

SKRIPSI
IMPLEMENTASI FIREBASE UNTUK PEMESANAN SERVIS MOTOR
BERBASIS ANDROID
STUDI KASUS DI KOTA YOGYAKARTA



AGUS TRI WAHYU

165410014

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AKAKOM
YOGYAKARTA
2020

SKRIPSI
IMPLEMENTASI FIREBASE UNTUK PEMESANAN SERVIS MOTOR
BERBASIS ANDROID
STUDI KASUS DI KOTA YOGYAKARTA

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

Studi Teknik Informatika
Sekolah Tinggi Manajemen Informatika dan Komputer
Akakom Yogyakarta

Disusun Oleh:

AGUS TRI WAHYU

Nomor Mahasiswa : 165410014

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AKAKOM YOGYAKARTA

2020

HALAMAN PERSETUJUAN

Judul : Implementasi Firebase Untuk Pemesanan Servis Motor
Berbasis Android Studi Kasus di Kota Yogyakarta

Nama : Agus Tri Wahyu

No. Mhs : 165410014

Jurusan : Teknik Informatika

Jenjang : Strata Satu (S1)

Semester : Ganjil (2020/2021)

Telah memenuhi syarat dan disetujui untuk diseminarkan di hadapan dosen
penguji tugas akhir

Yogyakarta, Desember 2020

Mengetahui

Dosen Pembimbing



(M.Guntara, Ir., M.T.)

HALAMAN PENGESAHAN

SKRIPSI

Implementasi Firebase Untuk Pemesanan Servis Motor Berbasis Android
Studi Kasus di Kota Yogyakarta

Telah dipertahankan di depan Dewan Penguji Skripsi dan dinyatakan diterima
untuk memenuhi syarat guna memperoleh Gelar Sarjana Komputer
STMIK Akakom Yogyakarta.

Yogyakarta, ..11.. Desember..... 2020

Mengesahkan

Dewan Penguji

1. M.Guntara, Ir., M.T.
2. Edi Iskandar, ST, M.Cs.

Tanda Tangan



Mengetahui,

Ketua Program Studi Teknik
Informatika

11 DEC 2020




Dini Fakta Sari, S.T., M.T

HALAMAN PERSEMBAHAN

Segala Puji Syukur kehadiran Tuhan yang Maha Esa, atas izin-Nya sehingga penulis dapat menyelesaikan naskah skripsi ini. Dengan bangga, saya persembahkan karya ini kepada:

1. Bapak Maryoto dan Ibu Supami tercinta yang tiada pernah hentinya selama ini memberikan do'a dan dorongan terimakasih atas semua perjuangan yang diberikan kepada anakmu, kerja keras kedua orang tua selalu saya jadikan motivasi, serta selalu memberikan dukungan penuh.
2. Saudara dan kerabat keluarga yang senantiasa mendukung penyelesaian naskah skripsi ini.
3. Kepada Bapak M.Guntara,Ir., M.T. selaku dosen pembimbing yang telah memberikan wejangan, arahan, waktu, dan ilmunya kepada saya dalam menyelesaikan karya ilmiah ini.
4. Para dosen penguji yang telah memberi masukan dan arahan demi kebaikan naskah ini.
5. Teman-teman yang terkumpul dalam grup PPT~Para Programmer Tjupu yang menjadi teman berjuang bersama dari awal semester hingga akhir semester.
6. Terimakasih kepada teman-teman seperjuangan di STMIK Akakom yang tidak dapat saya sebutkan satu-persatu yang telah membantu dan mensupport hingga dapat terselesainya karya ilmiah ini, terimakasih untuk kebersamaannya selama ini.

MOTTO

“Bukan saya yang hebat tetapi do’a bapak dan ibu saya yang hebat”

“Ketakutan adalah ilusi yang dibuat oleh pikiranmu sendiri, Ketakutan memang fitrah, tapi mengalahkan rasa takut adalah pilihan”

INTISARI

Kendaraan bermotor merupakan alat yang paling dibutuhkan sebagai media transportasi. Pesatnya kemajuan zaman, membuat sepeda motor sangat dibutuhkan sebagai media transportasi. Faktor yang dapat mengakibatkan masalah dalam berkendara adalah mendapati ban kendaraan bocor, mesin mati, servis motor. Smartphone saat ini telah menjadi sebuah kebutuhan sebagai sarana penyebaran informasi, dan beragam fungsionalitas lainnya yang berguna bagi kehidupan manusia

Aplikasi Servis Motor Online yang dibuat akan mempermudah dan membantu para pengguna sepeda motor dalam menemukan jasa servis terdekat dan sebagai sarana untuk membantu para pelaku usaha bengkel dalam menemukan pelanggan yang membutuhkan jasanya. Aplikasi dibuat dengan menggunakan bahasa pemrograman Java dengan memanfaatkan teknologi dan/atau API seperti Firebase, *Google Maps API*.

Aplikasi ini dapat digunakan pada platform Android dan bisa digunakan oleh siapapun yang memiliki android. Dengan adanya aplikasi ini, pengguna dan jasa servis menjadi lebih mudah dan terbantu dalam mempertemukan pengguna dan jasa servis, serta pengguna juga menjadi lebih hemat waktu dan tidak perlu antri ke bengkel, dan pengguna tidak repot menuju bengkel. Untuk kedepannya diharapkan ditambahkan fitur chat, fitur riwayat, dan perbaikan pada UI/UX agar tampilan aplikasi menjadi lebih menarik dan atraktif.

Kata kunci : *Android Studio, Firebase, Google Maps API, Java, Sepeda motor*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur bagi Allah SWT yang telah melimpahkan rahmat, karunia dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Firebase Untuk Pemesanan Aplikasi Servis Motor Berbasis Android”. Skripsi ini disusun untuk menyelesaikan Studi Jenjang Strata 1 (S1) Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.

Penulis menyadari sepenuhnya, telah mendapatkan banyak dukungan, bimbingan dan dorongan dari berbagai pihak yang telah menyumbangkan pikiran, waktu, tenaga dan sebagainya. Oleh karena itu, pada kesempatan ini dengan setulus hati penulis mengucapkan terimakasih kepada :

1. Bapak Ir. Totok Suprawoto, M.M., M.T., Selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.
2. Bapak Ir. M.Guntara, M.T. selaku dosen pembimbing Skripsi dan wakil ketua I Strata 1 Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.
3. Ibu Dini Fakta Sari, S.T., M.T selaku Ketua Departemen / Program Studi Teknik Informatika
4. Kedua orang tua beserta seluruh keluarga yang telah memberikan semangat dan dukungan berupa doa dan restu sehingga laporan skripsi ini dapat terselesaikan.

5. Seluruh dosen dan karyawan Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.
6. Teman-teman satu kelas yang turut membantu dan memberikan semangat dalam tersusunnya Laporan Skripsi ini.
7. Terhadap semuanya tidak kiranya penulis dapat membalasnya, hanya do'a serta puji syukur kepada Allah SWT, semoga memberikan balasan kebaikan kepada semuanya. Amin.

Laporan ini penulis susun sebagai salah satu syarat untuk menyelesaikan studi jenjang Strata 1 studi Teknik Informatika dan untuk memperoleh gelar Sarjana Komputer pada Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.

Penulis menyadari bahwa Laporan Skripsi ini tentu terdapat banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran sehingga dapat menjadi lebih baik lagi. Semoga Laporan Skripsi ini memberikan manfaat bagi penulis dan bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Desember 2020

Penulis

DAFTAR ISI

SKRIPSI.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSEMBAHAN	iv
MOTTO.....	v
INTISARI.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Ruang Lingkup	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II.....	6
2.1. Tinjauan Pustaka	6
2.2. Dasar Teori	8
2.2.1 Firebase.....	8
2.2.2 Android	10
2.2.3 Android Studio.....	10

2.2.4 Global Positioning System (GPS)	11
2.2.5 Google Maps API	11
BAB III	12
3.1. Data	12
3.2. Peralatan	12
3.2.1 Perangkat Lunak	12
3.2.2 Perangkat Keras	12
3.3. Prosedur Pengumpulan Data	13
3.4. Analisis Kebutuhan	13
3.4.1 Kebutuhan <i>Input</i>	14
3.4.2 Kebutuhan Proses	14
3.4.3 Kebutuhan <i>Output</i>	14
3.4.4 Hak Akses	15
3.5. Perancangan Sistem	15
3.5.1 Use case diagram	16
3.5.2 Sequence diagram	18
3.5.3 Activity diagram	24
3.5.4 Rancangan antarmuka	30
BAB IV	39
4.1 Implementasi Sistem	39
4.1.1 Halaman <i>Login</i> Pengguna	39
4.1.2 Halaman Daftar <i>Akun</i> Pengguna	42
4.1.3 Halaman <i>Akun</i> Pengguna	44
4.1.4 Halaman Bantuan Pengguna	46
4.1.5 Halaman Tanya Admin	47

4.1.6 Halaman Utama Pengguna.....	48
4.1.7 Halaman Pilih Lokasi.....	51
4.1.8 Halaman Pemesanan Pengguna	53
4.1.9 Halaman Daftar Jasa Servis	55
4.1.10 Halaman Masuk Jasa Servis	56
4.1.11 Halaman Akun Jasa Servis.....	57
4.1.12 Halaman Tanya Admin.....	58
4.1.13 Halaman Menu Utama Jasa Servis	59
4.1.14 Halaman Detail Pemesanan Jasa Servis.....	61
4.1.15 Halaman Firebase Database.....	62
BAB V.....	65
5.1 Kesimpulan.....	65
5.2 Saran.....	66
Daftar Pustaka	67
Lampiran	68

DAFTAR GAMBAR

Gambar 2. 1 <i>Arsitektur Sistem Firebase</i>	8
Gambar 2. 2 <i>Metode Menulis Data ke Firebase</i>	9
Gambar 2. 3 <i>Callback Kejadian dalam Pengambilan Data Firebase</i>	9
Gambar 3. 1 <i>Use Case Diagram Admin</i>	16
Gambar 3. 2 <i>Use Case Diagram jasa service</i>	17
Gambar 3. 3 <i>Use case diagram pengguna</i>	18
Gambar 3. 4 <i>Sequence diagram admin sistem</i>	19
Gambar 3. 5 <i>Sequence diagram jasa servis menu registrasi dan login</i>	20
Gambar 3. 6 <i>Sequence diagram jasa servis menu utama</i>	21
Gambar 3. 7 <i>Sequence diagram jasa servis menu akun</i>	21
Gambar 3. 8 <i>Sequence diagram pengguna menu registrasi dan login</i>	22
Gambar 3. 9 <i>Sequence diagram pengguna menu utama</i>	23
Gambar 3. 10 <i>Sequence diagram pengguna menu bantuan</i>	23
Gambar 3. 11 <i>Sequence diagram pengguna menu akun</i>	24
Gambar 3. 12 <i>Activity diagram admin sistem</i>	24
Gambar 3. 13 <i>Activity diagram menu login dan registrasi jasa servis</i>	25
Gambar 3. 14 <i>Activity diagram menu utama jasa servis</i>	26
Gambar 3. 15 <i>Activity diagram menu akun jasa servis</i>	27
Gambar 3. 16 <i>Activity diagram menu login dan registrasi pengguna</i>	27
Gambar 3. 17 <i>Activity diagram menu utama pengguna</i>	28
Gambar 3. 18 <i>Activity diagram menu akun atau profil pengguna</i>	29
Gambar 3. 19 <i>Activity diagram menu bantuan pengguna</i>	29
Gambar 3. 20 <i>Tampilan masuk dan pendaftaran pengguna</i>	30
Gambar 3. 21 <i>Tampilan menu utama dan navigation drawer pengguna</i>	31
Gambar 3. 22 <i>Tampilan menu akun dan chat pengguna</i>	32
Gambar 3. 23 <i>Tampilan menu tentang aplikasi dan bantuan pengguna</i>	33
Gambar 3. 24 <i>Tampilan menu pemesanan pengguna</i>	34
Gambar 3. 25 <i>Tampilan menu masuk dan daftar jasa servis</i>	35
Gambar 3. 26 <i>Tampilan menu navigation drawer dan akun jasa servis</i>	36
Gambar 3. 27 <i>Tampilan menu tanya admin dan info jasa servis</i>	37

Gambar 3. 28 Tampilan menu pemesanan pengguna	38
Gambar 4. 1 Halaman Masuk Pengguna.....	39
Gambar 4. 2 Kode Program Login Pengguna	41
Gambar 4. 3 Tampilan Daftar Akun Pengguna.....	42
Gambar 4. 4 Kode Program Daftar Akun Pengguna.....	43
Gambar 4. 5 Tampilan Halaman Akun Pengguna	44
Gambar 4. 6 Kode Program Akun Pengguna.....	45
Gambar 4. 7 Halaman Menu Bantuan Pengguna	46
Gambar 4. 8 Halaman Menu Tanya Admin	47
Gambar 4. 9 Tampilan Halaman Menu Utama Pengguna	48
Gambar 4. 10 Kode Program Menu Utama Pengguna.....	50
Gambar 4. 11 Tampilan Halaman Menu Pilih Lokasi	51
Gambar 4. 12 Kode Program Halaman Menu Pilih Lokasi	52
Gambar 4. 13 Tampilan Halaman Menu Pemesanan Pengguna	53
Gambar 4. 14 Kode Program Menu Pemesanan Pengguna	54
Gambar 4. 15 Tampilan Halaman Menu Daftar Jasa Servis	55
Gambar 4. 16 tampilan Halaman Menu Masuk Jasa Servis.....	56
Gambar 4. 17 Tampilan Halaman Menu Akun Jasa Servis	57
Gambar 4. 18 Tampilan Halaman Menu Tanya Admin.....	58
Gambar 4. 19 Tampilan Menu Utama Jasa Servis	59
Gambar 4. 20 Kode Program Menu Utama Jasa Servis.....	60
Gambar 4. 21 Tampilan Halaman Menu Detail Pemesanan Jasa Servis	61
Gambar 4. 22 Halaman Authentication Firebase	62
Gambar 4. 23 Halaman Realtime database	62
Gambar 4. 24 Kode Program JSON firebase	64

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka.....	7
---------------------------------	---

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kendaraan bermotor merupakan alat yang paling dibutuhkan sebagai media transportasi. Pesatnya kemajuan zaman, membuat kendaraan bermotor sangat dibutuhkan sebagai media transportasi. Kendaraan bermotor membuat efisiensi waktu dan tenaga karena diciptakan memang untuk membantu aktivitas manusia.

Faktor-faktor yang dapat mengakibatkan masalah dalam berkendara adalah mendapati ban kendaraan bocor, mesin mati, servis motor, masalah lampu, yang menyebabkan terganggunya perjalanan. Hal tersebut memengaruhi kebutuhan akan layanan kendaraan bermotor seperti bengkel *online*, sehingga dibutuhkan media informasi untuk membantu mencari lokasi pengendara motor yang mengalami kerusakan mesin. Ketidaktahuan pengendara bermotor perihal bengkel yang terdekat menjadi salah satu masalah yang sering di jumpai. Berkembangnya teknologi sekarang ini dibutuhkan media yang dapat membantu pengendara motor memperbaiki kendaraanya tanpa harus bingung mencari bengkel, sehingga jasa servis dari bengkel motor yang mendatangi pelanggannya atau bisa memanggil jasa servis untuk datang ke rumah pelanggan sehingga dapat dikerjakan di tempat.

Pada zaman modern ini, *smartphone* merupakan sebuah kebutuhan. Saat ini *smartphone* juga telah dilengkapi dengan fasilitas-fasilitas canggih yang

memungkinkan pengguna melakukan apa pun yang dibutuhkan seperti fasilitas kamera berkualitas tinggi dengan jumlah *resolusi* yang besar untuk mengambil gambar dengan hasil yang maksimal baik untuk foto maupun video, ditambah fitur lain seperti *GPS*, *Wi-Fi*, dan juga berbagi koneksi. Selain itu, saat ini *smartphone* juga telah dilengkapi dengan sistem informasi seperti *Android*, *IOS*, dan juga *Windows Phone*. Khusus untuk sistem operasi *Android* yang dikembangkan oleh *Google*, hampir semua *source code* untuk aplikasi-aplikasinya dipublikasikan atau *open source*, sehingga setiap orang dapat mengembangkan aplikasi tersebut sesuai dengan keinginan dan juga kebutuhan masing-masing pengguna.

Berdasarkan rumusan masalah tersebut perlu dibuat sebuah “Aplikasi Pemesanan Servis Motor Online”. Pengguna aplikasi ini mencari jasa servis dan dengan segera petugas jasa servis akan langsung mendatangi pengguna aplikasi yang mengalami kerusakan mesin maupun kerusakan ringan pada sepeda motornya kapanpun dan dimanapun, sehingga tidak memerlukan waktu yang lama dalam pencarian bengkel yang dapat memperbaiki kendaraannya dan tidak perlu susah payah mendorong motor menuju bengkel tersebut.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah di atas dapat dirumuskan permasalahannya adalah “Bagaimana merancang dan membangun aplikasi layanan pemesanan jasa servis motor berbasis android di kota Yogyakarta”.

1.3. Ruang Lingkup

Berdasarkan rumusan masalah di atas, dapat dijabarkan ruang lingkup penelitian adalah sebagai berikut :

1. Aplikasi menampilkan gambar, teks dan maps.
2. Membagikan lokasi pengguna memberikan informasi titik berhenti.
3. Data hanya dapat di *update* (diperbarui) oleh pengelola atau admin aplikasi, hal ini dilakukan sebagai pengaman aplikasi dan data.
4. Aplikasi ini jasa servis bisa mengetahui letak pengendara yang mengalami kerusakan, sehingga pihak jasa servis dapat menemukan letak pengendara yang mengalami kerusakan.
5. Jenis *user* yang digunakan pada aplikasi ini yaitu ada pengguna, dan jasa servis.
6. Aplikasi ini hanya melayani servis yang berada di kota Yogyakarta.

1.4. Tujuan Penelitian

Adapun tujuan penelitian adalah sebagai berikut :

1. Menghasilkan aplikasi untuk pencarian jasa servis yang berada di Kota Yogyakarta.
2. Membantu pengguna untuk memanggil jasa servis untuk ke lokasi terjadinya kerusakan atau servis ringan sepeda motor di rumah.
3. Mendekatkan antara pengguna dan jasa servis.

