

BAB I

PENDAHULUAN

A. Latar Belakang

Mutasi kendaraan sebagai salah satu sumber Pendapatan Asli Daerah (PAD) yang bersumber dari masyarakat, dimana pengelolaannya dahulu dilakukan oleh Badan Pendapatan Daerah, Kepolisian Daerah Republik Indonesia, dan PT. (Persero) Asuransi Kerugian Jasa Raharja yang berdasar pada Undang-Undang Republik Indonesia Nomor 28 Tahun 2009 Pasal 1, dan kini dalam pelaksanaan pemungutannya dilakukan di kantor SAMSAT (Sistem Administrasi Manunggal di bawah Satu Atap). Walikota parepare menilai potensi dari lahan sektor PAD kendaraan yang ada saat ini sangat memungkinkan untuk bertambah, sehingga Pendapatan Asli Daerah (PAD) dapat terus meningkat dan yang dilihat Kota Parepare merupakan suatu kota besar yang padat akan penduduk, tingginya mobilitas penduduk membuat jumlah kendaraan bertambah di setiap tahunnya.

Samsat Kota Parepare merupakan suatu sistem administrasi pelayanan publik yang diselenggarakan oleh tiga instansi pemerintahan dalam satu gedung (Hasan, 2020). Kerjasama yang dilakukan dalam satu atap ini meliputi beberapa instansi pemerintah diantaranya Polri, Dinas Pendapatan Daerah, dan PT Jasa Raharja. Polri memiliki fungsi penerbitan STNK dan Mutasi Kendaraan, Dinas Pendapatan Daerah menetapkan besarnya Pajak Kendaraan Bermotor (PKB) dan Bea Balik Nama Kendaraan Bermotor (BBN-KB), sedangkan PT Jasa Raharja mengelola Sumbangan Wajib Dana Kecelakaan Lalu Lintas Jalan (SWDKLLJ).

Mutasi kendaraan merupakan hal yang biasa terjadi. Mutasi selain fisik kendaraan yang biasanya berpindah lokasi sekaligus merupakan perpindahan lokasi pembayaran mutasi kendaraan bermotor namun masih dalam wilayah hukum Negara Kesatuan Republik Indonesia (NKRI) antar provinsi ataupun kabupaten/kota. Mengurus mutasi kendaraan identik dengan sesuatu yang ribet, berbelit-belit, buang-buang waktu, birokratis, pungli, calo dan sebagainya terlebih lagi apabila masyarakat baru saja membeli motor baru. Sehingga banyak orang yang menyerahkan pengurusan dokumen kendaraan tersebut kepada biro jasa. Mereka malas mengurus sendiri dokumen dari kendaraannya dan lebih suka membayar lebih mahal dengan menyerahkan proses mutasi kendaraan kepada biro jasa. Padahal jika mengetahui cara dan prosesnya, maka mengurus mutasi kendaraan sendiri tidak serumit dan sesulit yang dibayangkan tersebut. Namun apabila mengingat budaya tidak mau antre, ingin cepat dan tidak mau sulit menjadi salah satu faktor penghambat pengurusan mutasi kendaraan.

Selain itu, hal tersebut juga menjadi salah satu faktor-faktor pendukung menjamurnya praktik percaloan. Dengan keberadaan calo, otomatis pengguna jaspun berpikir semuanya bisa diselesaikan dengan cara praktis. Hal ini terbukti dengan kondisi pengguna jasa-jasa pelayanan yang terkadang lebih memilih mengeluarkan uang yang lebih besar untuk membayar pajak kendaraan bermotor, asalkan mereka bisa dengan mudah dan cepat. Dibandingkan harus mengeluarkan uang dari harga resmi dengan menjalani prosedur resmi yang mereka anggap melelahkan.

Permasalahan yang sering terjadi mengenai administrasi mutasi kendaraan keluar ini identik sangat tidak efektif, menggunakan waktu yang sangat lama, menghindari terjadinya kecurangan dalam administrasi sehingga masyarakat sering menyerahkan kepengurusannya melalui suatu jalur yang tidak resmi. Dari petugas bagian kepengurusan administrasi ini sering mengalami lama memproses administrasi mutasi kendaraan. Dengan permasalahan ini maka sangat dibutuhkan teknologi yang bisa mendukung pelayanan administrasi yang cepat dan efisien.

