3) Dari proses pengujian aplikasi menjelaskan bahwa perancangan aplikasi dapat digunakan dengan baik pada pengembangan perangkat lunak berbasis Android.
- 4) Aplikasi dapat memberikan informasi mengenai fasilitas dan jenis kendaraan, selain itu dapat melakukan penjawalan ketersediaan kendaraan secara *realtime* sehingga tanpa harus melakukan pembukuan secara manual.

5.2 Saran

Sebagai perbaikan untuk penelitian yang lebih lanjut, maka peneliti memberikan saran diantaranya:

- 1) Diharapkan pada penelitian selanjutnya hendaknya tidak hanya dapat diaplikasikan ke pengembangan perangkat lunak berbasis Android tetapi juga dapat diaplikasikan ke pengembangan perangkat lunak berbasis IOS.

- 2) Penambahan fitur sistem pembayaran dalam aplikasi pemesanan kendaraan.
- 3) Pada aplikasi *driver* pada menu status pemesanan dapat menampilkan untuk menspesifikasikan pesanan ke masing-masing *driver*.
- 4) Pada aplikasi *admin* dapat tersambung dengan *website*, sehingga jika kapasitas data detail pesanan banyak dapat mudah dilihat.
- 5) Pada aplikasi tidak hanya dapat menentukan titik lokasi penjemputan, juga dapat menentukan titik tujuan lokasi dari masing-masing *customer*.
- 6) Pada perancangan aplikasi selanjutnya tidak hanya dapat memesan kendaraan Mobil kepada satu tempat usaha rental Mobil melainkan dapat mencari keseluruhan tempat usaha rental Mobil yang ada di Bandar Lampung.
- 7) Ditambahkannya fitur GPS pada *maps location* untuk mengetahui posisi lokasi *driver*.

DAFTAR PUSTAKA

- Afaar. 2015. *Teknik Pengumpulan Data*. Diambil dari: [e-journal.uajy.ac.id/6933/4/MTA302033.pdf] (20 November 2017).
- Ariyanti, Rena dan Khairil. 2015. *Pemanfaatan Google Maps Api*. Universitas Dehasen Bengkulu Jurnal Media Infotama Vol. 11 No. 2.
- Arkhiansyah, Yuni. 2018. *Proyek Perangkat Lunak*. Bandar Lampung: Darmajaya (DJ) Press.
- Google, firebase. 2017. *Firestore Realtime Database dengan Android*. Diambil dari: [<https://firebase.google.com/docs/database>] (28 Desember 2017).
- Juansyah, Andi. 2015. *Pembangunan Aplikasi Child Tracker Berbasis Assisted*. Bandung: Jurnal Ilmiah Komputer dan Informatika. Edisi.1 Volume.1 ISSN:2089-9033.
- Neti. Ervi C. Rennu A. 2013. *Perancangan Aplikasi Rental Mobil Pada CV Karya Bersama Palembang*. Manajemen Informatika, STMIK-MDP.
- Neyfa, Bella Chintya dan Dony Tamara. 2016. *Perancangan Aplikasi E-Canteen Berbasis Android*. Depok : Jurnal Penelitian Komunikasi dan Opini Publik Vol. 20 No.1.
- Pratami, Riviana Dias dan Hendra Kurniawan. *Aplikasi Pencarian Tempat Makan Menggunakan Location Based Service Pada Android*, Jurusan Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji (UMRAH).
- Pressman, Ph.D. Roger S. 2012. *Pendekatan Praktisi Rekayasa Perangkat Lunak*. Edisi 7. Penerbit Andi. Yogyakarta.
- Raharjo, Budi. 2013. *Pengertian Reservasi Menurut Para Ahli*. Diambil dari: [<http://www.definisimenurutparaahli.com/pengertian-reservasi/>]. (5 November 2017).
- Sari, Deffi Rosdiana. 2011. *Membangun Aplikasi Sistem Pemesanan Dan Pembayaran Sewa Mobil Online Berbasis Web (Studi Kasus Di Rental Daras Corporation)*. Universitas Komputer Indonesia.
- Sukanto, Rosa A. dan M. Salahuddin. 2014. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika.
- Syahputra, Muhammad. 2016, "Perancangan Aplikasi Booking Taxi Secara Online Berbasis Android". Teknik Informatika Sekolah Tinggi Teknik Harapan Medan.

Wijaya, Sugeng Hadi. 2016. *Aplikasi Sistem Pemesanan Travel Berbasis Android pada PT Kirana Anugerah Terindah*. Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra Surabaya.

Yuni97. 2017. *Versi-Versi Android Hingga Saat Ini*. Diambil dari: [<https://www.codepolitan.com/versi-versi-android-hingga-saat-ini-59c4ba1c88350>] (6 April 2018).

