

**RANCANG BANGUN APLIKASI *BOOKING ONLINE*
BENGKEL MOTOR BERBASIS ANDROID**
(Studi kasus: Bengkel Oka Motor)

LAPORAN TUGAS AKHIR



ALFIAN DWI PRASETYA

5180421005

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

**RANCANG BANGUN APLIKASI *BOOKING ONLINE* BENGKEL MOTOR
BERBASIS ANDROID**
(Studi kasus: Bengkel Oka Motor)

Disusun Oleh

ALFIAN DWI PRASETYA

(5180421005)

Telah dipertahankan di depan Dewan Pengaji

pada tanggal 17 Februari 2024

Nama & Gelar

Joko Aryanto, S.Kom., M.Kom.
NIK. 110918168

Tanda tangan

Tanggal

21/02/2024

Ledy Elsera Astrianty, S.Kom., M.Kom.
NIK. 110221184

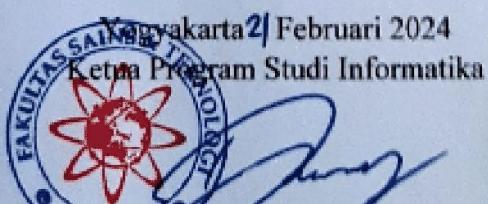
Pengaji I

20/02/2024

Sulistyo Dwi Sancoko S.Si.,M.Sc.
NIK. 110919179

Pengaji II
(Dosen Pembimbing)

21/02/2024



Dr. Donny Avianto, S.T., M.T.
NIK. 111015077

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini, saya:

Nama : Alfian Dwi Prasetya

NPM : 5180421005

Program Studi : Informatika

Program : Sarjana

Fakultas : Sains & Teknologi

Menyatakan bahwa tugas akhir dengan judul rancang bangun aplikasi *booking online* bengkel motor berbasis android ini adalah karya ilmiah asli saya dan belum pernah dipublikasikan oleh orang lain, kecuali yang tertulis sebagai acuan dalam naskah ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa yang diberikan Program Studi Informatika Fakultas Sains & Teknologi Universitas Teknologi Yogyakarta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada tanggal : 21 Februari 2024

Yang menyatakan



Alfian Dwi Prasetya

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT, dimana karena rahmat dan hidayahnya, penulisan laporan seminar tematik yang berjudul " rancang bangun aplikasi *booking online* bengkel motor berbasis android " dapat diselesaikan dengan lancar.Laporan tugas akhir ini disusun untuk memenuhi persyaratan memperoleh gelar sarjana pada program studi Informatika Fakultas Sains & Teknologi Universitas Teknologi Yogyakarta.Laporan tugas akhir ini tidak lepas dari segala bantuan, bimbingan, dorongan dan do'a dari berbagai pihak, yang pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Dr. Bambang Moertono S., M.M., Akt., C.A. selaku Rektor Universitas Teknologi Yogyakarta.
2. Dr. Endy Marlina, MT selaku Dekan Fakultas Sains & Teknologi Universitas Teknologi Yogyakarta.
3. Dr. Dony Avianto, MT selaku Ketua Program Studi Informatika.
4. Sulistyo Dwi Sancoko S.Si.,M.Sc.. selaku Dosen Pembimbing tugas akhir.
5. Oka Saputra, selaku pemilik bengkel Oka Motor.
6. Kedua Orang tua yang selalu memberikan doa dan dukungan bagi penulis dalam mengerjakan tugas akhir.

Penulis menyadari bahwa penulisan laporan tugas akhir ini masih jauh dari sempurna. Oleh karena itu penulis mohon kritik dan saran dari pembaca sebagai masukan yang berguna. Semoga penulisan proposal seminar tematik ini dapat bermanfaat bagi pihak yang membutuhkan.

Yogyakarta, 21 -02 - 2024



Alfian Dwi Prasetya

RANCANG BANGUN APLIKASI *BOOKING ONLINE* BENGKEL MOTOR BERBASIS ANDROID

(Studi kasus: Bengkel Oka Motor)

ALFIAN DWI PRASETYA

Program Studi Informatika, Fakultas Sains dan Teknologi

Universitas Teknologi Yogyakarta

Jl. Ringroad Utara Jombor Sleman Yogyakarta

E-mail : alviandwiprasetya10@gmail.com

ABSTRAK

Bengkel Oka Motor Motor merupakan salah satu bengkel yang memiliki banyak pelanggan di daerah Sleman, sehingga mengalami kesulitan dalam mengatur antrian dan memperkirakan estimasi waktu servis dari setiap pelanggannya, selama ini antrian dilakukan secara konvensional, hal ini tidak efisien dan terlalu membuang waktu dimana pelanggan tidak mengetahui kapan waktu reservasi pelayanan servis berakhir sehingga mempersulit pelanggan ketika datang ke bengkel. Untuk itu dibutuhkan suatu sistem yang dapat mengatasi malasah pada antrian dan waktu pelayanan servis oleh karena itu dibangun sebuah aplikasi pemesanan bengkel secara *online* berbasis *android* yang mempunyai fitur untuk menampilkan informasi seputar antrian, data pemesanan servis yang akan disimpan kedalam *database*. Dengan menggunakan metode *Scrum*. Hasil dari penelitian ini adalah sebuah aplikasi *android* untuk melakukan pemesanan antrian, servis bengkel secara *mobile*.

Kata Kunci : Layanan Motor, Sistem Antrian, Metode Scrum

ABSTRACT

Oka Motor Workshop is one of the workshops that has many customers in the Sleman area, so it has difficulty in managing queues and estimating the estimated service time of each customer, so far the queue has been carried out conventionally, this is inefficient and wastes too much time where customers do not know when the service reservation time ends, making it difficult for customers when coming to the workshop. For this reason, a system is needed that can overcome the problems in queues and service times, therefore an android-based online workshop booking application is built which has features to display information about queues, service order data which will be stored in the database. By using the Scrum method. The result of this research is an android application for booking queues, mobile workshop services.

Keywords: Motorbike Service, Queuing System, Scrum Method

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR.....	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
2.1 Rumusan Masalah	2
2.2 Batasan Masalah.....	2
2.3 Tujuan Penelitian	2
2.4 Manfaat Penelitian	2
2.5 Sistematika Penulisan	2
BAB II KAJIAN HASIL PENELITIAN DAN LANDASAN TEORI	5
2.1 Kajian Hasil Penelitian.....	5
2.2 Landasan Teori.....	9
2.2.1 Sistem.....	9
2.2.2 Booking.....	10
2.2.3 Bengkel Motor	10
2.2.4 Service.....	10
2.2.5 Aplikasi Mobile.....	11
2.2.6 Metode Scrum	11
2.2.7 Android	14
2.2.8 Flutter	14
2.2.9 Dart.....	15
2.2.10 Visual Studio Code	15
2.2.11 Database	16
2.2.12 Konsep Dasar Database.....	16
2.2.13 MySQL.....	17

2.2.14 XAMPP	17
2.2.15 Flowchart	17
2.2.16 Use Case Diagram.....	18
2.2.17 Activity Diagram.....	19
2.2.1 Entity Relationship Diagram (ERD)	20
2.2.2 Data Flow Diagram (DFD)	22
2.2.3 Black Box Testing.....	23
BAB III METODOLOGI PENELITIAN	24
3.1 Profil.....	24
3.1.1 Lokasi	25
3.1.2 Visi dan Misi	25
3.1.3 Struktur Organisasi.....	26
3.2 Bahan/Data	27
3.2.2 Data Yang Diperoleh.....	27
3.2.3 Metode Pengumpulan Data	28
3.3 Aturan Bisnis.....	29
3.4 Tahapan Penelitian	31
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	35
4.1 Analisis Sistem.....	35
4.1.2 Analisis Sistem Yang Diusulkan	35
4.1.3 Analisis Fungsional.....	36
4.1.4 Analisis non-fungsional.....	36
4.1 Arsitektur Sistem.....	37
4.2 Desain Sistem.....	38
4.2.1 Rancangan Unified Modeling Language Diagram.....	38
4.3 Desain Logik	43
4.3.1 Data Flow Diagram (DFD)	43
4.3.2 Entity Relationship Diagram (ERD)	44
4.3.3 Relasi Tabel.....	45
4.4 Perancangan Fisik	45
4.5 Desain Antarmuka.....	49
1.5.1 Desain Halaman login	50
1.5.2 Desain Halaman registrasi.....	50
1.5.3 Desain Halaman reset password.....	51

1.5.4 Desain Halaman utama.....	51
1.5.5 Desain Halaman motor.....	52
1.5.6 Desain Halaman registrasi motor	52
1.5.7 Desain Halaman reservasi	53
1.5.8 Desain Halaman reservasi layanan.....	53
BAB V IMPLEMENTASI HASIL	54
5.1 Implementasi	54
5.2 Source Code	57
5.3 Hasil	62
5.4 Pengujian Sistem.....	76
5.4.1 Rencana Pengujian	76
5.4.2 Pengujian <i>Black Box</i>	77
5.4.3 Pengujian Black Box Role Customer.....	77
5.4.4 Pengujian Black Box Role Admin	78
5.4.5 Kesimpulan Pengujian <i>Black Box</i>	79
BAB VI PENUTUP	80
6.1. Kesimpulan	80
6.2. Saran.....	80
DAFTAR PUSTAKA	81

DAFTAR GAMBAR

Gambar 3.1 Bengkel Oka Motor	24
Gambar 3.2 Lokasi Tempat Bengkel Oka Motor	25
Gambar 3.3 Struktur Organisasi Bengkel Oka Motor.....	26
Gambar 3.4 Data tabel yang diperoleh.....	27
Gambar 3.5 Profil Instagram motor oka	28
Gambar 3.6 Postingan Instagram motor oka.....	29
Gambar 3.7 Aturan bisnis sistem servis	30
Gambar 3.8 Tahap penelitian	32
Gambar 4.1 Flowchart Aplikasi Pendaftaran Service Bengkel Oka Motor	35
Gambar 4.2 Arsitektur Sistem.....	37
Gambar 4.3 Activity Diagram Mengelola Data Mekanik.....	39
Gambar 4.4 Activity Diagram Mengelola Jadwal Service.....	40
Gambar 4.5 Use Case Diagram.....	42
Gambar 4.6 DFD Pelayanan Service yang sedang berjalan.....	43
Gambar 4.7 Entity Relationship Diagram (ERD)	44
Gambar 4.8 Relasi Antar Tabel.....	45
Gambar 4.9 halaman login	50
Gambar 4.10 Halaman registrasi email.....	50
Gambar 4.11 Halaman reset password.....	51
Gambar 4.12 Halaman utama.....	51
Gambar 4.13 Halaman motor	52
Gambar 4.14 Halaman registrasi motor	52
Gambar 4.15 Halaman reservasi	53
Gambar 4.16 Halaman reservasi layanan	53
Gambar 5.1 Tabel users.....	54
Gambar 5.2 tabel jenismerk	55
Gambar 5.3 tabel merk	55
Gambar 5.4 tabel motor.....	56
Gambar 5.5 tabel reset password	56
Gambar 5.6 tabel pelayanan	56
Gambar 5.7 tabel reservasi	57
Gambar 5.8 tabel roles	57
Gambar 5.9 kode program untuk main Activity Admin	58
Gambar 5.10 kode program untuk main Activity user.....	58
Gambar 5.11 kode program untuk pelayanan	59
Gambar 5.12 Login Pengelola Admin/User.....	62
Gambar 5.13 Login Halaman utama	63
Gambar 5.14 Menu motor	64
Gambar 5.15 Registrasi motor	65
Gambar 5.16 Pelayanan	66
Gambar 5.17 Ajuan pelayanan	67
Gambar 5.18 Reservasi	68
Gambar 5.19 histori reservasi	69
Gambar 5.20 Halaman utama Admin.....	70

Gambar 5.21 Halaman Merk.....	71
Gambar 5.22 Halaman ajuan pelayanan.....	72
Gambar 5.23 Halaman <i>approve</i> reservasi	73
Gambar 5.24 Halaman reservasi	74
Gambar 5.25 Halaman daftar reservasi	75

DAFTAR TABEL

Tabel 2.1 Perbandingan Kajian Hasil Penelitian	7
Tabel 2.2 Simbol-Simbol Flowchart.....	18
Tabel 2. 3 Simbol pada <i>Activity Diagram</i>	20
Tabel 3.1 Data jenis jasa pelayanan & harga	27
Tabel 4.1 Skenario Aktor pada <i>Use Case Diagram</i>	36
Tabel 4. 2 Spesifikasi Hardware yang digunakan.....	36
Tabel 4.3 Kebutuhan perangkat lunak(software)	37
Tabel 4.4 skenario aktor <i>use case diagram</i>	41
Tabel 4.5 Rancangan Tabel Users.....	46
Tabel 4.6 Rancangan Jenis_Merk	46
Tabel 4.7 rancangan merk	47
Tabel 4.8 Rancangan motor	47
Tabel 4.9 Rancangan password reset	48
Tabel 4.10 Rancangan pelayanan.....	48
Tabel 4.11 Rancangan Reservasi	49
Tabel 4.12 Rancangan Roles.....	49

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang semakin maju menyebabkan terjadinya ada persaingan antar dunia usaha yang semakin ketat. Hal ini dikarenakan kemajuan teknologi yang sangat pesat menuntut kemampuan pimpinan perusahaan dalam mengalokasikan kebutuhan perusahaannya secara efektif dan efisien. Dalam rangka mendapatkan informasi yang tepat dan akurat, maka diperlukan suatu sistem yang dibuat menurut pola yang terpadu sesuai dengan kondisi dan kebutuhan perusahaan, salah satu pemanfaatan teknologi yang digunakan untuk menjaga hubungan baik antara perusahaan dengan pelanggan adalah sistem layanan *service* berbasis *mobile*.

Bengkel Oka Motor adalah perusahaan yang bergerak dalam bidang jasa *service* kendaraan bermotor yang telah berdiri sejak tahun 2003. Awalnya, Bengkel Oka Motor adalah bengkel umum yang melayani jasa *service* seluruh tipe dan merek kendaraan bermotor. Dalam melakukan pemesanan *service* kendaraan pelanggan harus datang langsung ke bengkel selanjutnya pelanggan menyiapkan kendaraan dan memberikan informasi *service* kepada admin hal ini mengakibatkan terjadi pengantian dan pelanggan tidak mengetahui berapa lama menunggu, sehingga menyebabkan pelanggan tidak dapat memprediksi waktu antrian. Masalah tersebut dapat diatasi dengan membuat aplikasi *booking online* motor untuk mempermudah bengkel dalam mengelola data layanan dan menghasilkan laporan yang akurat dengan adanya aplikasi *booking online* motor juga dapat mengurangi komplain dari pelanggan. Selain itu pemanfaatan teknologi informasi pada bengkel juga dapat meningkatkan pelayanan terhadap pelanggan. Dari latar belakang maka dilakukan penelitian dengan judul “Aplikasi *booking online* Motor Berbasis Mobile (Studi Kasus: Bengkel Oka Motor)” diharapkan dapat mempermudah pelanggan dalam memesan jasa *service* motor dimana saja dan kapan saja, pelanggan dapat menentukan jadwal *service* yang diinginkan oleh pelanggan dan instansi.

2.1 Rumusan Masalah

Berdasarkan latar belakang yang sudah dikemukakan maka rumusan masalah pada penelitian ini adalah bagaimana implementasi aplikasi *booking online service* motor pada bengkel Oka berbasis *Android*:

2.2 Batasan Masalah

Batasan masalah pada aplikasi yang dikembangkan menggunakan *android* sebagai berikut:

1. Jadwal *service* yang dilakukan costumer ditentukan oleh admin bengkel Oka Motor.
2. Aplikasi *booking* online motor yang akan dibuat ini hanya mencakup layanan *home service* dan layanan darurat khususnya pelanggan bengkel Oka motor.
3. Rancang bangun aplikasi hanya sampai pengujian aplikasi dengan menggunakan metode *blackbox testing*

2.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, tujuan penelitian ini adalah dapat menghasilkan Rancang Bangun Aplikasi *Booking online* bengkel motor Berbasis Android sehingga mempermudah tiap orang dalam melakukan pemesanan dan pengecekan layanan bengkel secara *online*.

2.4 Manfaat Penelitian

Manfaat penelitian yang diharapkan dari penelitian ini yaitu:

1. Bagi intansi Aplikasi *booking online* motor yang dibuat mampu mengubah sistem manual menjadi terkomputerisasi secara mobile.
2. Mempermudah pengolahan dan penyimpanan data sehingga menghasilkan laporan sesuai dengan yang dibutuhkan.
3. aplikasi *booking online service* motor ini juga dapat meningkatkan pelayanan dan mengurangi keluhan pelanggan.

2.5 Sistematika Penulisan

Terdapat enam bab pokok bahasan dalam sistematika dari penyusunan penelitian, yaitu:

BAB I PENDAHULUAN

Bab ini akan berisi mengenai latar belakang, perumusan masalah, batasan masalah, tujuan, dan manfaat penelitian yang menjadi pendahuluan dalam penggerjaan Laporan Tugas Akhir mengenai Rancang Bangun Apikasi *Booking Online* Bengkel motor berbasis *Android*, dengan menggunakan Metode Scrum.

BAB II TINJAUAN PUSTAKA DAN TEORI

Bab ini berisi tentang tinjauan pustaka dan kajian teori. Tinjauan pustaka hasil penelitian merupakan tinjauan pustaka tentang penelitian yang pernah dilakukan sebelumnya yang dapat dijadikan sebagai referensi dalam mengembangkan sistem yang akan peneliti bangun. Landasan teori berisi tentang dasar-dasar teori yang digunakan oleh peneliti.

BAB III METODE PENELITIAN

Bab ini Menjelaskan tentang gambaran umum bengkel Oka motor seperti profil, lokasi, visi & misi, struktur organisasi, data yang diperoleh, metode pengumpulan data, aturan bisnis, dan tahapan penelitian.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis & perancangan sistem yang akan diusulkan yang terdiri dari analisis sistem yang diusulkan, analisis fungsional dan non fungsional, arsitektur sistem, desain sistem, desain logic, dan desain antarmuka.

BAB V IMPLEMENTASI DAN HASIL SERTA PEMBAHASAN

Bab ini berisi penjelasan mengenai implementasi dari hasil rancangan sistem yang telah dibuat yang terdiri dari implementasi, source kode,dan hasil program yang dibuat dengan bahasa pemograman dart menggunakan metode scrum..

BAB VI PENUTUP

Bab ini berisi menyajikan kesimpulan dan saran dari keseluruhan hasil penelitian pembuatan sistem dari awal hingga akhir.

DAFTAR PUSTAKA

Berisi mengenai semua sumber bacaan yang dijadikan acuan dalam pembuatan laporan.

LAMPIRAN

Berisi dokumen-dokumen pendukung yang berkaitan dengan laporan Tugas Akhir.

BAB II

KAJIAN HASIL PENELITIAN DAN LANDASAN TEORI

2.1 Kajian Hasil Penelitian

Penelitian yang dilakukan oleh (Raharjo dkk, 2022) telah menghasilkan suatu rancangan aplikasi mobil derek berbasis *android* menggunakan metode *Scrum*. Aplikasi mobil derek dirancang dan dibangun dengan menggunakan *software Android Studio* serta menggunakan bahasa pemrograman *Java* dengan menggunakan metode pengembangan aplikasi *Scrum*. Aplikasi ini dapat melakukan pemesanan mobil derek dengan menampilkan *maps* menggunakan *mapbox* untuk mengetahui titik lokasi jemput, titik lokasi tujuan, dan lokasi supir mobil derek. Selain itu, aplikasi ini dapat menampilkan ketersediaan mobil derek yang berada pada sekitar pelanggan. Dalam penggunaan mobil derek masih terdapat masalah yang dialami oleh pelanggan dan supir mobil derek, yaitu susah untuk mencari posisi keberadaan pelanggan atau alamat yang belum tentu akurat. Dengan perancangan aplikasi pencarian mobil derek dapat menjadi sebuah solusi untuk menyelesaikan masalah. Selain itu juga dapat meningkatkan kualitas pelayanan dan mempermudah pengguna aplikasi pencarian mobil derek.

Penelitian yang dilakukan oleh (Andika, 2022) telah menghasilkan suatu perancangan aplikasi implementasi dengan metode scrum pada rancang bangun aplikasi e-konsulasi dokter berbasis *android*. Permasalahan aplikasi ini memungkinkan pasien berkonsultasi dengan mengirimkan pertanyaan melalui sistem E-Konsul lalu admin menjawab pertanyaan tersebut namun sistem yang ada tidak *real-time*. Permasalahan mengenai beberapa kesulitan yang dialami pasien dalam kegiatan pengobatannya dengan Dokter, perlu dirancang sistem berupa aplikasi e-konsultasi dokter yang memudahkan pasien dalam berkonsultasi dengan dokter yang diinginkan secara *daring* dan *real-time*. Yang tujuannya dapat menghemat waktu dan tenaga yang dikeluarkan antara dokter dan pasien.

Penelitian yang dilakukan oleh (Suwarno dkk, 2023) telah menghasilkan suatu rancang bangun aplikasi pengelolaan penyedia hasil pertanian menggunakan etode *Scrum* berbasis *android*, permasalahan yang sering dijumpai yaitu pemasaran hasil pertanian yang sulit dan sering kali barang dihargai rendah oleh

pengempul/tengkulak. Pemasaran hasil petani hanya dapat dilakukan dekat tempat lahan pertanian tersebut, dan proses transaksi tidak dapat langsung dilakukan antara penjual dan pembeli sehingga petani mengalami kerugian di akibatkan permainan harga yang dilakukan oleh pemasok atau tengkulak, maka dari itu berdasarkan permasalahan tersebut akan dibangun sistem informasi mengenai harga pasar hasil pertanian berbasis *android* (Tanimart). Dengan demikian, diharapkan dapat mengatasi kerugian dari para petani akibat permainan harga pasar.

Penelitian yang dilakukan oleh (Hadi, 2020) telah menghasilkan suatu implementasi metode *Scrum* pada aplikasi *marketplace* sektor pertanian berbasis *android*, permasalahan yang sering dijumpai yaitu para perantara mengambil persentase keuntungan sehingga mengakibatkan harga produk pertanian cukup tinggi, dan mengakibatkan harga padi dari hasil panen petani memiliki nilai yang rendah. Perbedaan harga saat berada di tangan pelanggan dengan harga saat panen memiliki selisih hingga 100%, petani juga tidak dapat memotong rantai perdagangan tersebut karena sudah menjadi tradisi, maka dari itu berdasarkan permasalahan tersebut, untuk mengurangi rantai perdagangan diperlukan suatu sistem yang dapat memberikan informasi serta tempat pemasaran untuk produk pertanian dan juga dapat digunakan untuk pembelian langsung oleh pelanggan. Sistem yang dapat memenuhi kebutuhan tersebut adalah *marketplace* dengan membuat tempat pemasaran secara online.