Salah satu tujuan dari bidang pelayanan adalah memberikan pelayanan terbaik kepada masyarakat yang membutuhkan. Baik instansi pemerintah maupun swasta saat ini tengah berlomba untuk meningkatkan inovasi pelayanan agar lebih berdaya guna dan berhasil guna melalui berbagai cara, salah satunya adalah dengan menerapkan teknologi khususnya teknologi informasi ke dalam sistem pelayanan mutasi kendaraan. Melalui penerapan teknologi informasi kedalam suatu mekanisme pelayanan yang ada, diharapkan masyarakat yang dilayani akan merasa puas dengan hasil maksimal dalam waktu pelayanan minimal.

Berdasarkan latar belakang permasalahan yang telah diuraikan diatas maka diidentifikasi masalahnya, dijadikanlah suatu sistem baru sebagai solusi permasalahan ini yang kemudian dituangkan dalam bentuk tugas akhir. Maka penulis mengusulkan dengan judul "**Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile***". Aplikasi ini sebagai sarana proses mutasi kendaraan keluar daerah, masyarakat lebih cepat mendapatkan hasil berkas yang diurus tanpa harus menunggu lama proses mutasi kendaraan keluar daerah.

B. Rumusan Masalah

Berdasarkan uraian dari latar belakang diatas maka dirumuskan masalah sebagai berikut :

1. Bagaimana mengetahui prosedur pembayaran mutasi keluar kendaraan bermotor antar daerah pada Kantor Samsat Kota Parepare?
2. Bagaimana merancang dan membangun suatu Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*?

C. Batasan Masalah

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang dan juga sesuai dengan latar belakang yang sudah diuraikan sebelumnya, maka penulis membatasi permasalahan penelitian. Batasan masalahnya antara lain :

1. Ruang lingkup dan uji coba implementasi penelitian ini difokuskan atau terpusat di wilayah Parepare, dalam hal ini sumber data diperoleh dari SAMSAT Kota Parepare.
2. Penelitian ini hanya terfokus pada pengelolaan data-data mutasi kendaraan keluar daerah.
3. Tidak dilakukan suatu pembahasan terhadap suatu data mutasi kendaraan lainnya dalam pengelolaan PAD di Kota Parepare.
4. Antarmuka pemrograman aplikasi dibangun dengan menggunakan bahasa pemrograman *Javascript*, *Dart*, dan *Android Flutter* serta database yang digunakan untuk menyimpan data adalah *MySQL*.

D. Tujuan Penelitian

Rumusan masalah yang telah didefinisikan sebelumnya, adapun tujuan dari penelitian dan perancangan sistem dalam penelitian ini adalah :

1. Untuk merancang dan membangun suatu Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*.
2. Untuk mengetahui suatu prosedur pembayaran mutasi keluar kendaraan bermotor antar daerah pada Kantor Samsat Kota Parepare.

E. Manfaat Penelitian

Salah satu fungsi dari penelitian adalah bisa memberikan manfaat dan kegunaan yang secara nyata akan diterima oleh berbagai pihak yang terkait dengan obyek penelitian secara luas. Dalam penelitian kali ini peneliti berharap bisa memberikan kegunaan. Penulisan tugas akhir ini memberikan manfaat ke beberapa pihak, antara lain :

1. Bagi Peneliti
 - a. Meningkatkan kemampuan berfikir tentang konsep seperti apa yang harus penulis angkat kedalam penelitian, tentunya berkaitan dekat dengan keadaan di Instansi yang menjadi tempat penelitian.
 - b. Meningkatkan wawasan mengenai langkah-langkah dan tata cara membuat sistem yang berguna sehingga dapat diukur melalui keefektifan dalam penggunaan dan keefisienan dalam pemanfaatan aplikasi.

- c. Sebagai tugas akhir untuk mendapatkan gelar sarjana strata 1 (S.1) pada Fakultas Teknik, Program Studi Informatika, Universitas Muhammadiyah Parepare.
2. Bagi Pemerintah dan Masyarakat
 - a. Memiliki sistem yang baru sebagai instrumen kunci kebijakan dalam mengawasi dan mengontrol serta meningkatkan kualitas pelayanan melalui inovasi digitalisasi sehingga dapat membantu Pemerintahan Daerah agar melangkah lebih baik didalam memberikan pelayanan kepada masyarakat.
 - b. Secara tidak langsung dapat menjadi bahan masukan untuk Direktorat Jenderal Perpajakan (DJP), Dinas Pendapatan Daerah dan Kantor Bersama SAMSAT agar dapat melakukan perbaikan terkait pemanfaatan teknologi informasi untuk menunjang pelayanan.
 - c. Mewujudkan keterlibatan masyarakat sebagai *social control* terhadap pengawasan pelaksanaan standar pelayanan yang diselenggarakan oleh setiap penyelenggara pelayanan publik melalui akses layanan gratis.
 3. Bagi Universitas/ Perguruan Tinggi
 - a. Menciptakan mahasiswa dan mahasiswi yang mampu menerapkan sebuah sistem yang menjadi dasar tempat penelitian.
 - b. Dapat membuat perguruan tinggi dikenal lebih luas oleh institusi swasta atau negeri yang dijadikan objek penelitian.
 - d. Penulisan tugas akhir ini diharapkan dapat dijadikan referensi akademis dari suatu pengembangan program studi Teknik Informatika Universitas Muhammadiyah Parepare selanjutnya.

F. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi ke dalam lima bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain. Setiap bab-nya terdiri dari beberapa sub tersendiri. Diawali dengan bab pendahuluan dan diakhiri dengan bab penutup yang berisi kesimpulan dan saran-saran. Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima bab sebagai berikut :

BAB I : PENDAHULUAN

Pada bab awal ini akan dijelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi uraian tentang teori-teori yang digunakan sebagai referensi dalam suatu “Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*”. Bab ini juga terdapat teori pengujian dua unit *testing*.

BAB III : METODOLOGI PENELITIAN

Pada bab ini berisi mengenai metode perancangan yang digunakan dalam membangun sistem ini. Pada bab ini juga membahas tentang identifikasi masalah, analisis kebutuhan aplikasi yang meliputi kebutuhan masukan, proses dan keluaran suatu aplikasi yang akan dibangun menggunakan UML.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil dari tahap analisis dan perancangan sesuai dengan metode yang digunakan pada alat dan sistem yang dibuat serta hasil dari pengujian serta bentuk-bentuk implementasi aplikasi beserta penerapannya.

BAB V : PENUTUP

Pada bab akhir ini berisi tentang kesimpulan dari penelitian yang dilakukan dan saran-saran yang diberikan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih maksimal.

DAFTAR PUSTAKA

Pada bagian ini akan dipaparkan tentang sumber-sumber literatur, tutorial, buku maupun situs-situs yang digunakan dalam pembuatan tugas akhir ini.

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Pada penelitian terkait ini adalah beberapa hasil dari yang dilakukan dari penelitian sebelumnya sebagai dasar acuan untuk menyelesaikan skripsi. Dari penelitian tersebut, peneliti menemukan beberapa penelitian yang mendukung untuk mengangkat topik yang berkaitan dengan penelitian penulis. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui internet. Berikut adalah tabulasi dari beberapa penelitian yang terkait.

Tabel 2.1 Penelitian Terdahulu

No	Nama Penulis (Tahun)	Judul Penelitian	Gambaran Umum
-----------	-------------------------------------	-------------------------	----------------------

1.	Ahmad Ali (2022)	“Aplikasi Antrian pada SAMSAT di Kota Palopo Berbasis <i>Android</i> ”	<p>Penelitian ini bertujuan untuk merancang Aplikasi Antrian di Samsat <i>Palopo Office</i> di <i>Android</i> yang mudah digunakan untuk mengolah data antrian yang dikirim oleh pengguna. Metode penelitian yang digunakan adalah <i>waterfall</i>. Dengan adanya aplikasi ini diharapkan dapat membantu masyarakat dan Samsat untuk ambil nomor antrian dan cek pajak kendaraan. Dari hasil pengujian sistem menggunakan Pengukuran skala <i>likert</i>, menunjukkan tingkat kelayakan sebesar 62,8%. Hasil instrumen uji validitas valid setiap item. Hasil uji reliabilitas instrumen adalah 0,989 dapat diandalkan.</p>
----	---------------------	--	--

No	Nama Penulis (Tahun)	Judul Penelitian	Gambaran Umum
2.	Wirantika Idris, Umar (2021)	“Sistem Informasi Pengelolaan Data Mutasi Kendaraan Bermotor Pada Samsat Halmahera Selatan Berbasis Web”	Tujuan dari penelitian ini adalah untuk mempermudah dalam proses pengelolaan data mutasi kendaraan bermotor dan dapat meningkatkan mutu pelayanan pada Samsat yang lebih efektif dan efisien. Sistem informasi data mutasi dirancang dengan pemrograman <i>PHP</i> dan <i>MySQL</i> sebagai basis datanya. Dengan dihasilkannya sistem informasi mutasi kendaraan bermotor pada samsat halmahera selatan berbasis <i>web</i> ini. Anggota samsat atau khususnya bagian administrasi yang bertanggung jawab mengenai mutasi kendaraan dapat memproses data atau formulir mutasi kendaraan secara terkomputerisasi.