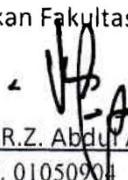
LAMPIRAN



SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.0211/DMJ/DFIK/BAAK/VII-18
Tentang
Dosen Pembimbing Skripsi
Program Studi S1 Teknik Informatika
REKTOR IIB DARMAJAYA

- Memperhatikan : 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi**.
2. Laporan dan usulan Ketua Program Studi **S1 Teknik Informatika**.
- Menimbang : 1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.
- Mengingat : 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.
- Menetapkan**
- Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.
- Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.
- Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 05 Juli 2018
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer


Dr. R.Z. Abdul Aziz, M.T.
NIK. 01050904

1. Kabiro. SDM
2. Ketua Jurusan S1 Teknik Informatika
3. Yang bersangkutan
4. Arsip

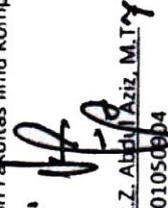
Lampiran : Surat Keputusan Rektor IIB Darmajaya
Nomor : SK. 0211/DMJ/DFIK/BAAK/VII-18
Tanggal : 05 Juli 2018
Perihal : Pembimbing Penulisan Skripsi
Program Studi Strata Satu (S1) Teknik Informatika

Judul Penulisan Skripsi dan Dosen Pembimbing
Program Studi Strata Satu (S1) Teknik Informatika

NO.	NAMA	NPM	JUDUL	PEMBIMBING
1	*Pikri Pit Pratama	1411010086	Customisable Notification Dalam Layanan Pelanggan Berbasis Android (Studi Kasus : Dealer Auto 2000)	Hariyanto Wibowo, S.Kom, M.T.I
2	*Ryan Cahya Nugraha	1411010051	Penggunaan Fitur Maps Location Dalam Perancangan Aplikasi Booking Order Rental Mobil Pada CV. Sabai Rent Car Berbasis Android	

Keterangan : * Surat Keputusan Perpanjangan

A.n. Rektor IIB Darmajaya
Dekan Fakultas Ilmu Komputer


Dr. R.Z. ABUL AZIZ, M.T.
NIK. 01050904



Bandar Lampung, 13 Desember 2017

Nomor : Penelitian.007/DMJ/DEKAN/BAAK/XII-17
Lampiran : -
Perihal : Permohonan Izin Penelitian

Kepada Yth,

CV. Sabai Rent Car

Di -

Jl. Pulau Belitung No.5 Sukabumi Kec. Sukabumi, Kota Bandar Lampung,
Prov. Lampung 35122

Dengan hormat,

Sehubungan dengan peraturan Akademik Institut Informatika dan Bisnis (IIB) bahwa mahasiswa/i Strata Satu (S1) yang akan menyelesaikan studinya diwajibkan untuk memiliki pengalaman kerja dengan melaksanakan Penelitian dan membuat laporan yang waktunya disesuaikan dengan kalender Informatik & Business Institute (IBI) Darmajaya.

Untuk itu kami mohon kerja sama Bapak/Ibu agar kiranya dapat menerima mahasiswa/i untuk melakukan Penelitian, yang pelaksanaannya dimulai dari tanggal **20 Desember 2017 s.d 31 Januari 2018** (selama satu bulan dua minggu)

Adapun mahasiswa/i tersebut adalah :

Nama : Ryan Cahya Nugraha
NPM : 1411010051
Jurusan : S1 Teknik Informatika
Jenjang : Strata Satu (S1)

Demikian permohonan ini dibuat, atas perhatian dan kerjasama yang baik kami ucapkan terimakasih.

Dean
Fakultas Ilmu Komputer,



Sriyanto S. Kom, MM
NIK. 00240800

Tembusan:

1. Jurusan S1 Teknik Informatika
2. Arsip.



CV.SABAI RENT CAR

**JL. PULAU BELITUNG NO.5 KEL.SUKABUMI,
KEC. SUKABUMI, KOTA BANDAR LAMPUNG**

25 Januari 2018

Dengan Hormat,

yang bertanda tangan dibawah ini,

Nama : Sultan

Jabatan : Owner / Pemilik CV.Sabai Rent Car

Menerangkan bahwa,

Nama : Ryan Cahya Nugraha

NPM : 1411010051

Jurusan : Teknik Informatika

Telah kami berikan izin untuk melaksanakan penelitian pada perusahaan kami sebagai syarat penyusunan skripsi dengan judul:

**“PENGUNAAN FITUR *MAPS LOCATION* DALAM PERANCANGAN
APLIKASI *BOOKING ORDER RENTAL MOBIL* PADA
CV.SABAI RENT CAR BERBASIS ANDROID”**

Demikian surat izin penelitian ini, kami buat agar dapat digunakan sebagaimana mestinya dan atas kerjasamanya kami ucapkan terima kasih.

Hormat Kami,



SULTAN

Owner / Sabai Rent Car

WAWANCARA

- Nama Narasumber : Sultan
- Jabatan / Posisi Narasumber : Owner / Pemilik CV.Sabai Rent Car
- Hari, Tanggal : Kamis, 25 Januari 2018
- Topik Wawancara : Informasi Mekanisme Pemesanan Rental Mobil
- Tempat : CV.Sabai Rent Car
-
- Peneliti : Bagaimana mekanisme pemesanan kendaraan rental Mobil itu seperti apa?
- Narasumber : Untuk cara pemesanan rental Mobil disini konsumen biasanya datang langsung kesini dan juga bisa menggunakan Telephone yang no HP nya ada di website kami, biasanya kalau konsumen yang memesan menggunakan Telephone itu konsumen yang ada di luar kota Bandar Lampung.
- Peneliti : Biasanya konsumen yang memesan kendaraan rental berada di daerah mana saja?
- Narasumber : Untuk pemesanan kendaraan rental ini banyak konsumen yang berada di kota Bandar Lampung namun juga banyak konsumen yang memesan dari luar kota untuk memesan rental untuk digunakan di Bandar Lampung.
- Peneliti : Berapa jumlah unit kendaraan rental yang dimiliki?
- Narasumber : Untuk jumlah kendaraan yang kami miliki yaitu 6 unit Avanza, 3 unit Innova, 2 unit Mobilio, 1 Unit BR-V, 1 Unit Rush, 1 Unit Calya.
- Peneliti : Apakah setiap hari unit kendaraan yang dimiliki selalu penuh untuk dipesan oleh konsumen?