Penelitian yang dilakukan oleh (Yanuaresta dkk, 2023) telah menghasilkan suatu analisis dan perancangan aplikasi Pengembangan *Integrated Ticketing System* untuk Meningkatkan Transparansi Pengelolaan Wisata Pantai Pasir Kencana dengan menggunakan metode *scrum*, permasalahan yang sering dijumpai yaitu Pengelolaan penjualan tiket saat ini masih menggunakan tiket yang berupa kertas karcis. Setiap unit penjualan tiket memiliki beberapa petugas atau operator yang bertanggung jawab terhadap penjualan tiket dan laporan pendapatan dari penjualan tiket tersebut, maka dari itu berdasarkan permasalahan tersebut dibutuhkan perancangan aplikasi sistem parkir Pengelolaan Wisata Pantai Pasir Kencana berbasis *android* untuk menyelesaikan masalah tersebut.

Berdasarkan hasil dari beberapa tinjauan pustaka tersebut, penulis menyimpulkan hasil dari penelitian terdahulu daa, tabel 2.1 berikut..

Tabel 2.1 Perbandingan Kajian Hasil Penelitian

No	Judul	penulis	Hasil & Kesimpulan
1	Perancangan Aplikasi Mobil Derek Berbasis Android Menggunakan Metode <i>Scrum</i>	Kukuh Primadito Raharjo, Gita Fadila Fitriana, Novian Adi Prasetyo (2022)	<ul style="list-style-type: none"> 1. Memudahkan pengguna/user untuk mendapatkan pertolongan pertama apabila kendaraan mobil mengalami kerusakan. 2. Memudahkan petugas mobil derek untuk menemukan pengguna/user yang mengalami kendaraan rusak melalui aplikasi ini. 3. Fitur: Aplikasi ini dapat melakukan pemesanan mobil derek dengan menampilkan <i>maps</i> menggunakan <i>mapbox</i> untuk mengetahui titik lokasi jemput, titik lokasi tujuan, dan lokasi supir mobil derek. Selain itu, aplikasi ini dapat menampilkan ketersediaan mobil derek yang berada pada sekitar pengguna.
2	perancangan aplikasi implementasi dengan metode scrum pada rancang bangun aplikasi e-konsultasi dokter berbasis <i>android</i> .	Rendi Andika (2022)	<ul style="list-style-type: none"> 1. Manfaat bagi pasien Mempersingkat waktu dalam proses pemeriksaan dengan Dokter yang dibutuhkan sekaligus dapat dilakukan dimanapun karena proses dilakukan secara <i>online</i>. 2. Manfaat bagi RSAL. DR. Komang Makes Belawan, meningkatkan keefektifan dan keefisienan pelayanan terhadap pasien khususnya pada pelayanan konsultasi dengan Dokter. Sehingga pasien dapat lebih puas terhadap pelayanan diberi oleh objek penelitnnian.

Lanjutan **Tabel 2.1** Perbandingan Kajian Hasil Penelitian

No	Judul	penulis	Hasil & Kesimpulan
2			<p>3. fitur:</p> <ul style="list-style-type: none"> - Aplikasi memiliki fitur janji temu dengan Dokter secara offline yang dapat digunakan user sesuai dengan ketersediaan jadwal dari Dokter yang bersangkutan. - Aplikasi dapat menampilkan jadwal praktek seluruh Dokter di RSAL. DR. Komang Makes Belawan melalui aplikasi yang nantinya akan dibangun.
3	Rancang bangun aplikasi pengelolaan penyedia hasil pertanian Menggunakan Metode Scrum berbasis android	Joko Suwarno, Galuh Saputri, Irham Anas (2023)	<ol style="list-style-type: none"> 1. Bagi Petani: Mempermudah petani dalam memasarkan produk hasil pertanian pada start-up aplikasi TaniMart berbasis android dan Membantu petani untuk menyampaikan informasi tentang kondisi pertanian diwilayah mereka masing-masing secara realtime. 2. bagi pengusaha: Mempermudah pengelolaan penyedia hasil pertanian. 3. fitur: -jual beli barang yang bisa dilakukan oleh sesama petani sehingga memudahkan untuk melakukan transaksi barang secara realtime -mengetahui harga barang secara riltime.
4	implementasi pada aplikasi <i>marketplace</i> sektor pertanian berbasis android	Steven Yuhansen Hadi (2020)	<ol style="list-style-type: none"> 1. Bagi Petani: menjual harga padi sesuai harga pasar yang berlaku. 2. Bagi developer: membangun sistem <i>marketplace</i> Duta Tani Store agar mendapatkan hasil kualitas aplikasi yang baik. <p>Fitur: mengetahui harga barang secara riltime.</p>
5	Pengembangan <i>Integrated Ticketing System</i> untuk Meningkatkan Transparansi Pengelolaan Wisata Pantai Pasir Kencana	Dianna Yanuarestaa, Ika Novita Dewib, Ramadhan Rakhmat Sanic (2023)	<ol style="list-style-type: none"> 1. user: Memudahkan user untuk menggunakan sistem parkir. 2. admin: Membantu mempermudah admin dalam melakukan rekap data informasi yang ada di lapangan. 3. fitur: pencarian area parker yang kosong bagi pengguna.

Data penelitian ini menghasilkan suatu aplikasi *booking online* berbasis *android* dengan menggunakan metode *scrum*. Permasalahan aplikasi ini dalam melakukan pemesanan *service* kendaraan pelanggan harus datang langsung ke bengkel selanjutnya pelanggan menyiapkan kendaraan dan memberikan informasi *service* kepada admin hal ini mengakibatkan terjadi pengantian dan pelanggan tidak mengetahui berapa lama menunggu, sehingga menyebabkan pelanggan tidak dapat memprediksi waktu antrian, maka pembuatan dari aplikasi diharapkan dapat mempermudah pelanggan dalam memesan jasa *service* motor dimana saja dan kapan saja, pelanggan dapat menentukan jadwal *service* yang diinginkan oleh pelanggan dan instansi. Manfaat bagi pengguna adalah dapat memudahkan pengguna untuk melakukan *booking service online* baik dimanapun dan kapanpun berada. Manfaat bagi Admin adalah dapat menggunakan aplikasi *booking online service* motor ini dengan lebih efisien versi mobile dan juga dapat meningkatkan pelayanan dan mengurangi keluhan pelanggan. Fitur yang terdapat pada aplikasi ini seperti pelayanan *service* kendaraan baik ringan maupun berat, serta untuk meminimalkan waktu antrian pelanggan dalam melakukan pelayanan pada Bengkel Oka motor.

2.2 Landasan Teori

2.2.1 Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan. Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Karim dkk, 2019) Dari beberapa kutipan di atas maka penulis dapat menyimpulkan bahwa sistem informasi adalah sistem di dalam suatu instansi atau organisasi perusahaan yang mempertemukan kebutuhan pengolahan transaksi harian dan memberikan laporan-laporan atau informasi yang dibutuhkan.

2.2.2 Booking

Booking yaitu pemesanan fasilitas yang diantaranya akomodasi, meal, seat pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, night club, *discountegue* dan sebagainya (Hidayat (2023) Sedangkan menurut (McLoad, 2019) yang dimaksud booking adalah proses, perbuatan, cara memesan (tempat, barang, dan sebagainya) kepada orang lain. Berdasarkan pengertian para ahli diatas dapat disimpulkan bahwa booking adalah suatu aktivitas yang dilakukan oleh konsumen sebelum membeli atau melakukan service untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem booking yang baik.

2.2.3 Bengkel Motor

Bengkel adalah tempat di mana seseorang mekanik melakukan pekerjaannya melayani jasa perbaikan dan perawatan kendaraan. Bengkel umum kendaraan bermotor adalah bengkel umum yang berfungsi untuk membetulkan, memperbaiki, dan merawat kendaraan bermotor agar tetap memenuhi persyaratan teknis dan layak jalan Fauzy (2020) Hal ini guna memenuhi tuntutan PP No. 44 Tahun 1993 tentang Kendaraan dan pengemudi pasal 126, 127, 128, dan 129, dinyatakan bahwa setiap kendaraan bermotor harus memenuhi persyaratan teknis dan kelayakan kendaraan bermotor. Menurut Yoga (2019), bengkel merupakan suatu usaha jenis wirausaha kecil dan menengah yang bergerak dalam bidang jasa pelayanan perbaikan, baik itu sepeda motor atau mobil. Usaha bengkel sepeda motor adalah usaha yang melakukan perbaikan sepeda motor agar dapat kembali berjalan dengan baik sesuai dengan keinginan pemilik atau bentuk asli dari sepeda motor tersebut (Yuda, 2020).

2.2.4 Service

Pengertian *service* menurut Simatupang (2019) adalah suatu aktivitas atau manfaat yang tidak memiliki wujud dan dapat diserahkan kepada orang lain yang bertindak sebagai customer dan tidak bersifat kepemilikan. Pengertian service menurut Kotler (2019) yaitu suatu tindakan, jasa, maupun kinerja yang ditawarkan kepada orang lain dan tidak memiliki wujud apapun serta tidak menimbulkan kepemilikan. Menurut beberapa pengertian para ahli yang telah dijabarkan diatas, maka dapat ditarik suatu kesimpulan dari pengertian service yaitu seluruh tindakan,

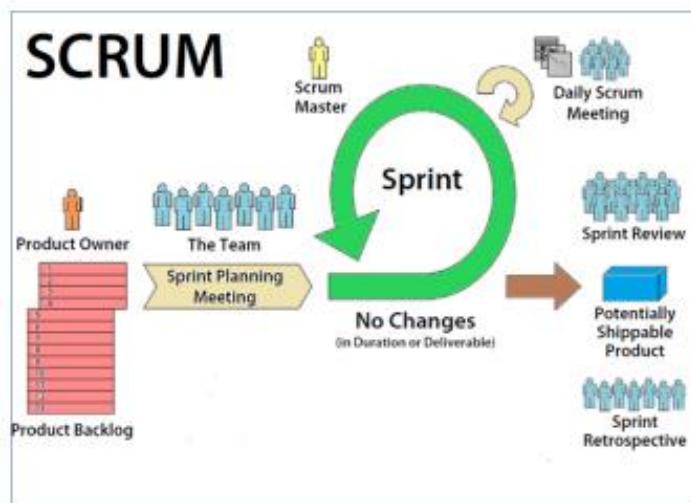
kinerja, manfaat ataupun aktivitas yang dapat diserahkan melalui satu pihak ke pihak lainnya, yang memiliki sifat dasar tidak berwujud dan tidak menimbulkan kepemilikan apapun.

2.2.5 Aplikasi Mobile

Aplikasi *mobile* merupakan perangkat lunak yang telah dirancang khusus untuk berada pada *mobile platform* (misalnya, *iOS*, *Android*, atau *windows Mobile*). Di sebagian besar, aplikasi *mobile* mencakup antarmuka pengguna yang mengambil keuntungan dari mekanisme interaksi unik yang disediakan oleh *mobile platform*, serta memiliki kemampuan pemrosesan lokal untuk mengumpulkan, menganalisa, dan memformat informasi dengan cara yang terbaik. Selain itu, aplikasi *mobile* menyediakan kemampuan penyimpanan yang terus-menerus dalam platform (Aditya et al., 2021)

2.2.6 Metode Scrum

Menurut Andipradana & Hartomo (2021) metode *Scrum* merupakan model pengembangan sistem menggunakan *agile method*, metode *agile method* sendiri adalah proses yang menerapkan siklus pendek yang berulang, melibatkan pengguna untuk membangun, memprioritaskan, dan memverifikasi kebutuhan. Pengembangan sistem menggunakan model *scrum* cocok untuk pengembangan sistem yang memiliki banyak perubahan.



Gambar 2.1 Scrum Workflow

Customer requirement merupakan kebutuhan pengguna yang akan digunakan untuk menyusun *product backlog* yang terdiri dari fitur yang akan digunakan pada aplikasi. Setelah menyusun *product backlog* kemudian menyusun *sprint backlog* yang berisi rencana bagaimana sprint akan dikerjakan, *sprint* dilakukan setelah selesai menyusun *sprint backlog* yang kemudian akan menghasilkan *product backlog* yang sudah selesai yang disebut increment.

A. Artefak Scrum

Artefak Scrum merepresentasikan pekerjaan atau nilai, bertujuan untuk menyediakan transparansi, dan kesempatan-kesempatan untuk peninjauan dan adaptasi (Andipradana & Hartomo, 2021) Scrum memiliki beberapa artefak yaitu.

1. *Product Backlog*

Product Backlog adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. *Product Owner* bertanggung jawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya. Pada awal pembuatannya, *product backlog* hanya berisi daftar kebutuhan dan fitur yang dibutuhkan pada saat itu. Seiring dengan berkembangnya produk dan lingkungan dimana produk tersebut berkembang maka isi dari *product backlog* tersebut juga berubah. *Product backlog* bersifat dinamis (Schwaber & Sutherland, 2013, hal. 235). *Product backlog* menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan terhadap produk di masa mendatang.

2. *Sprint backlog*

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di Sprint, juga di dalamnya terdapat rencana untuk mengembangkan produk dan merealisasikan *Sprint Goal*. *Sprint backlog* menampilkan semua pekerjaan yang dibutuhkan untuk mencapai *sprint goal* yang dibuat oleh tim pengembang. Tim pengembang dapat memodifikasi *sprint backlog* sepanjang sprint berlangsung dan *sprint backlog* dapat berubah kapanpun juga sepanjang *sprint* (Schwaber & Sutherland, 2013, hal. 235). Dengan bertambahnya

pekerjaan baru, maka tim pengembang dapat menambahkannya ke dalam *sprint backlog*. Kemudian estimasi sisa pekerjaan juga akan diperbarui.

3. *Sprint*

Sprint merupakan sebuah batasan waktu selama satu bulan atau kurang, dimana *increment* yang selesai sudah dapat digunakan. Setiap *sprint* memiliki definisi mengenai apa yang akan dikembangkan (Schwaber & Sutherland, 2013, hal. 235).

Increment

4. *Increment*

Increment adalah gabungan dari semua item *Product Backlog* yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari *increment* sprint sebelumnya. Setiap pihak harus mengerti definisi “selesai” dari suatu *increment*, walaupun definisi ini berbeda antar tim *scrum*. Setiap anggota harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Saat definisi selesai telah ditetapkan untuk tim scrum kemudian digunakan untuk memastikan apakah pekerjaan mengembangkan *increment*.

B. Ukuran Tim Scrum

Ukuran yang optimal pada tim *scrum* terdiri dari 5 hingga 9 orang, ukuran yang cukup kecil untuk dapat berkoordinasi dengan cepat dan cukup besar untuk menyelesaikan pekerjaan dalam *sprint* (Schwaber & Sutherland, 2013, hal. 235). Dari semua anggota tim *scrum* tersebut harus mencakup perancang, analisis bisnis, pengembang, penguji, dan lain-lain.

C. Kegiatan Scrum

Terdapat 4 kegiatan yang dilakukan pada *scrum* (Schwaber & Sutherland, 2013, hal. 236) yaitu:

1. *Sprint Planning*

Sprint Planning Merupakan rencana pekerjaan yang dilaksanakan di dalam *Sprint*, perencanaan dibuat oleh seluruh anggota tim *scrum*. *Sprint planning* dibatasi 8 jam untuk 1 hari pada sprint yang berdurasi 1 bulan.

2. *Daily Scrum*

Daily Scrum merupakan kegiatan yang dilakukan setiap hari selama maksimal 15 menit. Dilakukan untuk mensinkronisasikan yang dilakukan oleh tim *scrum* dan membuat rencana untuk 24 jam ke depan.

3. *Sprint Review*

Sprint review diadakan pada setiap akhir sprint untuk meninjau kembali *increment* dan merubah *product backlog* bila diperlukan.

4. *Sprint Retrospective*

Sprint Retrospective merupakan sebuah kesempatan bagi tim *scrum* untuk meninjau dirinya sendiri dan digunakan untuk membuat perencanaan mengenai peningkatan yang akan dilakukan pada sprint berikutnya.

2.2.7 **Android**

Android merupakan sistem operasi yang dirancang oleh Google dengan basis *kernel Linux* dan berbagai perangkat lunak lainnya seperti *Open Source*. Pada awal peluncurannya, *android* sendiri hanya ditargetkan untuk mengoperasikan perangkat pada kamera *digital*. Namun karena kurangnya minat pada kamera *digital*, *android* dikembangkan kembali sebagai sistem operasi pada *smartphone* yang akhirnya digunakan oleh banyak orang saat ini. Adapun pengertian *android* menurut para ahli sebagai berikut, “*Android* adalah *platform open source* yang komprehensif dan dirancang untuk *mobile devices*. *Android* dikatakan komprehensif karena menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. Sistem *Android* menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan” (Budiyanto, 2023)

2.2.8 **Flutter**

Flutter adalah sebuah *software development kit* (SDK) buatan *Google* yang berfungsi untuk membuat aplikasi *mobile phone* menggunakan Bahasa pemrograman *Dart*, baik untuk *android* maupun *IOS*. Dengan *flutter*, aplikasi *android* dan *IOS* dapat dibuat menggunakan basis kode dan Bahasa pemrograman yang sama, yaitu *dart*, Bahasa pemrograman yang juga diproduksi oleh *google* pada

tahun 2011. Sebelumnya, aplikasi murni (*native*) untuk *android* perlu dibuat menggunakan Bahasa *Java* atau *Kotlin*, sedangkan aplikasi *iOS* perlu dibuat menggunakan Bahasa pemrograman *Objective-C* atau *swift*. *Flutter* ditujukan untuk mempermudah dan mempercepat proses pengembangan aplikasi *mobile* yang dapat berjalan di atas *android* dan *iOS*, tanpa harus mempelajari dua Bahasa pemrograman secara terpisah. *Flutter* dapat dikatakan sebagai produk *google* yang masih relatif baru. Rilis perdana *flutter*, versi *Alpha* (v.0.0.6), dipublikasikan pada bulan Mei 2017, dan versi v.1.0 merupakan yang stabil saat diterbitkan oleh *google* (Solahuddin, 2022).

2.2.9 Dart

Menurut (Fadilla & Wiharko, 2023) Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Dart pertama kali dikenalkan pada 10 Oktober 2011. Versi 1.0 dari bahasa pemrograman ini baru dirilis pada bulan November 2013. Versi stabil terbaru adalah Dart 2.1, yang dirilis pada tanggal 15 November 2018. Dart dapat digunakan untuk membuat aplikasi server (berbentuk *commandline interface*), web, maupun *mobile* (Android dan iOS). *Dart* memiliki sintaks yang familiar, dan berbasis kelas, diketik secara opsional, dan *single threaded*. Ini memiliki model konkurensi yang disebut isolat yang memungkinkan eksekusi paralel. Selain menjalankan kode *Dart* di *browser* web dan mengubahnya menjadi *JavaScript*, Pengguna juga dapat menjalankan kode *Dart* di baris perintah. dihosting di mesin *virtual Dart*, memungkinkan klien dan bagian server dari aplikasi pengguna dikodekan dalam bahasa yang sama. Sintaks bahasanya sangat mirip dengan *Java*, *C#*, dan *JavaScript*. Salah satu tujuan utama *Dart* adalah agar bahasa itu tampak akrab. Ini adalah skrip *Dart* kecil, yang terdiri dari satu fungsi yang disebut *main*.

2.2.10 Visual Studio Code

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Microsoft Visual Studio dapat

digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *SilverLight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform *Microsoft .NET Framework* 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform *.NET Framework* 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk *.NET Framework* 1.1, dan Visual Studio 2002 ditujukan untuk *.NET Framework* 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0, (Permana & Romadlon, 2019).

2.2.11 Database

Database adalah sekumpulan file data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga data-data tersebut dapat diakses dengan mudah dan cepat, dan diproses menjadi sebuah informasi yang lebih bermanfaat. (Taqiyah, 2018) Database adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir sehingga mampu dan disimpan secara terintegrasi dengan menggunakan metode tertentu sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. Catatan yang mirip database sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi, dan kumpulan data yang berhubungan dengan bisnis.

2.2.12 Konsep Dasar Database

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang disimpan di dalamnya dan penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan diantara objek tersebut.

Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi data bentuk table-table yang saling berhubungan dimana setiap table terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antara table diwakili dengan menggunakan nilai yang sama antara table. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antara table.

2.2.13 MySQL

Menurut (Nasution, 2018), *MySQL* merupakan *software* yang tergolong sebagai DBMS (*database management system*) yang bersifat *open source*. *MySQL* sebenarnya produk yang berjalan pada *platform Linux*. Oleh karena itu sifatnya *open source*, maka *MySQL* dapat dijalankan pada semua *platform* baik *Windows* maupun *Linux*. Selain itu, *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi Multi User (Banyak Pengguna). Saat ini *database MySQL* telah digunakan hampir oleh semua programer *database*, apalagi dalam pemrograman web.

2.2.14 XAMPP

XAMPP adalah perangkat lunak gratis, yang mendukung banyak sistem operasi. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL, database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (General Public License) dan gratis. (Mahyasari, 2022).

2.2.15 Flowchart

Flowchart atau diagram alir merupakan representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri

dari sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan menerima input, pemrosesan *input*, dan diakhiri dengan menampilkan *output*. Dalam penulisan *Flowchart* dikenal dua model, yaitu sistem *flowchart* dan program *flowchart*. Sistem *Flowchart* adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file di dalam media tertentu, sedangkan program *flowchart* adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program.

Tabel 2.2 Simbol-Simbol Flowchart

No	Simbol	Keterangan
1.		<i>Flow Direction</i> : Arah alur menuju / menghubungkan ke simbol lainnya
2.		<i>Terminal</i> : Simbol awal atau akhir yang ada pada sistem
3.		<i>Decision</i> : Digunakan untuk memilih proses berdasarkan kondisi tertentu
4.		<i>Input / output</i> : Simbol <i>input / output</i> tanpa tergantung jenis peralatan
5.		<i>Processing</i> : Menunjukkan proses/ pengolahan yang sedang dikerjakan sistem

Sumber (Abdurahman dkk, 2018)

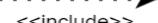
2.2.16 Use Case Diagram

Use case diagram menggambarkan kelakuan (behavior) sistem informasi yang akan dibuat. *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sugiarti, 2013: 41). Penamaan pada *use case* didefinisikan sesederhana mungkin dan mudah untuk dipahami. Ada dua hal utama dalam *use case*, yaitu 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. Meskipun simbol dari aktor berbentuk 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.

Simbol	Keterangan
	use case Menggambarkan fungsionalitas pada sistem sebagai unit yang bertukar pesan antar unit dengan aktor
	Aktor Abstraksi dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem
	Generalisasi simbol yang digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung
	Asosiasi Menggambarkan asosiasi antara aktor dengan usecase untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	include : menunjukkan suatu use case merupakan fungsionalitas dari use case lainnya
	exclude : menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika kondisi terpenuhi

Sumber Hendini (2016)

2.2.17 Activity Diagram

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (decision) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut (Sugiarti, 2013: 75). Berdasarkan definisi diatas, *activity diagram* merupakan gambar yang dibuat untuk menampilkan dan menjelaskan alur proses yang ada pada suatu sistem dimulai dari awal dijalankannya sistem, proses yang terjadi dan hasil dari proses tersebut. *Activity* diagram menggunakan simbol untuk menjelaskan proses bisnis yang dapat dilihat pada tabel 2.3 berikut.