3.	Ade Irma Suryani (2019)	“Sistem Informasi Biaya Balik Nama Kendaraan Bermotor Melalui Aplikasi <i>SMS Gateway</i> ”	<p>Aplikasi ini bertujuan untuk meningkatkan pelayanan publik dan efektifitas kinerja di kantor SAMSAT Kota Padang dengan cara memberikan informasi biaya BBNKB kepada masyarakat yang akan melakukan transaksi balik nama. Pengembangan <i>software</i> berbasis SDLC dengan metode <i>XP (Extreme Programming)</i>.</p> <p>Sistem ini dikembangkan dengan bahasa pemrograman <i>VB.Net</i>, dan database <i>MySQL</i> sebagai menyimpan data serta <i>Gammu</i> sebagai konfigurasi <i>SMS Gateway</i>. Hasil dari penelitian ini adalah sebuah aplikasi yang dapat digunakan bagi suatu pelayanan agar mengetahui jumlah biaya balik nama kendaraan bermotor, yang dapat meningkatkan efisiensi kerja pada kantor SAMSAT kota Padang.</p>
----	-------------------------------	---	--

B. Tinjauan Teori

1. Aplikasi

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam Bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu sub kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2017) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut Hartono (2020) aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Aplikasi menurut Dhanta dikutip dari Sanjaya (2019) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Menurut Jogiyanto dikutip oleh Ramzi (2017) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri.

2. E-Payment

E-Payment merupakan suatu model pembayaran yang memudahkan dan memberikan rasa kenyamanan kepada pengguna yang akan melakukan seluruh jenis pembayaran. Namun pengguna hanya memerlukan internet atau jaringan agar transaksi tersebut bisa berjalan dengan lancar sehingga tidak memerlukan adanya sebuah pertemuan.

Menurut Pram 2018 *e-Payment* merupakan metode pembayaran yang memiliki fasilitas internet sebagai sarana perantara. Sehingga sistem *e-Payment* sangat memudahkan dan membantu masyarakat dalam melakukan pembayaran pada semua jenis transaksi kapan pun dan dimana pun.

Menurut Nielsen 2019 bahwa *e-Payment* itu sendiri merupakan kegiatan transaksi yang dilakukan pada seluler, sehingga transaksi yang dilakukan oleh semua kalangan dapat dilakukan pada satu jenis perangkat saja. 38% banyak orang yang telah menggunakan layanan aplikasi seluler, dan 47% banyak orang yang menggunakan transaksi pada perangkat seluler. *E-Payment* digunakan pada saat ini untuk bertransaksi jarak jauh seperti *online shopping*, seiring dengan semakin bertumbuhnya penggunaan internet dan semakin banyaknya *e-commerce*. Yang termasuk dalam pembayaran elektronik adalah ATM, *e-money*, *internet banking*, *mobile banking* dan *QRIS*.

QRIS (Quick Response Code Indonesian Standard) adalah standar kode *QR* yang dikembangkan oleh Bank Indonesia untuk memfasilitasi pembayaran digital di Indonesia. *QRIS* memungkinkan berbagai penyelenggara jasa sistem pembayaran (PJSP) seperti bank dan dompet digital untuk menggunakan satu kode *QR* yang dapat diakses oleh semua aplikasi pembayaran yang terdaftar. Dengan demikian, *QRIS* bertujuan untuk menyederhanakan transaksi digital, meningkatkan inklusi keuangan, serta memajukan

ekosistem pembayaran non-tunai di Indonesia. *QRIS* biasanya berbentuk persegi putih kecil dengan bentuk geometris hitam (Putra, Budi 2020).



Gambar 2.1 *QRIS*

Adapun cara pembuatan *QRIS*, langkah pertama adalah memilih Penyedia Jasa Pembayaran (PJP) yang telah mendapat izin dari Bank Indonesia, seperti *GoPay*, *OVO*, *DANA*, *ShopeePay*, atau bank yang menawarkan layanan pembayaran berbasis *QR code*. Setelah itu, merchant harus mendaftar sebagai penyedia layanan melalui platform PJP yang dipilih, dengan menyediakan dokumen yang dibutuhkan. PJP kemudian akan melakukan verifikasi data-data merchant. Jika disetujui, merchant akan diberikan *QR code QRIS* unik yang dapat digunakan untuk menerima pembayaran dari semua aplikasi pembayaran yang mendukung *QRIS*.