Narasumber : Untuk ketersediaan unit kendaraan tergantung dari hari-hari tertentu, karena untuk pemesanan kendaraan rental oleh konsumen tidak bisa ditentukan, kadang hari ini hanya sebagian kendaraan yang keluar, kadang untuk besok kendaraan yang dipesan full.

Peneliti : Kesulitan apa yang dialami dari bisnis rental Mobil ini?

Narasumber : Kalau untuk kesulitan dalam bisnis rental ini biasanya untuk proses penjadwalannya karena kami harus mengetahui kendaraan yang sudah dipesan sebelumnya dengan kendaraan yang belum dipesan, dan juga untuk supir biasanya konsumen kebanyakan yang memesan hanya menginfokan lokasi alamat yang kurang jelas jadi untuk penjemputan ke konsumen supir harus membutuhkan waktu lama untuk mencari lokasi penjemputan.

Peneliti : Apakah tujuan rental Mobil ini bisa tujuan ke luar kota Bandar Lampung atau Provinsi Lampung?

Narasumber : Untuk tujuan perjalanan rental kami melayani tujuan dalam kota dan kabupaten yang ada di Lampung seperti Metro, Tanggamus, Natar, Kotabumi, Tulang Bawang, dll. Namun untuk tujuan perjalanan keluar provinsi Lampung kami belum melayani.

Peneliti : Apakah Rental Mobil ini hanya melayani menggunakan supir atau juga bisa rental kendaraan dengan lepas kunci?

Narasumber : Untuk pemesanan kendaraan dari rental kami semua sudah disediakan langsung menggunakan supir dan kami tidak melayani pemesanan kendaraan dengan lepas kunci.

Peneliti : Data-data apa saja yang dibutuhkan untuk melakukan pemesanan kendaraan rental Mobil ini?

Narasumber : Data yang diperlukan untuk pemesanan rental yaitu nama konsumen, no handphone, jenis kendaraan yang dipesan, tanggal pemesanan, tanggal selesai, jam penjemputan, jam penjemputan, lokasi penjemputan dan tujuan keberangkatan.

Peneliti : Untuk tarif rental Mobil biasanya dikenakan berapa dalam 1 hari?

Narasumber : Tarif biaya dalam 1 hari durasi 12 jam untuk dalam kota 450rb untuk Avanza, Mobilio, dan Calya, 500rb untuk BR-V, dan Rush, 650rb untuk Innova, itu sudah termasuk kendaraan, supir dan BBM belum termasuk biaya parkir. Untuk tujuan luar kota biaya tarif tergantung dari tujuannya.





Institut Informatika & Bisnis
DARMAJAYA

Yayasan Alfian Husin
 Jl. Zainal Abidin Pagar Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

PENDAFTARAN UJIAN SKRIPSI/TUGAS AKHIR*

Nama : Ryan Cahya Nugraha
 NPM : 1411010051
 Program Studi : Teknik Informatika
 Judul : Penggunaan Fitur Maps Location Dalam Perancangan Aplikasi Booking Order Rental mobil pada CV. Sabai Rent Car Berbasis Android.
 Pembimbing : Hariyanto Wibowo, S.Kom, M.T.I

Mendaftar untuk ujian yang

PERTAMA / ~~KEDUA~~ / ~~KETIGA~~ (Lingkari salah satu)

Tanggal Pendaftaran : 7 September 2018

Bandar Lampung, 7/09.....2018

Yang menerima Pendaftaran

 (.....)

Mahasiswa

 (Ryan Cahya Nugraha
 1411010051)

Catatan :
 Dibuat rangkap dua : Mahasiswa dan BAAK (masing-masing satu rangkap)
 *) coret yang tidak perlu

**PENGGUNAAN FITUR *MAPS LOCATION* DALAM PERANCANGAN
APLIKASI *BOOKING ORDER* RENTAL MOBIL PADA
CV.SABAI RENT CAR BERBASIS ANDROID**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER
Pada Jurusan Teknik Informatika
Institut Informatika Dan Bisnis Darmajaya Bandar Lampung



5/6/18
[Handwritten signature]
Aer Sedyap

Oleh

RYAN CAHYA NUGRAHA

NPM.1411010051

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2018**

**PENGGUNAAN FITUR *MAPS LOCATION* DALAM PERANCANGAN
APLIKASI *BOOKING ORDER* RENTAL MOBIL PADA
CV.SABAI RENT CAR BERBASIS ANDROID**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER
Pada Jurusan Teknik Informatika
Institut Informatika Dan Bisnis Darmajaya Bandar Lampung

Ace Rensi
01 / 2018 / k.
10



Ace Rensi
1/okt/2018
(Signature)