Tabel 2. 3 Simbol pada *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		Status awal	Sebagai simbol untuk mengawali suatu aktivitas.
2		Penggabungan / <i>join</i>	Penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
3.		Aktivitas	Menunjukkan aktivitas yang dilakukan sistem. Aktivitas biasanya diawali dengan kata kerja.
4.		Percabangan/ <i>decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
5.		Swimlane	Berfungsi untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
6.		Status Akhir	Status akhir yang dilakukan sistem. Semua diagram aktivitas memiliki sebuah status akhir

Sumber (Ahmad Ansori, 2022)

2.2.1 Entity Relationship Diagram (ERD)

Menurut NurmalaSari,dkk (2019), *Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Menurut Fridayanthie dan Mahdiati (2016), ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya. Berikut ini adalah simbol dan keterangan dari komponen ERD :

Tabel 2. 4 Notasi Dalam ERD

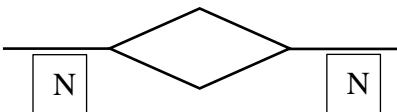
Nama	Gambar	Keterangan
Atribut		Field atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut Multinilai		Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Entitas		Entitas merupakan data inti yang akan disimpan
Garis		Penghubung relasi dan entitas dimana dikedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.
Relasi		Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

Sumber Rosa dan Shalahudin (2018).

Selanjutnya, jenis-jenis relasi dan simbolnya akan disajikan pada Tabel 2.3.

Tabel 2. 5 Relasi Dalam ERD

Nama	Simbol	Keterangan
Satu ke satu		Kerelasian jenis ini terjadi jika kejadian atau transaksi diantara dua entitas yang berhubungan hanya memungkinkan terjadi sebuah kejadian transaksi pada sebuah entitas
Banyak ke satu		Kerelasian jenis ini jika kejadian atau transaksi diantara dua entitas yang berhubungan hanya

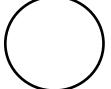
Nama	Simbol	Keterangan
		memungkinkan terjadi satu kali dalam entitas pertama dan dapat terjadi lebih dari satu kali kejadian atau transaksi pada entitas kedua, atau sebaliknya
Banyak ke Banyak		Transaksi diantara dua entitas yang berhubungan memungkinkan terjadi lebih dari satu kali dalam entitas pertama dan entitas kedua

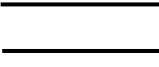
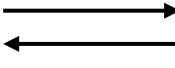
Sumber I Gusti Ngurah Suryantara (2016:2)

2.2.2 Data Flow Diagram (DFD)

Menurut Rusmawan (2019:52), *Data Flow Diagram* (DFD) suatu jaringan yang menggambarkan suatu sistem otomatis, manual atau gabungan dari keduanya yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan. DFD diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD merepresentasikan sistem dengan lengkap melalui beberapa level abstraksi. Level yang paling kecil (level 0) memberikan abstraksi yang lebih global, kemudian level-level berikutnya menunjukkan aliran data dan fungsi yang lebih detail. Simbol yang sering digunakan dalam DFD dijelaskan dalam Tabel 2.4.

Tabel 2. 6 Simbol Dalam DFD

Nama	Simbol	Keterangan
Proses		Aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, bisa berupa manual maupun terkomputerisasi.

Simpanan Data		<i>Data Store</i> adalah kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam <i>Data Store</i> .
Aliran Data		Simbol ini menunjukkan satu data tunggal atau kumpulan logis suatu data, selalu diawali atau diakhiri pada suatu proses
Entitas Luar		<i>External entity</i> / Entitas luar. Simbol ini menunjukkan orang, organisasi atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.

Sumber Rosa dan Shalahudin (2018).

2.2.3 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya. Pada *Black Box Testing* ini dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Blackbox testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi domain. Jadi metode ini bertujuan untuk memeriksa, setelah tahap akhir proyek, apakah perangkat lunak atau aplikasi berfungsi dengan baik, dan melayani penggunanya secara efisien. Biasanya, penguji mencari fungsi yang hilang atau salah; antarmuka, kinerja, inisialisasi program dan kesalahan keluar; struktur data atau kesalahan akses basis data eksternal, (Siadari, 2022).

BAB III

METODOLOGI PENELITIAN

3.1 Profil

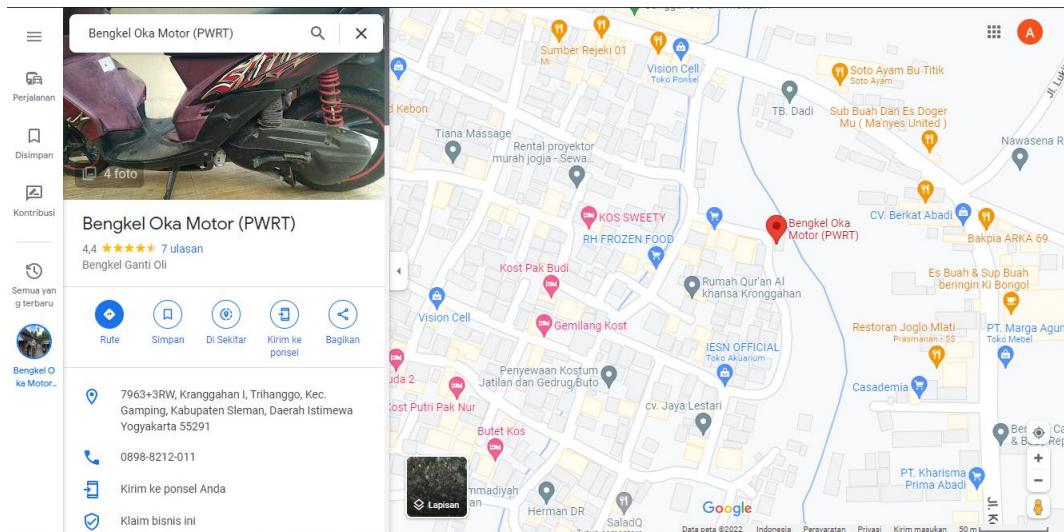
Oka Motor adalah perusahaan yang berjalan dibidang jasa *service* motor beralamat di Jalan Kranggahan I, Trihanggo, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Berdiri pada tahun 2003. Didirikan oleh Oka Saputra setelah menikah dengan menggunakan uang hasil patungan dengan istrinya. Awal mula didirikannya Oka Motor karena dulu ayah dari Oka Saputra juga membuka usaha jasa *service* motor kecil-kecilan, lalu Oka Saputra berinisiatif untuk membuat bengkel barunya sendiri setelah menikah. Sampai saat ini, perkembangan bengkel Oka Motor sangat pesat, bahkan dapat dibilang Oka Motor adalah salah satu bengkel yang sering dikunjungi costumer. Selain menyediakan jasa *service* motor, Oka Motor juga menyediakan *spare part original* dan berbagai macam aksesoris untuk berbagai jenis motor. Dalam kegiatan operasionalnya, Oka Motor ini melayani penjualan secara retail dan wholesale. Oka motor ini juga belum dapat mengakses informasi dengan cepat dikarenakan cara operasional masih manual dan belum terkomputerisasi dengan baik.



Gambar 3.1 Bengkel Oka Motor

3.1.1 Lokasi

Bengkel Oka Motor berada di Jalan Kranggahan I, Trihanggo, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Letak lokasi dapat dilihat pada aplikasi google maps seperti Gambar 3.2 berikut.



Gambar 3.2 Lokasi Tempat Bengkel Oka Motor

3.1.2 Visi dan Misi

1. Visi

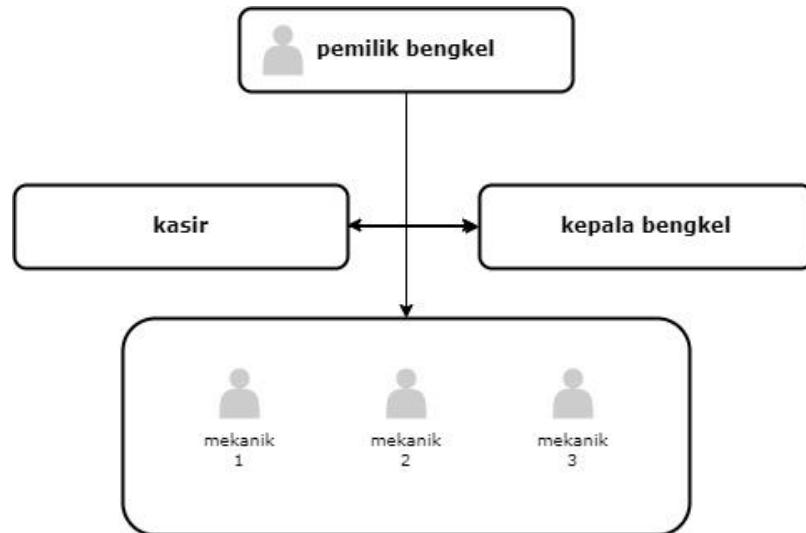
Menjadi bengkel umum terbaik dan terlengkap, pilihan pertama setelah bengkel resmi untuk melakukan perawatan dan perbaikan kendaraan. Menjadi bengkel umum terbaik dan terlengkap, pilihan pertama setelah bengkel resmi untuk melakukan perawatan dan perbaikan kendaraan.

2. Misi

- Melengkapi fasilitas bengkel sesuai dengan kemajuan teknologi otomotif.
- Melaksanakan pelatihan SDM secara berkesinambungan untuk meningkatkan kemampuan personel bengkel agar dapat menangani segala jenis dan merk kendaraan.
- Mengembangkan standarisasi, agar tercipta hasil kerja, biaya perawatan dan perbaikan serta pelayanan yang memuaskan bagi pelanggan bengkel Oka Motor.

3.1.3 Struktur Organisasi

Berdasarkan bagan struktur organisasi pada Gambar 3.3 maka dapat dijelaskan mengenai Tugas pokok dan fungsi setiap bagian organisasi. Penjabaran dari tugas pokok dan fungsi setiap bagian adalah sebagai berikut:



Gambar 3.3 Struktur Organisasi Bengkel Oka Motor

A. Pemilik bengkel

Adapun tugas dari pemilik bengkel adalah sebagai berikut:

1. Mengambil Keputusan setiap kegiatan di Bengkel Oka Motor
2. Verifikasi surat permintaan barang
3. Menyetujui laporan penjualan, pembelian dan laporan persediaan barang.

B. Kepala Bengkel

1. Bertanggung jawab atas segala sesuatu yang terjadi di bengkel.
2. Mengatur segala kegiatan bengkel.
3. Mengadakan pemilihan pegawai

C. Kasir

1. Memposting transaksi sesuai dengan kredit.
2. Membuat laporan sesuai dengan tugas dan dilaporkan pada kepala bengkel.
3. Bertanggung jawab atas segala kekurangan dan kelebihan yang ada.

4. Membuat laporan keuangan berupa pengeluaran untuk pembelian/belanja, serta hasil pemasukan dari bengkel untuk kemudian diserahkan kepada kepala bengkel. Mekanik.

D. Mekanik

1. Merawat dan memperbaiki sepeda motor pelanggan yang masuk ke bengkel.
2. Memberikan kualitas terbaik dalam melakukan service motor.
3. Memberikan penjelasan kepada pelanggan tentang kerusakan kerusakan yang terjadi pada sepeda motor pelanggan.

3.2 Bahan/Data

Bahan data yang sudah diperoleh akan digunakan untuk melakukan penelitian ini dengan melakukan survei, observasi dan wawancara pada tempat bengkel Oka Motor di Jalan Kranggahan I, Trihanggo, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta untuk mengumpulkan data yang berkaitan dengan bengkel Oka Motor. Berikut ini adalah sample data yang ada di tampilkan dalam sistem.

3.2.2 Data Yang Diperoleh

Berikut ini data yang di peroleh dari bengkel Oka Motor mulai dari jenis jasa pelayanan, dan harga yang sudah ditetapkan oleh pemilik bengkel Oka Motor yang akan di tampilkan dalam Tabel 3.1.

Tabel 3.1 Data Jenis Jasa Pelayanan & Harga

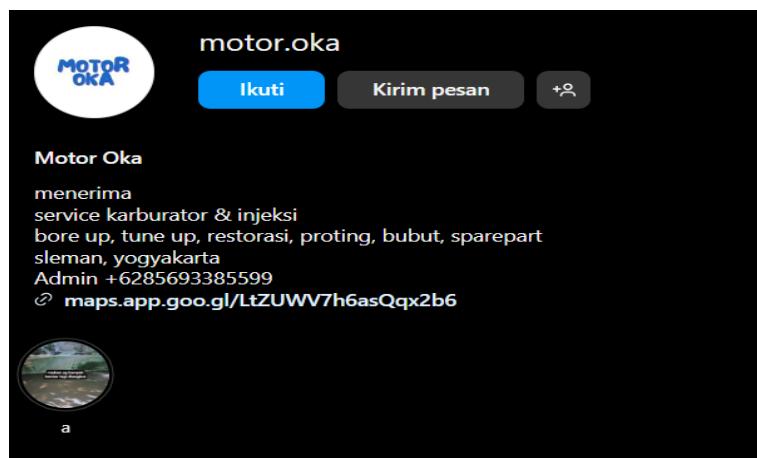
No	Jenis Jasa Pelayanan	Harga
1	Servis Karburator	Rp.30.000
2	Servis Ringan	Rp.75.000
3	Servis injeksi	Rp.35.000
4	Bore up	Rp.500.000
5	Tune up	Rp.200.000
6	Bubut	Rp.150.000
7	Ganti oli mesin	Rp. 40.000
8	Ganti oli gardan	Rp.20.000

Gambar 3.4 Data penelitian yang diperoleh

3.2.3 Metode Pengumpulan Data

Proses pengumpulan data yang dilakukan untuk digunakan sebagai data uji dari penelitian ini adalah dengan melakukan interview, wawancara dan observasi ke lokasi penelitian. Berikut ini tahapan mengumpulkan data yang akan dilakukan sebagai berikut:

- Membuka social media motor oka untuk mendapatkan informasi terbaru terkait dengan jenis servis pada motor oka seperti pada Gambar 3.5 berikut ini <https://www.instagram.com/motor.oka/?hl=id>



Gambar 3.5 Profil Instagram motor oka

- Di dalam akun Instagram @motor.oka tersebut terdapat postingan tampilan jenis servis yang ada pada motor oka ini dan juga macam-macam servis yang ditampilkan pada halaman instagramnya yaitu seperti Gambar 3.6 berikut ini.

C. Observasi

Observasi dengan melakukan pengamatan langsung oleh peneliti pada lokasi tempat penelitian ini yaitu bengkel Oka Motor. Observasi ini lakukan untuk mengumpulkan data jenis servis, harga servis, jenis sparepart, dan harga sparepart yang ada pada bengkel Oka Motor tersebut.

D. Wawancara

Wawancara yang dilakukan dengan orang-orang yang berkaitan dengan pemilik bengkel motor Oka Saputra, pegawainya, dan juga masyarakat sekitar sebagai costumer yang sering melakukan jasa servis kendaran pada bengkel oka motor,

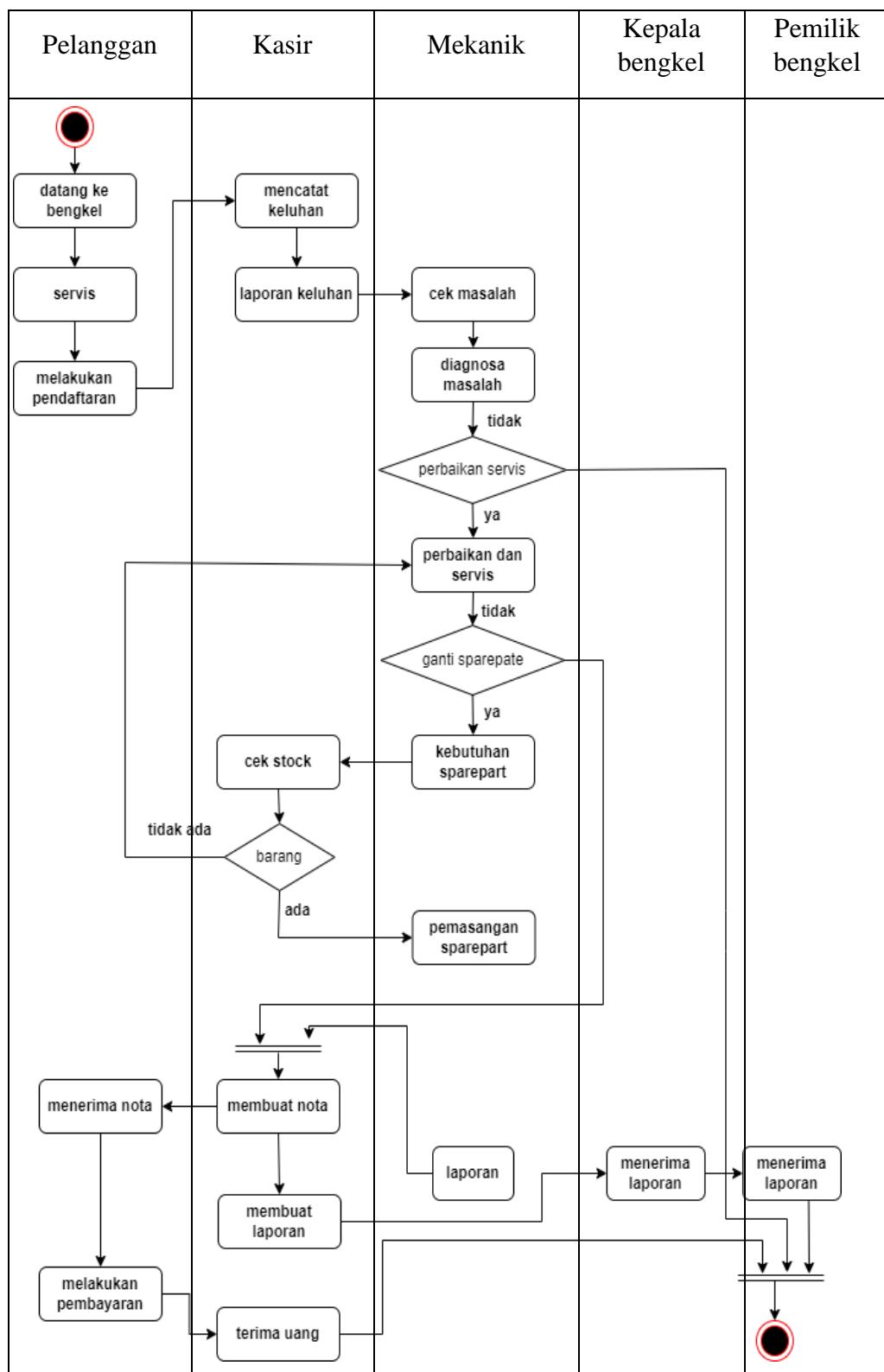
untuk mengumpulkan data yang akan dijadikan bahan untuk membantu penelitian yang sedang dilakukan ini.



Gambar 3.6 Postingan Instagram motor oka

3.3 Aturan Bisnis

Pada aturan bisnis yang berlaku dalam Penelitian ini mengenai bagaimana pelanggan, kasir, mekanik, kepala bengkel, pemilik bengkel mengoperasikan sistem. berikut merupakan alur proses yang ada pada sistem yang digambarkan dalam diagram yang dapat dilihat pada Gambar 3.5 berikut: Gambar 3.7 menjelaskan proses bisnis servis pada sistem pelayanan jasa servis. Pelanggan diarahkan untuk mendaftarkan dan memberikan keluhan kendaraannya, setelah itu pihak kasir akan melaporkan keluhan kendaraan yang ingin di servis ke mekanik, selanjutnya mekanik akan melakukan pengecekan kendaraan dan setelah menemukan diagnosis masalahnya mekanik akan melakukan servis kendaraan sebagaimana mestinya sesuai dengan keluhan dan kerusakan yang ada, jika ada yang perlu diganti dari *sparepart* kendaraan pelanggan.



Gambar 3.7 Aturan bisnis sistem servis

mekanik akan konfirmasi kepada pelanggan apakah ingin diganti saat ini atau tidak? Jika tidak mekanik akan melakukan servis saja, tetapi jika iya mekanik akan melakukan pengecekan terlebih dahulu kebagian kasir apakah *sparepart* yang dibutuhkan tersedia? jika iya mekanik akan melakukan penggantian, jika tidak tersedia mekanik meminta pelanggan mencarinya sendiri di toko lain, dan mekanik menyelesaikan pekerjaannya, setelah itu mekanik melaporkan ke bagian kasir untuk dicatat pada nota rincian harga servis dan harga pergantian *sparepart* dari kendaraan milik pelanggan tersebut jika barang diambil dari stock bengkel, lalu kasir memberitahukan kepada pelanggan rincian pengeluaran dan biaya-biaya yang harus dibayar setelah itu pelanggan melakukan pembayaran, dan kasir memberikan nota lembar ke 1 untuk pelanggan, lembar ke 2 untuk disimpan dibagian kasir untuk data laporan harian yang nanti nya akan di berikan kepada kepala toko dan pemilik bengkel.