3. Mutasi Kendaraan

Mutasi kendaraan adalah usaha untuk mengurus administrasi mengenai perpindahan identifikasi kendaraan bermotor dari daerah asal ke daerah tujuan yang disesuaikan dengan kepindahan dari alamat pemilik kendaraan yang baru. Kegiatan

melakukan mutasi kendaraan akan membuat nama dan alamat pada surat kendaraan seperti STNK dan BPKB akan berubah dari nama dan alamat pemilik lama kendaraan menjadi nama dan alamat pemilik baru kendaraan bermotor tersebut. Begitu pula dengan plat nomor kendaraan bermotor tersebut juga akan berubah. Proses mutasi kendaraan bermotor dibagi menjadi 2 (dua) yaitu mutasi kendaraan bermotor pada satu daerah dan mutasi kendaraan bermotor lain daerah.

4. Samsat Kota Parepare

Satuan Administrasi Menunggal Satu Atap (SAMSAT) Kota Parepare merupakan salah satu administrasi pelaksanaan teknis pemerintah daerah Provinsi Sulawesi Selatan di bidang pungutan pendapatan daerah yang merupakan gabungan dari beberapa instansi di dalamnya yaitu instansi Kepolisian, Dinas Pendapatan Daerah Provinsi Sulawesi Selatan serta PT. Asuransi Kecelakaan Jasa Raharja dituntut untuk lebih meningkatkan suatu pendapatan asli daerah. Satuan Administrasi Menunggal Satu Atap (SAMSAT) Kota Parepare yang terletak ± 1 Km dari Kantor Walikota Kota Parepare. Berlokasi Jl. Jend. Sudirman Parepare, Sulawesi Selatan, Indonesia 91261.

a. Visi dan Misi Samsat Kota Parepare

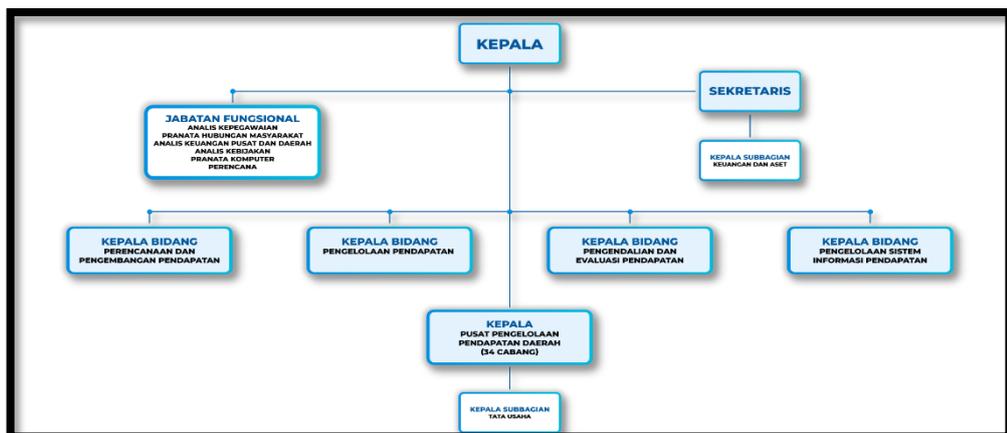
1) Visi

“Terwujudnya Pelayanan Prima Dalam Pengurusan Administrasi dan Regident Kendaraan Bermotor Melalui Satu Keterpaduan Pelayanan POLRI, PEMDA, dan JASA RAHARJA Pada Samsat Parepare”.

2) Misi

- a) Memberikan Pelayanan Kepada Masyarakat dengan Menjunjung Tinggi Etika Profesi.
- b) Melaksanakan Proses Administrasi Kendaraan Bermotor Secara Tepat dan Cepat.
- c) Mewujudkan Aparat Pelaksana Samsat yang Bersih, Jujur, Cakap dan bertanggung jawab serta profesional.
- d) Meningkatkan Kesadaran Masyarakat dalam Membayar Pajak.
- e) Penataan suatu Arsip Kendaraan yang Tertib untuk Memudahkan Identitas dan Keamanan Dokumen.

b. Struktur Organisasi Samsat Kota Parepare



Gambar 2.2 Struktur Organisasi Samsat Kota Parepare

(Sumber : <https://bapendasulsel.web.id/>)