Oleh

RYAN CAHYA NUGRAHA

NPM.1411010051

2/10/18
(Signature)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG**

2018



FORMULIR

KARTU SEMINAR PROPOSAL

NAMA : Ryan Cahya Nugraha

NPM : 1411010051

PROGRAM STUDI : ~~TI / SI / MI / SK / TK / MA / AK / AD *~~

FAKULTAS : ~~ILMU KOMPUTER / ILMU BISNIS DAN EKONOMI~~

FOTO 3X4

No.	Tanggal	Nama Mahasiswa	Judul	Paraf Pembahas
1	22 / 2017 / 11	Putri Ayu Wulandari	Penerapan data mining untuk memprediksi mahasiswa yang beresiko mendapatkan status peruntukan perkuliahan dengan algoritma naive bayes	
2	22 / 2017 / 11	Ayu Silvia	aplikasi mobile untuk notifikasi kegiatan dosen menggunakan metode bound robin (studi kasus : ib Darmajaya.)	
3	22 / 2017 / 11	widia agustriani	Rancang bangun aplikasi pembelajaran matematika diskrit berbasis android.	
4	28 / 2017 / 11	Firdaus	Perancangan dan pembuatan aplikasi visualisasi 3D rangkaian atom pada senyawa hidrokarbon berbasis android	
5	28 / 2017 / 11	Bakti Sanudra	Penerapan augmented reality pembelajaran kompas dan pratelan mobil berbasis android	
6				

* Coret yang tidak perlu

Catatan :

Mahasiswa wajib menghadiri seminar proposal minimal 5(lima) kali sebelum tampil seminar proposal

Bandar Lampung,
Ka. Jurusan

[Yuni Arkhiansyah, r1 Kom]
NIK. 00480802



DARMAJAYA

Yayasan Alfian Husin

Jl. Zainal Abidin Pagar Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

BIRO ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)

FORM KONSULTASI/BIMBINGAN SKRIPSI/~~TUGAS AKHIR~~ *)

NAMA : Ryan Cahya Nugraha
 NPM : 1411010051
 PEMBIMBING I : Hariyanto Wibowo, S.Kom., M.T.I
 PEMBIMBING II :
 JUDUL LAPORAN : Penggunaan Fitur Maps Location Dalam Perancangan Aplikasi Booking Order Rental Mobil Pada CU. Sabai Rent Car Berbasis Android.
 TANGGAL SK : s.d (6+2 bulan)

No	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	29/3/18	+ Labor Bulanan	
2	12/4/18	+ Analisis awal / tynan per forum	
3	18/4/18	+ Uebatke syabe, Metale	
4	2/5/18	+ Proyeke syabe Use Case, dan design syabe	
5	14/5/18	+ Uj syabe yg dibyru	
6	16/5/18	+ Puberok syabe	
7	23/5/18	+ Supula & Sase by Cerna Sem bullean	
8	5/6/18	+ Aca Rqhu Sykue	
9			
10			

*) Coret yang tidak perlu

Bandar Lampung, 14 - Agustus 2018
Ketua Jurusan

(Yuni Ardhiansyah, M. Kom)
 NIK. 00480802

Lampiran Code Source Program :

Login.Java

```
public class Login extends
AppCompatActivity implements
View.OnClickListener {

    FirebaseAuth mAuth;
    EditText editTextEmail,
editTextPassword;
    ProgressBar progressBar;

    @Override
    protected void
    onCreate(Bundle
    savedInstanceState) {

super.onCreate(savedInstanceSt
    ate);

    setContentView(R.layout.activi
ty_login);

    getSupportActionBar().setTitle
    ("Halaman Login");

        mAuth =
    FirebaseAuth.getInstance();

        editTextEmail =
    (EditText)
    findViewById(R.id.editTextEmai
l);
        editTextPassword =
    (EditText)
    findViewById(R.id.editTextPass
word);
        progressBar =
    (ProgressBar)
    findViewById(R.id.progressbar)
    ;

    findViewById(R.id.daftar).setO
    nClickListener(this);

    findViewById(R.id.masuk).setO
    nClickListener(this);

        Button btn2 = (Button)
    findViewById(R.id.daftar);

    btn2.setOnClickListener(new
    View.OnClickListener() {
        @Override
```

```
        public void
    onClick(View v) {

            Intent i2 =

new
    Intent(getApplicationContext()
    , daftarlayout.class);

    startActivity(i2);
        }
    });
    }

        private void userLogin() {
            String email =
editTextEmail.getText().toStri
ng().trim();
            String password =
editTextPassword.getText().toS
tring().trim();

            if (email.isEmpty()) {

editTextEmail.setError("Masuka
n Email Anda Terlebih
Dahulu");

editTextEmail.requestFocus();
                return;
            }

            if
    (!Patterns.EMAIL_ADDRESS.match
    er(email).matches()) {

editTextEmail.setError("Mohon
Masukan Email Anda Yang
Benar");

editTextEmail.requestFocus();
                return;
            }

            if
    (password.isEmpty()) {

editTextPassword.setError("Mas
ukan Password Anda");

editTextPassword.requestFocus(
);
                return;
            }

            if (password.length()
    < 6) {

editTextPassword.setError("Pan
jang Kata Sandi Minimum Harus
```

```

6");
editTextPassword.requestFocus(
);
        return;
    }

progressBar.setVisibility(View
.VISIBLE);

 mAuth.signInWithEmailAndPasswo
rd(email,
password).addOnCompleteListener(
new
OnCompleteListener<AuthResult>
() {
        @Override
        public void
onComplete(@NonNull
Task<AuthResult> task) {

progressBar.setVisibility(View
.GONE);
            if
(task.isSuccessful()) {
                finish();
                Intent
intent = new
Intent(Login.this,
MainActivity.class);

intent.addFlags(Intent.FLAG_AC
TIVITY_CLEAR_TOP);

startActivity(intent);
            } else {

Toast.makeText(getApplicationContext()
context(), "Email / Password
Anda Salah Atau Belum
Terdaftar",
Toast.LENGTH_SHORT).show();
            }
        }
    });

    @Override
    protected void onStart() {
        super.onStart();

        if
(mAuth.getCurrentUser() !=
null) {
            finish();
            startActivity(new
Intent(this,