1.5. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini diharapkan dapat membantu pengguna untuk dapat mengakses informasi yang berkaitan dengan letak

pengguna berada , sehingga dapat mempertemukan jasa servis dalam mencari pengguna yang akan dapat memperbaiki kendaraan mereka tanpa harus mencari letak bengkel, dikarenakan dalam aplikasi ini pihak jasa servis yang akan mendatangi pengguna kendaraan yang mengalami kerusakan.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan skripsi adalah sebagai berikut :

1. Bagian awal skripsi terdiri dari halaman judul, halaman pengesahan, halaman *motto*, halaman persembahan, intisari, kata pengantar, daftar isi, daftar gambar, daftar tabel dan daftar lampiran.
2. Bagian isi skripsi terdiri dari lima bab yaitu.

BAB I PENDAHULUAN

Pada bab ini mencakup latar belakang, perumusan masalah, ruang lingkup, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini mencakup tinjauan pustaka dan dasar teori. Tinjauan pustaka membahas mengenai uraian tentang kajian berbagai pustaka yang kemudian hasil dari kajian tersebut akan dihubungkan dengan masalah yang sedang diteliti dalam penyusunan skripsi. Sedangkan dasar teori menjelaskan definisi – definisi dari yang terkait dengan pembuatan naskah skripsi.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai sistem, penjelasan analisis kebutuhan sistem, perancangan sistem.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini mencakup implementasi dan uji coba sistem serta pembahasan pada bagian implementasi dan uji coba sistem. Pada bagian ini juga menguraikan tentang implementasi sistem yang dianggap penting atau inti dari penelitian yang sesuai dengan rancangan dan berdasarkan komponen-komponen yang dipakai.

BAB V PENUTUP

Pada bab ini berisi tentang kesimpulan serta menjawab permasalahan yang dihadapi sampai mengetahui keunggulan dan kekurangan sistem yang dirancang, serta hasil implementasi diikuti pengujian telah mencapai tujuan yang diinginkan dalam pembuatan aplikasi yang dibuat.

DAFTAR PUSTAKA

Pada bagian ini berisi tinjauan pustaka, dasar teori, daftar pustaka yang menjadi sumber-sumber dalam penyusunan naskah.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

“Teknologi *Global positioning System (GPS)* Untuk Pelapor dan Penjemputan Sampah Berbasis Android” oleh Saeful Bahri, Satia Suhada, dan Jamal Maulana Hudin (2019). Penelitian ini mengambil teknologi yang digunakan *GPS* yang digunakan untuk menentukan sebuah lokasi tumpukan sampah.

“Aplikasi Bengkel Online Menggunakan *Global Positioning System (Gps)* Berbasis Android Pada Cv. Rumah Otomotif” oleh Badri Zaki, dan Syahrizal Dwi Putra (2018). Hasil dari penelitian ini maka dengan adanya aplikasi bengkel online menggunakan *GPS* ini para pengendara bermotor dapat melakukan *booking* untuk mendapatkan jasa maupun produk, aplikasi bengkel ini tempat yang tepat untuk para pemilik bengkel mendapatkan pelanggan.

“Perancangan Aplikasi Mobile Bengkelku Sebagai Informasi Alamat Bengkel Resmi Sepeda Motor Di kota Yogyakarta Menggunakan *GPS* Berbasis Android” oleh Rizqi Fitriansyah Antasari, dan Kusrini”. Penelitian ini menghasilkan aplikasi pencarian bengkel resmi sepeda motor dan menentukan jalur yang akan dilalui oleh pengguna menuju bengkel, aplikasi bengkelku dibuat melalui tahap analisis kebutuhan dan analisis kelayakan.

“Perancangan Aplikasi Untuk Mendapatkan Mekanik Sepeda Motor Menggunakan *Google Maps Api*” oleh Ratna Salkiawati, M.Hadi Prayitno, dan Dadi Ulul Wilhadad (2019). Penelitian ini menghasilkan aplikasi untuk mendapatkan mekanik sepeda motor.

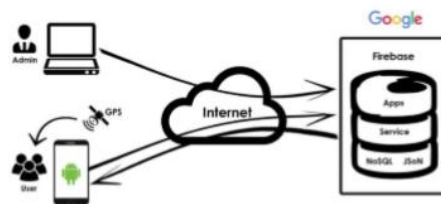
Tabel 2.1. Tinjauan Pustaka

No	Sumber	Judul	Teknologi	Hasil
1.	Saeful Bahri, Satia Suhada, dan Jamal Maulana Hudin (2019)	Teknologi <i>Global positioning System (GPS)</i> Untuk Pelapor dan Penjemputan Sampah Berbasis Android	<i>Global Positioning System (GPS), Android, Black box</i>	aplikasi ini memanfaatkan Black Box sebagai pengujiannya, <i>Global positioning system</i> pada google maps
2.	Zaki Badri dan Dwi Syahrizal (2018)	Aplikasi Bengkel Online Menggunakan <i>Global Positioning System (Gps)</i> Berbasis Android Pada Cv. Rumah Otomotif	<i>Global Positioning System (GPS), Android, waterfall</i>	Aplikasi bengkel online
3.	.Rizqi Fitriansyah Antasari dan Kusrini	Perancangan Aplikasi Mobile Bengkelku Sebagai Informasi Alamat Bengkel Resmi Sepeda Motor Dikota Yogyakarta Menggunakan GPS Berbasis Android	<i>Global Positioning System (GPS), Android, Google Maps</i>	Aplikasi “Bengkelku” berbasis android dibuat melalui tahap analisis yaitu dengan menggunakan analisis kebutuhan dan analisis kelayakan.
4.	Ratna S, M Hadi Prayitno dan Dadi Ulul W	Perancangan Aplikasi Untuk Mendapatkan Mekanik Sepeda Motor Menggunakan Google Maps Api	<i>Google Maps Api, Location Based Service</i>	Aplikasi yang mempermudah para pengendara motor untuk mendapatkan mekanik terdekat.
5.	Agus Tri Wahyu	Implementasi Firebase Untuk Pemesanan Servis Sepeda Motor Berbasis Android Studi Kasus Di Kota Yogyakarta	<i>Google Maps Api, Firebase, Android</i>	Aplikasi layanan pemesanan servis motor berbasis Android

2.2. Dasar Teori

2.2.1 Firebase

Firebase yakni model layanan yang bekerja di belakang layar dan menghubungkan aplikasi *mobile* ke *cloud storage*. *Firebase Realtime Database* adalah database yang di-host di *cloud*. Data disimpan sebagai *JSON* dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika anda membuat aplikasi lintas-*platform* dengan *SDK Android*, *iOS*, dan *JavaScript*, semua *klien* akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis. (firebase, 2018).



Gambar 2.1 Arsitektur Sistem Firebase

Semua data *Firebase Realtime Database* disimpan sebagai objek *JSON*. Bisa dianggap basis data sebagai *JSON tree* yang di-host di awan. Tidak seperti basis data *SQL*, tidak ada tabel atau rekaman. Ketika ditambahkan ke *JSON tree*, data akan menjadi simpul dalam struktur *JSON* yang ada. Meskipun basis data menggunakan *JSON tree*, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe *JSON* yang tersedia untuk membantu anda menulis lebih banyak kode yang bisa dipertahankan. Ada empat metode untuk menulis data ke *Firebase Realtime Database*:

Metode	Penggunaan umum
<code>setValue()</code>	Menulis atau mengganti data ke jalur yang didefinisikan, seperti <code>users/<user-id>/<username></code> .
<code>push()</code>	Tambahkan ke daftar data. Setiap kali Anda memanggil <code>push()</code> , Firebase akan menghasilkan ID unik, seperti <code>user-posts/<user-id>/<unique-post-id></code> .
<code>updateChildren()</code>	Memperbarui beberapa kunci untuk jalur yang didefinisikan tanpa mengganti semua data.
<code>runTransaction()</code>	Memperbarui data kompleks yang bisa rusak karena pembaruan bersamaan.

Gambar 2. 2 Metode Menulis Data ke *Firebase*

Untuk operasi tulis dasar, Anda bisa menggunakan `setValue()` untuk menyimpan data ke referensi yang ditetapkan, menggantikan data yang ada di jalur tersebut. Fungsi dalam pengambilan data melalui *Firebase*:

Listener	Callback kejadian	Penggunaan biasa
<code>ValueEventListener</code>	<code>onDataChange()</code>	Membaca dan mendengarkan perubahan untuk seluruh konten jalur.
<code>ChildEventListener</code>	<code>onChildAdded()</code>	Mengambil daftar item atau mendengarkan penambahan daftar item. Disarankan untuk digunakan dengan <code>onChildChanged()</code> dan <code>onChildRemoved()</code> untuk memantau perubahan daftar.
	<code>onChildChanged()</code>	Mendengarkan perubahan pada item dalam daftar. Gunakan dengan <code>onChildAdded()</code> dan <code>onChildRemoved()</code> untuk memantau perubahan daftar.
	<code>onChildRemoved()</code>	Mendengarkan item yang dibuang dari daftar. Gunakan dengan <code>onChildAdded()</code> dan <code>onChildChanged()</code> untuk memantau perubahan daftar.
	<code>onChildMoved()</code>	Gunakan dengan data diurutkan untuk mendengarkan perubahan dalam prioritas item.

Gambar 2. 3 Callback Kejadian dalam Pengambilan Data *Firebase*

Untuk menambahkan *listener* kejadian, gunakan metode `addValueEventListener()` atau `addListenerForSingleValueEvent()`. Untuk menambahkan *listener* kejadian anak, gunakan metode `addChildEventListener()`. Metode `onDataChange()` untuk membaca cuplikan statis konten pada jalur tertentu, seperti yang telah ada pada saat kejadian. Metode ini terpicu satu kali ketika *listener* terpasang dan terpicu lagi setiap kali terjadi perubahan data, termasuk anaknya. *Callback* kejadian meneruskan cuplikan yang berisi semua data di lokasi tersebut, termasuk data anak. Jika tidak ada data, cuplikan yang dikembalikan adalah `null`. Metode `onDataChange()`

dipanggil setiap kali terjadi perubahan data pada referensi *database* yang ditetapkan, termasuk perubahan ke anaknya. (Firebase, 2018).

2.2.2 Android

Android merupakan *OS (Operating System) Mobile* yang tumbuh ditengah *OS* lainnya yang berkembang dewasa ini. *OS* lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, *OS* yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.(Hermawan, 2011 : 1).

2.2.3 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment / IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada *IntelliJ IDEA*. Selain sebagai editor kode dan fitur *developer IntelliJ* yang andal, *Android Studio* menawarkan banyak fitur yang meningkatkan produktivitas. Anda dalam membuat aplikasi *Android*, seperti: Sistem build berbasis *Gradle* yang fleksibel, *Emulator* yang cepat dan kaya fitur, lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat *Android*, terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi, Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel, *framework* dan fitur

pengujian yang lengkap, fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya, dukungan C++ dan NDK, dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.

2.2.4 Global Positioning System (GPS)

Dalam penelitian Mulyadi.B,dkk yang berjudul “Aplikasi Sistem Pemesanan Jasa *Laundry (E-laundry)* Berbasis Android”, GPS (*Global Positioning System*) adalah suatu sistem *navigasi* menggunakan lebih dari 24 satelit *MEO (Medium Earth Orbit* atau *Middle Earth Orbit*) yang mengelilingi bumi sehingga penerima-penerima sinyal di permukaan bumi dapat menangkap sinyalnya. *GPS* mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan arah, dan waktu satelit mengorbit pada ketinggian 12.000 *mil* di atas bumi dan mampu mengelilingi bumi dua kali dalam 24 jam.

2.2.5 Google Maps API

Google Maps API adalah suatu *library* yang berbentuk *javascript* yang berguna untuk memodifikasi peta yang ada di *Google Maps* sesuai kebutuhan. Dengan adanya *Google Maps API* ini pengguna dapat memaksimalkan kemampuan pemetaan *Google* untuk tujuan-tujuan dan manfaat tertentu secara spesifik. (Elian, dkk. 2012:2 dalam penelitian yang dilakukan oleh Daniel Oktodeli Sihombing yang berjudul : “Perancangan Aplikasi *Web* Untuk Pencarian Lokasi Dan Rute Rumah Sakit Berbasis *Google Maps API*”).

BAB III

METODE PENELITIAN

3.1. Data

Data yang dibutuhkan dalam sistem aplikasi jasa servis motor ini dibutuhkan data berupa pengguna dan jasa servis.

3.2. Peralatan

3.2.1 Perangkat Lunak

Perangkat lunak (*software*) merupakan program – program perangkat pendukung dalam menjalankan perangkat keras. *Software* sebagai penerjemah suatu bahasa mesin (analog) atau bahasa tingkat rendah ke bahasa tingkat tinggi yang akhirnya menghasilkan suatu informasi yang dapat dikenal oleh manusia (*user*). Adapun kebutuhan perangkat lunak pada sistem ini adalah :

1. Sistem Operasi *Windows 8 (64 Bit)*
2. *Android Studio 3.5.1*
3. *Android SDK*
4. Browser (*Google Chrome*)
5. *Firebase*
6. *Java*
7. *Rational Rose*
8. *JSON*

3.2.2 Perangkat Keras

Perangkat keras (*hardware*) merupakan suatu peralatan fisik komputer yang digunakan untuk menjalankan program. Sistem perangkat keras terdiri dari unit

masukkan, unit pengolahan dan unit keluaran. Adapun perangkat keras yang digunakan dalam menjalani sistem ini adalah komputer dengan spesifikasi sebagai berikut :

1. Laptop dengan spesifikasi :

- a. *RAM 2GB*
- b. *Processor Intel Core i3-2310M CPU @ 2,1 GHz VGA Intel Graphics*
- c. *VGA Intel HD Graphic 3000*
- d. *Harddisk 500 GB*

2. Smartphone dengan spesifikasi :

- a. *OS dengan menggunakan Android versi 5.0*
- b. *RAM 2GB dan Storage 16 GB*

3.3. Prosedur Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah dengan menggunakan metode studi kepustakaan. Studi kepustakaan adalah kegiatan untuk menghimpun informasi yang relevan dengan topik atau masalah yang menjadi objek penelitian. Informasi tersebut dapat diperoleh dari buku-buku, karya ilmiah, tesis, disertasi, *ensiklopedia*, *internet*, dan sumber-sumber lain. Dengan melakukan studi kepustakaan, peneliti dapat memanfaatkan semua informasi dan pemikiran-pemikiran yang *relevan* dengan penelitiannya.

3.4. Analisis Kebutuhan

Analisis kebutuhan mencakup pekerjaan-pekerjaan penentuan kebutuhan atau kondisi yang harus dipenuhi dalam suatu produk baru atau perubahan produk, yang mempertimbangkan berbagai kebutuhan yang bersinggungan antar berbagai

pemangku kepentingan. Kebutuhan dari hasil analisis ini harus dapat dilaksanakan, diukur, diuji, terkait dengan kebutuhan bisnis yang teridentifikasi, serta didefinisikan sampai tingkat detail yang memadai untuk *desain* sistem.

3.4.1 Kebutuhan *Input*

Kebutuhan *input* yang diperlukan pada system ini yaitu :

1. Input nama jasa servis
2. Input lokasi
3. Input *username*
4. Input *password*
5. Input nama pengguna / *user*
6. Input nomor telepon
7. Input kerusakan yang terjadi pada motor

3.4.2 Kebutuhan Proses

Kebutuhan proses yang diperlukan pada sistem ini yaitu :

1. Sistem menampilkan *API maps*
2. Menampilkan pengguna yang melakukan pemesanan
3. Menampilkan lokasi pengguna

3.4.3 Kebutuhan *Output*

Kebutuhan *output* yang diperlukan pada sistem ini yaitu :

1. Menampilkan posisi pelanggan
2. Menampilkan *Maps API* dan *MapBox*
3. Jasa servis memperbaiki motor konsumen sesuai lokasi yang disesuaikan

3.4.4 Hak Akses

Jenis *user* yang memiliki hak akses pada aplikasi ini sebagai berikut :

1. Admin sistem pada aplikasi ini memiliki hak akses untuk mengelola *database* pada aplikasi, admin dapat menentukan jasa servis dan menentukan *registrasi* pengguna, kemudian admin dapat tambah, edit, hapus data dari pengguna atau jasa servis melalui *database firebase*.
2. Pengguna pada aplikasi ini memiliki hak akses untuk melakukan pemesanan pada aplikasi sehingga pengguna menentukan kerusakan yang di *input* kan dan menentukan titik lokasi berhenti kemudian pengguna dapat melakukan *chat* dengan admin, pengguna dapat melihat bantuan penggunaan aplikasi pada menu bantuan, kemudian pengguna dapat melihat profil.
3. Jasa servis yang terdapat pada aplikasi ini yaitu dapat menerima orderan, dapat melihat profil, dapat melakukan *chat* dengan admin, dan sebagai jasa layanan yang memperbaiki kerusakan yang terjadi pada pelanggannya sehingga jasa servis mendapatkan pesanan maka menuju lokasi yang telah ditentukan.

3.5. Perancangan Sistem

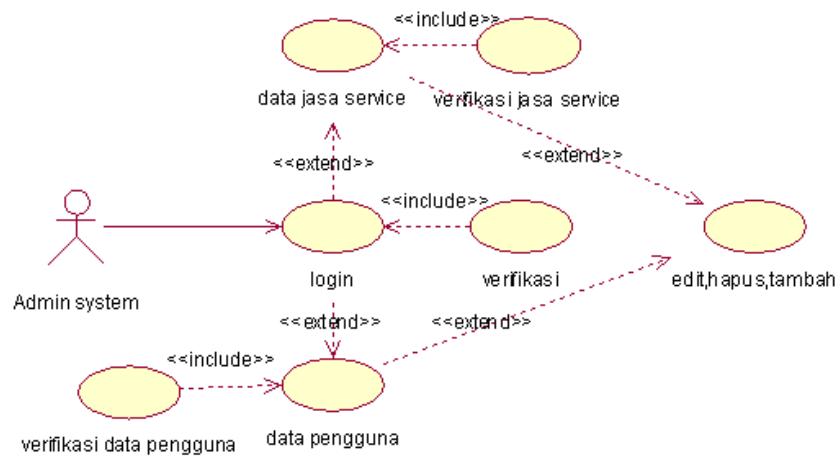
Perancangan sistem merupakan suatu tahapan setelah analisis dari siklus pengembangan sistem yang didefinisikan dari kebutuhan – kebutuhan fungsional dan persiapan untuk rancang bangun *implementasi* yang menggambarkan bagaimana suatu sistem dibentuk. Dalam perancangan ini terdiri atas rancangan

use case diagram, sequence diagram, activity diagram, relasi antar tabel dan rancangan antarmuka.