3.4 Tahapan Penelitian

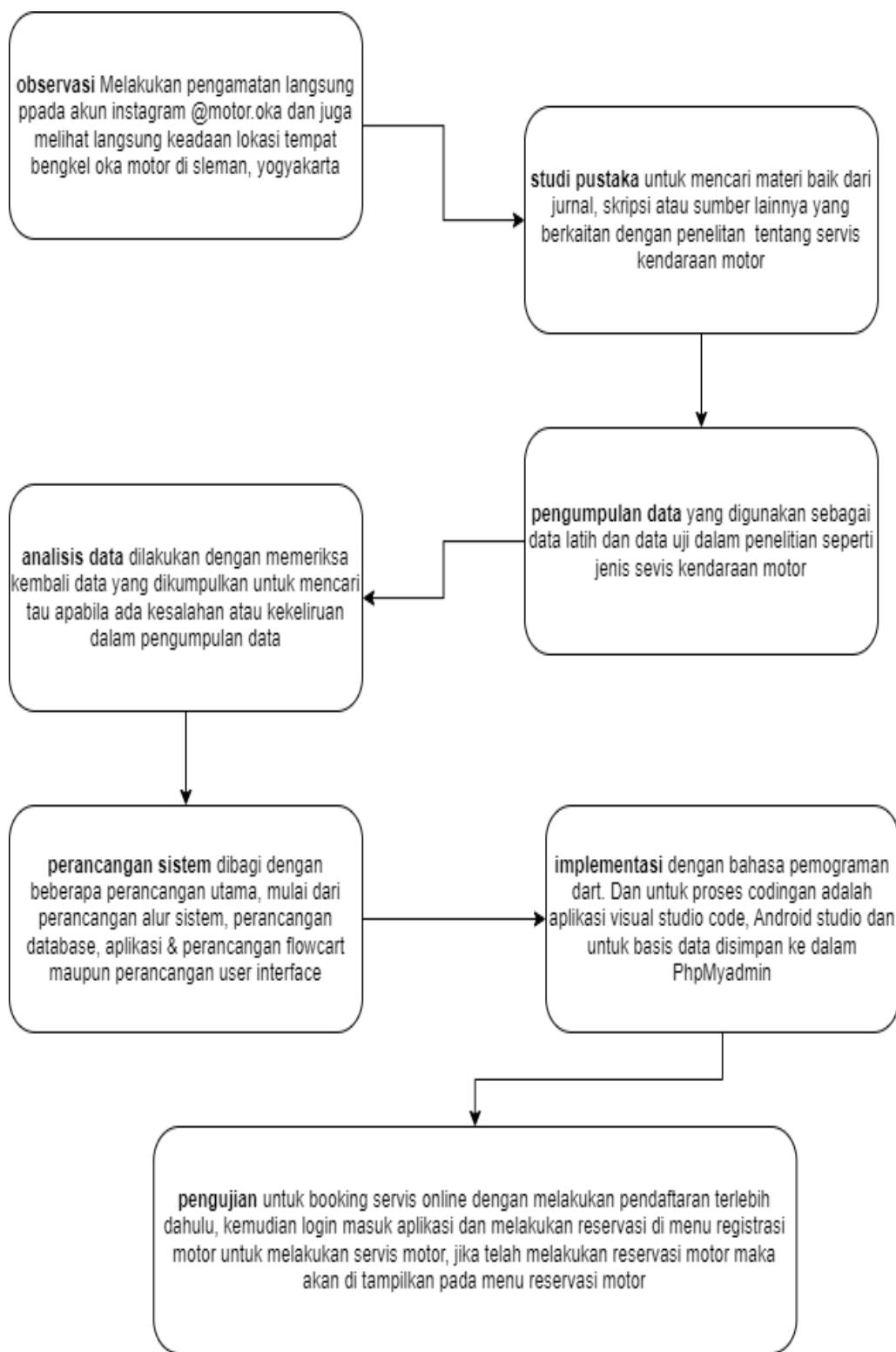
Tahapan penelitian dilakukan oleh peneliti dalam melakukan penelitian untuk menyediakan solusi dari permasalahan yang ada, berikut ini merupakan tahap penelitian yang dapat dilihat pada halaman selanjutnya di Gambar 3.6:

A. Observasi

Observasi dengan melakukan pengamatan langsung oleh peneliti pada akun Instagram @motor.oka dan juga melihat langsung lokasi tempat yang sedang dilakukan penelitian ini yaitu bengkel oka motor Sleman Yogyakarta. Observasi ini lakukan untuk mengetahui kondisi atau keadaan dan untuk melakukan wawancara dengan pemilik dan para pegawai juga masyarakat yang ada di bengkel oka motor tersebut. Sehingga dengan begitu rencana sistem yang akan dibuat dapat sesuai dengan kebutuhan yang diperlukan.

B. Studi Pustaka

Studi Pustaka dilakukan dengan mencari materi baik dari jurnal, skripsi atau sumber lainnya yang berkaitan dengan penelitian yang sedang dilakukan yang akan dijadikan sebagai panduan untuk menyelesaikan penelitian ini.



Gambar 3.8 Tahap penelitian

C. Pengumpulan Data

Proses pengumpulan data dilakukan dengan mengumpulkan data yang akan digunakan sebagai data latih dan data uji dalam penelitian. Data yang dikumpulkan seperti servis karburator & injeksi, restorasi, tune up, bor up, porting, bubut, sparepart dan juga harga sesuai pemesanan yang ditampilkan di media social Instagram @motor.oka atau lainnya dan juga dari hasil observasi.

D. Analisis Data

Analisis data dilakukan dengan memeriksa kembali data yang sudah dikumpulkan untuk mencari tau apabila ada kesalahan atau kekeliruan dalam pengumpulan data yang tidak sesuai agar dapat diperbaiki kembali sebelum dilanjutkan ke tahap berikutnya.

E. Perancangan Sistem

Perancangan sistem dibagi dengan beberapa perancangan utama, mulai dari perancangan alur sistem, perancangan database, perancangan flowchart maupun perancangan user interface. Perancangan user interface berguna untuk merancang tampilan aplikasi agar pengguna mudah menggunakan aplikasi sistem serta mengurangi kemungkinan human error dalam penggunaan sistem. Perancangan alur sistem dan flowchart dilakukan dengan menggunakan perangkat lunak draw io sedangkan database dibangun di phpMyAdmin.. Dalam perancangan ini terdapat tabel, tampilan form dan laporan yang ada pada aplikasi atau sistem ini yaitu sebagai berikut.

1. Tampilan Form

- a) Form tampilan login
- b) Form tampilan registrasi
- c) Form tampilan lupa password
- d) Form tampilan home
- e) Form tampilan registrasi motor
- f) Form tampilan motor
- g) Form tampilan reservasi
- h) Form tampilan history layanan

2. Tabel perancangan sistem
 - a) Tabel jenismerk
 - b) Tabel pemesanan
 - c) Tabel merk
 - d) Tabel motor
 - e) Tabel password resset
 - f) Tabel pelayanan
 - g) Tabel reservasi
 - h) Tabel roles
 - i) Tabel users
3. Report/laporan yang akan ditampilkan pada sistem/aplikasi berupa laporan reservasi

F. Pengujian

Pengujian dilakukan dengan menggunakan metode *Black box* yaitu dengan menguji semua fitur untuk memastikan fungsionalitas dari sistem yang telah dibangun sebelum digunakan. Dengan melakukan pengujian booking atau pemesanan pada aplikasi. Pertama pemesan perlu melakukan registrasi sebagai user agar dapat login ke dalam aplikasi, pada tampilan home jika ingin melakukan pelayanan servis, untuk *booking* servis *online* dengan melakukan pendaftaran terlebih dahulu, kemudian login masuk aplikasi dan melakukan reservasi di menu registrasi motor untuk melakukan servis motor, jika telah melakukan reservasi motor maka akan di tampilkan pada menu reservasi motor.

BAB IV

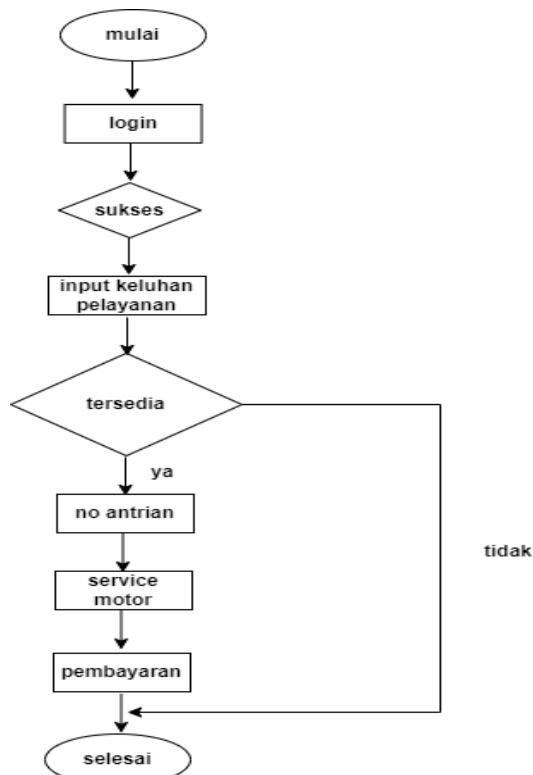
ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Analisis sistem ini dilakukan untuk menggali kekurangan yang ada di dalam penelitian yang pernah dilakukan sebelumnya. Sehingga kekurangan ini dapat dioptimalisasi dengan beberapa update yang akan dilakukan. Selain itu, analisis sistem ini juga bertujuan untuk mempertahankan kelebihan-kelebihan dari penelitian sebelumnya untuk dapat kembali digunakan dalam penelitian saat ini.

4.1.2 Analisis Sistem Yang Diusulkan

Setelah menganalisis dan mengetahui sistem antrian yang ada maka langkah selanjutnya adalah membuat rancangan sistem antrian yang akan dibangun. Perancangan ini dimaksudkan untuk memudahkan dalam melakukan pembangunan sistem antrian dan juga menjelaskan sebuah proses bekerjanya sebuah sistem dalam melakukan pengolahan data input menjadi data output. Rancangan tersebut dapat dilihat seperti pada Gambar 4.2.



Gambar 4.1 Flowchart Aplikasi Pendaftaran Service Bengkel Oka Motor

Gambar 4.1 menjelaskan tentang rancangan aplikasi pendaftaran servis akan di jalankan, dimana terdapat perbedaan dalam proses datang, konsumen mendapatkan kemudahan dengan hanya input keluhan, mendapatkan nomor antrian dan akan mendapat notifikasi jika sudah servis.

4.1.3 Analisis Fungsional

Analisis fungsional merupakan analisis fungsional yang terkait dengan fasilitas yang dibutuhkan oleh sistem secara umum. Bagian ini berisi tabel yang berisi kebutuhan fungsional (fitur dan kemampuan yang dimiliki sistem) hasil analisis.

Tabel 4.1 Skenario Aktor pada *Use Case Diagram*

No	Pengguna	Keterangan
1	Admin	<ul style="list-style-type: none"> ➤ Mengelola data master ➤ Menyetujui <i>service</i> ➤ Menginputkan barang ➤ Menyetujui pembayaran
2	Login	<ul style="list-style-type: none"> ➤ Menerima <i>service</i> ➤ Melakukan reservasi <i>service online</i>
3	Dashboard	<ul style="list-style-type: none"> ➤ Menampilkan hasil reservasi <i>service online</i>

4.1.4 Analisis non-fungsional

Analisis non fungsional yang dihasilkan dari penelitian dibagi menjadi dua dimana hal tersebut mencakup kebutuhan sistem baik dari sisi perangkat keras maupun perangkat lunak agar sistem yang dibangun dapat beroperasi dengan lancar yang dijelaskan sebagai berikut:

1. Kebutuhan perangkat keras (*hardware*)

Dalam proses membangun sistem pengamanan data selama melakukan penelitian, dibutuhkan perangkat keras yang dapat dilihat pada tabel.

Tabel 4. 2 Spesifikasi Hardware yang digunakan

No.	Nama	Spesifikasi
1	Laptop	Asus X441b AMD A9, AMD Dual core A9-9425 2,7 Ghz, Ram 4GB, HDD 500Gb, Win 10
2	Handphone	Xiaomi Redmi Note 8, Processor Octa Core Max 2.01 Ghz 8, Ram 4gb

2. Kebutuhan Perangkat Lunak (*Software*)

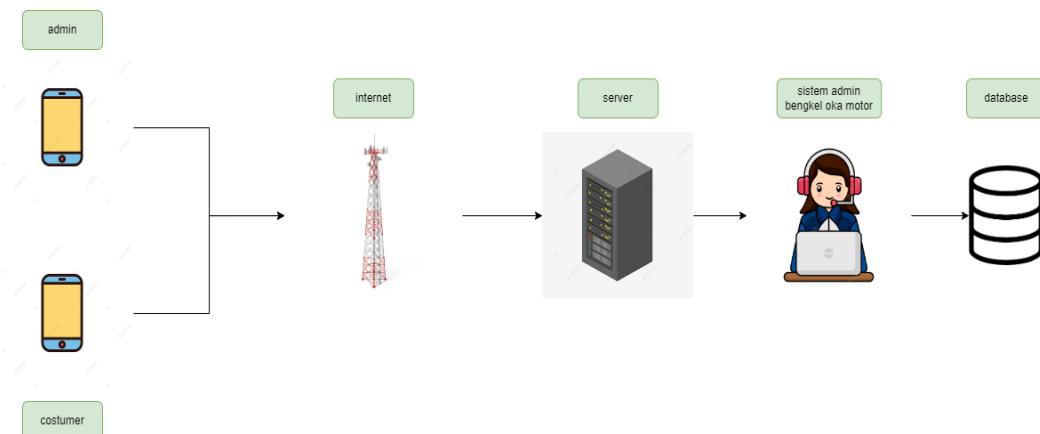
Dalam proses membangun sistem selama penelitian berlangsung, dibutuhkan perangkat lunak yang dapat dilihat pada tabel.

Tabel 4.3 Kebutuhan perangkat lunak(software)

No.	Nama	Spesifikasi
1	Sistem Operasi	<i>Windows 10</i>
2	<i>Browser</i>	<i>Google Chrome</i>
3	Xampp	<i>Xampp Version 3.2.2</i>
4	<i>Database</i>	<i>Mysql</i>
5	Desain	<i>Draw.io dan Figma</i>
6	<i>Office</i>	<i>Microsoft Office Word 2019</i>
7	<i>Text Editor</i>	<i>Visual Studio Code</i>

4.1 Arsitektur Sistem

Arsitektur sistem adalah penjelasan tentang alur proses dari sistem yang dibuat yaitu sistem pemesanan *Booking* aplikasi *Online* Oka motor. Arsitektur sistem menjelaskan bahwa terdapat 2 hak akses yaitu admin dan costumer. Admin bertugas untuk mengelola sistem pemesanan *Booking* aplikasi *Online* dan untuk costumer melakukan pemesanan reservasi pada aplikasi *Booking Online* bengkel motor oka Sleman Yogyakarta.



Gambar 4.2 Arsitektur Sistem

4.2 Desain Sistem

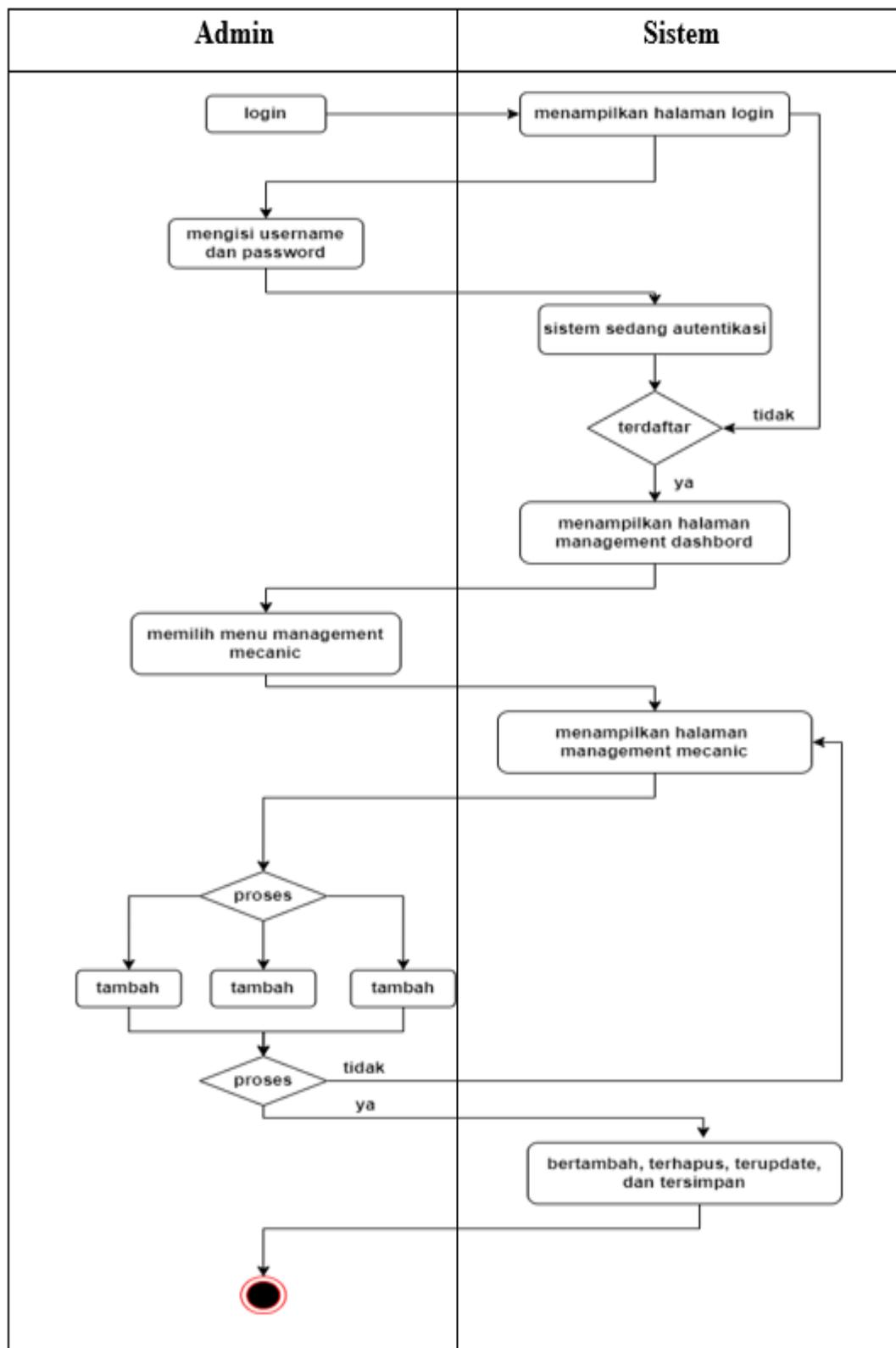
Tahap selanjutnya yang dilakukan dalam penelitian ini adalah desain sistem. Desain sistem bertujuan untuk mempermudah dalam membangun sistem dan juga digunakan untuk dokumentasi. Perancangan adalah gambaran secara terperinci dari metode–metode perancangan dalam pembangunan suatu perangkat lunak. Berdasarkan uraian tersebut, perancangan sistem merupakan tahapan dari pembuatan/perancangan yang akan diimplementasikan dalam perancangan suatu perangkat lunak yang memiliki fungsional tertentu yang menjadi suatu kesatuan yang utuh sehingga dapat berfungsi dengan baik.

4.2.1 Rancangan Unified Modeling Language Diagram

Perancangan UML yang dilakukan yaitu pembuatan *Use Case Diagram*, *Activity Diagram*.

a. Activity Diagram Mengelola Data Mekanik

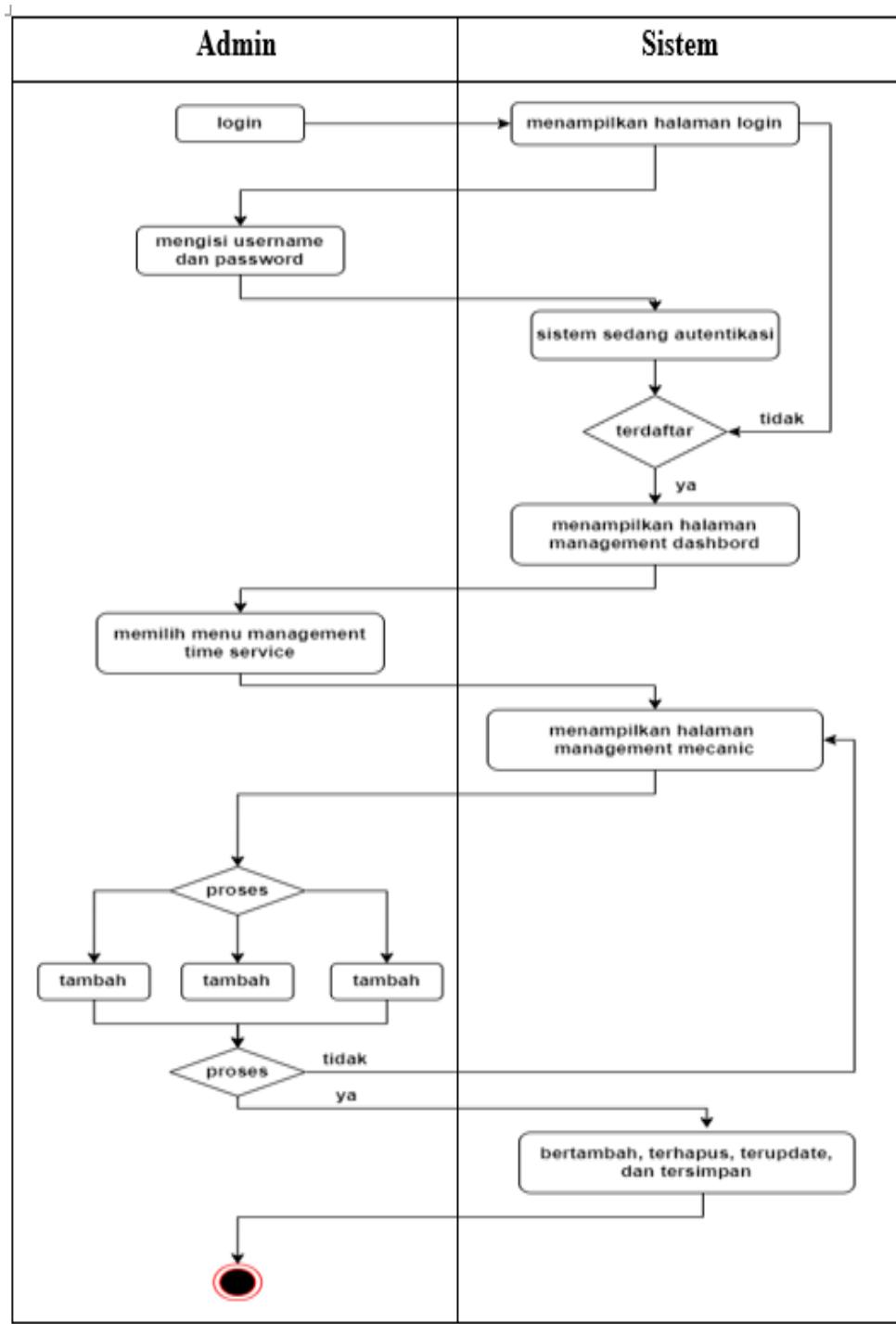
Pada bagian ini dijelaskan bagaimana mengelola data mekanik yang digambarkan dengan proses dimulai dari admin melakukan login kemudian sampai dengan update data mekanik. Deskripsi secara lengkap dari deskripsi *activity diagram* untuk mengelola data mekanik dijelaskan pada Gambar 4.3 *activity diagram* dapat dijelaskan tahapan tentang pengelolaan data mekanik, admin login ke dalam sistem, dan menampilkan halaman login selanjutnya admin akan mengisi *username* dan *password*, dan sistem akan mengautentikasi, jika *username* sudah terdaftar maka sistem akan menampilkan halaman utama. jika tidak maka sistem akan kembali ke halaman login Admin dapat melakukan tambah, edit, dan hapus sistem jika proses dilakukan maka data akan bertambah, terupdate, terhapus dan tersimpan jika tidak kembali ke halaman *management mechanic* admin telah selesaimekelola mekanik, dijelaskan tentang deskripsi dari aktor pada *activity diagram* mengelola data mekanik.



Gambar 4.3 Activity Diagram Mengelola Data Mekanik

b. Activity Diagram Mengelola Jadwal Service

Pada bagian ini dijelaskan bagaimana mengelola data jadwal servis yang digambarkan dengan proses dimulai dari admin melakukan login kemudian sampai dengan update data jadwal servis.