```

```

MainActivity.class));
        }
    }

    @Override
    public void onClick(View
view) {
        switch (view.getId())
        {
            case R.id.daftar:
                finish();

startActivity(new Intent(this,
daftarlayout.class));
                break;

            case R.id.masuk:
                userLogin();
                break;
        }
    }
}

```

Daftar.Java

```

public class daftarlayout
extends AppCompatActivity
implements
View.OnClickListener {
    List<User> user;

    ProgressBar progressBar;
    EditText namalengkap,
whatsapp, editTextEmail,
editTextPassword;
    DatabaseReference
databaseReference;

    private FirebaseAuth
 mAuth;
    private Button daftar;
    private SliderPagerAdapter
 mAdapter;
    private SliderIndicator
 mIndicator;

    private SliderView
 sliderView;
    private LinearLayout
 mLinearLayout;
    String nama, email, nowa,
password;

    @Override
    protected void
onCreate(Bundle
savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

setContentView(R.layout.daftar
layout);

        namalengkap =
        (EditText)
        findViewById(R.id.namalengkap)
        ;
        whatsapp = (EditText)
        findViewById(R.id.whatsapp);
        editTextEmail =
        (EditText)
        findViewById(R.id.editTextEmail);
        editTextPassword =
        (EditText)
        findViewById(R.id.editTextPassword);
        progressBar =
        (ProgressBar)
        findViewById(R.id.progressbar)
        ;
        databaseReference =
        FirebaseDatabase.getInstance()
        .getReference("User");

        mAuth =
        FirebaseAuth.getInstance();
        daftar = (Button)
        findViewById(R.id.daftar);

        findViewById(R.id.daftar).setOnClickListener(this);

        findViewById(R.id.textViewLogin)
        .setOnClickListener(this);

        getSupportActionBar().setTitle
        ("Daftar");

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        sliderView =
        (SliderView)
        findViewById(R.id.sliderView);
        mLinearLayout =
        (LinearLayout)
        findViewById(R.id.pagesContainer);
        setupSlider();
    }

    private void

```

```

registerUser() {
    String nama =
namalengkap.getText().toString()
    .trim();
    String nomor =
whatsapp.getText().toString()
    .trim();
    String email =
editTextEmail.getText().toString()
    .trim();
    String password =
editTextPassword.getText().toString()
    .trim();

        if (nama.isEmpty()) {
namalengkap.setError("Masukan
Nama Anda Terlebih Dahulu");

namalengkap.requestFocus();
            return;
        }

        if (nomor.isEmpty()) {
whatsapp.setError("Masukan
Nomor / WhatsApp Anda Terlebih
Dahulu");

whatsapp.requestFocus();
            return;
        }

        if (email.isEmpty()) {
editTextEmail.setError("Masukan
Email Anda");

editTextEmail.requestFocus();
            return;
        }

        if
        (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
editTextEmail.setError("Mohon
Masukan Email Anda Yang
Benar");

editTextEmail.requestFocus();
            return;
        }

        if
        (password.isEmpty()) {
editTextPassword.setError("Masukan
Password Anda");

```

```

editTextPassword.requestFocus(
);
        return;
    }

    if (password.length()
< 6) {

editTextPassword.setError("Min
imum length of password should
be 6");

editTextPassword.requestFocus(
);
        return;
    }

progressBar.setVisibility(View
.VISIBLE);

mAuth.createUserWithEmailAndPa
ssword(email,
password).addOnCompleteListener(new
OnCompleteListener<AuthResult>
()) {
        @Override
        public void
onComplete(@NonNull
Task<AuthResult> task) {

progressBar.setVisibility(View
.GONE);

            if
(task.isSuccessful()) {
                addUser();
                finish();

startActivity(new
Intent(daftarlayout.this,
MainActivity.class));
            } else {

                if
(task.getException()
instanceof
FirebaseAuthUserCollisionExcep
tion) {

                    Toast.makeText(getApplicationC
ontext(), "Pendaftaran Akun
Berhasil",
                    Toast.LENGTH_SHORT).show();

                } else {

```

```

                    Toast.makeText(getApplicationC
ontext(),
                    task.getException().getMessage
(),
                    Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    @Override
    public void onClick(View
view) {
        switch (view.getId())
        {
            case R.id.daftar:
                registerUser();
                break;

            case
R.id.textViewLogin:
                finish();

                startActivity(new Intent(this,
Login.class));
                break;
        }
    }

    private void addUser() {
        //getting the values
to save
        nama =
namalengkap.getText().toString
().trim();
        nowa =
whatsapp.getText().toString().
trim();
        email =
editTextEmail.getText().toStri
ng().trim();
        password =
editTextPassword.getText().toS
tring().trim();

        if
(!TextUtils.isEmpty(nama)) {

            String id =
mAuth.getUid();

            User user = new
User(id, nama, nowa, email,
password);