3.5.1 Use case diagram

Use Case Diagram adalah gambaran *graphical* dari beberapa atau semua *actor, use case, dan interaksi* diantaranya yang memperkenalkan suatu sistem.

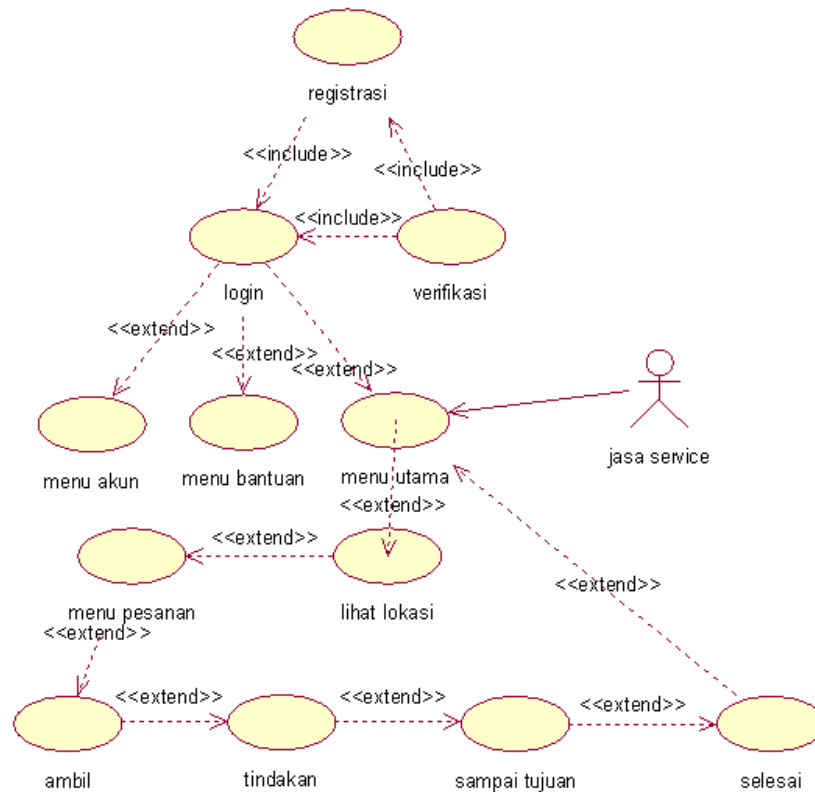
1. Use Case Diagram Admin



Gambar 3. 1 Use Case Diagram Admin

Pada *Use Case Diagram* gambar 3.1 menggambarkan bahwa admin sistem ketika melakukan perubahan data edit, hapus, maupun tambah maka admin harus login terlebih dahulu ke dalam *database firebase* sehingga admin dapat merubah data jasa servis dan data pengguna.

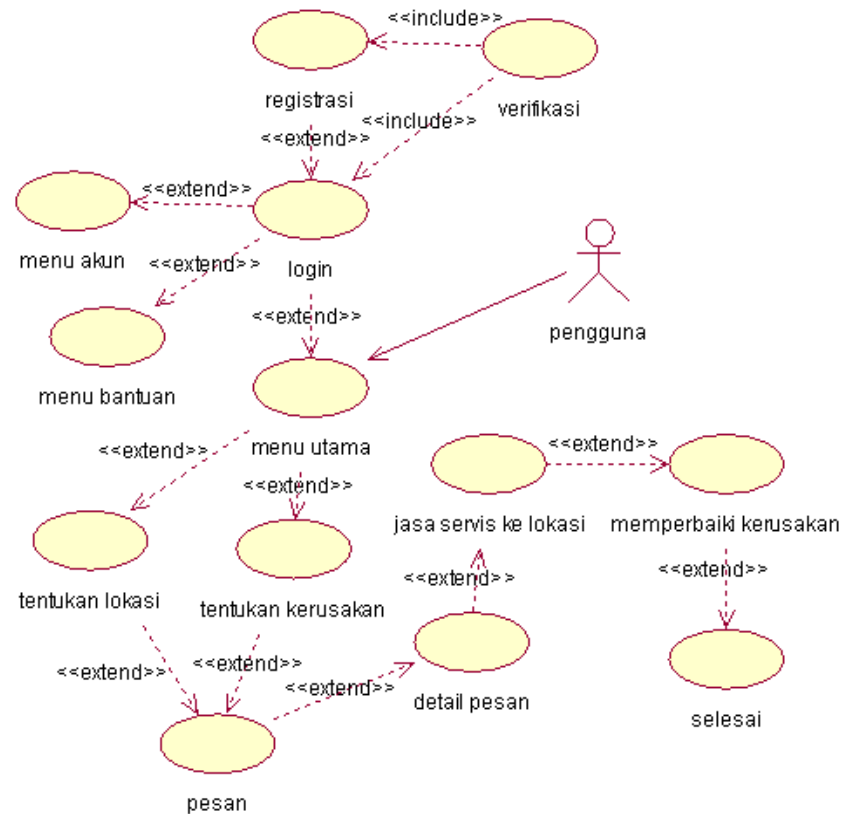
2. Use Case Diagram Jasa Service



Gambar 3. 2 Use Case Diagram jasa service

Pada *Use Case Diagram* jasa servis yang terdapat pada gambar 3.2 menjelaskan bahwa jasa servis melakukan *registrasi* terlebih dahulu sebelum melakukan *login* dan setelah itu dapat menerima pesanan dari pelanggan dan menyelesaikan pekerjaan jika sudah selesai.

3. Use Case Diagram Pengguna



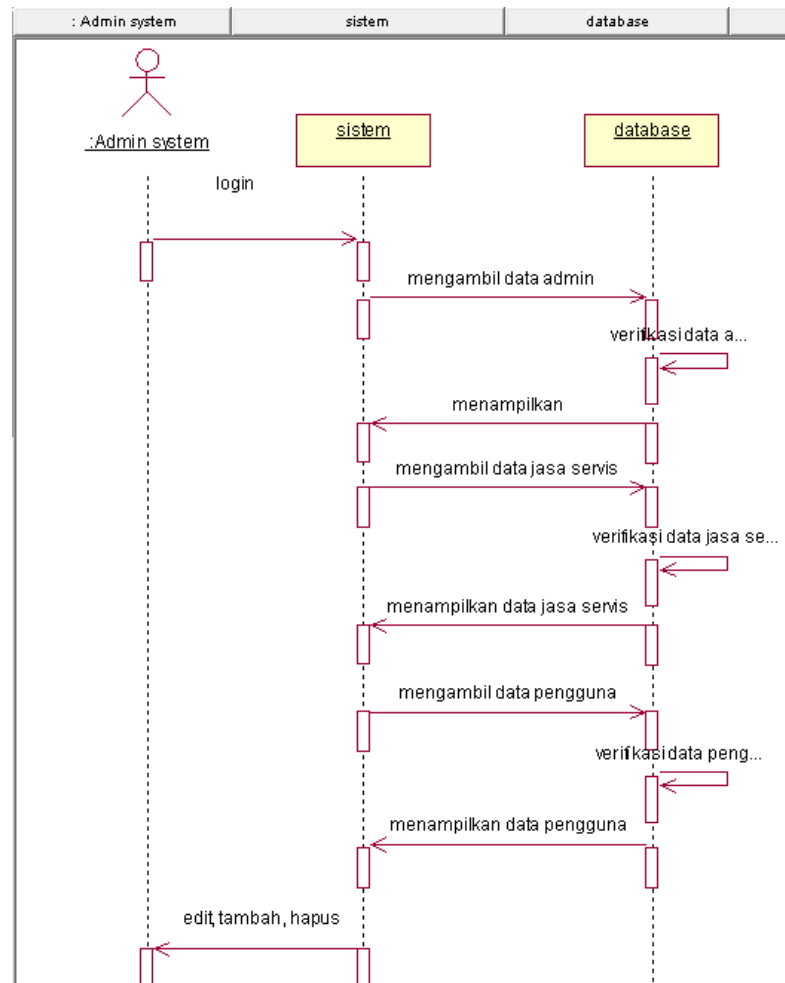
Gambar 3. 3 Use case diagram pengguna

Pada gambar 3.3 menjelaskan bahwa pengguna melakukan *registrasi* setelahnya *login* ke aplikasi dan setelah itu maka pengguna dapat ke menu utama selanjutnya dapat melakukan pesan dan selanjutnya menentukan lokasi dan kerusakan setelah itu maka jasa servis akan menuju ke lokasi yang ditentukan.

3.5.2 Sequence diagram

Sequence diagram merupakan diagram yang menggambarkan kolaborasi dinamis antara sejumlah object.

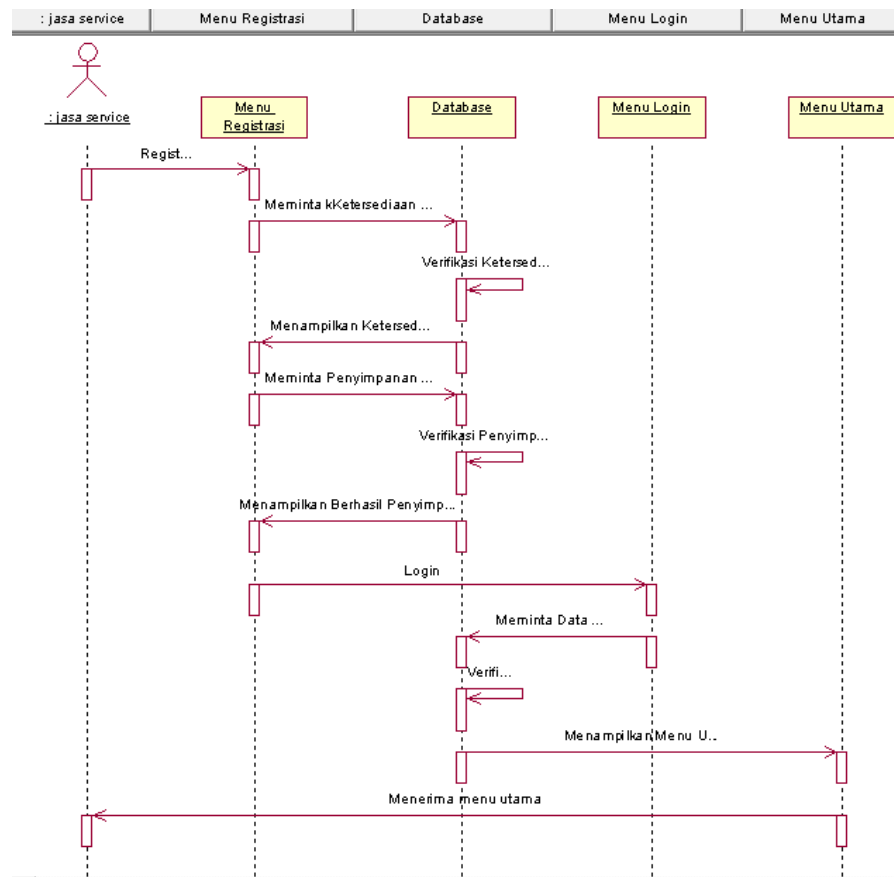
1. Sequence Diagram Admin Sistem



Gambar 3. 4 Sequence diagram admin sistem

Pada bagian *sequence diagram* gambar 3.4 pada admin sistem ini pengguna dan jasa servis dapat menggunakan aplikasi ketika *registrasi* diverifikasi oleh admin, dan admin dapat edit data, tambah data dan hapus data.

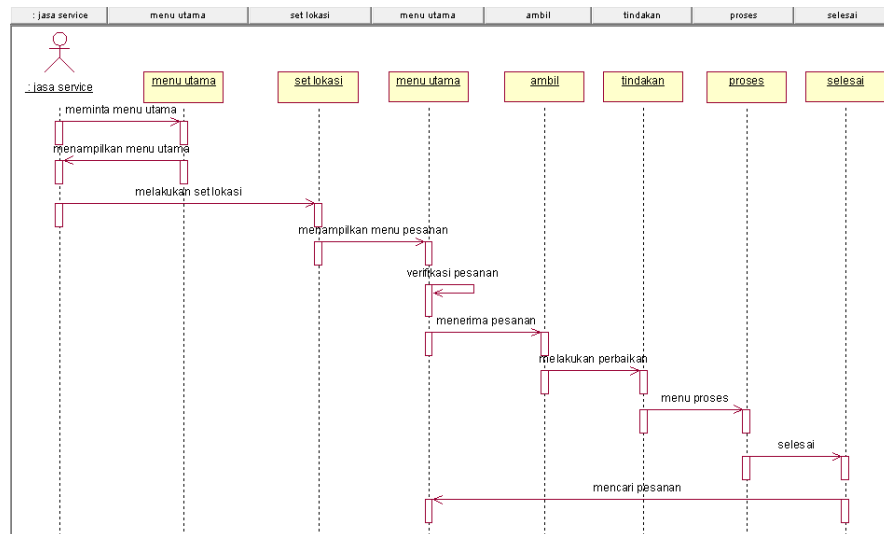
2. Sequence Diagram Registrasi dan Login Jasa Servis



Gambar 3. 5 Sequence diagram jasa servis menu registrasi dan login

Pada bagian sequence *diagram* gambar 3.5 menu *registrasi* jasa servis ini sebelum menggunakan aplikasi maka jasa servis melakukan *registrasi* setelah itu jasa servis *login* pada aplikasi.

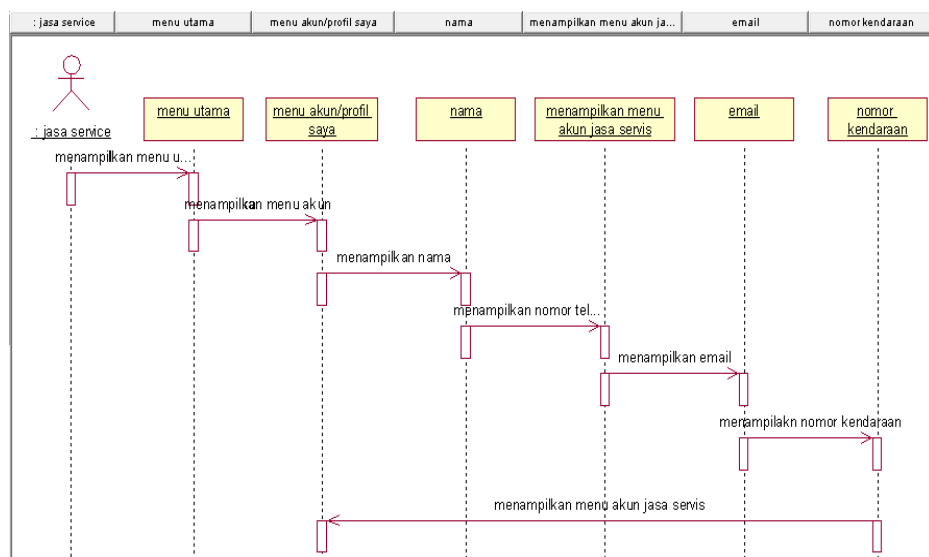
3. Sequence Diagram Jasa Servis Menu Utama



Gambar 3. 6 Sequence diagram jasa servis menu utama

Pada bagian *sequence diagram* gambar 3.6 menu utama pada jasa servis terdapat list untuk menerima pesanan dari pengguna ketika terdapat pesanan dan setelah itu maka jasa servis menuju lokasi yang ditentukan oleh pengguna.

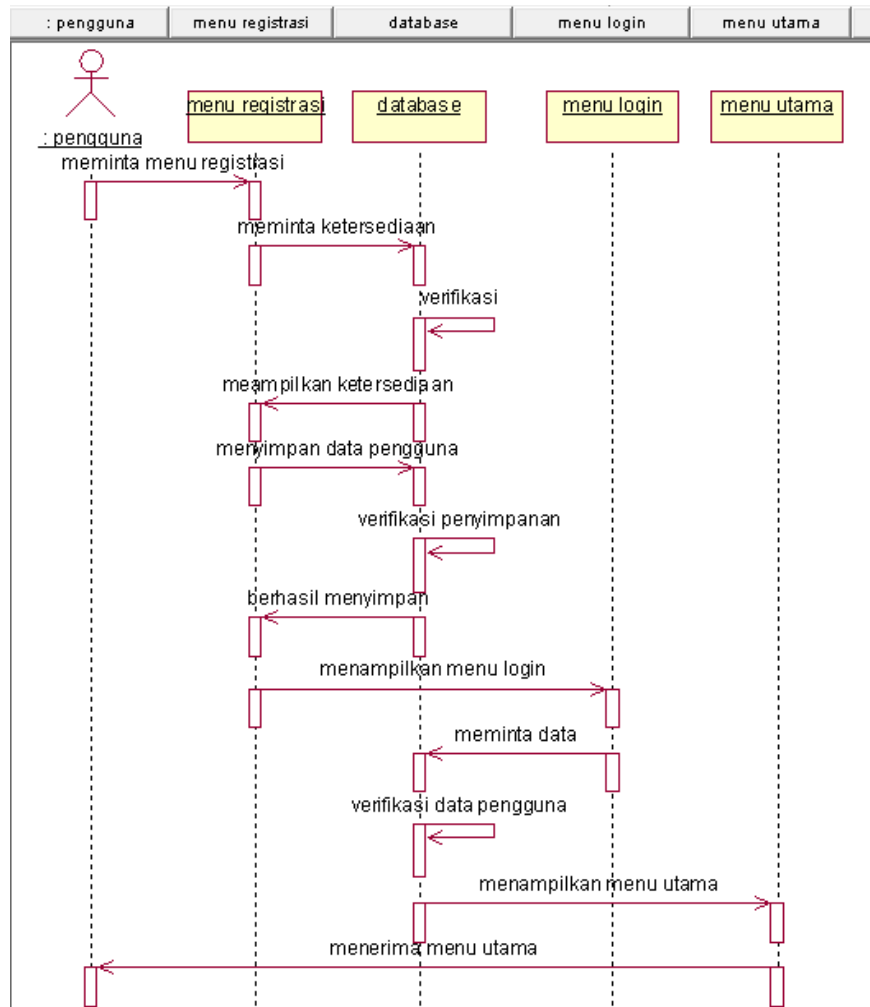
4. Sequence Diagram Menu Akun Jasa Servis



Gambar 3. 7 Sequence diagram jasa servis menu akun

Pada bagian *sequence diagram* gambar 3.7 yaitu pada menu akun jasa servis yang dapat digunakan untuk merubah data akun yang sudah terdaftar dengan data yang baru dari jasa servis.