Gambar 4.4 Activity Diagram Mengelola Jadwal Service

Pada Gambar 4.4 *activity* diagram dapat dijelaskan tahapan tentang mengelola jadwal servis yang dimulai dari login admin sampai dengan *update* data. Deskripsi secara lengkap dari *activity* diagram mengelola jadwal servis dapat dilihat pada Gambar 4.4 *activity* diagram, admin login kedalam sistem, sistem menampilkan halaman login, selanjutnya admin mengisi *username* dan *password*, sistem mengautentikasi, jika terdaftar menampilkan halaman utama, jika tidak kembali ke halaman login, selanjutnya admin dapat melakukan tambah, edit, dan hapus sistem jika proses itu dilakukan data bertambah, terupdate, terhapus dan tersimpan, jika tidak kembali ke halaman *management* mekanik admin telah selesai mengelola jadwal servis.

c. Usecase Diagram

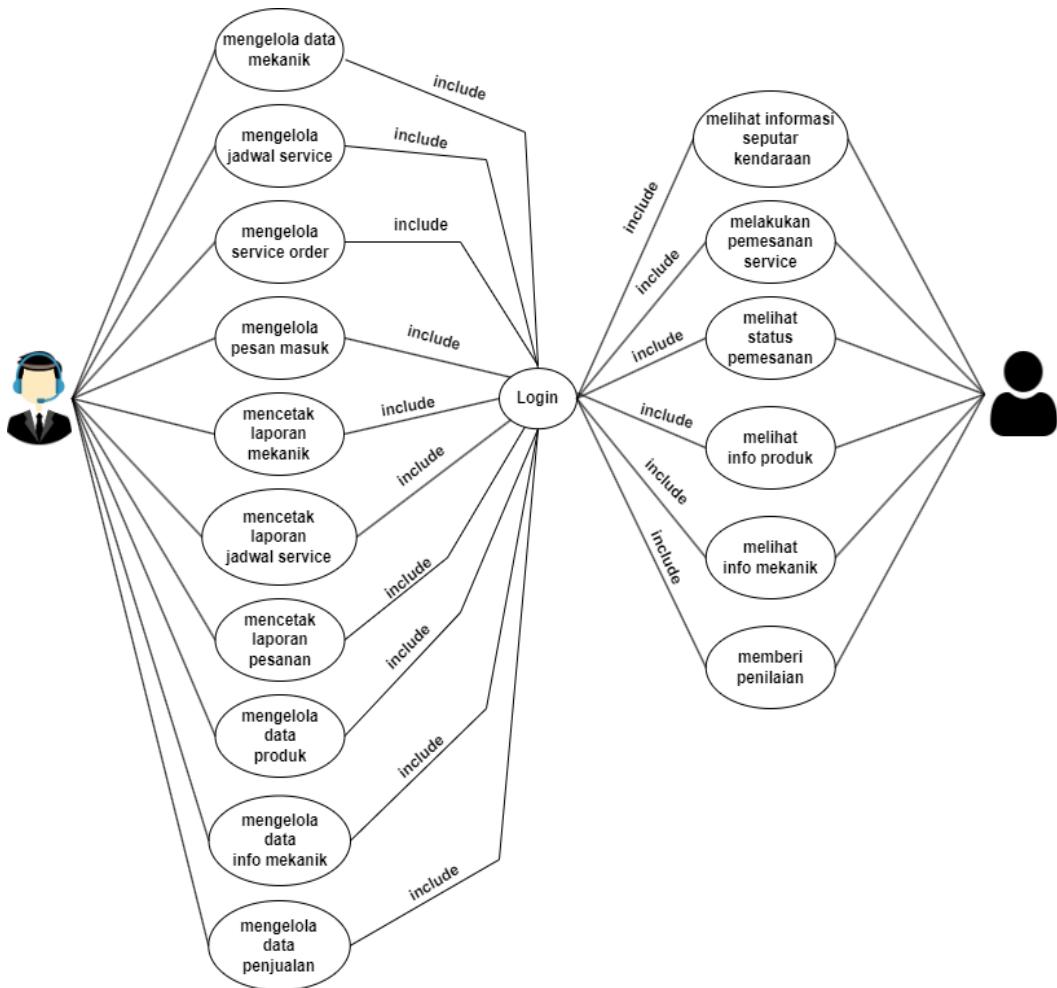
Skenario *use case* mendeskripsikan actor dan fungsi yang digambarkan dalam *use case* diagram. Berikut deskripsi actor yang ada dalam *use case* pelayanan *Service Kendaraan* dalam aplikasi *Booking Service Online* kendaraan motor. Dijelaskan pada tabel 4.4 berikut.

Tabel 4.4 skenario aktor *use case* diagram

No	Use Case	Deskripsi
1	admin	Admin merupakan aktor yang dapat mengatur pemberian informasi mekanik, daftar mekanik, daftar waktu pengkerjaan service, mengkonfirmasi waktu pengkerjaan service, daftar order service, mengkonfirmasi order pelanggan masuk, mengkonfirmasi kedatangan pelanggan, mengkonfirmasi service telah selesai
2	Pelangan	Pelanggan merupakan aktor yang dapat melihat seputar informasi, melakukan pesanan service, melihat status pesanan service

Selanjutnya, diagram *use case* untuk aplikasi ini dijelaskan pada Gambar 4.5 Berikut.

1. Admin
 - a) Login : proses masuk ke dalam sistem untuk dapat melakukan semua aktivitas
 - b) Home : proses yang dilakukan oleh admin seperti melihat dan mengedit informasi.



Gambar 4.5 Use Case Diagram

- c) Menu Utama : proses yang dilakukan admin seperti, melihat, mencari, menambah, mengedit menghapus dan mencetak laporan data mekanik, data waktu *service*, dan pesanan *service* pelanggan
- d) Sistem : proses yang dapat dilakukan oleh admin seperti, melihat, mencari, menambah, mengedit, dan mencetak laporan data mekanik, data waktu *service*, dan pesanan *service* pelanggan.

2. Pelanggan

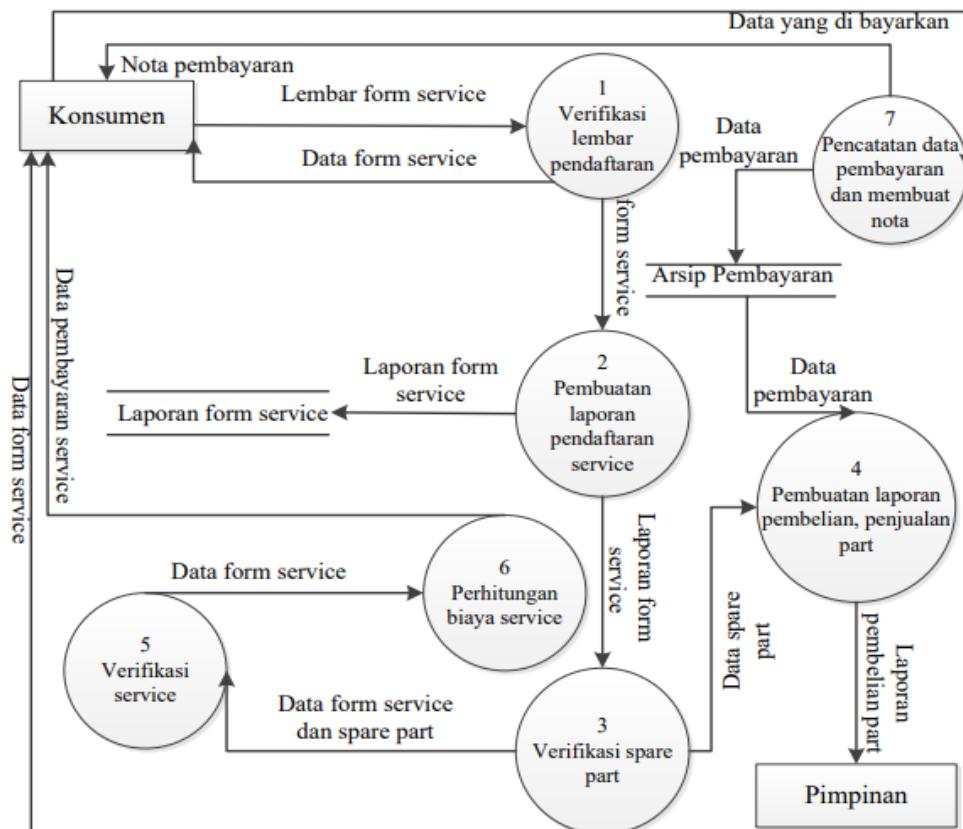
- a) Login : proses masuk ke dalam sistem untuk dapat melakukan semua aktivitas
- b) Home : proses yang dapat dilakukan oleh pelanggan seperti, melihat informasi pemesanan, dan melakukan pemesanan *service*.
- c) Menu utama : proses yang dapat dilakukan oleh pelanggan seperti, melihat informasi dan melakukan pemesanan.

4.3 Desain Logik

Desain Logikal digambarkan dengan menggunakan Data Flow Diagram (DFD). DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Rancangan sistem yang berjalan berupa perancangan Entity Relationship Diagram (ERD) menggunakan draw.io, sistem yang dibangun merupakan Sistem aplikasi *Booking Service Online* bengkel Oka motor.

4.3.1 Data Flow Diagram (DFD)

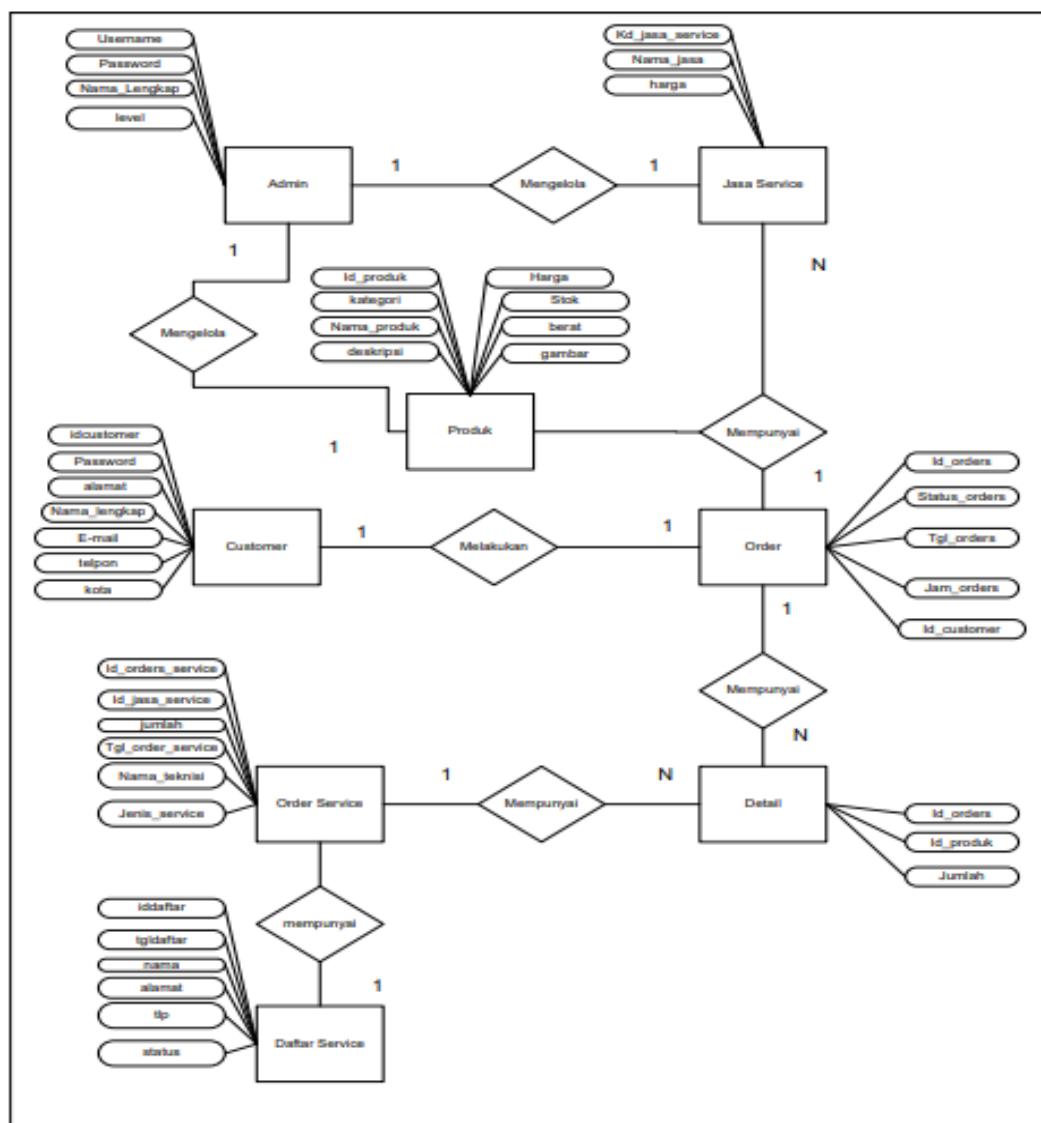
Pada Data Flow Diagram (DFD) akan diuraikan tentang gambaran dan penjelasan tentang sistem yang berjalan saat ini. Data Flow Diagramnya adalah sebagai berikut.



Gambar 4.6 DFD Pelayanan Service yang sedang berjalan.

4.3.2 Entity Relationship Diagram (ERD)

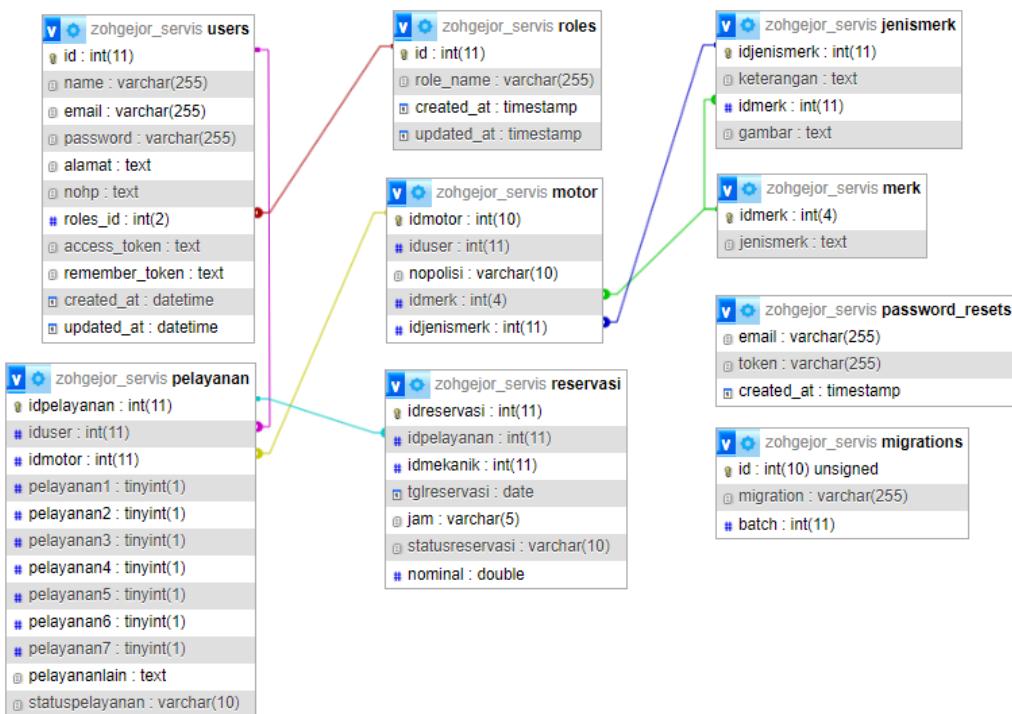
Entity Relationship Diagram adalah diagram yang dapat mengekspresikan keseluruhan data logis dari struktur penggambaran basis data dan menjelaskan tentang hubungan antar entitas yang digunakan dalam pembuatan sistem aplikasi *Booking Service Online* bengkel Oka motor menggunakan metode scrum berbasis mobile. Penggambaran ERD bertujuan untuk menggambarkan himpunan-himpunan entitas dan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh dengan lebih sistematis. ERD dapat dilihat pada gambar 4. 6 berikut.



Gambar 4.7 Entity Relationship Diagram (ERD)

4.3.3 Relasi Tabel

Relasi tabel menjelaskan relasi Primary Key dan Foreign Key dari tabel-tabel yang digunakan dalam pembuatan sistem. Dari rancangan tabel yang dibuat, seperti relasi antara tabel users dan tabel pelayanan yang saling berhubungan untuk mengelola jenis data pelayanan servis yang dilakukan oleh user, kemudian relasi antara tabel users dan tabel roles yang saling berhubungan untuk mengelola jenis data users dan admin, kemudian relasi antara tabel motor dan tabel pelayanan yang saling berhubungan untuk mengelola jenis data reservasi yang dilakukan oleh user, kemudian relasi antara tabel motor dan tabel merk yang saling berhubungan untuk mengelola jenis data merk motor, relasi antar tabel yang terdapat dalam database users dapat dilihat pada Gambar 4.7 berikut.



Gambar 4.8 Relasi Antar Tabel

4.4 Perancangan Fisik

Pada tahap ini dilakukan pendefinisian basis data yang akan disimpan meliputi struktur tabel, tipe data dan keterangan tiap-tiap *field*. Adapun struktur tabel dalam rancang bangun aplikasi *booking online* berbasis *android* menggunakan metode scrum ada 5 tabel sebagai berikut:

a. Perancangan Fisik Tabel

Dalam perancangan struktur tabel juga dilakukan transformasi struktur data yang akan disimpan dengan membuat spesifikasi struktur tiap berkas data.

1. Tabel User:

Nama tabel : users

Primary key : id

Foreign Key : Roles_id

Tabel 4.5 Rancangan Tabel Users

No	Field	Type	Keterangan
1	Id*	Integer (11)	Primary Key
2	Name	Varchar (25)	
3	Email	Varchar (25)	
4	Password	Varchar (25)	
5	Alamat	Varchar (25)	
6	Nohp	Integer (11)	
7	Roles_id**	Integer (11)	Foreign Key
8	access_token	Text	
9	remember_token	Text	
10	Created_at	Datetime	
11	Update_at	Datetime	

Pada tabel user motor diatas memiliki 11 atribut, yaitu; id, name, email, password, alamat, nohp, roles_id, access_token, remember_token, created_at, dan update_at.

2. Tabel jenis merk

Nama tabel : jenismerk

Primary key : Id_jenismerk

Tabel 4.6 Rancangan Jenismerk

No	Field	Type	Keterangan
1	Idjenismerk *	Int (11)	Primary Key
2	keterangan	Text	

Tabel 4.7 Rancangan Jenismerk (lanjutan)

No	Field	Type	Keterangan
3	Idmerk	Int (11)	
4	gambar	Text	

Pada tabel jenismerk motor diatas memiliki 4 atribut, yaitu; idjenismerk, keterangan, idmerk, dan gambar.

3. Tabel merk

Nama tabel : merk

Primary key : id_merk

Tabel 4.8 rancangan merk

No	Field	Type	Keterangan
1	Id_merk*	Int (4)	Primary Key
2	Jenismerk	Text	

Pada tabel merk diatas memiliki 2 atribut, yaitu; id_merk, dan jenismerk.

4. Tabel motor

Nama tabel : Motor

Primary key : idmotor

Foreign Key : Iduser, Idmerk

Tabel 4.9 Rancangan motor

No	Field	Type	Keterangan
1	Idmotor *	Int (11)	Primary Key
2	Iduser **	Int (11)	Foreign Key
3	nopolisi	Varchar (10)	
4	Idmerk **	Int (11)	Foreign Key
5	idjenismerk	Int (11)	

Pada tabel motor diatas memiliki 5 atribut, yaitu; idmotor, iduser, nopolisi, idmerk dan idjenismerk.

5 Tabel password resset

Nama tabel : password_resets

Foreign Key : email

Tabel 4.10 Rancangan password reset

No	Field	Type	Keterangan
1	email **	Varchar (25)	Foreign Key
2	token	Varchar (25)	
3	Created_at	timestamp	

Pada tabel password_reset diatas memiliki 3 atribut, yaitu; email, token dan created_at.

6 Tabel Pelayanan

Nama tabel : pelayanan

Primary key : idpelayanan

Foreign Key : Iduser, Idmotor

Tabel 4.11 Rancangan pelayanan

No	Field	Type	Keterangan
1	Idpelayan *	Int (10)	Primary Key
2	Iduser **	Int (10)	Foreign Key
3	Idmotor **	Int (10)	Foreign Key
4	Servis Karburator	Tinyint (10)	
5	Servis Ringan	Tinyint (10)	
6	Servis injeksi	Tinyint (10)	
7	Bore Up	Tinyint (10)	
8	Tune Up	Tinyint (10)	
9	Bubut	Tinyint (10)	
10	Ganti oli mesin	Tinyint (10)	
11	Pelayananlain	Text	
12	Statuspelayanan	Varchar (10)	

Pada tabel pelayanan diatas memiliki 12 atribut, yaitu; id_pelayanan, iduser, idmotor, Pelayanan 1, Pelayanan 2, Pelayanan 3, Pelayanan 4, Pelayanan 5, Pelayanan 6, Pelayanan 7, Pelayananlain, dan stastuspelayanan.

7 Tabel Reservasi

Nama tabel : Reservasi

Primary key : idreservasi

Foreign Key : Idpelayanan, Idmekanik

Tabel 4.12 Rancangan Reservasi

No	Field	Type	Keterangan
1	Idreservasi *	Int (11)	Primary Key
2	Idpelayanan **	Int (11)	Foreign Key
3	Idmekanik **	Int (11)	Foreign Key
4	Tglreservasi	Date	
5	Jam	Varchar (10)	
6	Statusreservasi	Varchar (10)	

Pada tabel reservasi diatas memiliki 6 atribut, yaitu; idreservasi, idpelayanan, idmekanik, tglreservasi, jam, dan statusreservasi.

8 Tabel Roles

Nama tabel : Reservasi

Primary key : id

Tabel 4.13 Rancangan Roles

No	Field	Type	Keterangan
1	Id *	Int (11)	Primary Key
2	Role_name	Varchar (25)	
3	Created_at	Timestamp	
4	Update_at	Timestamp	

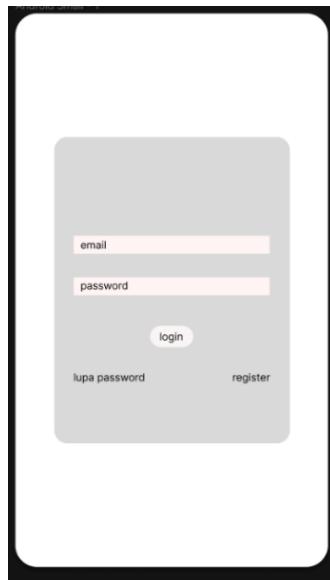
Pada tabel Roles diatas memiliki 4 atribut, yaitu; id, role_name, created_at, dan update_at .

4.5 Desain Antarmuka

Desain antarmuka berisi tentang rancangan atau desain antarmuka. Desain antarmuka menjelaskan mengenai perancangan antarmuka sebelum diimplementasikan menjadi sistem. Desain antarmuka ini yang selanjutnya akan diimplementasikan menjadi antarmuka mobile.

1.5.1 Desain Halaman Login

Desain halaman login adalah langkah buat admin dan kostumer untuk dapat login ke dalam sistem. Desain halaman login dapat dilihat pada Gambar berikut.