```

```

databaseReference.child(id).setValue(user);

namalengkap.setText("");

whatsapp.setText("");

editTextEmail.setText("");

editTextPassword.setText("");

Toast.makeText(this,
"Pendaftaran Akun Berhasil",
Toast.LENGTH_LONG).show();
    } else {

Toast.makeText(this, "Please
enter a name",
Toast.LENGTH_LONG).show();
    }
}

private void setupSlider()
{

sliderView.setDurationScroll(800);
    List<Fragment>
fragments = new ArrayList<>();

fragments.add(FragmentSlider.newInstance("http://www.menucool.com/slider/prod/image-slider-1.jpg"));

fragments.add(FragmentSlider.newInstance("http://www.menucool.com/slider/prod/image-slider-2.jpg"));

fragments.add(FragmentSlider.newInstance("http://www.menucool.com/slider/prod/image-slider-3.jpg"));

fragments.add(FragmentSlider.newInstance("http://www.menucool.com/slider/prod/image-slider-4.jpg"));

mAdapter = new
SliderPagerAdapter(getSupportFragmentManager(), fragments);

sliderView.setAdapter(mAdapter);

mIndicator = new
SliderIndicator(this,
mLinearLayout, sliderView,

```

```

R.drawable.indicator_circle);

mIndicator.setPageCount(fragments.size());
    mIndicator.show();
}
}

```

Avanza.Java

```

public class AvanzaLayout
extends AppCompatActivity {
    FirebaseAuth
currentFirebaseUser =
FirebaseAuth.getInstance().getCurrentUser();
    String uid =
currentFirebaseUser.getUid();

@Override
protected void onCreate(
Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.avanza_layout);

getSupportActionBar().setTitle("Toyota Avanza");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

Button btn = (Button)
findViewById(R.id.formulir);

btn.setOnClickListener(new
View.OnClickListener() {
@Override
public void
onClick(View v) {
Bundle bundle
= new Bundle();

bundle.putString("uid", uid);

bundle.putString("mobil",
"Toyota Avanza");
Intent intent
= new
Intent(AvanzaLayout.this,
formulirlayout.class);

intent.putExtras(bundle);

```

```

startActivity(intent);
        }
    });
}
}

```

Formulir.Java

```

public class formulirlayout
extends AppCompatActivity {
    List<Booking> booking;
    EditText namalengkap,
lokasil, whatsapp, email ;
    TextView tglpesan,
tglselesai, waktu1, status,
Stok;
    Spinner jeniskendaraan,
spinnerkota;

    FirebaseAuth mAuth;
    DatabaseReference
databaseReference,
databaseReference1,
databaseReference2,
databaseReference3;
    private Button formulir;
    private TextView
startDateDisplay;
    private TextView
endDateDisplay;
    private Calendar
startDate;
    private Calendar endDate;

    static final int
DATE_DIALOG_ID = 0;

    private TextView
activeDateDisplay;
    private Calendar
activeDate;

    private SliderPagerAdapter
mAdapter;
    private SliderIndicator
mIndicator;

    private SliderView
sliderView;
    private LinearLayout
mLinearLayout;
    List<Driver> bookings;

    FirebaseDatabase database
=
FirebaseDatabase.getInstance()

```

```

;
    DatabaseReference mref =
database.getReference("Armada"
);
    String Text;

    TextView waktu;
    @Override
    protected void onCreate
(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.formul
irlayout);
    bookings = new
ArrayList<>();
    databaseReference =
FirebaseDatabase.getInstance()
.getReference("Booking");
    databaseReference3 =
FirebaseDatabase.getInstance()
.getReference("Status");

    formulir = (Button)
findViewById(R.id.formulir);

    startDateDisplay =
(TextView)
findViewById(R.id.tglpesan);

    startDate =
Calendar.getInstance();

    startDateDisplay.setOnClickListener(new
View.OnClickListener() {
        public void
onClick(View v) {
showDateDialog(startDateDispla
y, startDate);
        }
    });

    endDateDisplay =
(TextView)
findViewById(R.id.tglselesai);

    endDate =
Calendar.getInstance();

    endDateDisplay.setOnClickListener(new
View.OnClickListener() {

```

```

        public void
onClick(View v) {

showDateDialog(endDateDisplay,
endDate);
    }
    });

    sliderView =
(SliderView)
findViewById(R.id.sliderView);
    mLinearLayout =
(LinearLayout)
findViewById(R.id.pagesContain
er);
    setupSlider();

    waktu = (TextView)
findViewById(R.id.waktu);

getSupportActionBar().setTitle
("Formulir Booking");

    Button btn = (Button)
findViewById(R.id.ceklokasi);

btn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View v) {

        Intent i1 =
new
Intent(getApplicationContext()
,MapsActivity.class);

startActivity(i1);
    }
    });

    AutoCompleteTextView
textView =
findViewById(R.id.lokasi);

    Intent lokasi =
getIntent();
    String message =
lokasi.getStringExtra("searchS
tring");

textView.setText(message);

    namalengkap =
(EditText)
findViewById(R.id.namalengkap)
;
    spinnerkota =
(Spinner)
findViewById(R.id.spinnerkota)
;
    tglpesan = (TextView)
findViewById(R.id.tglpesan);
    tglselesai =
(TextView)
findViewById(R.id.tglselesai);
    jeniskendaraan =
(Spinner)
findViewById(R.id.jniskendaraa
n);
    waktul = (TextView)
findViewById(R.id.waktu);
    lokasil = (EditText)
findViewById(R.id.lokasi);
    whatsapp = (EditText)
findViewById(R.id.whatsapp);
    email = (EditText)
findViewById(R.id.email);
    status = (TextView)
findViewById(R.id.statuskendar
aan);
    Stok = (TextView)
findViewById(R.id.stok);

formulir.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View v) {
        addBooking();
    }
    });

    Bundle bundle =
getIntent().getExtras();

    FirebaseUser
currentFirebaseUser =
FirebaseAuth.getInstance().get
CurrentUser();
    String mail =
currentFirebaseUser.getEmail();
;
    email.setText(mail);

    databaseReferencel =
FirebaseDatabase.getInstance()
.getReference("Armada").child(
getIntent().getStringExtra("mo
bil"));