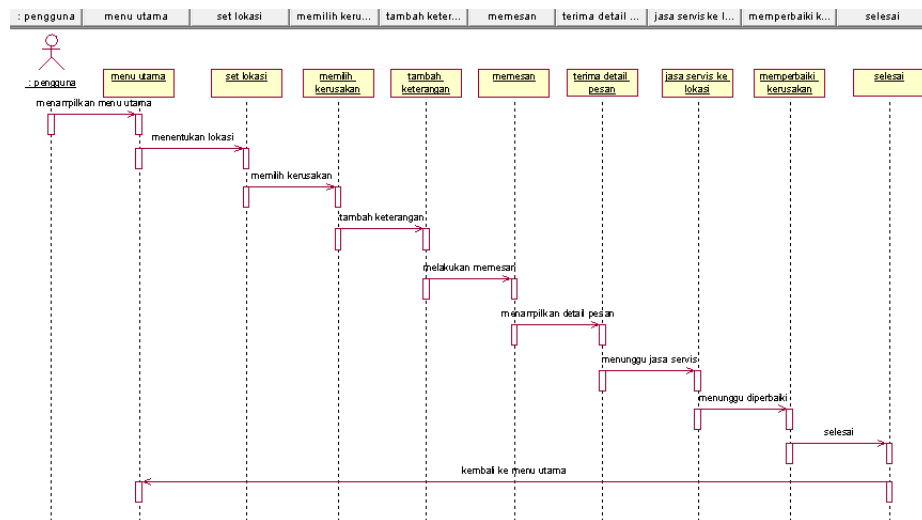
5. Sequence Diagram Registrasi dan Login Pengguna



Gambar 3. 8 Sequence diagram pengguna menu registrasi dan login

Pada gambar 3.8 *sequence diagram* menu registrasi pengguna yaitu ketika pengguna melakukan *registrasi* pada aplikasi dan setelah itu maka pengguna dapat *login* pada aplikasi setelah melakukan *registrasi* dan dapat menggunakan aplikasinya.

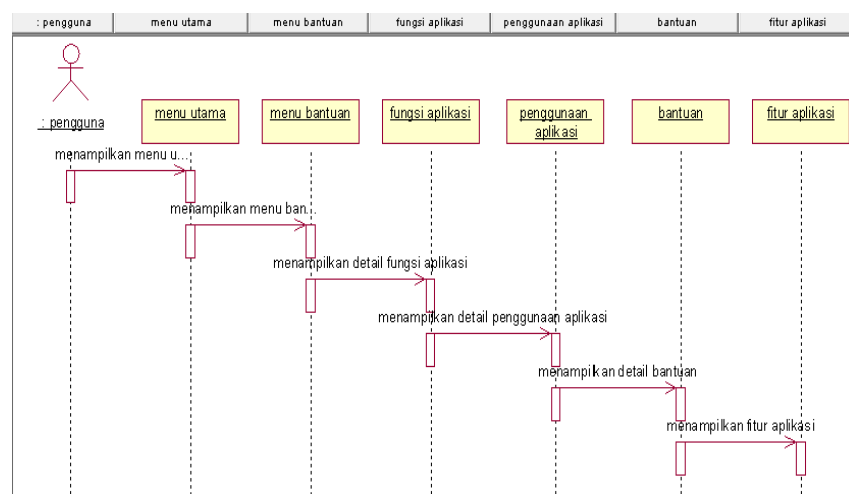
6. Sequence Diagram Menu Utama Pengguna



Gambar 3. 9 Sequence diagram pengguna menu utama

Pada gambar 3.9 menu utama *sequence diagram* pengguna dapat melakukan *order* setelah masuk ke aplikasi dalam menu utama atau menu *order* pengguna dapat menentukan lokasi kemudian pengguna dapat menentukan kerusakannya.

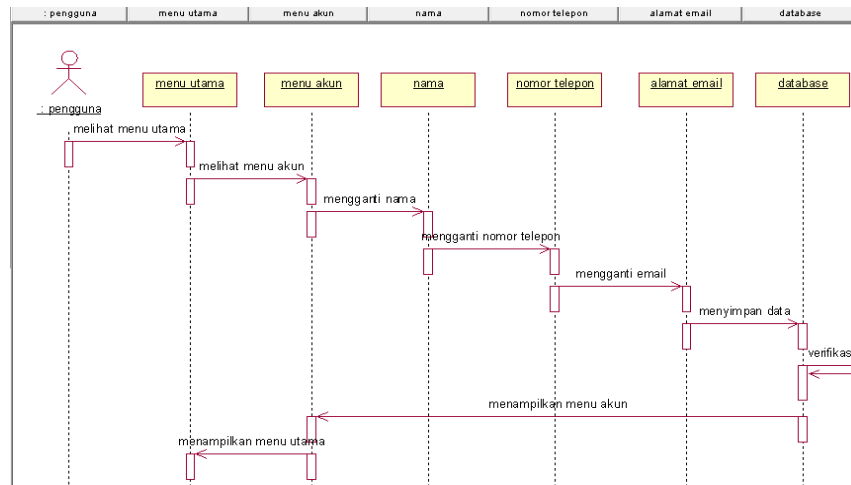
7. Sequence Diagram Menu Bantuan Pengguna



Gambar 3. 10 Sequence diagram pengguna menu bantuan

Pada gambar 3.10 bagian *sequence diagram* menu bantuan yang terdapat pada pengguna ketika mengalami masalah atau bantuan penggunaan pada aplikasi.

8. Sequence Diagram Menu Akun Pengguna

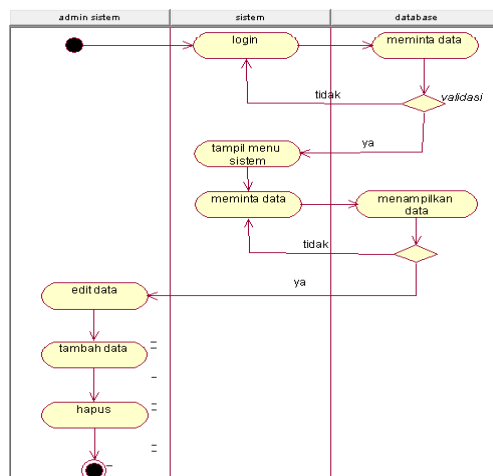


Gambar 3. 11 Sequence diagram pengguna menu akun

Pada bagian *sequence diagram* menu akun pengguna dapat merubah data atau melihat data atau profil yang terdaftar pada aplikasi tersebut.

3.5.3 Activity diagram

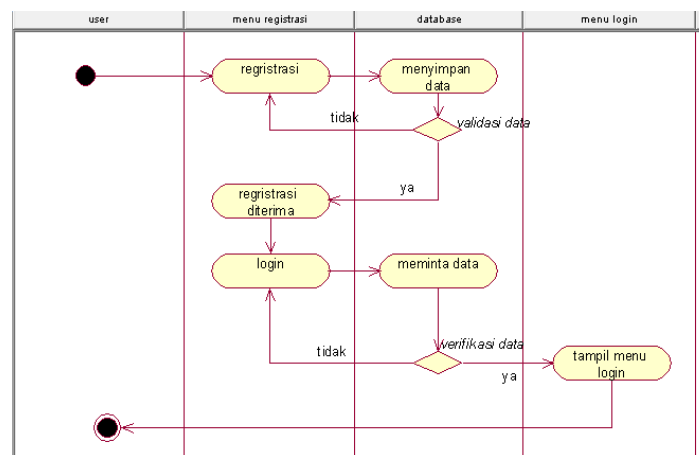
1. Activity Diagram Admin Sistem



Gambar 3. 12 Activity diagram admin sistem

Pada gambar 3.12 bagian ini admin sistem melakukan *login* untuk mengelola *database* untuk masuk maka *database* akan menampilkan menu sistem dan menampilkan data dari jasa servis dan pengguna, admin dapat menghapus, menambah, dan mengedit data.

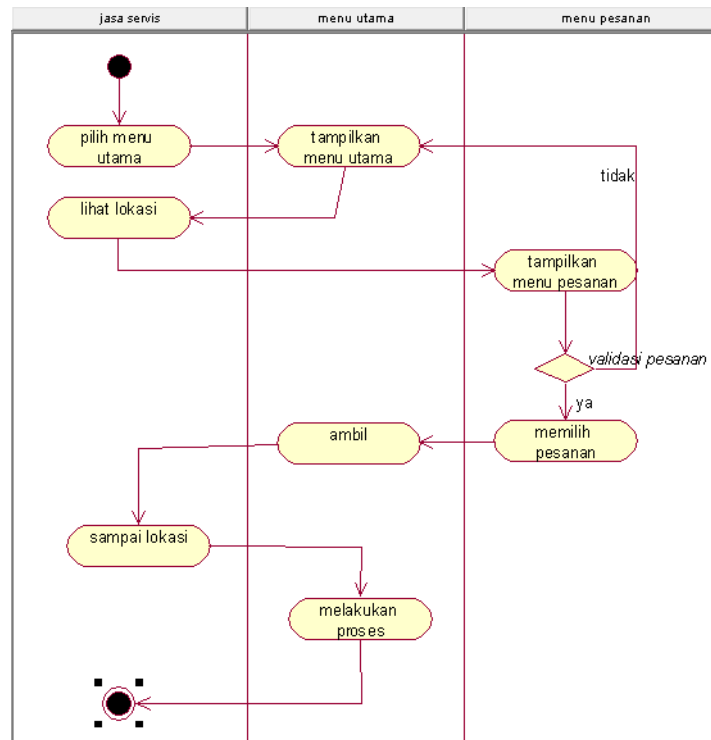
2. Activity Diagram Registrasi dan Login Jasa Servis



Gambar 3. 13 Activity diagram menu login dan registrasi jasa servis

Pada gambar 3.13 bagian *activity diagram login* jasa servis ini sebelum *login* maka jasa servis melakukan *registrasi* terlebih dahulu dan setelah menyimpan data dan diverifikasi oleh *database* setelah *registrasi* diterima maka jasa servis *login* dan meminta data ke *database* ketika data ada maka jasa servis dapat tampilan menu utama.

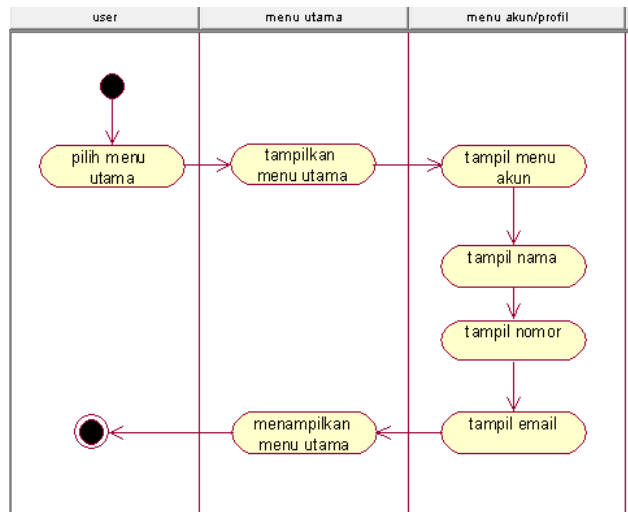
3. Activity Diagram Jasa Servis Menu Utama



Gambar 3. 14 Activity diagram menu utama jasa servis

Pada bagian menu utama gambar 3.14 jasa servis maka dilakukan *activity* jasa servis memilih menu utama dan kemudian jasa servis dapat melihat lokasi kemudian setelah menerima pesanan maka jasa servis dapat mengambilnya dan ketika tidak diambil maka akan kembali ke menu utama dan ketika diambil maka jasa servis menuju ke lokasi selanjutnya jasa servis melakukan perbaikan dan selesai.

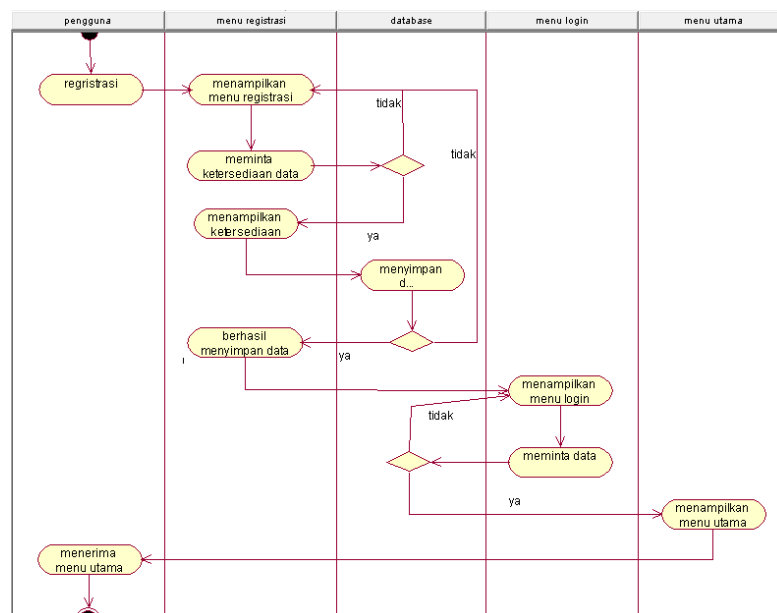
4. Activity Diagram Jasa Servis Menu Akun



Gambar 3. 15 Activity diagram menu akun jasa servis

Pada *activity diagram* gambar 3.15 yaitu menu akun pada jasa servis maka dapat melakukan *activity* melihat atau mengganti data *akun* pada menu *akun*.

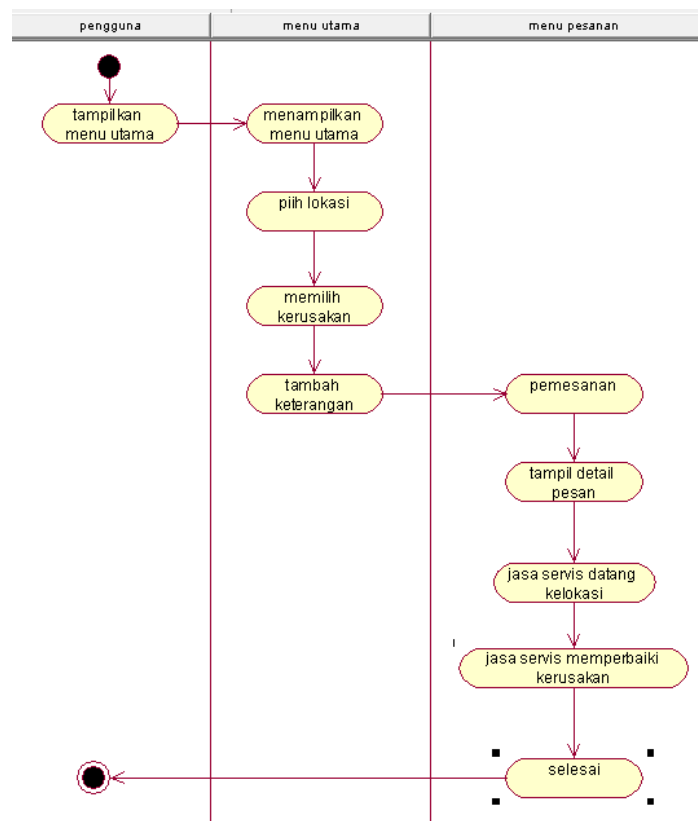
5. Activity Diagram Pengguna Menu Registrasi dan Login



Gambar 3. 16 Activity diagram menu login dan registrasi pengguna

Pada bagian gambar 3.16 *activity diagram login* pengguna ini sebelum *login* maka pengguna melakukan *registrasi* terlebih dahulu dan setelah menyimpan data dan diverifikasi oleh admin setelah *registrasi* diterima maka pengguna dapat *login*.

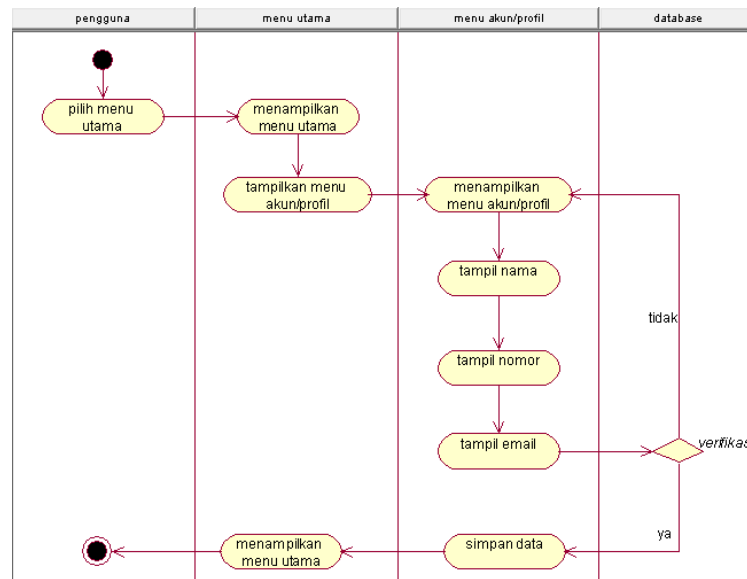
6. Activity Diagram Pengguna Menu Utama



Gambar 3. 17 Activity diagram menu utama pengguna

Pada gambar 3.17 bagian ini yang terdapat pada menu utama pengguna maka pengguna dapat melakukan pemesanan dengan *activity* yaitu pengguna menentukan lokasi selanjutnya pengguna dapat menentukan kerusakan pada motornya dan pengguna dapat menambah keterangan lebih detailnya setelah itu maka pengguna dapat memesan pada aplikasi selanjutnya pengguna menerima detail pemesanan.