Gambar 4.9 halaman login

1.5.2 Desain Halaman Registrasi

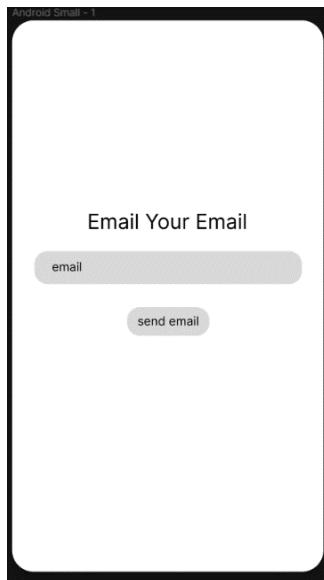
Desain halaman registrasi adalah langkah kostumer untuk mendaftarkan akun pada aplikasi *Booking Service Online* bengkel Oka motor . Desain halaman registrasi dapat dilihat pada Gambar berikut.



Gambar 4.10 Halaman registrasi email

1.5.3 Desain Halaman Reset Password

Desain halaman reset password adalah langkah buat kostumer untuk dapat melakukan reset password apabila terjadi lupa password. Desain halaman reset password dapat dilihat pada Gambar berikut.



Gambar 4.11 Halaman reset password

1.5.4 Desain Halaman Utama

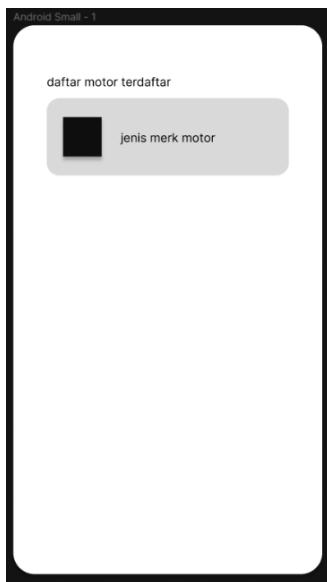
Desain halaman registrasi adalah tampilan aplikasi pada aplikasi *Booking Service Online* bengkel Oka motor . Desain halaman utama dapat dilihat pada Gambar berikut.



Gambar 4.12 Halaman utama

1.5.5 Desain Halaman Motor

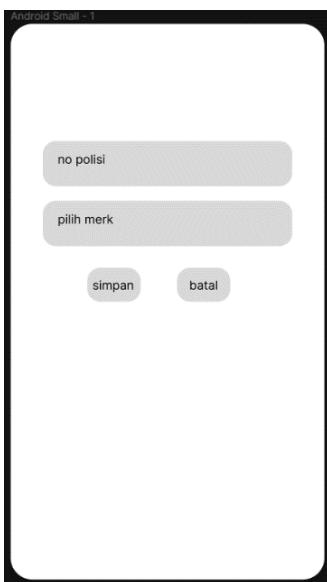
Desain halaman motor adalah langkah buat mengetahui jenis-jenis motor yang ada didaftar reservasi. Desain halaman motor dapat dilihat pada Gambar berikut.



Gambar 4.13 Halaman motor

1.5.6 Desain Halaman Registrasi Motor

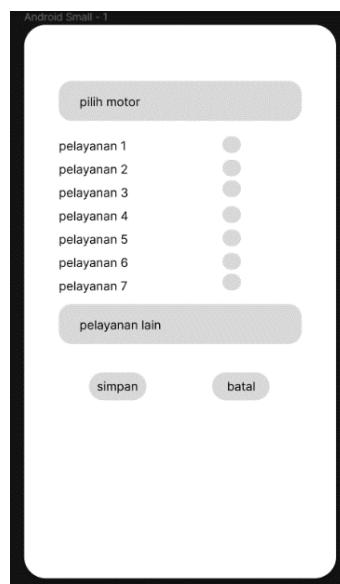
Desain halaman registrasi motor adalah tampilan untuk melakukan reservasi pada aplikasi *Booking Service Online* bengkel Oka motor. Desain halaman registrasi motor dapat dilihat pada Gambar berikut.



Gambar 4.14 Halaman registrasi motor

1.5.7 Desain Halaman Reservasi

Desain halaman reservasi adalah tampilan untuk melakukan reservasi tahap selanjutnya pada aplikasi *Booking Service Online* bengkel Oka motor. Desain halaman registrasi motor dapat dilihat pada Gambar berikut.



Gambar 4.15 Halaman reservasi

1.5.8 Desain Halaman Reservasi Layanan

Desain halaman reservasi layanan adalah tampilan untuk menampilkan hasil reservasi kostumer pada aplikasi *Booking Service Online* bengkel Oka motor. Desain halaman registrasi motor dapat dilihat pada Gambar berikut.



Gambar 4.16 Halaman reservasi layanan

BAB V

IMPLEMENTASI HASIL

5.1 Implementasi

Pada bagian ini menguraikan tentang implementasi dari perancangan menggunakan komponen/tool/bahasa pemrograman yang dipakai. Hal-hal yang perlu di tunjukan pada bagian ini adalah implementasi potongan program pokok yang dibangun sesuai dengan algoritma atau flowchart pada rancangan dan menampilkan script inti dari sistem. Gambar 5.1 menampilkan struktur database yang digunakan dalam relasi ini.

1. Tabel Users

Pada Gambar 5.1 merupakan implementasi tabel users. Tabel users digunakan untuk menyimpan data costumer, dan data pengelola (admin) dengan role dari setiap datanya yang terdaftar dalam sistem. Data tersebut digunakan untuk masuk ke dalam sistem sesuai dengan ketentuan masing-masing rolenya pengelola (admin) sebagai pengelola sistem, costumer sebagai pengguna harus melakukan pendaftaran terlebih dahulu kemudian untuk melakukan reservasi pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

The screenshot shows the 'Structure' tab of MySQL Workbench for the 'users' table. The table has 11 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8_general_ci		No	None			Change Drop More
3	email	varchar(255)	utf8_general_ci		No	None			Change Drop More
4	password	varchar(255)	utf8_general_ci		No	None			Change Drop More
5	alamat	text	utf8_general_ci		Yes	NULL			Change Drop More
6	nohp	text	utf8_general_ci		Yes	NULL			Change Drop More
7	roles_id	int(2)			No	None			Change Drop More
8	access_token	text	utf8_general_ci		Yes	NULL			Change Drop More
9	remember_token	text	utf8_general_ci		Yes	NULL			Change Drop More
10	created_at	datetime			No	current_timestamp()			Change Drop More
11	updated_at	datetime			No	current_timestamp()			Change Drop More

At the bottom, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'.

Gambar 5.1 Tabel Users

2. Tabel Jenismerk

Pada Gambar 5.2 merupakan implementasi tabel jenismerk. Tabel jenismerk digunakan untuk menyimpan data seperti jenismerk, keterangan, idmerk dan gambar motor. Data tersebut digunakan untuk menampilkan sebelum melakukan reservasi pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idjenismerk	int(11)		No	None		AUTO_INCREMENT		Change Drop More
2	keterangan	text	utf8mb4_general_ci	No	None				Change Drop More
3	idmerk	int(11)		No	None				Change Drop More
4	gambar	text	utf8mb4_general_ci	No	None				Change Drop More

Gambar 5.2 Tabel Jenismerk

3. Tabel merk

Pada Gambar 5.3 merupakan implementasi tabel jenismerk. Tabel jenismerk digunakan untuk menyimpan data seperti idmerk, dan jenismerk. Data tersebut digunakan untuk menampilkan sebelum melakukan reservasi pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idmerk	int(4)		No	None		AUTO_INCREMENT		Change Drop More
2	jenismerk	text	utf8mb4_general_ci	No	None				Change Drop More

Gambar 5.3 Tabel Merk

4. Tabel motor

Pada Gambar 5.4 merupakan implementasi tabel motor. Tabel motor digunakan untuk menyimpan data seperti idmotor, iduser, nopolisi, idmerk, dan idjenismerk. Data tersebut digunakan untuk menampilkan seberapa banyak jumlah costumer dalam melakukan reservasi pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idmotor 🔑	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	iduser 🔑	int(11)			No	None			Change Drop More
3	nopolisi	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
4	idmerk 🔑	int(4)			No	None			Change Drop More
5	idjenismerk	int(11)			No	None			Change Drop More

Gambar 5.4 Tabel Motor

5. Tabel reset password

Pada Gambar 5.5 merupakan implementasi tabel reset password. Tabel reset password digunakan untuk menyimpan data hasil dari reset password yang dilakukan oleh costumer.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	email 🔑	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
2	token	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	created_at	timestamp			Yes	NULL			Change Drop More

Gambar 5.5 Tabel Reset Password

6. Tabel pelayanan

Pada Gambar 5.6 merupakan implementasi tabel pelayanan. Tabel pelayanan digunakan untuk menyimpan data seperti idpelayanan, iduser, idmotor, pelayanan1, pelayanan2, pelayanan3, pelayanan4, pelayanan5, pelayanan6, pelayanan7, pelayananlain, dan statuspelayanan. Data tersebut digunakan untuk melakukan reservasi jenis pelayanan pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idpelayanan 🔑	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	iduser 🔑	int(11)			No	None			Change Drop More
3	idmotor 🔑	int(11)			No	None			Change Drop More
4	pelayanan1	tinyint(1)			Yes	0			Change Drop More
5	pelayanan2	tinyint(1)			Yes	0			Change Drop More
6	pelayanan3	tinyint(1)			No	0			Change Drop More
7	pelayanan4	tinyint(1)			No	0			Change Drop More
8	pelayanan5	tinyint(1)			Yes	0			Change Drop More
9	pelayanan6	tinyint(1)			No	0			Change Drop More
10	pelayanan7	tinyint(1)			No	0			Change Drop More
11	pelayananlain	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
12	statuspelayanan	varchar(10)	utf8mb4_general_ci		Yes	baru			Change Drop More

Gambar 5.6 Tabel Pelayanan

7. Tabel reservasi

Pada Gambar 5.7 merupakan implementasi tabel reservasi. Tabel reservasi digunakan untuk menyimpan data seperti idreservasi, idpelayanan, idmekanik, tglreservasi, jam, dan stastusreservasi. Data tersebut digunakan untuk mengetahui status reservasi yang dilakukan oleh costumer dalam melakukan reservasi pada aplikasi *Booking Online* pelayanan servis bengkel Oka motor.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idreservasi	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	idpelayanan	int(11)			No	None			Change Drop More
3	idmekanik	int(11)			No	None			Change Drop More
4	tglreservasi	date			No	None			Change Drop More
5	jam	varchar(5)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	statusreservasi	varchar(10)	utf8mb4_general_ci		No	baru			Change Drop More

Gambar 5.7 Tabel Reservasi

8. Tabel roles

Pada Gambar 5.8 merupakan implementasi tabel roles. Tabel roles digunakan untuk menyimpan data dan membedakan dari beberapa akun seperti admin, mekanik, dan costumer.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	role_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	created_at	timestamp			Yes	NULL			Change Drop More
4	updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 5.8 Tabel Roles

5.2 Source Code

Pada Bagian ini menampilkan implementasi potongan kode program yang digunakan untuk membuat aplikasi *Booking Online* pelayanan servis bengkel Oka motor, implementasi dalam tulisan ini menggunakan software visual studio code dan android studio seperti pada gambar berikut:

1) Kode program halaman login

Berikut ini merupakan kode program dari halaman login yang digunakan untuk membuat tampilan halaman login sebagai Gambar berikut.

```
child: Text(_isLogin ? 'Login' : 'Signup'),
),
Row(
    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
    children: [
        TextButton(
            onPressed: () {
                Navigator.push(context,
MaterialPageRoute(
                    builder: (context) {
                        return const ForgotPassword();
                },
            ),
        ),
    ],
}
```

Gambar 5.9 kode program untuk halaman login

2) Kode program home page

Berikut ini merupakan kode program dari home page yang digunakan untuk membuat tampilan home page sebagai Gambar berikut:

```
Container(
margin: const EdgeInsets.only(top: 5), height: 120,
width: double.infinity, child: Card(
margin: const EdgeInsets.only(top: 5, bottom: 5),
color: const Color.fromARGB(255, 217, 219, 224), elevation: 5,
shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10),
), child: Column(mainAxisAlignment: MainAxisAlignment.center,
crossAxisAlignment: CrossAxisAlignment.center, children: [
const Text('Member Dashboard', style: TextStyle( fontSize: 20,
fontWeight: FontWeight.bold,
color: Colors.black, )), const SizedBox(height: 5),
const Text('Bengkel Oka Motor', style: TextStyle(
```

Gambar 5.10 kode program untuk home page

3) Kode program halaman motor

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman motor sebagai Gambar berikut:

```
factory Motor.fromMap(Map<String, dynamic> map) {
    return Motor(
        idmotor: map['idmotor'] != null ? map['idmotor'] as int : null,
        iduser: map['iduser'] as int,
        nopolisi: map['nopolisi'] as String,
        idmerk: map['idmerk'] as int,
        idjenismerk: map['idjenismerk'] as int,);}String toJson() => json.encode(toMap());
factory Motor.fromJson(String source) =>
Motor.fromMap(json.decode(source) as Map<String, dynamic>);
} required this.idjenismerk,});
```

Gambar 5.11 kode program untuk motor

4) Kode program registrasi motor

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman registrasi motor sebagai Gambar berikut:

```
await MotorDio().postmotor(motor).then((value) {
    value["status"] == false
    ? ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar( content: Text(
            'Register Failed, cek kembali data anda'))))
    : ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar( content: Text('Register Success'))));
Future.delayed( const Duration(seconds: 5)
// ignore: avoid_types_on_closure_parameters
, () { Navigator.pushAndRemoveUntil(
context, MaterialPageRoute( builder: (BuildContext context) =>
const HomePage(). ), (route) => false. ); });
});
```

Gambar 5.12 kode program untuk registrasi motor

5) Kode program ajuan layanan

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman ajuan layanan sebagai Gambar berikut:

```
const Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [ Text(
    'Daftar Ajuan Layanan' ),
  ],
), isLoading
? const Center( child: CircularProgressIndicator(),
): Expanded( child: pelayanan.isEmpty
? const Center( child: Text('Tidak ada item'),
: ListView.builder( itemCount: pelayanan.length,
return ItemPelayananWidget( pelayanan: pelayanan[index],
handleRefresh: refreshData, );
} ),
```

Gambar 5.13 kode program untuk ajuan pelayanan

6) Kode program daftar ajuan layanan

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman daftar ajuan layanan sebagai Gambar berikut:

```
factory PelayananModel.fromMap(Map<String, dynamic> map) {
  return PelayananModel(
    idpelayanan: map['idpelayanan'] != null ?
    map['idpelayanan'] as int : null,
    iduser: map['iduser'] as int,
    idmotor: map['idmotor'] as int,
    pelayanan1: map['pelayanan1'] as bool,
    pelayanan2: map['pelayanan2'] as bool,
    pelayanan3: map['pelayanan3'] as bool,
```

Gambar 5.14 kode program untuk daftar ajuan layanan

7) Kode program reservasi

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman reservasi sebagai Gambar berikut:

```

const Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [Text(
'Daftar Reservasi'),],),
isLoading? const Center(
    child: CircularProgressIndicator(),
): Expanded(child: reservasi.isEmpty
? const Center(child: Text('Tidak ada item'),
): ListView.builder(itemCount: reservasi.length,
itemBuilder: (context, index) {
    return ItemReservasiWidget(reservasi: reservasi[index],
        handleRefresh: refreshData,);
}),

```

Gambar 5.15 kode program untuk reservasi

8) Kode program histori reservasi

Berikut ini merupakan kode program dari pelayanan yang digunakan untuk membuat tampilan halaman histori reservasi sebagai Gambar berikut:

```

const Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
children: [Text(
'Daftar Reservasi'),],),
isLoading
? const Center(child: CircularProgressIndicator(),)
: Expanded(child: reservasi.isEmpty
? const Center(child: Text('Tidak ada item'),
): ListView.builder(itemCount: reservasi.length,
itemBuilder: (context, index) {
    return ItemReservasiHistoryWidget(
        reservasi: reservasi[index],
        handleRefresh: refreshData,
    );
},

```

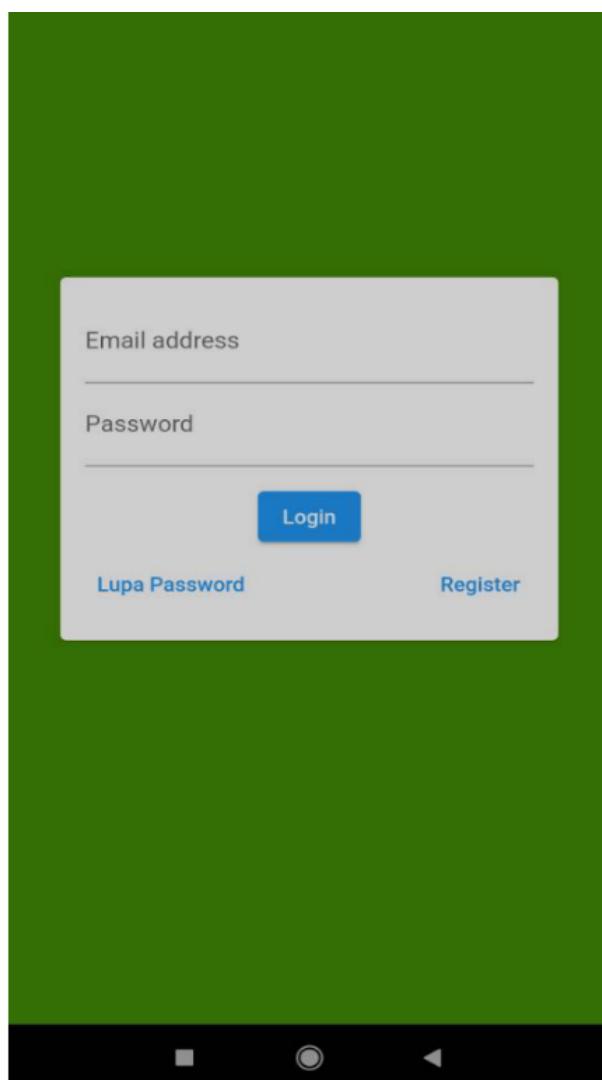
Gambar 5.16 kode program untuk histori reservasi

5.3 Hasil

Pada bagian ini menampilkan hasil dari sistem yang dibuat dengan menampilkan gambar hasil screenshot tampilan aplikasi android yaitu pada gambar berikut ini.

1. Tampilan Login pengelola (admin) / User

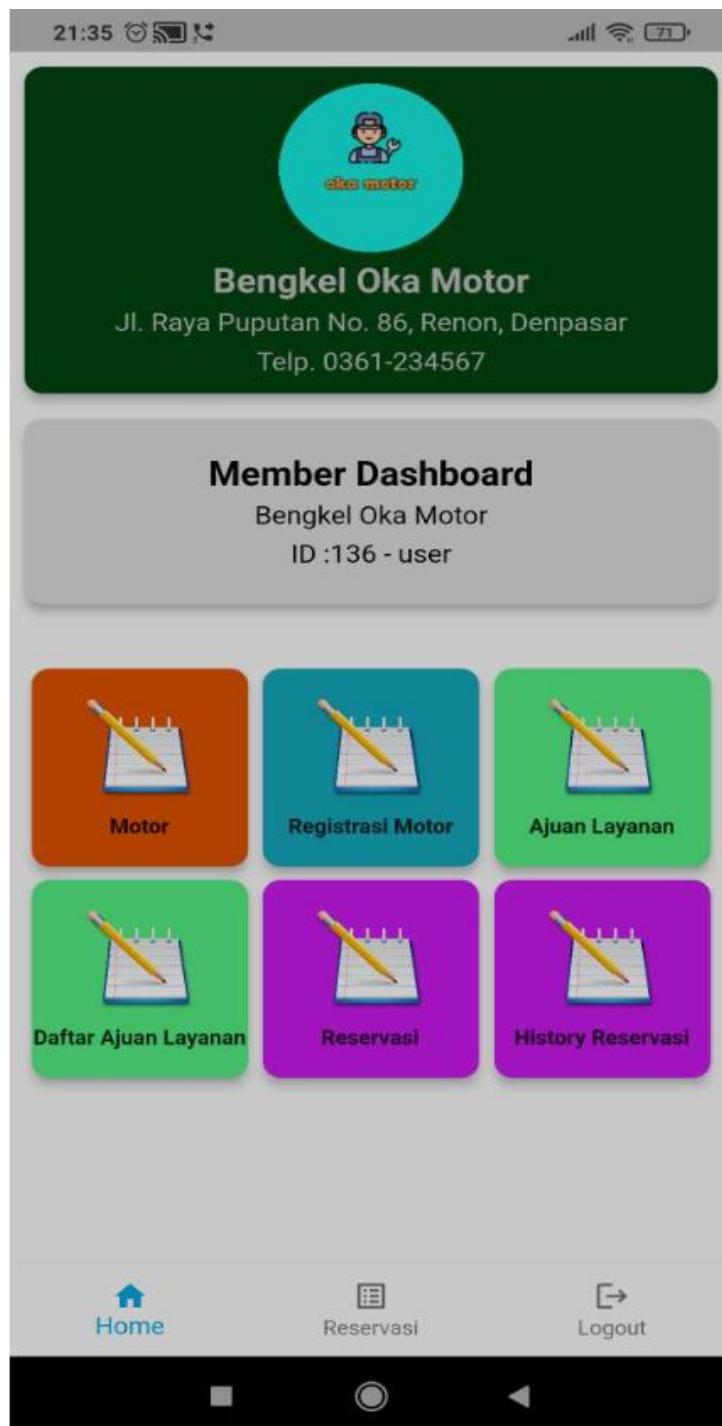
Berikut ini adalah Tampilan Halaman login yang digunakan oleh pengelola (admin/user) dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer & menyetujui reservasi servis sebagai admin. Login dengan menggunakan email dan password yang sudah terdaftar, seperti pada Gambar 5.17 berikut.