```

```

databaseReference1.addValueEventListener(new
onChangeListener(new
ValueEventListener() {
    @Override
        public void
onDataChange(DataSnapshot
dataSnapshot) {
            Armada aramda
=
dataSnapshot.getValue(Armada.class);

status.setText(aramda.getDeskripsi());

Stok.setText(aramda.getStok());
;
        }

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

    databaseReference2 =
FirebaseDatabase.getInstance()
.getReference("User").child(getIntent().getStringExtra("uid"));

databaseReference2.addValueEventListener(new
onChangeListener(new
ValueEventListener() {
    @Override
        public void
onDataChange(DataSnapshot
dataSnapshot) {
            User user =
dataSnapshot.getValue(User.class);

namalengkap.setText(user.getUserName());
        }

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

mref.addValueEventListener(new
ValueEventListener() {
        @Override
            public void
onDataChange(final
DataSnapshot dataSnapshot) {

                List<String> areas = new
ArrayList<String>();
                final
List<String> stok = new
ArrayList<String>();
                final
List<String> deskripsi = new
ArrayList<String>();

                for
(DataSnapshot areaSnapshot:
dataSnapshot.getChildren()) {
                    String
areaName =
areaSnapshot.getKey();
                    String
datastok = (String)
areaSnapshot.child("stok").getValue();
                    String
datadeskripsi = (String)
areaSnapshot.child("deskripsi").getValue();
                    deskripsi.add(datadeskripsi);
                    stok.add(datastok);
                    areas.add(areaName);
                }

                Spinner
areaSpinner = (Spinner)
findViewById(R.id.jniskendaraan);

                ArrayAdapter<String>
areasAdapter = new
ArrayAdapter<String>(formulirlayout.this,
android.R.layout.simple_spinner_item, areas);

                areasAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

                areaSpinner.setAdapter(areasAdapter);

                areaSpinner.setOnItemClickListener(new

```

```

AdapterView.OnItemSelectedListener() {
    @Override
    public
    void
    onItemSelected(AdapterView<?>
parent, View view, int
position, long id) {
        Text =
parent.getItemAtPosition(position).toString();

Stok.setText(stok.get(position));

status.setText(deskripsi.get(position));
    }
    @Override
    public
    void
    onNothingSelected(AdapterView<?> parent) {
    }
    }
    @Override
    public void
    onCancelled(DatabaseError
databaseError) {
    }
    });
}

    public void setTime(View
view){
        Calendar calendar =
Calendar.getInstance();
        int hour =
calendar.get(Calendar.HOUR);
        int minute =
calendar.get(Calendar.MINUTE);
        TimePickerDialog
timePickerDialog;
        timePickerDialog= new
TimePickerDialog(formulirlayout
t.this, new
TimePickerDialog.OnTimeSetList
ener() {
            @Override
            public void
            onTimeSet(TimePicker view, int
hourOfDay, int minute) {
                waktu.setText(hourOfDay+": "+mi
nute+" WIB");
            }
        }, hour,minute, true);

        timePickerDialog.show();
    }
    @Override
    public boolean
    onCreateOptionsMenu(Menu menu)
    {
        getMenuInflater().inflate(R.me
nu.main, menu);
        return true;
    }
    @Override
    public boolean
    onOptionsItemSelected(MenuItem
item) {
        int id =
item.getItemId();

        if (id ==
R.id.action_settings) {
            startActivity(new
Intent(Settings.ACTION_LOCATIO
N_SOURCE_SETTINGS));
        }

        return
super.onOptionsItemSelected(it
em);
    }

    private void
updateDisplay(Textview
dateDisplay, Calendar date) {
        dateDisplay.setText(
new
StringBuilder()
.append(date.getDisplayName(Ca
lendar.DAY_OF_WEEK,
Calendar.LONG,
Locale.getDefault()))
.append(" ")
.append(date.get(Calendar.DAY_
OF_MONTH)).append(" ")
.append(date.getDisplayName(Ca
lendar.MONTH, Calendar.LONG,
Locale.getDefault()))
.append(" ")