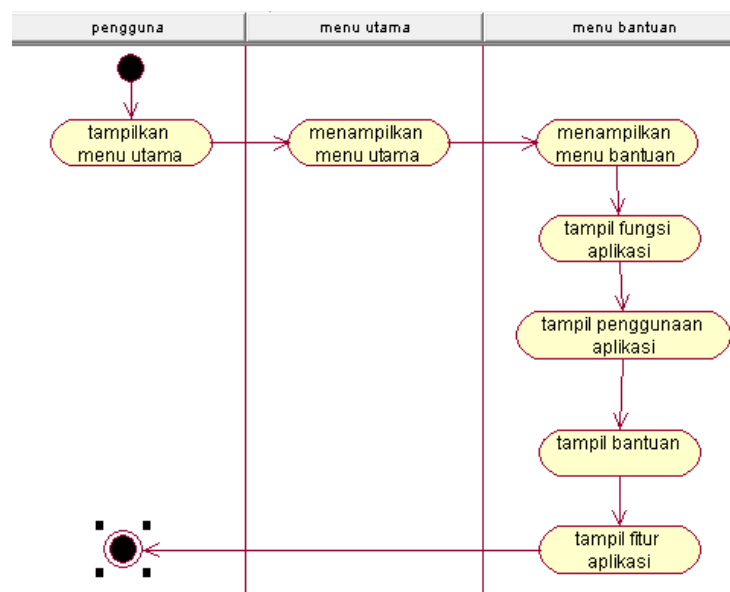
7. Activity Diagram Pengguna Menu Akun atau Profil



Gambar 3. 18 Activity diagram menu akun atau profil pengguna

Pada gambar 3.18 bagian menu *akun* pengguna dapat merubah atau melihat data yang tersimpan pada database ketika pengguna melakukan perubahan data maka terdapat *verifikasi* simpan data.

8. Activity Diagram Pengguna Menu Bantuan



Gambar 3. 19 Activity diagram menu bantuan pengguna

Pada gambar 3.19 diatas terdapat aktivitas pada menu bantuan untuk pengguna yang digunakan untuk melihat ketika terjadi masalah dan dapat digunakan sebagai petunjuk penggunaan aplikasi pada menu bantuan pengguna terdapat *list* tampil data *profil* pada menu pengguna.

3.5.4 Rancangan antarmuka

1. Menu Masuk dan Pendaftaran pengguna



Gambar 3. 20 Tampilan masuk dan pendaftaran pengguna

Pada gambar 3.20 desain rancangan antar muka yang digunakan untuk *login* pada aplikasi *user* atau pengguna sebelum melakukan masuk pengguna diharuskan pendaftaran terlebih dahulu mengisi data seperti gambar 3.20.

2. Menu Utama dan *Navigation Drawer* pengguna



Gambar 3. 21 Tampilan menu utama dan *navigation drawer* pengguna

Pada tampilan gambar 3.21 yaitu menu utama pengguna dimana pada menu tersebut pengguna melakukan pemesanan pada aplikasi dengan menentukan lokasi pengguna kemudian pengguna memilih kerusakan pada motor, pengguna juga dapat menambahkan keterangan tambah dengan lengkap sehingga memudahkan jasa servis menemukannya, pada menu *navigation drawer* pengguna dapat membukanya untuk melihat menu *profile*, *chat*, *bantuan*, *tentang aplikasi* dan *logout*.

3. Menu *Akun* dan *Chat* pengguna



Gambar 3. 22 Tampilan menu *akun* dan *chat* pengguna

Pada gambar 3.22 yaitu tampilan *akun* pengguna pada menu tersebut terdapat menu yang digunakan untuk memperbarui data pengguna, dan pada menu *chat* pengguna berfungsi untuk menanyakan atau meminta bantuan kepada admin jika terdapat kendala pada penggunaan aplikasi.

4. Menu Tentang Aplikasi dan Bantuan pengguna



Gambar 3. 23 Tampilan menu tentang aplikasi dan bantuan pengguna

Pada gambar 3.23 yaitu menu tentang aplikasi yang berfungsi untuk menampilkan informasi versi pada aplikasi pengguna, dan menu bantuan pengguna yang digunakan untuk menambahkan informasi dan cara penggunaan aplikasi Serline tersebut.

5. Menu Pemesanan Pengguna



Gambar 3. 24 Tampilan menu pemesanan pengguna

Gambar 3.24 yaitu tampilan menu detail pemesanan pengguna yang sesuai dengan apa yang di pesan pengguna dengan informasi yang terdapat pada menu pemesanan tersebut tidak dapat di klik selesai oleh pengguna jika belum ada tindakan oleh jasa servis.

6. Menu Masuk dan Pendaftaran Jasa Servis



Gambar 3. 25 Tampilan menu masuk dan daftar jasa servis

Pada gambar 3.25 desain rancangan antar muka yang digunakan untuk masuk pada aplikasi *user* atau pengguna sebelum melakukan masuk jasa servis diharuskan pendaftaran terlebih dahulu mengisi data seperti gambar 3.25 dengan menambahkan nama lengkap, nomor kendaraan, nomor telepon, *email*, dan *password*.

7. Menu *Navigation Drawer* dan Akun Jasa Servis



Gambar 3. 26 Tampilan menu *navigation drawer* dan akun jasa servis

Pada tampilan gambar 3.26 yaitu menu menu *navigation drawer* jasa servis dapat membukanya untuk melihat menu *profile*, tanya admin, info aplikasi dan keluar, dan pada tampilan menu selanjutnya yaitu menu akun jasa servis yang menampilkan informasi nama lengkap, nomor kendaraan, nomor telepon, dan email jasa servis

8. Menu Tanya Admin dan Info Aplikasi Pengguna



Gambar 3. 27 Tampilan menu tanya admin dan info jasa servis

Pada gambar 3.27 yaitu menu tanya admin ketika jasa servis membutuhkan bantuan atau mengalami kendala, selanjutnya menu info aplikasi yang berfungsi untuk menampilkan informasi versi pada aplikasi jasa servis.

9. Menu Utama dan Menu Detail Pemesanan Jasa Servis



Gambar 3. 28 Tampilan menu pemesanan pengguna

Pada gambar 3.28 pada menu utama jasa servis menampilkan daftar list pengguna yang melakukan pemesanan pada aplikasi Serline ketika *list* di klik maka akan menampilkan detail pemesanan pada jasa servis sehingga jasa servis dapat melakukan tindakan proses yaitu menerima pesanan yang sudah dipesan oleh pengguna, setelah itu maka jasa servis dapat menuju pada titik yang telah ditentukan oleh pengguna dengan rute yang ada pada menu bantuan *google maps* setelah sampai tujuan jasa servis mengambil tindakan yaitu melakukan perbaikan sesuai yang dipesan oleh pengguna ketika jasa servis selesai mengerjakan maka dapat menyelesaikan pada aplikasi Serline.

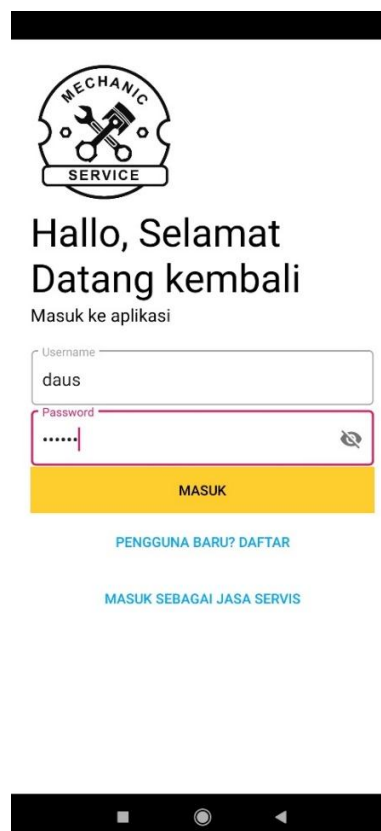
BAB IV

IMPLEMENTASI DAN PEMBAHASAN SISTEM

4.1 Implementasi Sistem

Aplikasi Servis *Online* (SERLINE) berbasis *mobile* menggunakan teknologi *android studio* telah diimplementasikan pada program aplikasi *mobile* dengan menggunakan bahasa pemrograman *Java*. Berdasarkan hasil analisis dan perancangan sistem yang telah dibuat pada bab III sebelumnya. Maka *implementasi* merupakan tahap yang berisi tentang pengoperasian dari sistem pada keadaan yang sebenarnya. Berikut merupakan *implementasi* dari sistem yang ada :

4.1.1 Halaman *Login* Pengguna



Gambar 4. 1 Halaman Masuk Pengguna

Dapat dilihat pada gambar 4.1 halaman masuk merupakan halaman yang digunakan untuk membatasi hak akses pengguna dan jasa servis, pengguna dan jasa servis yang boleh memasuki halaman melalui masuk ini adalah pengguna memiliki hak akses pengguna dan jasa servis, adapun fitur yang tersedia adalah tombol masuk.

```

checkUser.addListenerForSingleValueEvent(new
ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            username.setError(null);
            username.setEnabled(false);
            String passwordFromDB =
dataSnapshot.child(userEnteredUsername).child("password").getValue(String.class);

            if
(passwordFromDB.equalsIgnoreCase(userEnteredPassword)
) {
                username.setError(null);
                username.setEnabled(false);
                auth = FirebaseAuth.getInstance();

                auth.signInWithEmailAndPassword(userEnteredUsername,
userEnteredPassword)

                .addOnCompleteListener(Login.this, new
                OnCompleteListener<AuthResult>() {
                    @Override
                    public void
onComplete(@NonNull Task<AuthResult> task) {
                        if (!task.isSuccessful()) {
                            } else {
                                Bundle bundle =
new Bundle();

                                bundle.putString("username", userEnteredUsername);

                                bundle.putString("pass", userEnteredPassword);
                                    if
(!userEnteredUsername.equals("admin@gmail.com") &&!userEnteredPassword.equals("admin123")) {
                                        startActivity(new Intent(Login.this, Home.class)

```

```

        .putExtra("emailpass", bundle));
                                                                    finish();
    }else{
        startActivity(new Intent(Login.this,Home.class)

        .putExtra("emailpass", bundle));
            finish();
    }}}});

        String nameFromDB =
dataSnapshot.child(userEnteredUsername).child("name")
        .getValue(String.class);
        String usernameFromDB =
dataSnapshot.child(userEnteredUsername).child("userna
me").getValue(String.class);
        String phoneNoFromDB =
dataSnapshot.child(userEnteredUsername).child("phoneN
o").getValue(String.class);
        String emailFromDB =
dataSnapshot.child(userEnteredUsername).child("email"
).getValue(String.class);

        Intent intent = new
Intent(getApplicationContext(), Home.class);
        intent.putExtra("name", nameFromDB);
        intent.putExtra("username",
usernameFromDB);
        intent.putExtra("email",
emailFromDB);
        intent.putExtra("phoneNo",
phoneNoFromDB);
        intent.putExtra("password",
passwordFromDB);
        startActivity(intent);
    } else {

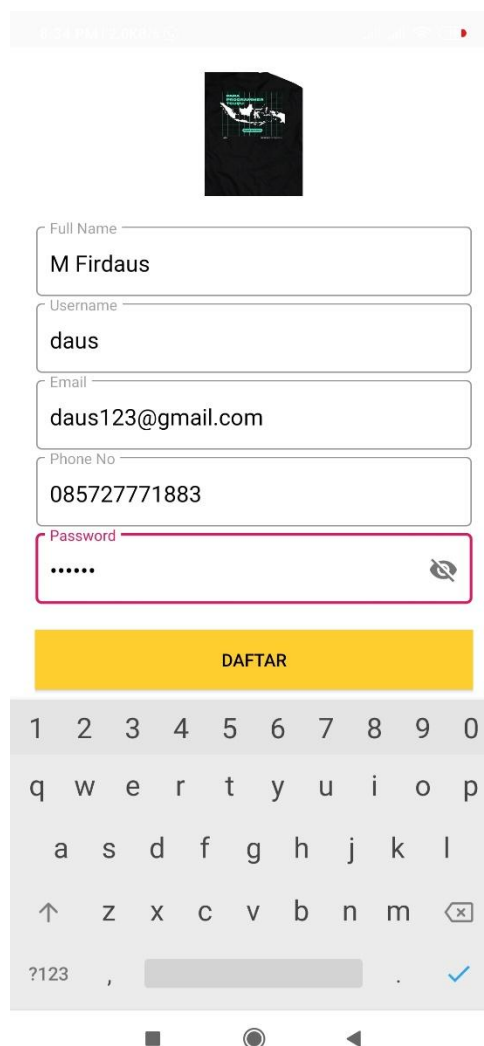
password.setError("Wrong Password");
        password.requestFocus();
    }
    } else {
        username.setError("No such
User exist");
        username.requestFocus();
    }
}
@Override
public void onCancelled(@NonNull DatabaseError
databaseError) {
}});

```

Gambar 4. 2 Kode Program *Login* Pengguna

Pada gambar 4.2 yaitu kode program yang digunakan memverifikasi *username* dan *password* pengguna untuk masuk ke aplikasi Serline, ketika pengguna akan login dengan *username* dan *password* berbeda maka tidak bisa, kode program pada gambar 4.2 juga digunakan untuk mengecek status itu sebagai pengguna atau jasa servis.

4.1.2 Halaman Daftar Akun Pengguna



The image shows a mobile application registration screen. At the top, there is a header with the text 'Serline' and a red notification dot. Below the header is a small image of a smartphone. The main form consists of several input fields: 'Full Name' with the value 'M Firdaus', 'Username' with the value 'daus', 'Email' with the value 'daus123@gmail.com', and 'Phone No' with the value '085727771883'. The 'Password' field is currently empty and is highlighted with a red border, indicating it is required. Below the form is a yellow button labeled 'DAFTAR'. At the bottom, a virtual keyboard is visible, showing the letters 'a s d f g h j k l' on the second row.

Gambar 4. 3 Tampilan Daftar Akun Pengguna

Pada gambar 4.3 yaitu tampilan yang digunakan untuk mendaftar sebagai pengguna baru sebelum dilakukan masuk pada aplikasi Serline maka untuk menyimpan data pengguna ke *database firebase*.

```

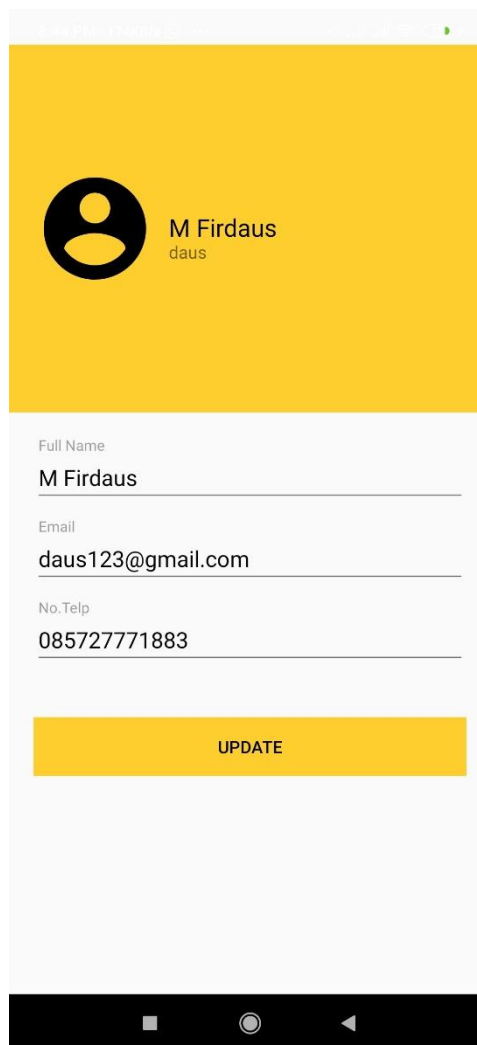
private void CreateUserAccount(final String name,
String username, String email, String phoneNo, String
password) {
 mAuth.createUserWithEmailAndPassword(email,password) .
addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull
Task<AuthResult> task) {
if (task.isSuccessful()) {
showMessage("Account
created");
updateUserInfo(name,pickedImgUri,mAuth.getCurrentUser
());
}
else{
showMessage("account creation failed" +
task.getException().getMessage());
regBtn.setVisibility(View.VISIBLE);
}
});
}
private void checkAndRequestForPermission() {
if
(ContextCompat.checkSelfPermission(SignUp.this,
Manifest.permission.READ_EXTERNAL_STORAGE)
!= PackageManager.PERMISSION_GRANTED)
{if
(ActivityCompat.shouldShowRequestPermissionRationale(
SignUp.this,
Manifest.permission.READ_EXTERNAL_STORAGE)) {
Toast.makeText(SignUp.this, "Please
accept for required permission",
Toast.LENGTH_SHORT).show();
} else {
ActivityCompat.requestPermissions(SignUp.this,
new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PReqCode);
}}
else
openGallery();}

```

Gambar 4. 4 Kode Program Daftar Akun Pengguna

Kode program pada gambar 4.4 yaitu untuk membuat akun baru pada aplikasi Serline untuk menyimpan kedalam *database firebase*, sehingga ketika semua data di isi lengkap maka akun bisa dibuat jika tidak maka akun gagal dibuat, dan potongan kode program pada gambar 4.4 terdapat potongan program untuk melakukan *check permissions* penggunaan galeri pengguna untuk mengambil gambar foto akun.