Gambar 5.17 Login Pengelola Admin/User

2. Tampilan Menu Utama Costumer

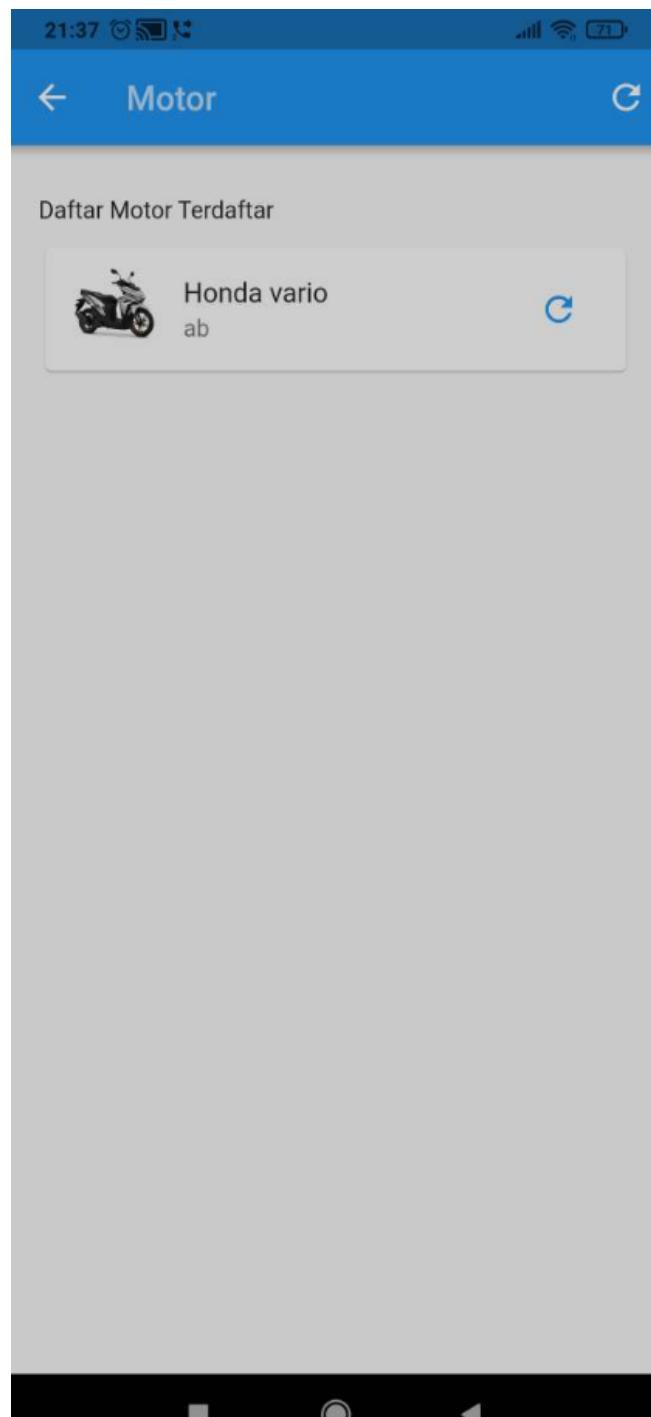
Berikut ini adalah Tampilan Halaman menu utama customer yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.18 berikut.



Gambar 5.18 Login Halaman utama

3. Tampilan Menu Motor

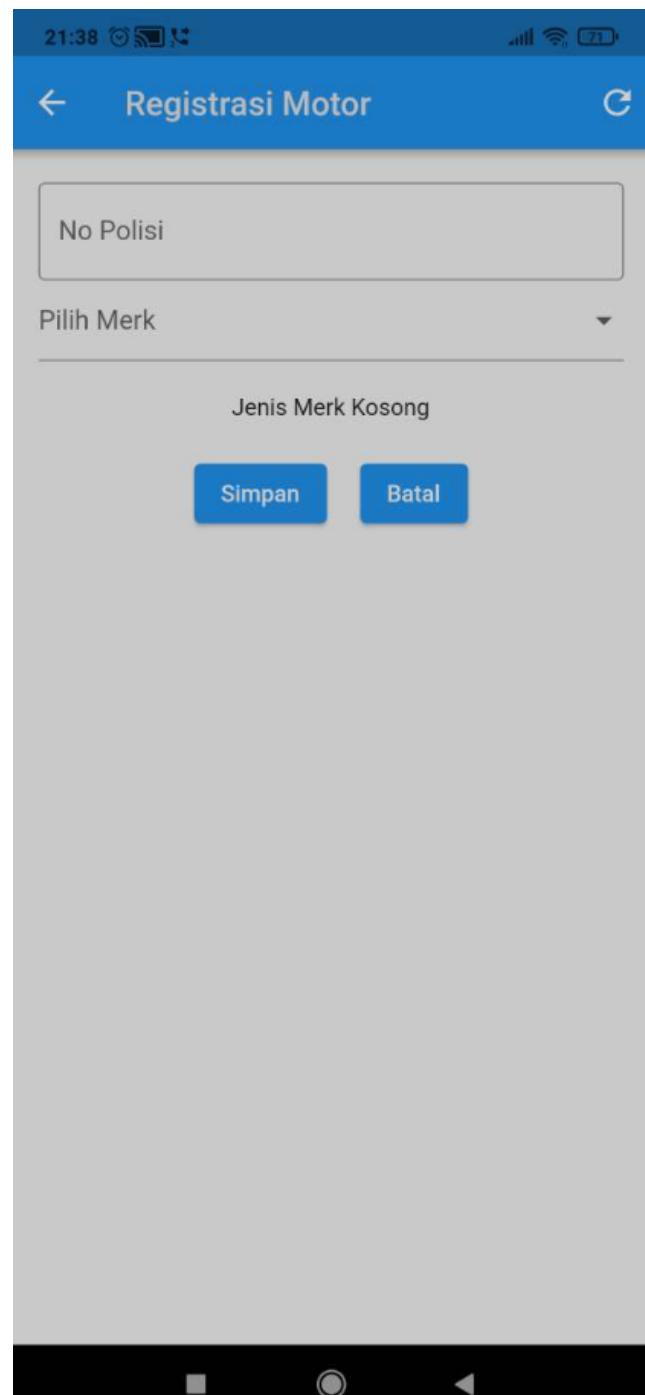
Berikut ini adalah Tampilan Halaman menu motor yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.19 berikut.



Gambar 5.19 Menu motor

4. Tampilan Menu Registrasi Motor

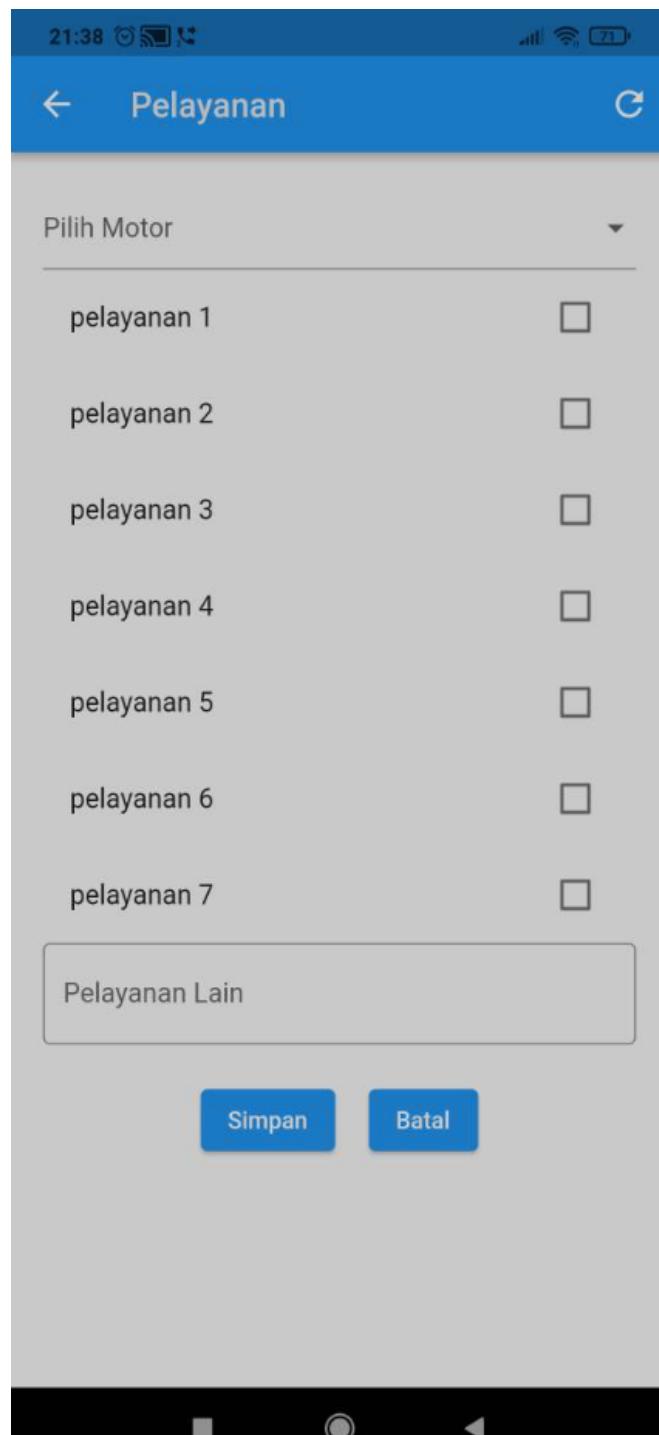
Berikut ini adalah Tampilan Halaman menu registrasi motor yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.20 berikut.



Gambar 5.20 Registrasi motor

5. Tampilan Menu Pelayanan

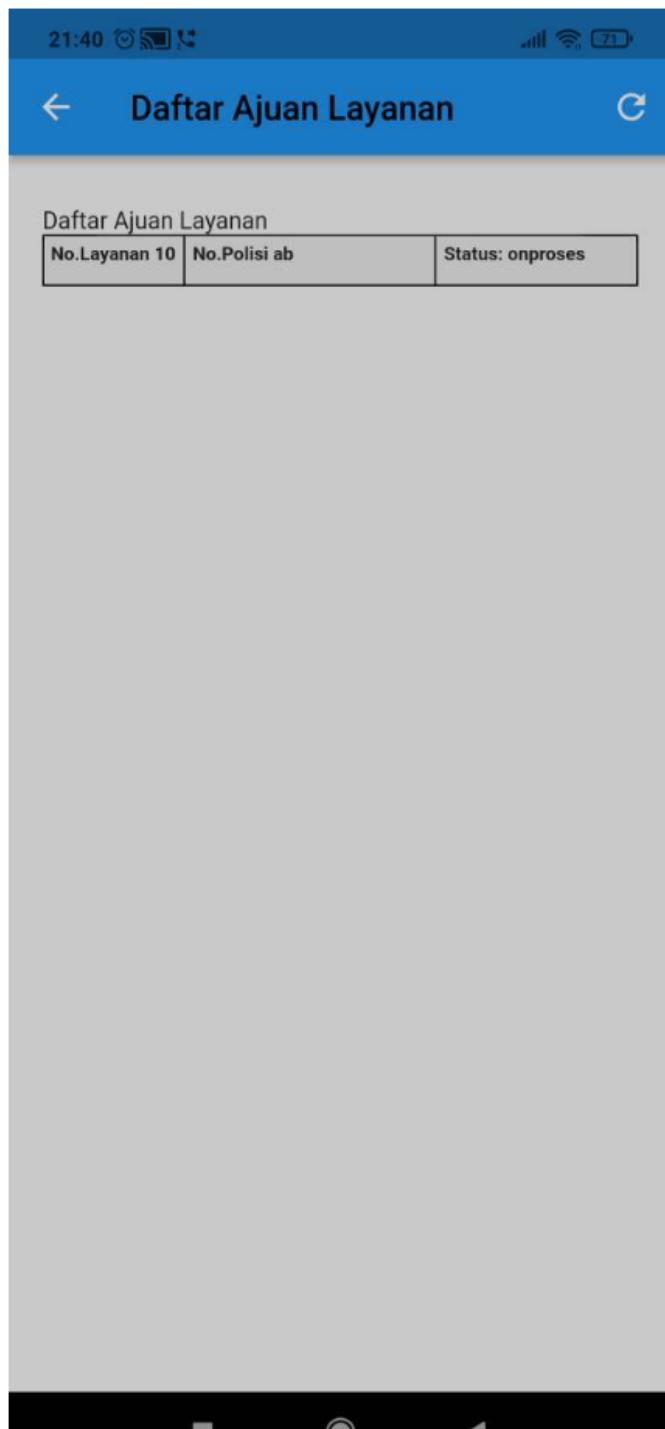
Berikut ini adalah Tampilan Halaman menu ajuan pelayanan yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.21 berikut.



Gambar 5.21 Pelayanan

6. Tampilan Menu Daftar Ajuan Pelayanan

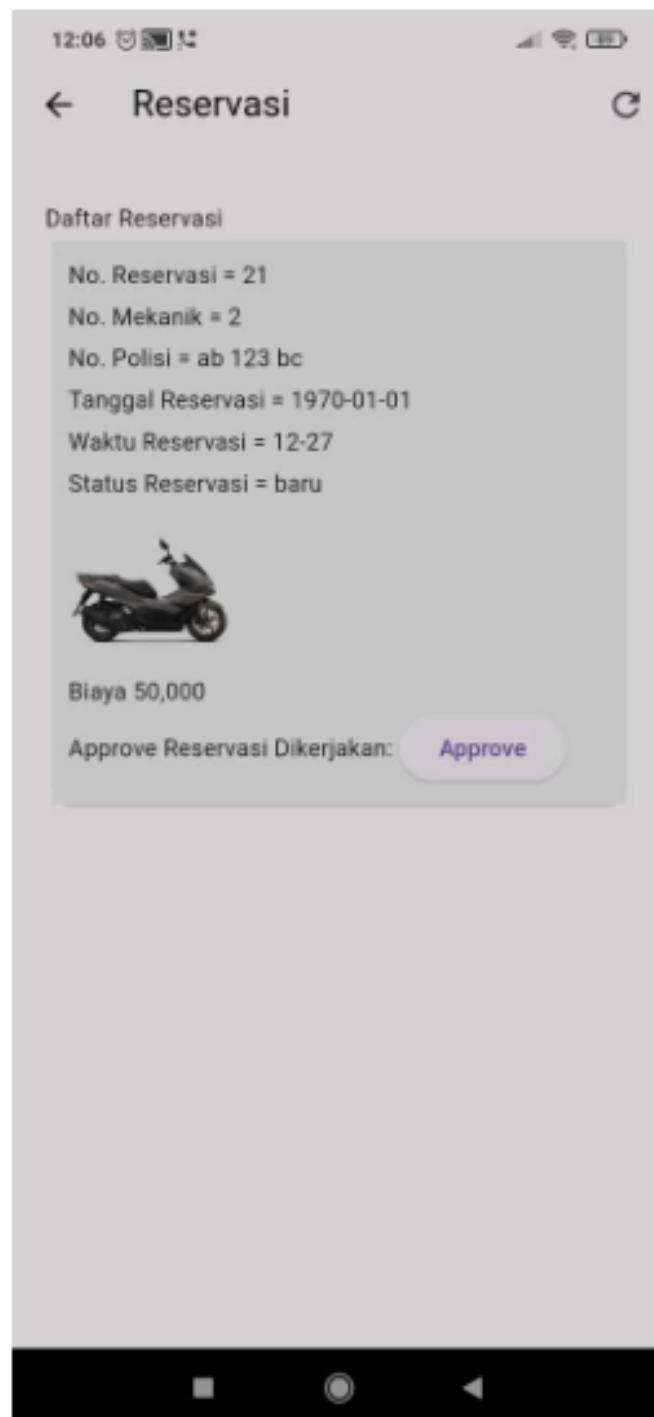
Berikut ini adalah Tampilan Halaman menu ajuan pelayanan yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.22 berikut.



Gambar 5.22 Ajuan pelayanan

7. Tampilan Menu Reservasi

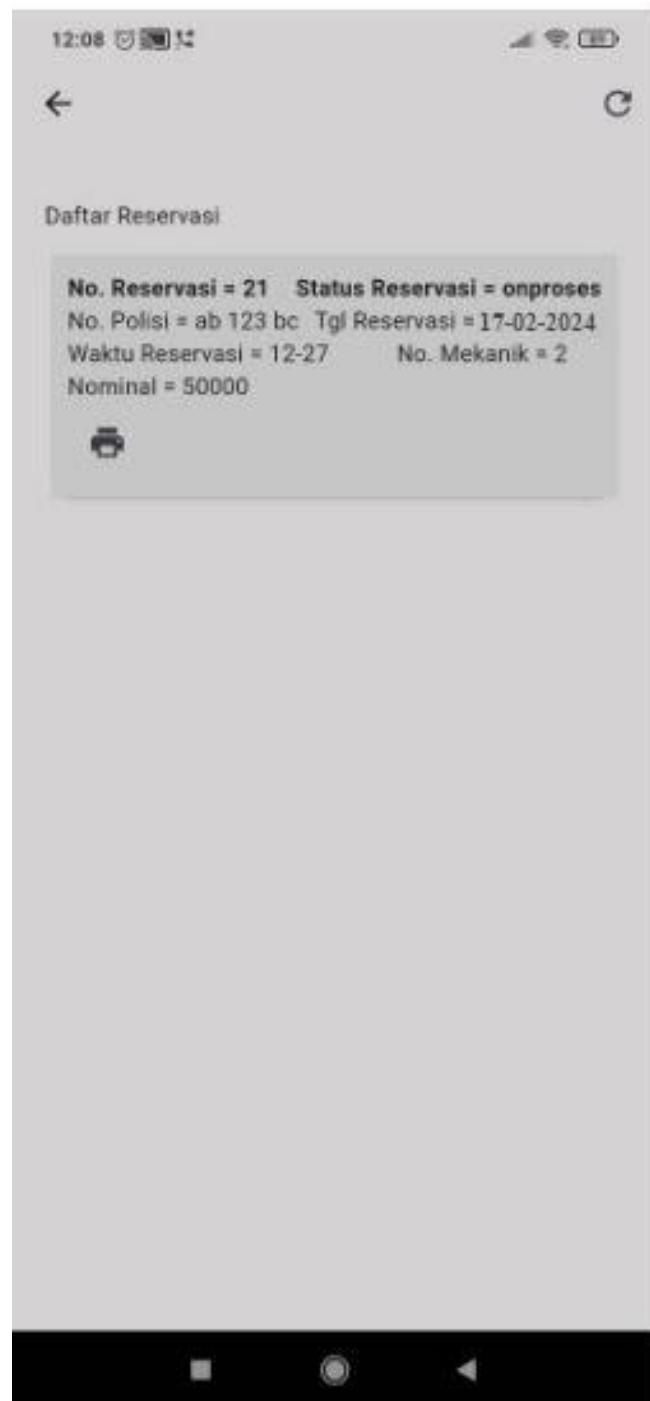
Berikut ini adalah Tampilan Halaman menu reservasi yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.23 berikut.



Gambar 5.23 Reservasi

8. Tampilan Menu History Reservasi

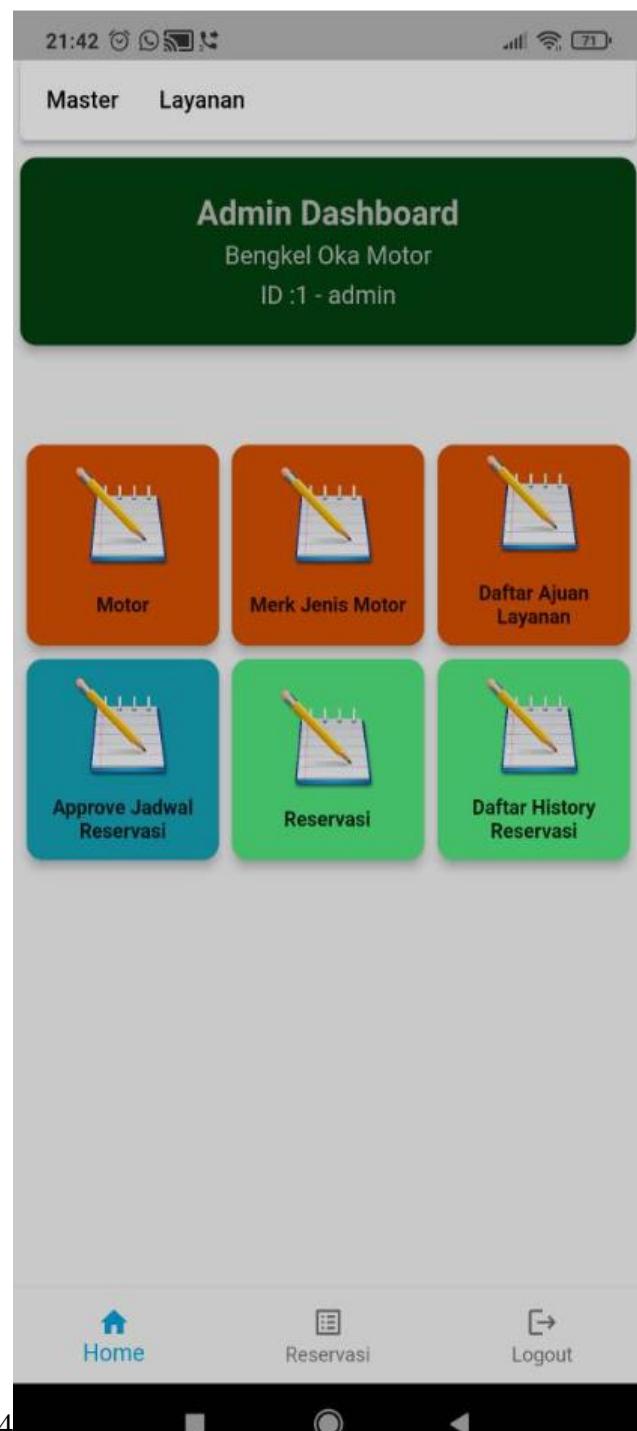
Berikut ini adalah Tampilan Halaman menu history reservasi yang digunakan oleh costumer dari aplikasi *Booking Online* bengkel motor untuk melakukan reservasi sebagai costumer seperti pada Gambar 5.24 berikut.



Gambar 5.24 histori reservasi

9. Tampilan Menu Halaman Utama Admin

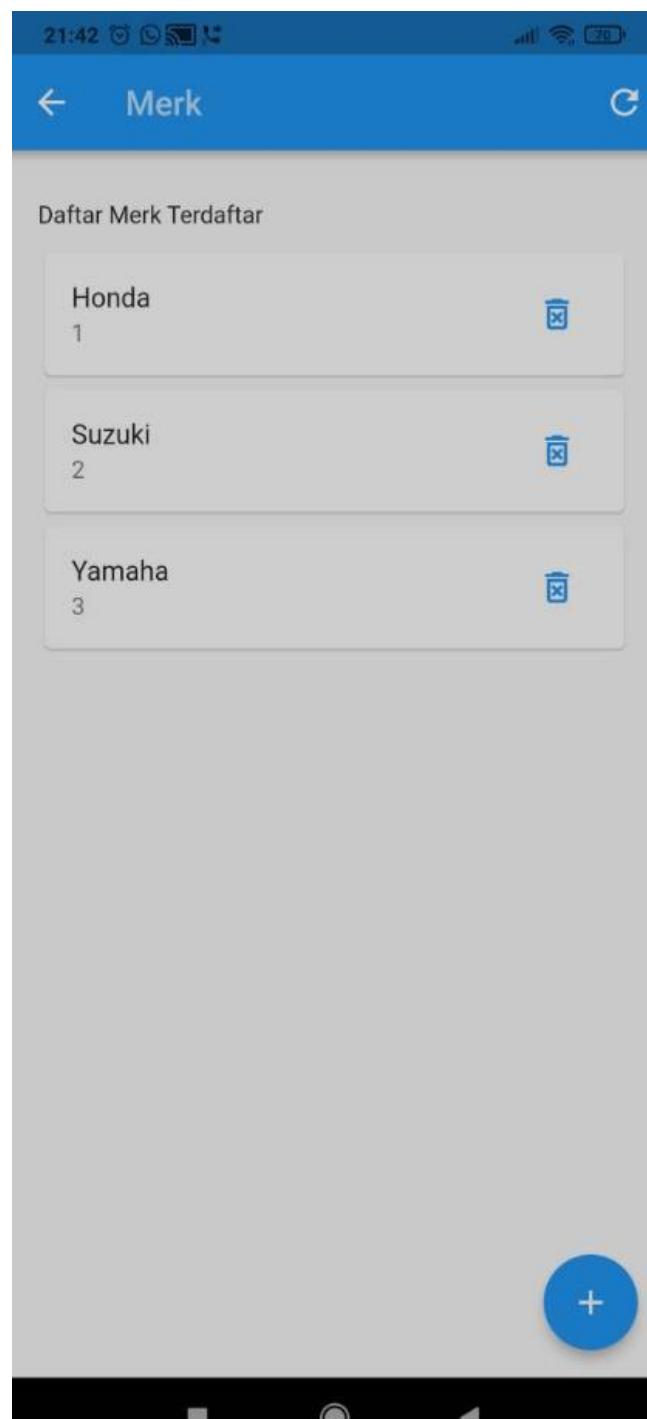
Berikut ini adalah Tampilan Halaman menu utama yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.25 berikut.