C. Tinjauan Perangkat Lunak

1. Android



Gambar 2.3 Logo *Android*

(Sumber : <https://2.bp.blogspot.com/>)

Android merupakan OS (*Operating System*) *mobile* yang tumbuh ditengah OS lainnya yang berkembang pada zaman sekarang (Hermawan, 2019). OS lainnya seperti *windows mobile*, *Iphone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Dalam konteks aplikasi *e-payment QRIS*, *Android* memanfaatkan komponen seperti *Activity* untuk antarmuka pengguna, *Service* untuk proses latar belakang, dan *BroadcastReceiver* untuk notifikasi dan pembaruan sistem. Aplikasi ini berinteraksi dengan server *backend* melalui API untuk memproses pembayaran *QRIS*, yaitu sistem kode *QR* standar untuk pembayaran digital di Indonesia. Ketika pengguna melakukan mutasi kendaraan, aplikasi akan mengautentikasi data kendaraan, melakukan verifikasi pembayaran *QRIS*, dan menyinkronkan informasi mutasi dengan database. Penggunaan komponen *Android* seperti *Intent* dan *ContentProvider* memungkinkan aplikasi untuk mengelola suatu data dan komunikasi dengan aplikasi lain, serta memastikan integrasi yang mulus dengan sistem pembayaran dan administrasi kendaraan.

2. *Flutter*



Gambar 2.4 Flutter

(Sumber : <https://www.cloudsavvyit.com/>)

Flutter merupakan salah satu *framework* atau *Software Development Kit* (SDK) untuk pengembangan aplikasi *mobile* yang dapat berjalan di sistem operasi *ios* dan *android*. *Flutter* diciptakan oleh *Google* dan dirilis pertama kali pada bulan Mei 2017 (versi alpha). Pada Desember 2018, *Google* meluncurkan *flutter* 1.0.0 (versi stabil). Dalam aplikasi *e-payment QRIS*, *Flutter* memungkinkan pembuatan *UI* yang menarik dan fungsional dengan *widget-widget* yang dapat disesuaikan. Aplikasi ini mengintegrasikan API untuk melakukan transaksi *QRIS*, yaitu sistem kode *QR* standar yang memfasilitasi pembayaran digital di Indonesia. *Flutter* mengelola status aplikasi dan navigasi menggunakan *state management* dan *routing*, serta memanfaatkan paket seperti *http* atau *dio* untuk komunikasi dengan *server backend*. Proses mutasi kendaraan melibatkan pengumpulan data kendaraan dan identitas pengguna, verifikasi pembayaran *QRIS*, dan sinkronisasi dengan database atau otoritas terkait. *Flutter* mendukung pemrograman reaktif yang memungkinkan aplikasi untuk memperbarui *UI* secara dinamis berdasarkan status transaksi dan hasil permohonan mutasi kendaraan, menjamin pengalaman pengguna yang mulus dan responsif.

3. *Android Studio*



Gambar 2.5 *Android Studio*

(Sumber : <https://trainingeltasa.com/>)

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform android*. *Android studio* ini berbasis pada *IntelliJ IDEA*, sebuah IDE untuk bahasa pemrograman *Java*. Bahasa pemrograman utama yang digunakan adalah *Java*, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk *deploy* ke perangkat *android*. *Android Studio* juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan lebih profesional yang telah tersedia didalamnya *Android Studio IDE, Android SDK tools* (Smith, 2020).

Dalam suatu aplikasi *e-payment QRIS*, *Android Studio* digunakan untuk merancang antarmuka pengguna dengan XML, serta menulis logika aplikasi dalam bahasa pemrograman *Java* atau *Kotlin*. Proses mutasi kendaraan dalam aplikasi ini melibatkan autentikasi pengguna, verifikasi transaksi *QRIS*, dan sinkronisasi data mutasi dengan server atau otoritas terkait, yang semuanya dikelola dan diuji menggunakan alat yang disediakan oleh *Android Studio* untuk memastikan aplikasi berfungsi dengan baik di berbagai perangkat *Android*.

4. *Java Development Kit (JDK)*



Gambar 2.6 *Java Development Kit (JDK)*
(Sumber : <https://cdn.educba.com/>)

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi *Java*. *JDK* merupakan *superset* dari *JRE*, berisikan segala sesuatu di *JRE* ditambahkan *compiler* dan *debugger* yang diperlukan untuk membangun aplikasi. *Android Studio SDK* dikembangkan dengan menggunakan bahasa pemrograman *java*. Demikian aplikasi *android* juga dikembangkan *Java*. Sehingga *Java Development Kit (JDK)* adalah komponen yang harus diinstal. Pengembangan *android* membutuhkan instalasi baik versi 6 atau 7 Edisi Standar *Kit Java Platform* Pembangunan. *Java* disediakan di kedua paket pengembangan (*JDK*) dan *runtime (JRE)*.