4.1.3 Halaman Akun Pengguna



Gambar 4. 5 Tampilan Halaman Akun Pengguna

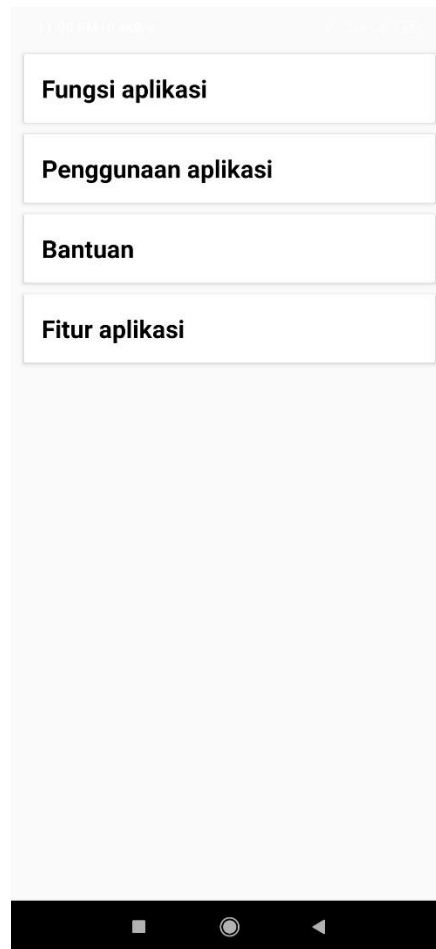
Pada gambar 4.5 yaitu tampilan yang terdapat pada menu akun pengguna yang menampilkan data pengguna ketika pendaftaran akun pada menu akun itu dapat dilakukan perubahan data seperti nama pengguna, *email* pengguna, dan nomor telepon.

```
private void update(View view) {  
  
    if(isNameChanged()){  
        Toast.makeText(this, "Data berhasil di  
update", Toast.LENGTH_LONG).show();  
    }  
    if(isEmailChanged()){  
        Toast.makeText(this, "Data berhasil di  
update", Toast.LENGTH_LONG).show();  
    }  
    if(isTeleponNoChanged()){  
        Toast.makeText(this, "Data berhasil di  
update", Toast.LENGTH_LONG).show();  
    }  
}  
  
private void showAllUserData() {  
  
    fullNameLabel.setText(_name);  
    usernameLabel.setText(_username);  
    fullName.getEditText().setText(_name);  
    email.getEditText().setText(_email);  
    phoneNo.getEditText().setText(_phoneNo);  
}
```

Gambar 4. 6 Kode Program Akun Pengguna

Gambar 4.6 merupakan potongan kode program yang digunakan untuk melakukan memperbarui data pengguna dan untuk menampilkan data dari *database* ke menu akun pengguna.

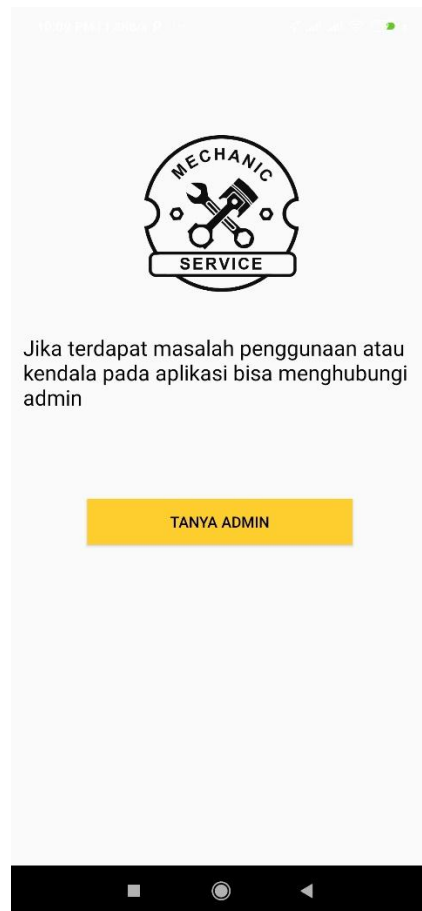
4.1.4 Halaman Bantuan Pengguna



Gambar 4. 7 Halaman Menu Bantuan Pengguna

Pada halaman menu bantuan pengguna pada gambar 4.7 yang digunakan untuk menampilkan informasi bantuan untuk pengguna dengan bentuk tampilan *recyclerview*.

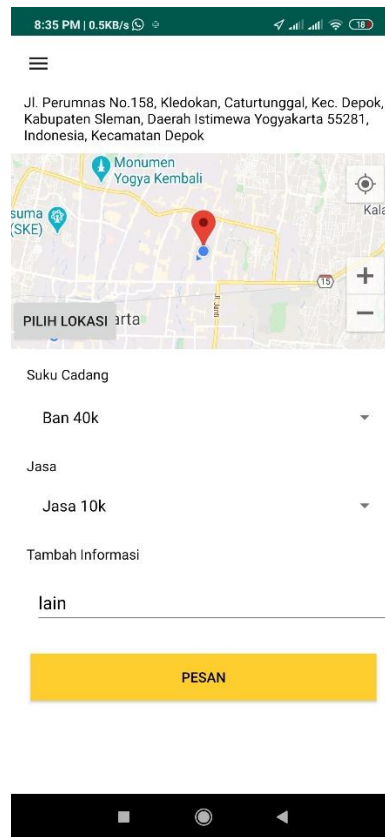
4.1.5 Halaman Tanya Admin



Gambar 4. 8 Halaman Menu Tanya Admin

Pada tampilan menu tanya admin yang digunakan untuk menghubungi jika terdapat masalah pada penggunaan aplikasi Serline.

4.1.6 Halaman Utama Pengguna



Gambar 4. 9 Tampilan Halaman Menu Utama Pengguna

Pada tampilan halaman menu utama pengguna yaitu digunakan untuk membuat menu pesanan pada menu ini maka pengguna menentukan lokasi kerusakan dengan pilih lokasi selanjutnya pengguna menentukan pergantian suku cadang yang dibutuhkan dengan adanya jasa penggantian suku cadang atau tambal ban, pengguna dapat menambahkan informasi untuk memberitahu jasa servis.

```

pesan.setOnClickListner(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = getIntent();
        String _username =
intent.getStringExtra("username");
String spinn = simpleSpinner.getSelectedItem().toString();
        int total=0;
        if (spinn !=null){
            member.setSukuCadang(spinn);
reffer.child(String.valueOf(_username)).setValue(member);
            if(spinn.equals("Tidak ada")){
                total +=0;
            }else if(spinn.equals("Ban 40k")){
                total+=40000;
            }else if(spinn.equals("Busi 20k")){
                total+=20000;
            }else if(spinn.equals("Lampu Depan 30k")){
                total+=30000;}}
String spin = costumSpinner.getSelectedItem().toString();
        if (spin !=null){
            member.setJasa(spin);
reffer.child(String.valueOf(_username)).setValue(member);
            if(spin.equals("Jasa 10k")){
                total+=10000;
            }else if(spin.equals("Tambal Ban 15k")){
                total+=15000;
            }else if(spin.equals("Servis 50k")){
                total+=50000;}}
            if(resAlamat.getLatitude() != null&&
resAlamat.getLongitude() !=null){}
            String
ket=pesanKeterangan.getText().toString();
            if (ket != null) {
                member.setKeterangan(ket);
reffer.child(String.valueOf(_username)).setValue(member);
                Intent intent2 = new
Intent(getApplicationContext(), Pemesanan.class);
intent2.putExtra("latitude",resAlamat.getLatitude());
intent2.putExtra("longitude",resAlamat.getLongitude());
                intent2.putExtra("alamat",
tvAlamat.getText().toString());
                intent2.putExtra("sukuCadang", spinn);
                intent2.putExtra("jasa", spin);
                intent2.putExtra("keterangan", ket);
                intent2.putExtra("user", _username);
                intent2.putExtra("tot",String.valueOf(total));
                AgendaModel data = new AgendaModel();
                data.setAlamat(tvAlamat.getText().toString());
data.setLatitude(String.valueOf(resAlamat.getLatitude()));

```

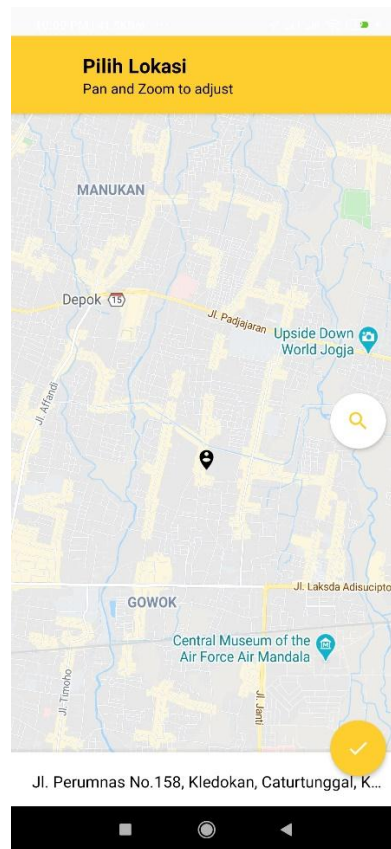


```
data.setLongitude(String.valueOf(resAlamat.getLongitude()));
    intent2.putExtra("resAlamat", Gson.to(data));
    setResult(AppCompatActivity.RESULT_OK, intent2);
    if(resAlamat.getLatitude()!=null &&
resAlamat.getLongitude()!=null) {
        startActivity(intent2);
        finish();
    }else{
        showMessage("Lokasi belum dipilih");
    }
});}
```

Gambar 4. 10 Kode Program Menu Utama Pengguna

Pada potongan kode program pada gambar 4.10 yaitu menu utama pengguna yang digunakan untuk melakukan proses pemesanan pengguna diharuskan menginputkan lokasi selanjutnya pengguna menentukan suku cadang yang digunakan, selanjutnya pengguna memilih menu jasa servis, pengguna dapat menambah keterangan kemudian klik tombol pesan, maka pengguna menunggu jasa servis sampai ke lokasi yang telah ditentukan oleh pengguna.

4.1.7 Halaman Pilih Lokasi



Gambar 4. 11 Tampilan Halaman Menu Pilih Lokasi

Pada gambar 4.11 yaitu untuk menentukan lokasi pengguna pada saat mengalami gangguan pada kendaraan sehingga untuk menginformasikan kepada jasa servis lokasi pengguna.

```

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

    getWindow().requestFeature(Window.FEATURE_ACTION_BAR);
    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        actionBar.hide();
    }
    Mapbox.getInstance(this,
    "pk.eyJ1IjoiYWdlc3RyaSIsImEiOiJja2U0ZXl2d24wbXZiMnVxaHkldnM0NHpoIn0.tM_RFoJB7lrG9h4s62bZEq");
    setContentView(R.layout.activity_place_picker);
    ButterKnife.bind(this);
    SupportMapFragment mapFragment =
    (SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map_view);
    if (mapFragment != null) {
        mapFragment.getMapAsync(this);
    }
    gps = new LocationTrack(this);
    currentLocation = new LatLng(gps.getLatitude(),
    gps.getLongitude());
    btnPick.setOnClickListener(v -> {
        Intent returningIntent = new Intent();
        AgendaModel data = new AgendaModel();
        data.setAlamat(resAddress);

        data.setLatitude(String.valueOf(pickLoc.latitude));

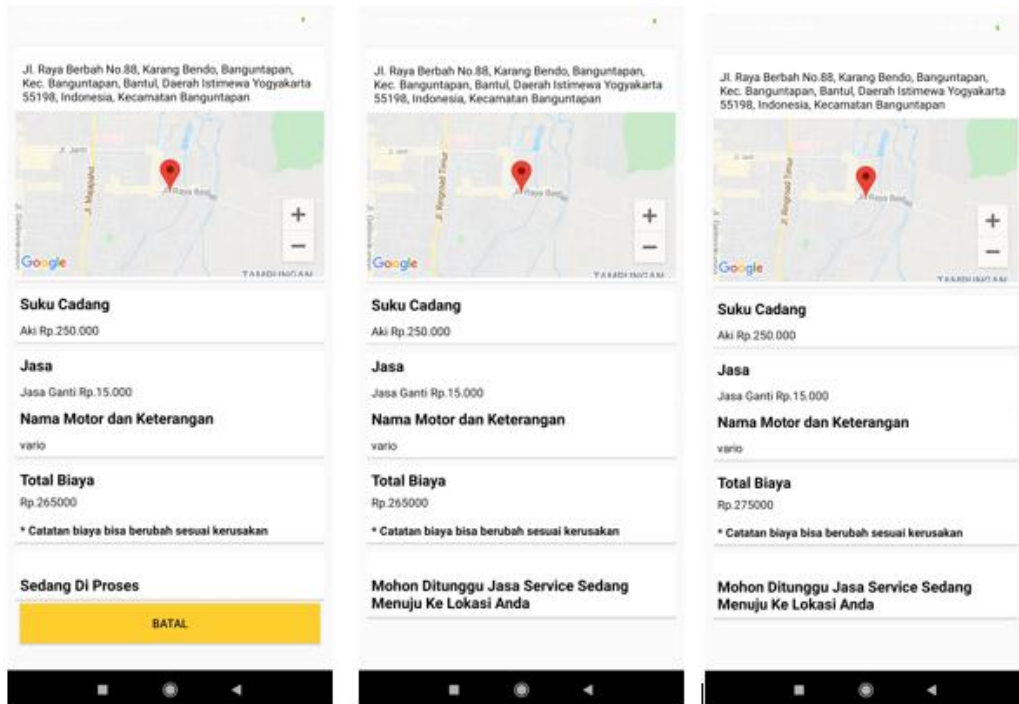
        data.setLongitude(String.valueOf(pickLoc.longitude));
        returningIntent.putExtra("resAlamat",
        Gson.toJson(data));
        setResult(AppCompatActivity.RESULT_OK,
        returningIntent);
        finish();
    });
}

```

Gambar 4. 12 Kode Program Halaman Menu Pilih Lokasi

Potongan kode program pada gambar 4.12 untuk menampilkan lokasi yang akan dimasukan oleh pengguna sehingga lokasi pengguna dapat disimpan pada database untuk ditampilkan ke jasa servis, pada potongan kode program pilih menu lokasi pengguna dapat mencari kata kunci alamat yang ada di *map*.

4.1.8 Halaman Pemesanan Pengguna



Gambar 4. 13 Tampilan Halaman Menu Pemesanan Pengguna

Pada gambar 4.13 yaitu detail menu pemesanan pengguna untuk menampilkan hasil dari pemesanan pada menu ini ketika belum dilakukan tindakan proses oleh jasa servis maka pengguna bisa membatalkan ketika pesanan sudah diproses jasa servis maka tidak dapat dibatalkan dan pengguna menunggu jasa servis sampai lokasi yang sudah ditentukan, dan pada menu pemesanan dapat berubah total harga jika terdapat tambahan biaya dari jasa servis untuk kerusakan yang tidak diketahui sebelumnya.

```
reference =
FirebaseDatabase.getInstance().getReference().child("O
rderan").child(user);
reference.addValueEventListener(new
ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
```

```

if(dataSnapshot.child("status").getValue()!=null) {
    status =
dataSnapshot.child("status").getValue().toString();
    if(status.equals("sudah")){
        tindakan.setText("Mohon Ditunggu Jasa
Service Sampai Di Lokasi Anda");
btnBatal.setVisibility(View.INVISIBLE);
        temp=0;
    }else{
        btnBatal.setVisibility(View.VISIBLE);
        tindakan.setText("Sedang Di Proses");
    }
}
}else if(temp==0){
    Intent intent2 = new
Intent(Pemesanan.this,Login.class);
    temp=1;
    startActivity(intent2);
    finish();
}
}
@Override
public void onCancelled(@NonNull DatabaseError
databaseError) {
}
});
btnBatal.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        reff =
FirebaseDatabase.getInstance().getReference().child("O
rderan");
        reff.child(user).removeValue();
        Intent intent = new Intent(Pemesanan.this,
Login.class);
        startActivity(intent);
        finish();
    }
});

```

Gambar 4. 14 Kode Program Menu Pemesanan Pengguna

Pada gambar 4.14 potongan kode program menu pemesanan pengguna menampilkan data yang disimpan pada *database firebase*, yang di pesan oleh pengguna ketika pesanan belum di proses atau diambil jasa servis pengguna dapat membatalkan jika sudah di proses oleh jasa servis maka pesanan tidak bisa di batal kan.

4.1.9 Halaman Daftar Jasa Servis



PT. SERLINE

agus

ab 1234 as

085727778888

agus@gmail.com

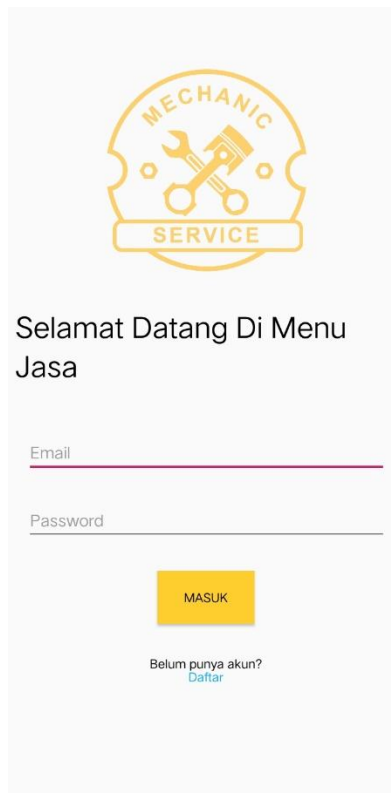
.....

DAFTAR

Gambar 4. 15 Tampilan Halaman Menu Daftar Jasa Servis

Pada gambar 4.15 yaitu tampilan yang digunakan untuk mendaftar sebagai jasa servis baru sebelum dilakukan login pada aplikasi Serline maka untuk menyimpan data jasa servis ke *database firebase*.

4.1.10 Halaman Masuk Jasa Servis

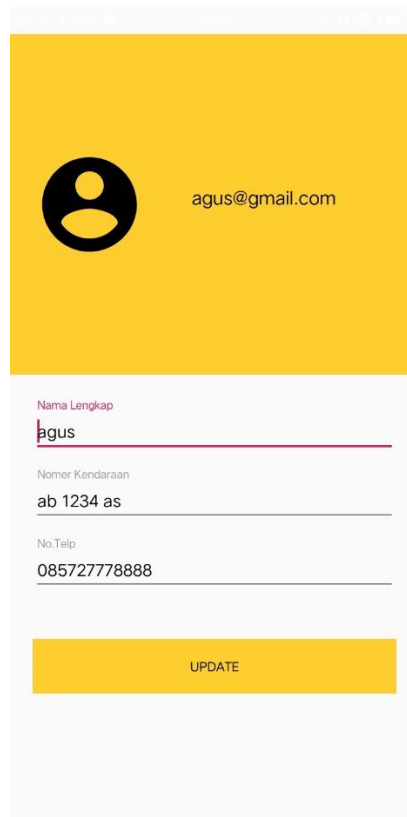


The image shows a login page for 'MECHANIC SERVICE'. At the top center is a circular logo with a yellow border. Inside the circle, the word 'MECHANIC' is written in a semi-circle at the top, and 'SERVICE' is written in a semi-circle at the bottom. In the center of the circle are icons of a wrench and a piston. Below the logo, the text 'Selamat Datang Di Menu Jasa' is displayed. Underneath, there are two input fields: 'Email' and 'Password', each with a horizontal line below it. A yellow button with the text 'MASUK' is positioned below the password field. At the bottom, the text 'Belum punya akun?' is followed by a blue link labeled 'Daftar'.