Gambar 5.25 Halaman utama Admin

10. Tampilan Menu Merk

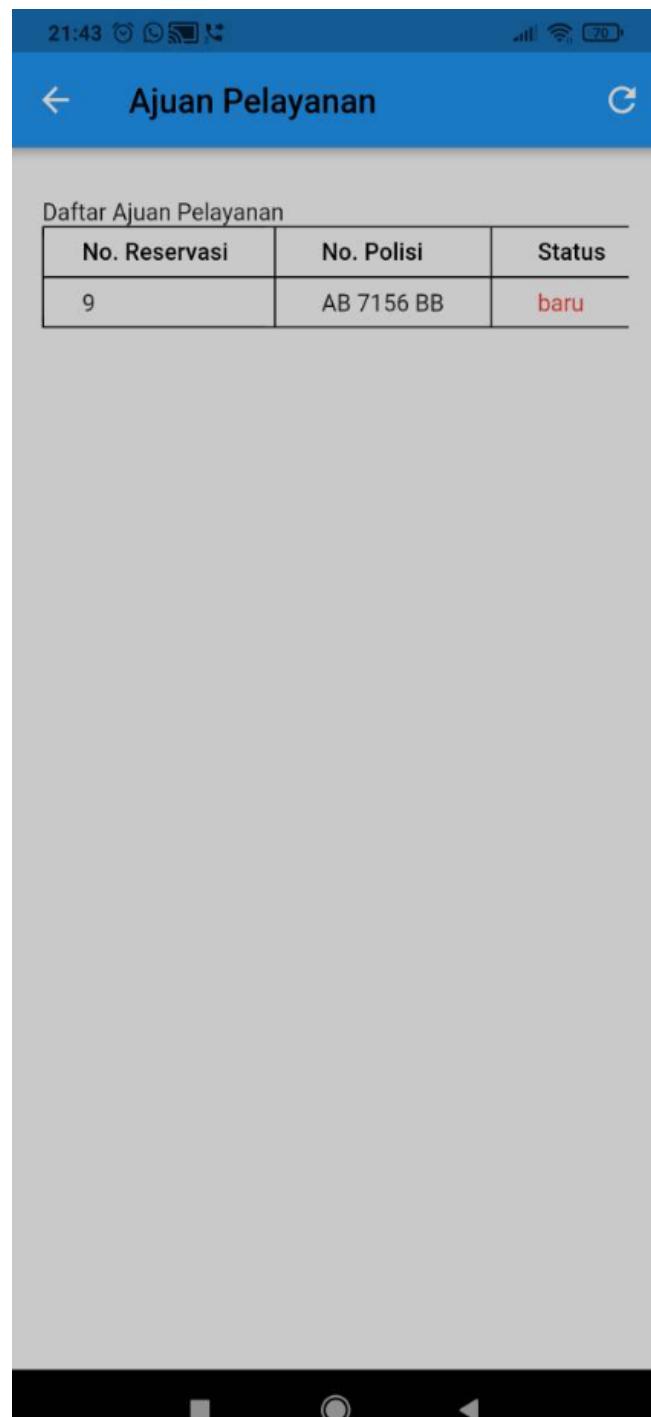
Berikut ini adalah Tampilan Halaman menu merk yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.26 berikut.



Gambar 5.26 Halaman Merk

11. Tampilan Menu Ajuan Pelayanan

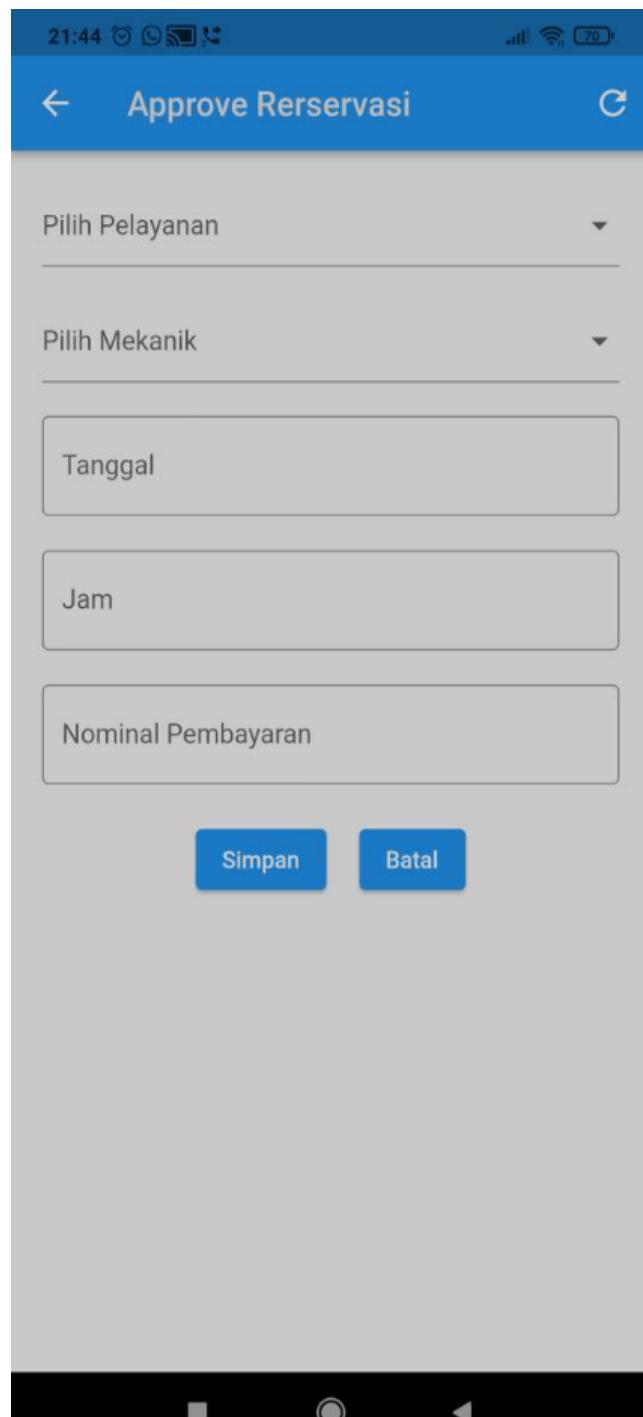
Berikut ini adalah Tampilan Halaman menu ajuan pelayanan yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.27 berikut.



Gambar 5.27 Halaman ajuan pelayanan

12. Tampilan Menu Approve Reservasi

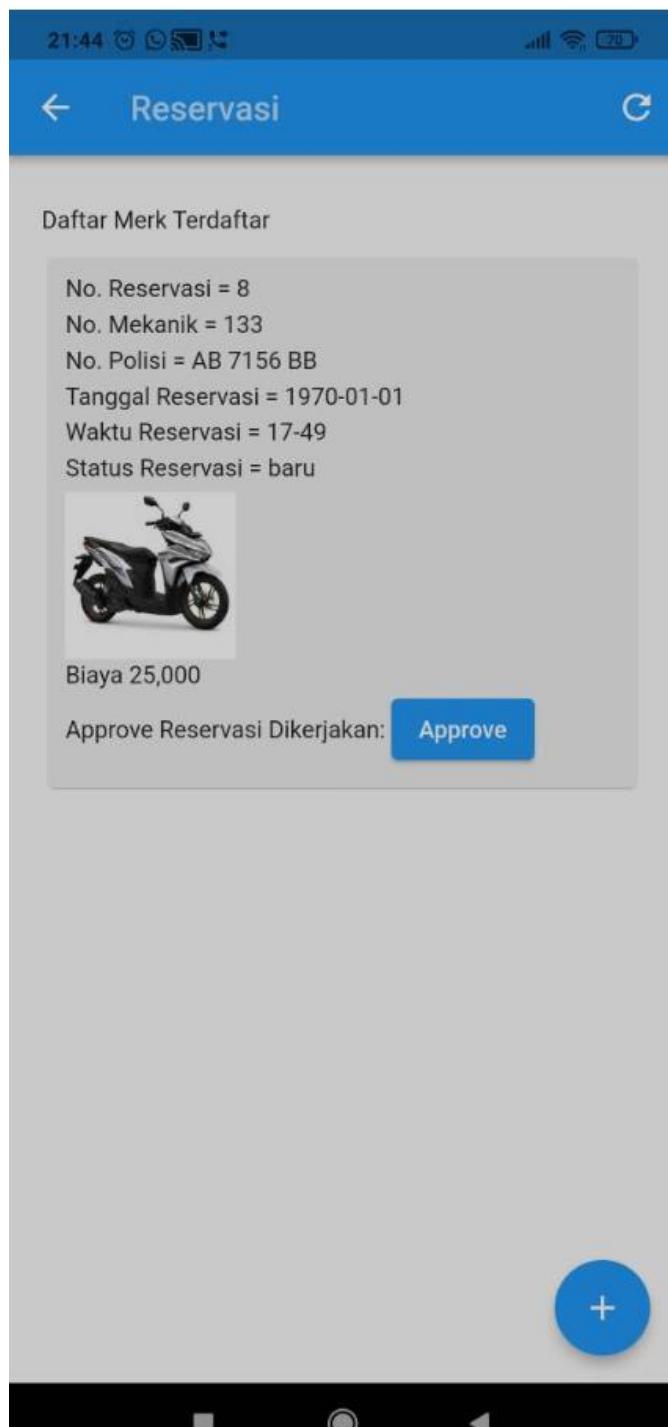
Berikut ini adalah Tampilan Halaman menu *approve reservasi* yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.28 berikut.



Gambar 5.28 Halaman *approve reservasi*

13. Tampilan Menu Reservasi

Berikut ini adalah Tampilan Halaman menu reservasi yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.29 berikut.



Gambar 5.29 Halaman reservasi

14. Tampilan Menu Daftar Reservasi

Berikut ini adalah Tampilan Halaman menu reservasi yang digunakan oleh admin dari aplikasi *Booking Online* bengkel motor untuk melakukan pengelolaan data reservasi sebagai admin seperti pada Gambar 5.30 berikut.



Gambar 5.30 Halaman daftar reservasi

5.4 Pengujian Sistem

Pengujian sistem bertujuan untuk mengetahui kelemahan perangkat lunak dan mengetahui kesesuaian implementasi sistem terhadap perencanaannya guna menjamin kualitas sistem tersebut. Tujuan lain dalam pengujian ini adalah untuk meningkatkan keamanan sistem dan menjamin data penting tetap aman. Pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian Black Box. Pengujian Black Box lebih berfokus pada pengujian fungsional perangkat lunak.

5.4.1 Rencana Pengujian

Rencana pengujian yang dilakukan terhadap fungsi-fungsi yang ada pada sistem. Rencana pengujian ini bertujuan untuk mengetahui setiap *role* atau menu dapat berjalan sesuai yang diharapkan. Berikut rencana pengujian pada sistem ini.

Tabel 5.1 Rencana Pengujian Sistem *Role Customer*.

Kelas Uji	Butir Uji	Jenis Pengujian
<i>Login Customer</i>	Verifikasi <i>email</i>	<i>Black Box</i>
	Verifikasi <i>password</i>	<i>Black Box</i>
	Tombol <i>login</i>	<i>Black Box</i>
Halaman Motor		<i>Black Box</i>
Halaman Registrasi Motor	Menampilkan halaman registrasi motor yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman Ajuan Layanan	Menampilkan halaman jenis reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman Daftar Ajuan Layanan	Menampilkan halaman hasil ajuan layanan yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman Reservasi	Menampilkan halaman reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman History Reservasi	Menampilkan halaman history reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>

Tabel 5.2 Rencana Pengujian Sistem *Role Admin*

Kelas Uji	Butir Uji	Jenis Pengujian
Halaman <i>Login Admin</i>	Verifikasi <i>email</i>	<i>Black Box</i>
	Verifikasi <i>password</i>	<i>Black Box</i>
	Tombol <i>login</i>	<i>Black Box</i>
Halaman Motor	Menampilkan halaman jenis motor	<i>Black Box</i>
Halaman Merk Motor	Menampilkan halaman jenis merk motor	<i>Black Box</i>
Halaman Ajuan Layanan	Menampilkan halaman jenis reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman <i>Approve Jadwal Reservasi</i>	Menampilkan halaman pengajuan layanan yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman Reservasi	Menampilkan halaman reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>
Halaman Daftar History Reservasi	Menampilkan halaman daftar history reservasi yang dilakukan <i>costumer</i>	<i>Black Box</i>

5.4.2 Pengujian *Black Box*

Pengujian *black box* ini dilakukan berdasarkan dari perencanaan pengujian yang sudah dilakukan sebelumnya. Adapun pengujian *black box* dari masing-masing *role* adalah sebagai berikut.

5.4.3 Pengujian *Black Box Role Customer*

Pengujian *Black box pada role customer* ini bertujuan untuk menguji terhadap fungsi-fungsi yang ada pada aplikasi. Berikut merupakan pengujian *black box* yang dilakukan untuk *role customer*.

Pengujian *Black Box Role Customer*

Tabel 5.3 Pengujian *Black Box Role Customer*

Uji Coba Sistem	Kasus Uji Coba	Hasil yang Diharapkan	Keterangan
<i>Login Customer</i>	Tekan tombol <i>login</i> dengan <i>email</i> dan <i>password</i> benar.	Berhasil <i>login</i> sistem.	Terpenuhi

Tabel 5.4 Pengujian *Black Box Role Customer* (lanjutan)

Uji Coba Sistem	Kasus Uji Coba	Hasil yang Diharapkan	Keterangan
	Tekan tombol <i>login</i> dengan <i>email</i> dan <i>password</i> salah.	Tidak berhasil <i>login</i> dan muncul peringatan <i>email</i> atau <i>password</i> salah.	Terpenuhi
Halaman Motor	Tekan tombol button motor	Berhasil menampilkan halaman jenis motor	Terpenuhi
Halaman Registrasi Motor	Tekan tombol button registrasi motor	Berhasil menampilkan halaman registrasi motor, (<i>Customer</i> diminta mengisi registrasi motor untuk melakukan reservasi)	Terpenuhi
Halaman Ajuan Layanan	Tekan tombol button ajuan layanan	Berhasil menampilkan halaman ajuan layanan (<i>Customer</i> diminta mengisi ajuan layanan untuk melakukan reservasi)	Terpenuhi
Halaman Daftar Ajuan Layanan	Tekan tombol button daftar ajuan layanan	Berhasil menampilkan halaman ajuan daftar layanan (apakah status layanan baru atau selesai)	Terpenuhi
Halaman Reservasi	Tekan tombol button reservasi	Berhasil menampilkan halaman reservasi (apakah reservasi telah selesai atau tidak)	Terpenuhi
Halaman History Reservasi	Tekan tombol button reservasi	Berhasil menampilkan halaman history reservasi yang dilakukan <i>costumer</i>	Terpenuhi

5.4.4 Pengujian Black Box Role Admin

Pengujian *Black box* pada role admin ini bertujuan untuk menguji terhadap fungsi-fungsi yang ada pada aplikasi. Berikut merupakan pengujian *black box* yang dilakukan untuk role admin

Tabel 5.5 Pengujian Black Box Role Admin

Uji Coba Sistem	Kasus Uji Coba	Hasil yang Diharapkan	Keterangan
<i>Login Admin</i>	Tekan tombol <i>login</i> dengan <i>username</i> dan <i>password</i> benar.	Berhasil <i>login</i> sistem.	Terpenuhi
	Tekan tombol <i>login</i> dengan <i>username</i> dan <i>password</i> salah.	Tidak berhasil <i>login</i> dan muncul peringatan <i>username</i> atau <i>password</i> salah.	Terpenuhi
Halaman Motor	Tekan tombol button motor	Berhasil menampilkan halaman jenis motor	Terpenuhi
Halaman Jenis Merk Motor	Tekan tombol button merk motor	Berhasil menampilkan halaman Jenis Merk Motor (admin bisa menambahkan jenis merk motor)	Terpenuhi
Halaman Daftar Ajuan Layanan	Tekan tombol button daftar ajuan layanan	Berhasil menampilkan halaman daftar ajuan layanan (yang dilakukan <i>costumer</i>)	Terpenuhi
Halaman Approve Jadwal Reservasi	Tekan tombol button <i>approve</i> jadwal reservasi	Berhasil menampilkan halaman <i>approve</i> jadwal reservasi (setuju atau tidak reservasi yang dilakukan oleh admin)	Terpenuhi
Halaman Reservasi	Tekan tombol button reservasi	Berhasil menampilkan halaman reservasi (apakah reserverasi telah selesai atau tidak)	Terpenuhi
Halaman Daftar History Reservasi	Tekan tombol button history reservasi	Berhasil menampilkan halaman history reservasi	Terpenuhi

5.4.5 Kesimpulan Pengujian Black Box

Setelah melakukan pengujian menggunakan metode *black box* pada kedua hak akses yaitu *customer* dan admin maka diperoleh kesimpulan bahwa pada setiap kasus uji coba yang sudah dipraktikkan hasilnya sesuai dengan yang ada pada perencanaan.

BAB VI

PENUTUP

6.1. Kesimpulan

Hasil analisis dari sistem aplikasi *booking online* bengkel motor berbasis android dengan studi kasus di bengkel Oka Motor ini yang telah dilaksanakan penulis dapat diambil kesimpulan yaitu:

1. Berdasarkan latar belakang yang sudah dikemukakan maka rumusan masalah pada penelitian ini adalah bagaimana implementasi aplikasi *booking online service* motor pada bengkel Oka berbasis *Android*.
2. Tujuan dalam penelitian ini adalah untuk meminimalkan waktu antrian pelanggan dalam melakukan pelayanan pada Bengkel Oka motor.

6.2. Saran

Sistem aplikasi *booking online* bengkel motor berbasis android dengan studi kasus di bengkel Oka Motor ini diharapkan akan terus dikembangkan supaya menjadi sistem yang bermanfaat bagi banyak masyarakat Sleman Yogyakarta. Saran untuk pengembang lebih lanjut agar dapat membantu sistem ini semakin berkembang, yaitu:

1. Menambahkan Fitur chat antara costumer dan admin
2. Menambahkan Fitur Notifikasi pemberitahuan kepada pihak admin apabila ada costumer yang melakukan booking online pada aplikasi bengkel motor

DAFTAR PUSTAKA

- Abdurahman, M., Safi, M., & Abdullah, M. H. (2018). Sistem Informasi Pengolahan Data Balita Berbasis Website Pada Kantor UPT-KB Kec. Ternate Selatan. *IJIS-Indonesian Journal On Information System*, 3(2), 85–92.
- Aditya, R., Pranatawijaya, V. H., & Putra, P. B. A. A. (2021). Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype. *Journal of Information Technology and Computer Science*, 1(1), 47–57.
- Ahmad Ansori. (2022). *Pengertian Activity Diagram : Tujuan, Simbol, dan Contohnya*. <https://www.ansoriweb.com/2020/03/pengertian-activity-diagram.html>
- Andika, R. (2022). *Implementasi Scrum Pada Rancang Bangun Aplikasi E-Konsultasi Dokter Berbasis Android (Studi Kasus Pada RSAL Dr. Komang Makes Belawan)*.
- Andipradana, A., & Hartomo, K. D. (2021). Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum. *Jurnal Algoritma*, 18(1), 161–172.
- Budiyanto, A. (2023). Perancangan Aplikasi Pembukuan Keuangan Warung Sembako Jakarta Timur Berbasis Manajemen Keuangan dengan Android. *Jurnal Esensi Infokom: Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, 7(1), 90–94.
- Fadilla, R., & Wiharko, T. (2023). Penerapan Metode Forward Chaining Dalam Sistem Pakar Deteksi Kerusakan Hardware Komputer Berbasis Android. *Digital Transformation Technology*, 3(2), 408–417.
- Fauzy, D. A., Iskandar, I., Rahmadhan, J., & Priambodo, R. (2020). Aplikasi Bengkel Motor Dengan Sistem Pakar Menggunakan Metode Forward Chaining. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(1), 89–96.
- Hadi, S. Y. (2020). Implementasi Metode Scrum Pada Aplikasi Marketplace Sektor Pertanian Berbasis Android. *Bachelor Thesis*.
- Hidayat, T., Sari, E., Munawir, M., & Susmanto, S. (2023). Sistem Informasi Reservasi dan Pemesanan Penginapan pada Wisma Syariah Lambhuk Kota Banda Aceh. *Karya Ilmiah Fakultas Teknik (KIFT)*, 3(4), 245–252.

- Karim, A., Mariana, A. R., & Ahmadi, A. (2019). Implementasi Sistem Informasi Cut Off Menggunakan FINA Pada Poly Jaya Pratama. *Jurnal Sisfotek Global*, 9(2).
- Mahyasari, R. (2022). *Pengertian XAMPP Menurut Para Ahli Lengkap Serta Fungsinya*.
- Nasution, E. A. (2018). Perancangan Aplikasi Pendataan Produksi Tiang Pancang PT. Pilaren Menggunakan Vb. Net Dan Mysql dengan Metode System Development Life Cycle. *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, 3(1), 1–14.
- Permana, A. Y., & Romadlon, P. (2019). Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode Sdlc Pada Pt. Mandiri Land Prosperous Berbasis Mobile. *Jurnal Sigma*, 10(2), 153–167.
- Raharjo, K. P., Fitriana, G. F., & Prasetyo, N. A. (2022). Perancangan Aplikasi Mobil Derek Berbasis Android Menggunakan Metode Scrum. *Jurnal Ilmiah Informatika Global*, 13(1).
- Siadari, C. (2022). *Pengertian Black Box Testing menurut para ahli*. Kumpulan Pengertian. Retrieved July.
- Simatupang, J. (2019). Perancangan sistem informasi jasa servis kendaraan dan penjualan suku cadang pada Jaya Bersama. *Jurnal Intra Tech*, 3(1), 1–9.
- Solahuddin, S. (2022). Model Percangan Aplikasi Makanan Ringan Cemilan Cemilan Akbar Store Berbasis Android. *Jurnal Global Multicom Komaksi*, 1(1), 16–22.
- Suwarno, J., Saputri, G., & Anas, I. (2023). Penerapan Aplikasi Pengelolaan Penyedia Hasil Pertanian Berbasis Website dengan Metode Extreme (Studi Kasus: Kelompok Tani Desa Susuk Mulya). *Jurnal Ilmu Komputer*, 6(3), 80–84.