5. *Software Development Kit*



Gambar 2.7 *Software Development Kit (SDK)*
(Sumber : <https://cdn.educba.com/>)

Java Development Kit (JDK) adalah distribusi Teknologi *Java* oleh *Oracle Corporation*, pada proses pengembangan dan pembuatan sebuah aplikasi *Android*, terdapat bantuan untuk membuatnya yaitu dengan menggunakan suatu perangkat bantuan berbasis perangkat lunak yang dikenal sebagai *Android SDK*. *Android SDK* merupakan suatu perangkat API (*Application Programming Interface*) yang dibutuhkan untuk memulai pengembangan dan pembuatan aplikasi pada *platform Android*, dalam proses tersebut digunakan bahasa pemrograman *Java*. Dengan adanya *Android SDK* dapat membantu *developer* untuk mengembangkan aplikasi yang bukan merupakan bawaan dari *mobilephone* atau *smartphone* tersebut.

6. *Visual Studio Code*



Gambar 2.8 *Visual Studio Code*
(Sumber : <https://trainingeltasa.com/>)

Visual Studio Code adalah suatu editor *source code* yang dikembangkan oleh Microsoft untuk *Windows*, *Linux* dan *MacOS*. Ini termasuk dukungan untuk *debugging*, *GIT Control* yang disematkan, penyorotan *sintaks*, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. *Visual Studio Code* gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi *proprietary* (Lardinois, 2021).

D. Tahapan Perancangan Sistem

1. Use Case

Use case diagram adalah abstraksi dari interaksi antara sistem dan *actor*.

Use case bekerja mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2018).

2. Kamus Data

Tabel 2.2 Simbol-Simbol Dalam Kamus Data

No	Simbol	Arti
1	=	Disusun atau terdiri atas
2	+	Dan
3	[I]	Baik ...atau...
4	{ <i>n</i> }	n kali diulang/ bernilai banyak
5	()	Data operasional
6	*...*	Batas komentar

Kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan informasi yang mengalir pada sistem dengan lengkap dari suatu sistem informasi. Kamus data di buat berdasarkan arus data yang mengalir pada *Data Flow*

Diagram (DFD). Sukamto dan Shalahuddin, (2019), "Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)".

Kamus data merupakan tempat penyimpanan dari aliran-aliran data, *file-file* dan proses-proses dalam sebuah sistem. Bagian ini menjelaskan secara detail proses-proses yang terjadi disetiap proses, *file* dan struktur data mengenai model sistem yang digambarkan. Kamus data digunakan untuk menjelaskan semua data yang mengalir atau digunakan dalam sistem, yaitu mengenai arus data yang masuk ke dalam sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Kamus data adalah repositori (*manual atau computed-based*) yang berisi informasi tentang berbagai objek data muncul pada setiap DFD. tidak ada format standar untuk kamus data. Notasi menyimpan data dalam DFD menunjukkan adanya salah satu item data yang lebih dari data yang tersimpan. Sebuah kamus data hanya dapat memberikan gambaran sederhana dari data. (Sharma, 2020)

Ada berbagai jenis data, masing-masing mampu memegang hanya jenis tertentu dari informasi :

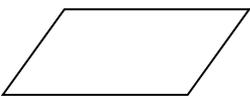
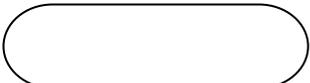
- a. *Integer*: mampu menahan hanya seluruh nomor tunggal.
- b. *Real* : mampu menahan 1 angka dengan *decimal*.
- c. *Date*: hanya bisa menampung tanggal.
- d. *String*: bisa menahan teks, termasuk nomor dan tanda baca.
- e. *Boolean*: hanya bisa memegang nilai '*true*' or '*false*'. Sebuah kamus data yang khas terlihat seperti ini.

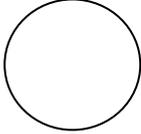
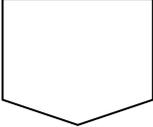
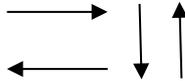
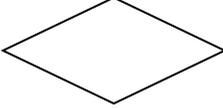
3. Bagan Alir Program

Bagan alir program merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir sistem. Menurut Ade Hastuty (2021) “Bagan alir program (*program flowchart*) merupakan bagan alir yang mirip dengan suatu bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem”

Supardi (2017), “*flowchart* merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir”. Adapun simbol-simbol dari *Flowchart* adalah sebagai berikut :

Tabel 2.3 Simbol *Flowchart*

SIMBOL	KETERANGAN
	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.