Gambar 4. 16 tampilan Halaman Menu Masuk Jasa Servis

halaman masuk merupakan halaman yang digunakan untuk membatasi hak akses jasa servis, jasa servis yang boleh memasuki halaman melalui login ini adalah jasa servis memiliki hak akses, adapun fitur yang tersedia adalah tombol *login* pada gambar 4.16.

4.1.11 Halaman Akun Jasa Servis

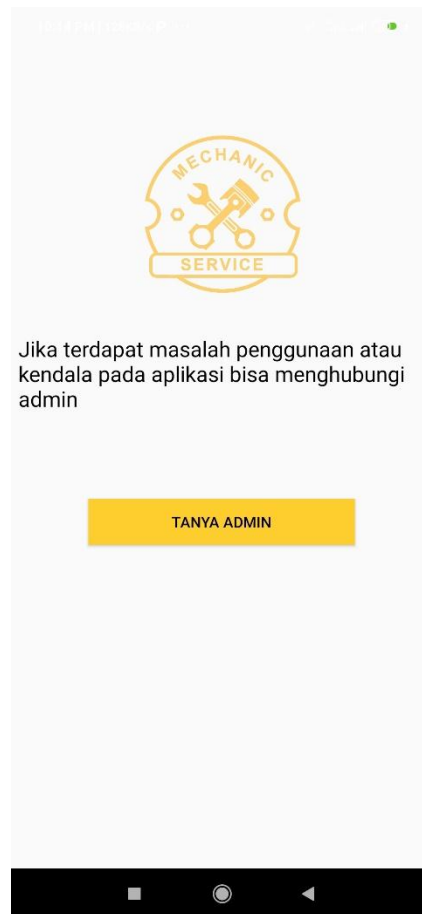


The screenshot shows a mobile application interface for a service account. At the top, there is a yellow header bar containing a black profile icon and the email address 'agus@gmail.com'. Below the header, there are three text input fields with labels in red: 'Nama Lengkap' (containing 'agus'), 'Nomer Kendaraan' (containing 'ab 1234 as'), and 'No.Telp' (containing '085727778888'). At the bottom of the form is a yellow button labeled 'UPDATE'.

Gambar 4. 17 Tampilan Halaman Menu Akun Jasa Servis

Pada gambar 4.17 yaitu tampilan yang terdapat pada menu akun jasa servis yang menampilkan data jasa servis ketika pendaftaran akun pada menu akun itu dapat dilakukan perubahan data seperti nama jasa servis, nomor kendaraan jasa servis, dan nomor servis.

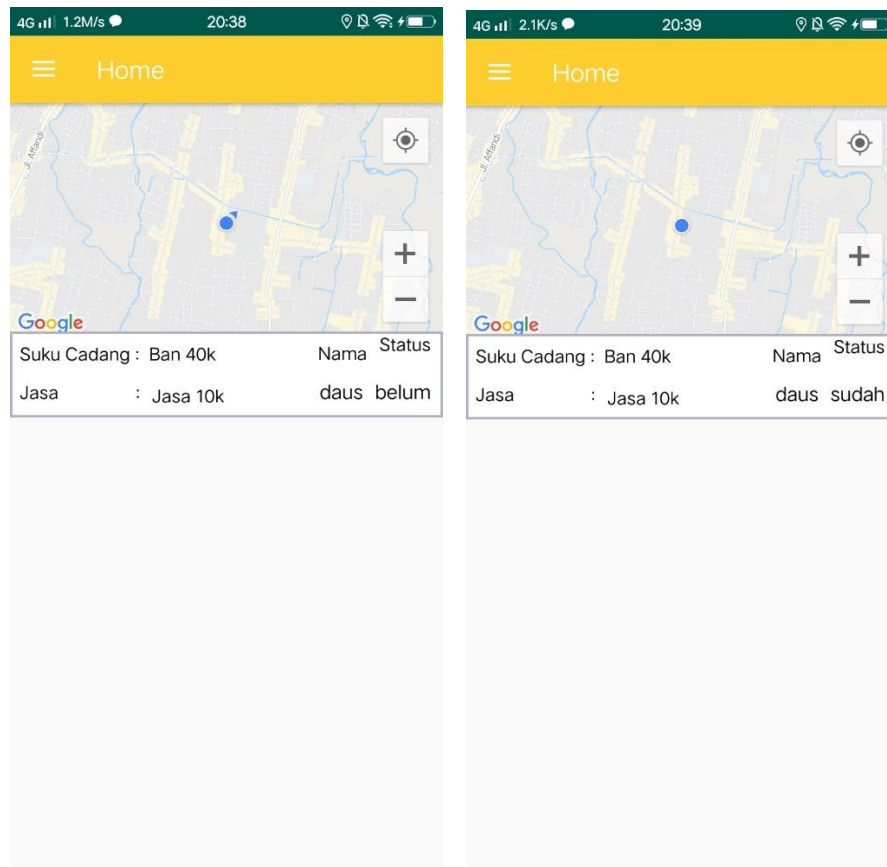
4.1.12 Halaman Tanya Admin



Gambar 4. 18 Tampilan Halaman Menu Tanya Admin

Pada tampilan menu tanya *admin* yang digunakan untuk menghubungi jika terdapat masalah pada penggunaan jasa servis di aplikasi Serline.

4.1.13 Halaman Menu Utama Jasa Servis



Gambar 4. 19 Tampilan Menu Utama Jasa Servis

Pada Gambar 4.19 yaitu tampilan menu utama pada jasa servis pada menu itu terdapat *list* pemesanan yang dipesan oleh pengguna dengan tampilan list maka jasa servis dapat melihat pesanan jika jasa servis klik *list* itu maka jasa servis dapat memproses pesanan pengguna, tampilan map untuk melihat posisi jasa servis pada saat itu.

```
class BookItemView extends RecyclerView.ViewHolder{
    private TextView mSukuCadang;
    private TextView mJasa;
    private TextView mUsername;
    private String key;
    public BookItemView(ViewGroup parent){
        super (LayoutInflater.from(mContext) .
            inflate(R.layout.book_list_item,
                parent, false));
```

```

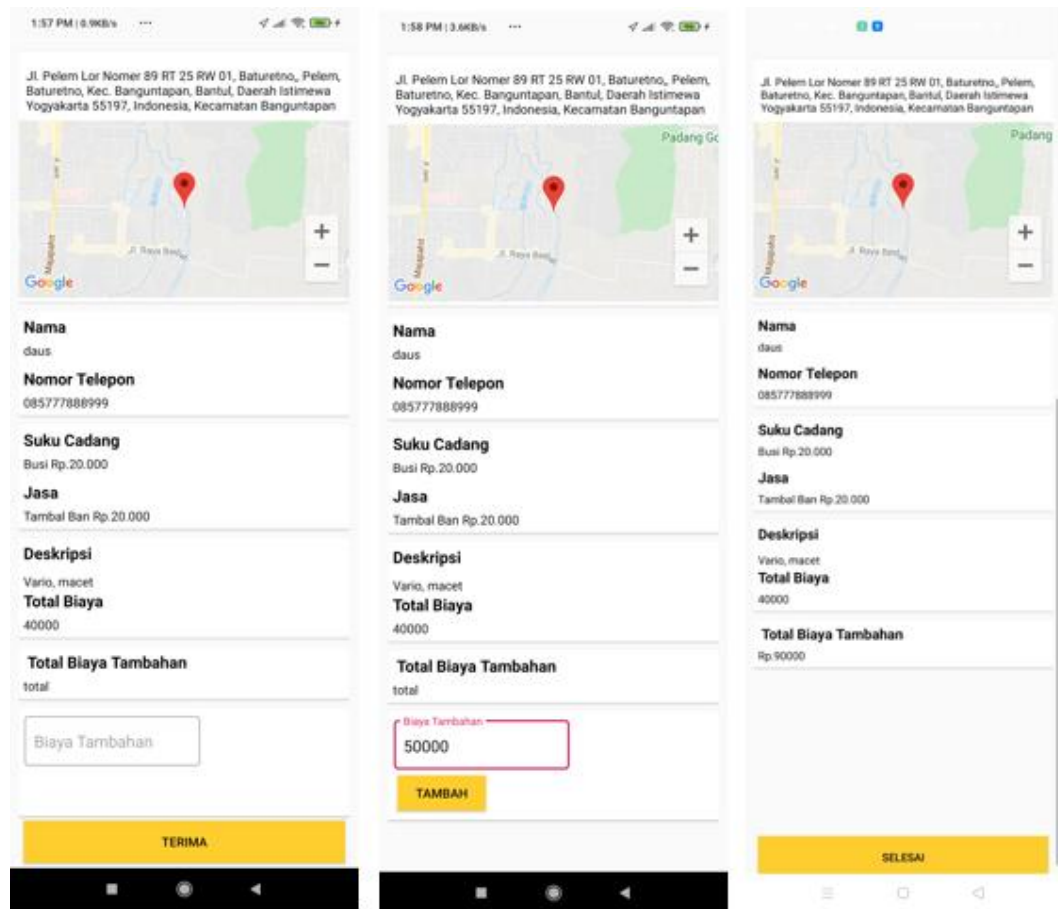
        mSukuCadang = (TextView)
itemView.findViewById(R.id.tvSukuCadang);
        mJasa = (TextView)
itemView.findViewById(R.id.tvJasa);
        mUsername = (TextView)
itemView.findViewById(R.id.tvUsername);
    }
    public void bind(Pesan book, String key){
        mSukuCadang.setText(book.getSukuCadang());
        mJasa.setText(book.getJasa());
        mUsername.setText(book.getUsername());
        this.key = key;
    }
}
public void onBindViewHolder(@NonNull BookItemView
holder, int position) {
    holder.bind(mBookList.get(position),
mKeys.get(position));
    holder.itemView.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(v.getContext(),
DetailPesan.class);
            i.putExtra("nama" , mKeys.get(position));
            v.getContext().startActivity(i);
        }
    });
}
}

```

Gambar 4. 20 Kode Program Menu Utama Jasa Servis

Potongan kode program pada gambar 4.20 yaitu pada menu utama jasa servis pada menu utama jasa servis maka terdapat *recyclerview* yang di gunakan untuk menampilkan data pemesana yang di lakukan oleh pengguna, sehingga jasa servis dapat memilih pesanan yang di pesan oleh pengguna.

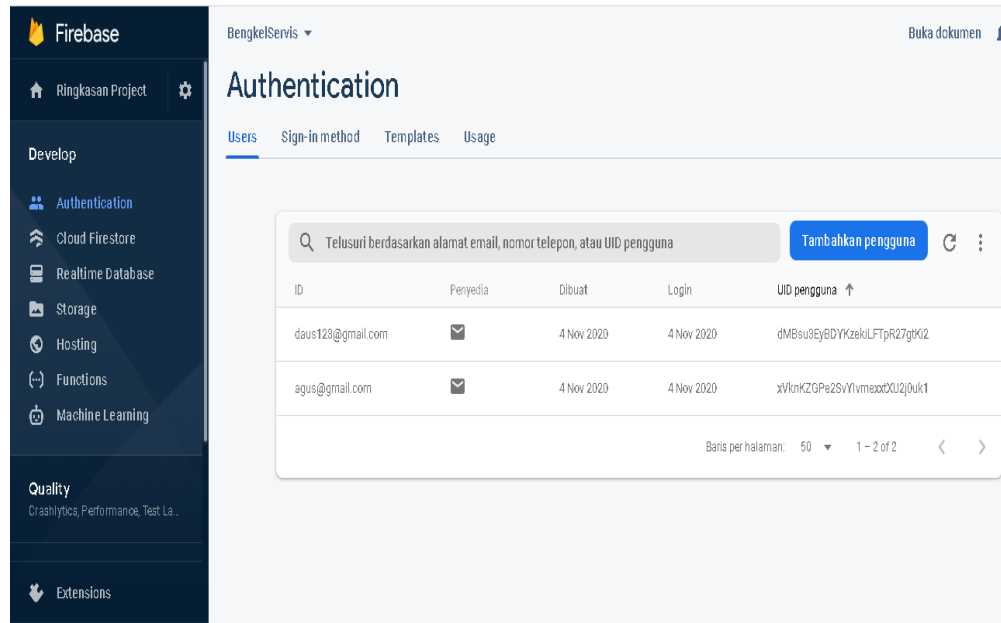
4.1.14 Halaman Detail Pemesanan Jasa Servis



Gambar 4. 21 Tampilan Halaman Menu Detail Pemesanan Jasa Servis

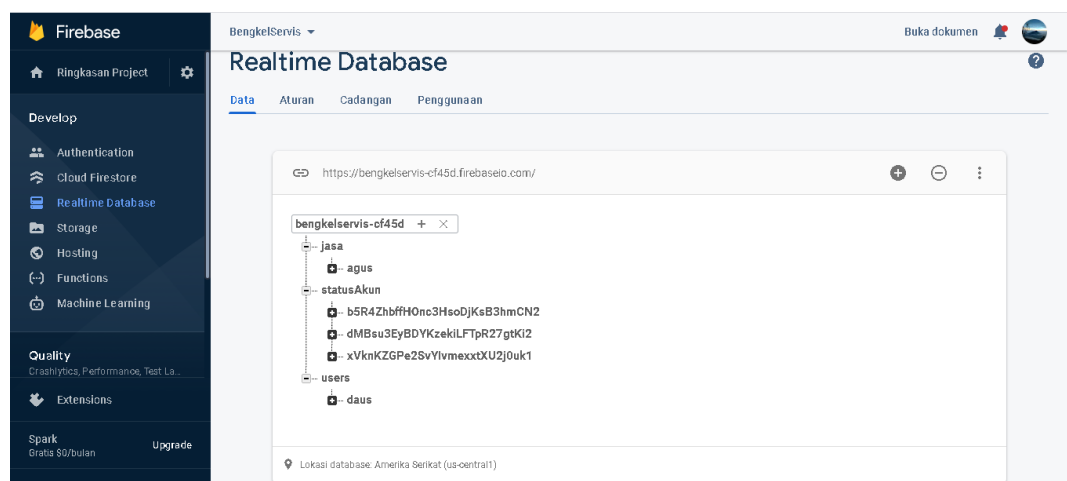
Pada tampilan menu Detail Pemesanan jasa servis dapat memproses pesanan pengguna yang telah dipesan pada map pemesanan maka jasa servis dapat melihat rute atau melihat lokasi dengan *google map*, jika jasa servis memproses pesanan pelanggan maka jasa servis menuju lokasi yang sudah ditentukan pengguna ketika selesai perbaikan kerusakan maka jasa servis dapat menyelesaikan pesanan itu, dan jasa servis dapat menambahkan biaya tambahan jika terdapat kerusakan yang tidak sesuai dengan yang ditentukan pengguna.

4.1.15 Halaman Firebase Database



Gambar 4. 22 Halaman *Authentication Firebase*

Pada gambar 4.22 yaitu tampilan *firebase authentication* yaitu untuk *verifikasi* akun pengguna atau user jika akan masuk ke aplikasi *Serline database authentication* menyimpan *email* pengguna dan jasa servis.



Gambar 4. 23 Halaman *Realtime database*

Gambar 4.23 yaitu *realtime database* pada *firebase* yang digunakan untuk menyimpan data pengguna, jasa servis, pesanan, dan status akun pada *realtime database* akan merefresh jika terdapat kegiatan daftar baru maupun pesanan yang dibuat oleh pengguna.

```

{
  "project_info": {
    "project_number": "523787624561",
    "firebase_url": "https://bengkelservis-
cf45d.firebaseio.com",
    "project_id": "bengkelservis-cf45d",
    "storage_bucket": "bengkelservis-cf45d.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id":
"1:523787624561:android:812cdc37d7d7c79f64bbd0",
        "android_client_info": {
          "package_name": "com.example.bengkelservis"
        }
      },
      "oauth_client": [
        {
          "client_id": "523787624561-
capf82mlto3lp7ajqdtj7bsr371lqd2g.apps.googleusercontent.
com",
          "client_type": 1,
          "android_info": { "package_name":
"com.example.bengkelservis",
          "certificate_hash":
"81bab9883f181698d45d25ba715680af2cc7c530"
        }
      },
      {
        "client_id": "523787624561-
md39g9130sgd3s7n4qrqo096eqcd8b1q.apps.googleusercontent.
com",
        "client_type": 3
      }
    ],
    "api_key": [
      {
        "current_key": "AIzaSyA1_xSyiUwPkPEFP-
UDZGLJl63s3E7JWiI"
      }
    ],
  ],
}

```

```

    "services": {
      "appinvite_service": {
        "other_platform_oauth_client": [
          {
            "client_id": "523787624561-
md39g9l30sgd3s7n4qrqo096eqcd8b1q.apps.googleusercontent.com",
            "client_type": 3
          }
        ]
      }
    }
  ],
  "configuration version": "1"

```

Gambar 4. 24 Kode Program JSON firebase

Pada gambar 4.24 yaitu *kode* program yang digunakan untuk koneksi antara *android studio* dengan *firebase database* dengan ada *kode* unik yang digunakan untuk mengetahui *id* pada *project android studio* dengan *firebase* pada *kode program* terdapat nomor *project* dan nama *project* yang sudah dikoneksikan sehingga dapat digunakan untuk *authentication*, *realtime database* dan *store* pada *firebase*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi firebase database dan google maps api bahwa dalam mengembangkan aplikasi servis *online* berbasis *android mobile* ini, maka dapat diambil kesimpulan bahwa :

1. Aplikasi implementasi *Firestore Api* pada pemesanan jasa servis berbasis *android* dapat menyimpan data pengguna dan jasa servis pada database *firebase*.
2. Aplikasi pada pemesanan servis online menggunakan bahasa pemrograman *Java* untuk *implementasi firebase api* dan *google maps api*.
3. Aplikasi servis *online* menggunakan *Global Positioning System GPS* ini untuk mengetahui letak para pengendara kendaraan bermotor yang mengalami gangguan atau pemesanan servis *online*, tanpa harus mengantri dan tanpa mencari bengkel terdekat.
4. Aplikasi servis online *implementasi firebase* ini dapat menampilkan data dari *firebase database* untuk ditampilkan di aplikasi atau *frontend*.
5. Aplikasi telah berhasil mempertemukan antara pengguna dengan jasa servis pada pemesanan dengan memanfaatkan teknologi *Google Maps API*, untuk mempertemukan antara pengguna dan jasa servis

5.2 Saran

Dengan selesainya penelitian mengenai aplikasi *Servis Online* (Serline) ini, saran yang dapat diberikan yaitu :

1. Diharapkan pengembangan dapat menambahkan fitur *chat* atau komunikasi antara pengguna dengan jasa servis pada aplikasi.
2. Diharapkan pengembangan dapat memberikan daftar riwayat pesanan.
3. Diharapkan pengembangan dapat memberikan notifikasi pesanan.
4. Diharapkan pengembangan aplikasi dapat menggunakan pembayaran di aplikasi.
5. Diharapkan pengembang dapat mengembangkan tampilan menu atau *UI/UX* aplikasi lebih *modern* atau lebih *responsive*.