```

```

")
.append(date.get(Calendar.YEAR
)).append(" ");
}

    public void
showDateDialog(Text<view
dateDisplay, Calendar date) {
    activeDateDisplay =
dateDisplay;
    activeDate = date;

showDialog(DATE_DIALOG_ID);
}

    private
DatePickerDialog.OnDateSetList
ener dateSetListener = new
DatePickerDialog.OnDateSetList
ener() {
    @Override
    public void
onDateSet(DatePicker view, int
year, int monthOfYear, int
dayOfMonth) {
        activeDate =
Calendar.getInstance();

activeDate.set(Calendar.YEAR,
year);

activeDate.set(Calendar.MONTH,
monthOfYear);

activeDate.set(Calendar.DAY_OF
_MONTH, dayOfMonth);

updateDisplay(activeDateDispla
y, activeDate);

unregisterDateDisplay();
    }
};

    private void
unregisterDateDisplay() {
    activeDateDisplay =
null;
    activeDate = null;
}

    @Override
    protected Dialog
onCreateDialog(int id) {
    switch (id) {
        case
DATE_DIALOG_ID:
            return new
DatePickerDialog(this,
dateSetListener,
activeDate.get(Calendar.YEAR),
activeDate.get(Calendar.MONTH)
,
activeDate.get(Calendar.DAY_OF
_MONTH);
            }
            return null;
        }

    @Override
    protected void
onPrepareDialog(int id, Dialog
dialog) {
        super.onPrepareDialog(id,
dialog);
        switch (id) {
            case
DATE_DIALOG_ID:
                ((DatePickerDialog)
dialog).updateDate(activeDate.
get(Calendar.YEAR),
activeDate.get(Calendar.MONTH)
,
activeDate.get(Calendar.DAY_OF
_MONTH);
                break;
            }
        }

        private void addBooking()
{
            String nama =
namalengkap.getText().toString
().trim();
            String tanggalpesan =
tglpesan.getText().toString().
trim();
            String tanggalselesai
=
tglselesai.getText().toString(
).trim();
            String waktujpt =
waktul.getText().toString().tr
im();
            String lokasijpt =
lokasil.getText().toString().t
rim();
            String nowa =
whatsapp.getText().toString().
trim();
            String email=
email.getText().toString().tri
m();

```

```

        if (nama.isEmpty()) {
            namalengkap.setError("Masukan Nama Anda");

            namalengkap.requestFocus();
            return;
        }

        if (lokasijpt.isEmpty()) {
            lokasi1.setError("Masukan Lokasi Penjemputan");

            lokasi1.requestFocus();
            return;
        }

        if (tanggalpesan.isEmpty()) {
            tglpesan.setError("Masukan Tanggal Pesan");

            tglpesan.requestFocus();
            return;
        }

        if (tanggalselesai.isEmpty()) {
            tglselesai.setError("Masukan Tanggal Selesai");

            tglselesai.requestFocus();
            return;
        }

        if (waktujpt.isEmpty()) {
            waktu1.setError("Masukan Waktu Penjemputan");

            waktu1.requestFocus();
            return;
        }

        if (nowa.isEmpty()) {
            whatsapp.setError("Masukan Nomor / Whatsapp");

            whatsapp.requestFocus();
            return;
        }

        if (email1.isEmpty()) {
            email.setError("Masukan Email");

            email.requestFocus();
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email1).matches()) {
            email.setError("Mohon Masukan Email Anda Yang Benar");

            email.requestFocus();
            return;
        }

        if (!TextUtils.isEmpty(nama)) {
            String id =
                databaseReference.push().getKey();

            Booking booking =
                new Booking();

            booking.setBookingId(id);

            booking.setBookingEmail(email1);

            booking.setBookingKendaraan(jenisKendaraan.getSelectedItem().toString());

            booking.setBookingLokasi(lokasijpt);

            booking.setBookingWaktu(waktujpt);

            booking.setBookingTglpesan(tanggalpesan);

            booking.setBookingTglselesai(tanggalselesai);

            booking.setBookingNowa(nowa);

            booking.setBookingNama(nama);

            booking.setBookingTujuan(spinnerkota.getSelectedItem().toString());

```

```

databaseReference.child(id).setValue(booking);

        Status status =
new Status();

status.setBookingId(id);

status.setUserEmail(email1);

status.setDriverId("");

status.setStatuspembayaran("");
;

databaseReference3.child(id).setValue(status);

namalengkap.setText("");

tglpesan.setText("");

tglselesai.setText("");

waktu1.setText("");

lokasil.setText("");

whatsapp.setText("");
        email.setText("");

Toast.makeText(this,
"Pembukaan Armada Berhasil",
Toast.LENGTH_LONG).show();
        startActivity(new
Intent(formulirlayout.this,
MainActivity.class));
        } else {
Toast.makeText(this, "Mohon Masukan Nama Anda",
Toast.LENGTH_LONG).show();
        }
        }

        private void setupSlider()
{

sliderView.setDurationScroll(800);
        List<Fragment>
fragments = new ArrayList<>();

fragments.add(FragmentSlider.newInstance("http://www.diooda.com/slider/prod/image-slider-1.jpg"));

fragments.add(FragmentSlider.newInstance("http://www.diooda.com/slider/prod/image-slider-2.jpg"));

        mAdapter = new
SliderPagerAdapter(getSupportFragmentManager(), fragments);

sliderView.setAdapter(mAdapter);

        mIndicator = new
SliderIndicator(this,
mLinearLayout, sliderView,
R.drawable.indicator_circle);

mIndicator.setPageCount(fragments.size());
        mIndicator.show();
        }

        private void updatedata(){
        String stokdata =
Stok.getText().toString();
        String deskripsi =
status.getText().toString();

if(!TextUtils.isEmpty(stokdata)){

mref.child(Text).child("stok")
        .setValue(stokdata);

mref.child(Text).child("deskripsi")
        .setValue(deskripsi);

        Toast.makeText(this, "Berhasil Update Data",
        Toast.LENGTH_SHORT).show();
        }
        else{

        Toast.makeText(this, "Nilai tidak bisa kosong",
        Toast.LENGTH_SHORT).show();
        }
        }

        @Override
        protected void onStart() {
        super.onStart();
        }
}

```