Daftar Pustaka

- Anggraini, Dewita, 2019. Aplikasi Kartu Menuju Sehat untuk Anak Menggunakan Teknologi Firebase *Cloud Messaging* Berbasis Android. Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.
- Badri, Zaki, Dwi S, 2018. Aplikasi Bengkel Online Menggunakan *Global Positioning System* (Gps) Berbasis Android Pada Cv. Rumah Otomotif.
- Bahri, Saeful., Satia S dan Jamal M.H., 2019. Teknologi *Global positioning System* (GPS) Untuk Pelapor dan Penjemputan Sampah Berbasis Android.
- Developer *Android* .Mengenal *Android Studio* <https://developer.android.com/studio/intro/index.html?hl=id> (diakses 22 November 2019).
- Firestore, 2018 *Firestore Authentication* <https://firebase.google.com/docs/auth?hl=id> (diakses 22 november 2019).
- Fitriansyah, Rizqi Antasari, Kusrini. 2012. Perancangan Aplikasi *Mobile* Bengkelku Sebagai Informasi Alamat Bengkel Resmi Sepeda Motor Dikota Yogyakarta Menggunakan *GPS* Berbasis *Android*.
- Kumalawati, Reni, 2019. Aplikasi Pencarian Jasa Pemeliharaan Bangunan Berbasis *Android*. Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta.
- Salkiawati, Ratna., M.Hadi Prayitno dan Dadi Ulul Wilhadad, 2019. Perancangan Aplikasi Untuk Mendapatkan Mekanik Sepeda Motor Menggunakan *Google Maps API*.
- Saputra, Ivan Alfatih, 2017. Aplikasi Layanan Bengkel Mobil Berbasis *Android* Di Kota Bandar Lampung. Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Lampung.

Lampiran

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_main);
    topAnim =
AnimationUtils.loadAnimation(this,R.anim.top_animation);
    bottomAnim = AnimationUtils.loadAnimation(this,
R.anim.bottom_animation);
    image = findViewById(R.id.imageView1);
    logo = findViewById(R.id.textView1);
    slogan = findViewById(R.id.textView2);
    image.setAnimation(topAnim);
    logo.setAnimation(bottomAnim);
    slogan.setAnimation(bottomAnim);
    checkPermission();
    cekLogin();

    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            intent = new Intent(MainActivity.this, Login.class);
            Pair[] pairs = new Pair[2];
            pairs[0] = new Pair<View, String>(image, "logo_image");
            pairs[1] = new Pair<View, String>(logo, "logo_text");
            //wrap the call in API level 21 or higher
            if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.LOLLIPOP) {
                ActivityOptions options =
ActivityOptions.makeSceneTransitionAnimation(MainActivity.this, pairs);
                checkPermission();
                cekLogin();
            }
            startActivity(intent);
            finish();
        }
    }, SPLASH_SCREEN);
}
private void checkPermission() {
    if (!PermissionsManager.areLocationPermissionsGranted(this)) {
        permissionsManager = new PermissionsManager(this);
        permissionsManager.requestLocationPermissions(this);
    }
    else {
    }
}
}
```

Potongan kode program Splash screen

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sign_up_jasa);

    jasaNama = findViewById(R.id.regNamaJasa);
    jasaNomer = findViewById(R.id.regNmrJasa);
    jasaTelp = findViewById(R.id.regNoTelpJasa);
    jasaEmail = findViewById(R.id.regEmailJasa);
    jasaPassword = findViewById(R.id.regPassJasa);
}
```

```

regBtn = findViewById(R.id.regBtnJasa);
loadingProgress = findViewById(R.id.regprogressBar);
loadingProgress.setVisibility(View.INVISIBLE);
mDatabase = FirebaseDatabase.getInstance().getReference();
mAuth = FirebaseAuth.getInstance();

regBtn.setOnClickListener(new View.OnClickListener() {
    Intent intent = new Intent(SignUpJasa.this, LoginJasa.class);
    public void onClick(View v) {

        regBtn.setVisibility(View.INVISIBLE);
        loadingProgress.setVisibility(View.VISIBLE);
        rootNode = FirebaseDatabase.getInstance();
        reference = rootNode.getReference("jasa");
        final String nama = jasaNama.getText().toString();
        final String nomerkendaraan =
jasaNomer.getText().toString();
        final String nomertelepon = jasaTelp.getText().toString();
        final String email = jasaEmail.getText().toString();
        final String password = jasaPassword.getText().toString();

        if (nama.isEmpty() || nomerkendaraan.isEmpty() ||
nomertelepon.isEmpty() || email.isEmpty() || password.isEmpty() ||
pickedImgUri == null) {
            showMessage("Please Verify all fields");
            regBtn.setVisibility(View.VISIBLE);
            loadingProgress.setVisibility(View.INVISIBLE);
        } else {
            CreateUserAccount(email, nama, password);
        }
        JasaHelperClass helperClass = new JasaHelperClass("jasa",
nama, nomerkendaraan, nomertelepon, email, password);
        reference.child(nama).setValue(helperClass);
    }
});
ImgJasaPhoto = findViewById(R.id.regJasaPhoto);
ImgJasaPhoto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (Build.VERSION.SDK_INT >= 22) {
            checkAndRequestForPermission();
        } else {
            openGallery();
        }
    }
});
}

```

Potongan kode program Daftar Jasa Servis

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    sessionManager = new SessionManager(LoginJasa.this,
SessionManager.SESSION_JASA);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_login_jasa);
    jasaEmail = findViewById(R.id.email_jasa);
    jasaPassword = findViewById(R.id.password_jasa);
    btnLogin = findViewById(R.id.btn_jasa);
    loginProgress = findViewById(R.id.login_progress);
    mAuth = FirebaseAuth.getInstance();
    HomeJasa = new Intent(this,

```

```

com.example.bengkelservis.Jasa.Home.class);
    currentUser = mAuth.getCurrentUser();

    btnRegister = findViewById(R.id.reg_btn_jasa);
    btnRegister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent registerActivity = new
Intent(getApplicationContext(), SignUpJasa.class);
            startActivity(registerActivity);
            finish();
        }
    });
    loginProgress.setVisibility(View.INVISIBLE);
    btnLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            loginProgress.setVisibility(View.VISIBLE);
            btnLogin.setVisibility(View.INVISIBLE);
            final String email = jasaEmail.getText().toString();
            final String password = jasaPassword.getText().toString();
            if (email.isEmpty() || password.isEmpty()) {
                showMessage("Please Verify All Field");
                btnLogin.setVisibility(View.VISIBLE);
                loginProgress.setVisibility(View.INVISIBLE);
            } else {
                signIn(email, password);
            }
        }
    });
}
private void signIn(String email, String password) {
    mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                loginProgress.setVisibility(View.VISIBLE);
                btnLogin.setVisibility(View.INVISIBLE);
                updateUI();
            } else {
                showMessage(task.getException().getMessage());
                btnLogin.setVisibility(View.VISIBLE);
                loginProgress.setVisibility(View.INVISIBLE);
            }
        }
    });
}
}

```

Potongan kode program Masuk Jasa Servis

```

public void setConfig(RecyclerView recyclerView, Context context,
List<Pesan> books, List<String> keys){
    mContext = context;
    mBooksAdapter = new BooksAdapter(books, keys);
    recyclerView.setLayoutManager(new LinearLayoutManager(context));
    recyclerView.setAdapter(mBooksAdapter);}
class BookItemView extends RecyclerView.ViewHolder{
    private TextView mSukuCadang;
    private TextView mJasa;
    private TextView mUsername;
    private TextView mStatus;
    private String key;
    public BookItemView(ViewGroup parent){

```

```

        super(LayoutInflater.from(mContext).
            inflate(R.layout.book_list_item, parent, false));
        mSukuCadang = (TextView)
itemView.findViewById(R.id.tvSukuCadang);
        mJasa = (TextView) itemView.findViewById(R.id.tvJasa);
        mUsername = (TextView) itemView.findViewById(R.id.tvUsername);
        mStatus = (TextView) itemView.findViewById(R.id.tvStatus);
    }
    public void bind(Pesan book, String key){
        mSukuCadang.setText(book.getSukuCadang());
        mJasa.setText(book.getJasa());
        mUsername.setText(book.getUsername());
        mStatus.setText(book.getStatus());
        this.key = key;
    }
}
class BooksAdapter extends RecyclerView.Adapter<BookItemView>{
    private List<Pesanan> mBookList;
    private List<String> mKeys;
    public BooksAdapter(List<Pesanan> mBookList, List<String> mKeys) {
        this.mBookList = mBookList;
        this.mKeys = mKeys;
    }
    public BookItemView onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        return new BookItemView(parent);
    }
    public void onBindViewHolder(@NonNull BookItemView holder, int
position) {
        holder.bind(mBookList.get(position), mKeys.get(position));
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(v.getContext(),
DetailPesanan.class);
                i.putExtra("nama" , mKeys.get(position));
                String _username = mKeys.get(position);
                if(_username!=null)
                    reff =
database.getInstance().getReference().child("Orderan").child(_username)
;
                if (reff != null) {
                    mAuth = FirebaseAuth.getInstance();
                    currentUser = mAuth.getCurrentUser();
                    reff.addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                            if (dataSnapshot.exists()) {
                                status =
dataSnapshot.child("status").getValue(String.class);
                                diambilOleh =
dataSnapshot.child("diambilOleh").getValue(String.class);
                            }
                        }
                        @Override
                        public void onCancelled(@NonNull DatabaseError
databaseError) {
                            }
                        });
                    }
                    if(status!=null && status.equals("sudah") &&
diambilOleh.equals(currentUser.getId())){

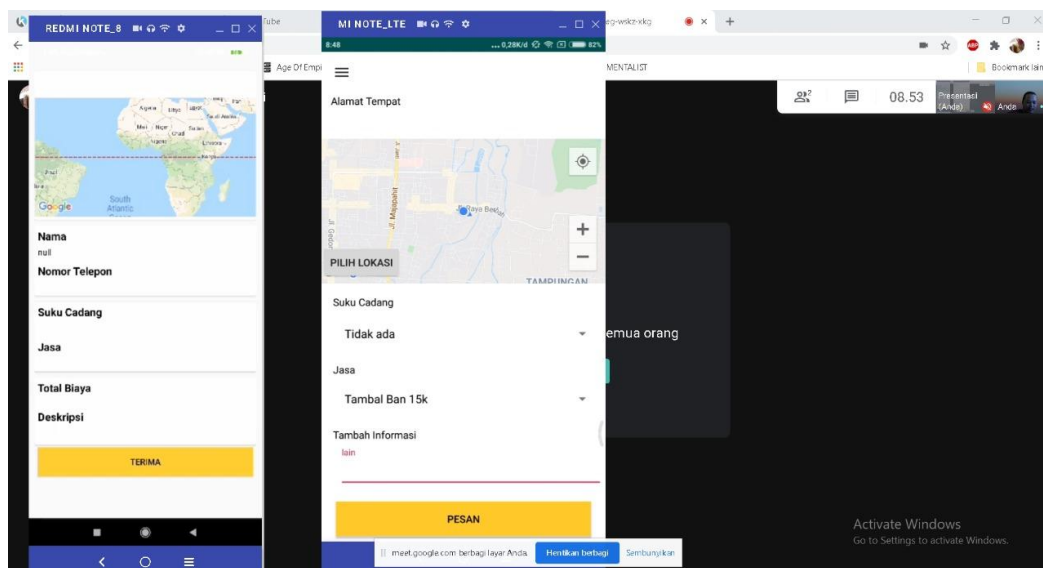
```

```

        v.getContext().startActivity(i);
    }else if(status!=null && status.equals("belum")) {
        v.getContext().startActivity(i);
    }else if(status!=null && status.equals("sudah")){
        Toast.makeText( v.getContext(), "Sudah di Ambil
        Jasa Servis", Toast.LENGTH_LONG).show();
    }));}
    public int getItemCount() {
        return mBookList.size();
    }
}

```

Potongan kode program Daftar Pesanan pada menu utama Jasa Servis



Bimbingan pengujian aplikasi menggunakan google meet

PEMBERITAHUAN SEBELUM UJIAN :										
Pengkumpulan akhir dokumen Tugas Akhir/Skripsi melewati semester berjalan,mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya.										
A	B	C	D	E	F	G	H	I	J	K
1	PEMBERITAHUAN SEBELUM UJIAN : Pengkumpulan akhir dokumen Tugas Akhir/Skripsi melewati semester berjalan,mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya.									
2										
3	KRITERIA KELULUSAN UJIAN SIDANG / PENDADARAN									
4										
5										
6	1. Lulus ujian tanpa syarat, disebut kriteria 1.									
7	2. Lulus bersyarat, disebut kriteria 2, yaitu dengan sedikit perbaikan atau penyempurnaan text dan atau program dalam waktu maksimum tanggal 26 Januari 2021 dan tidak ada ujian lagi. Jika dalam waktu yang ditentukan mahasiswa tersebut tidak dapat menyelesaikan maka, mahasiswa yang bersangkutan dianggap tidak lulus ujian.									
8	3. Tidak lulus ujian sidang/pendadaran, disebut kriteria 3, dijelaskan, disarankan Ketua Tim Penguji untuk mempelajari ulang materi, merombak program/teks, atau mengganti judul.									
9										
10										
11	Ketentuan bagi peserta yang tidak lulus ujian sidang / pendadaran.									
12	1) Mahasiswa wajib menepuh ujian sidang/pendadaran ulang									
13	2) Kesempatan ujian sidang/pendadaran ulang hanya diberikan dalam rentang waktu maksimum 6 bulan, setelah ujian sidang/pendadaran									
14	3) Jika sampai batas waktu maksimum 6 bulan tersebut belum dapat diajukan/diselesaikan, maka calon peserta ujian dinyatakan sebagai mahasiswa peserta Skripsi/TGA baru, dengan segala ketentuan yang berlaku bagi peserta baru									
15	4) Mahasiswa yang akan menepuh ujian sidang/pendadaran ulang ini diwajibkan membayar biaya ujian setara 2 SKS praktik, sesuai tahun angkatan									
16										
17										
18										
19	Yogyakarta, _____									
20	Memahami dan bersedia									
21	Mengetahui dan menyetujui atas,									
22										

Ketentuan pelaksanaan pendadaran

	A	B	C	D	E	F	G	H
1								
2	Catatan Pendadaran							
3								
4								
5	Hari, tanggal	:	Kamis, 26 November 2020					
6	Waktu	:	08.00					
7	Nama	:	Agus Tri Wahyu					
8	No. Mahasiswa / Jurusan	:	165410014 / Teknik Informatika					
9	Nama Dosen	:	Edi Iskandar, ST, M.Cs.					
10			Penguji					
11	Hal yang harus diperbaiki	:						
12		1.	Laporan diperbaiki sistem penulisan sesuai dengan catatan yang ada dilaporan					

Catatan pendadaran penguji pendadaran

fx =IMPORTRANGE("https://docs.google.com/spreadsheets/d/16NbFhnMZMs4Y5nZYbXdRG8h9b1JKNVKbnrrUsFAR-ro/edi

	A	B	C	D	E	F	G	I
1								
2	Catatan Pendadaran							
3								
4								
5	Hari, tanggal	:	Agus Tri Wahyu					
6	Waktu	:	08.00					
7	Nama	:	Agus Tri Wahyu					
8	No. Mahasiswa / Jurusan	:	165410014 / Teknik Informatika					
9	Nama Dosen	:	Edi Iskandar, ST, M.Cs.					
10			Pembimbing					
11	Hal yang harus diperbaiki	:						
12		1.	Perhatikan spasi antar header/sub jeader dengan body teks					OK
13		2.	Batasan wilayah dan owner aplikasi milik sendiri, masukkan di ruang lingkup					OK
14		3.	Pengembangan kedepan masukkan di saran					OK
15		4.	Analisis kebutuhan : gmap API dan MapBox					OK
16		5	lihat catatan di naskah					
17								

Catatan pembimbing pendadaran

	A	B	C	D	E	F
1						
2	KEPUTUSAN HASIL UJIAN PENDADARAN					
3						
4	Sesuai dengan hasil sidang pendadaran pada tanggal			26 November 2020	maka	
5						
6	Nama Mahasiswa	Agus Tri Wahyu				
7	NIM / Program Studi	165410014 / Teknik Informatika				
8	Jenjang	S1				
9						
10		dinyatakan	LULUS	dengan kriteria	2	
11						
12						
13						
14	Ketua Penguji	Edi Iskandar, ST, M.Cs.				
15						
16						

Keputusan hasil ujian pendadaran

The screenshot shows a web browser window with the URL elearning.akakom.acid/mod/assign/view.php?id=11274. The page title is 'E L A' and the user is identified as '165410014 Agus Tri Wahyu'. The main content area is titled 'Feedback' and contains the following information:

- Grade: Competent
- Graded on: Monday, 7 December 2020, 7:59 AM
- Graded by: M Guntara
- Feedback comments:
 - Berdasarkan pemeriksaan, maka naskah hasil revisi pasca pendadaran ini dinyatakan disetujui.
 - Untuk selanjutnya DIJILID. untuk tanda tangan selama masa COVID ini silahkan dititipkan bag. ADAK.
 - Cetak komentar ini sbg bukti acc revisi
 - Selamat dan terimakasih!

Bukti hasil acc revisi pasca ujian pendadaran