SIMBOL	KETERANGAN
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang masih sama.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses.
	Simbol keputusan, digunakan untuk suatu penyelesaian kondisi didalam program.

4. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) menurut Munawar dalam buku “Pemodelan Visual dengan UML” (2017 : 17) menyatakan UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Sedangkan menurut menurut Adi Nugroho (2019:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan

untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode Unified Modelling Language (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan suatu sistem atau perangkat lunak yang akan dikembangkan, yaitu :

a. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language (UML)* :

- 1) *Structural things*, bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
- 2) *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang ruang dan waktu.
- 3) *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta *subsistem-subsistem*.
- 4) *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi (*relationship*)

Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya *statis* atau *dinamis*. Ke 9 jenis diagram dalam UML itu adalah diagram kelas, diagram objek bersifat statis, *use case*

diagram, sequence diagram, statechart diagram, activity diagram, component diagram, deployment diagram. (Shohiq, 2021)

Tabel 2.4 *Table Use Case Diagram*

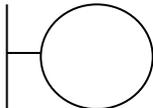
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

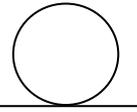
Tabel 2.5 *Symbol Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 2.6 *Symbol Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Objekct</i> <i>(Partisipan)</i>	<i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma.

2		<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3		<i>Life line</i>	<i>Life line</i> mengidentifikasi keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>lifeline</i> adalah garis putus-putus <i>vertical</i> yang ditarik oleh sebuah <i>object</i>
4		<i>Collaborations</i>	<i>Activication</i> dinotasikan sebagai kotak segi empat yang digambar pada sebuah <i>lifeline</i> <i>action</i> mengidentifikasi sebuah <i>object</i> yang akan beraksi.
NO	GAMBAR	NAMA	KETERANGAN
5		<i>Boundary</i>	<i>Boundary</i> terletak diantara system dengan dunia di sekelilingnya. Semua <i>form</i> , laporan-laporan, antara muka ke perangkat keras seperti <i>printer</i> atau <i>scenner</i> dan antara muka ke system lainnya adalah termasuk dalam kategori.

6		<i>Control</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Entity</i>	<i>Entity</i> digunakan untuk menengani informasi yang akan disimpan secara permanen.

E. Metode Pengujian Perangkat Lunak

Untuk metode pengujian yang dilakukan pada penelitian ini adalah metode pengujian langsung yaitu dengan menggunakan pengujian *White Box* dan *Black Box*. *White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. *Black Box* Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang.

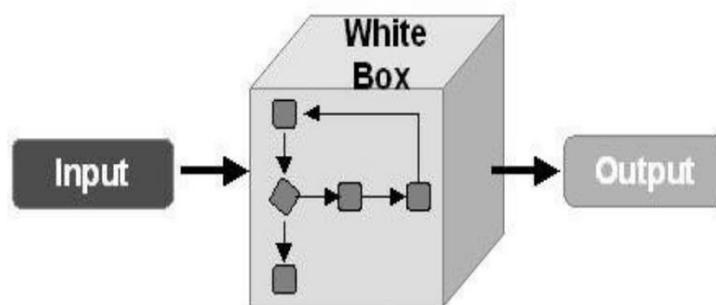
Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *typical*, kritis atau abnormal. Pengujian harus mencakup *unit testing*, yang mengecek *validasi* dari *prosedur* dan fungsi-fungsi secara *independen* dari komponen sistem yang lain. Kemudian modul *testing* harus menyusul dilakukan untuk mengetahui apakah

penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit *testing*, yaitu *Black Box Testing* adalah suatu metode pengujian perangkat lunak dimana penguji mengevaluasi fungsionalitas dari aplikasi tanpa melihat ke dalam struktur internal. Pengujian ini berfokus pada verifikasi jalur logika internal, kondisi cabang, dan aliran data dalam perangkat lunak (Patton, 2021).

1. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *test case* yang menggunakan struktural untuk mendapatkan *test case*, *test* ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika).



Gambar 2.9 Pengujian *White Box*
(Sumber : <https://www.codester.com/>)

Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* menginputkan *password user* yang salah, untuk kesalahan semacam ini akan

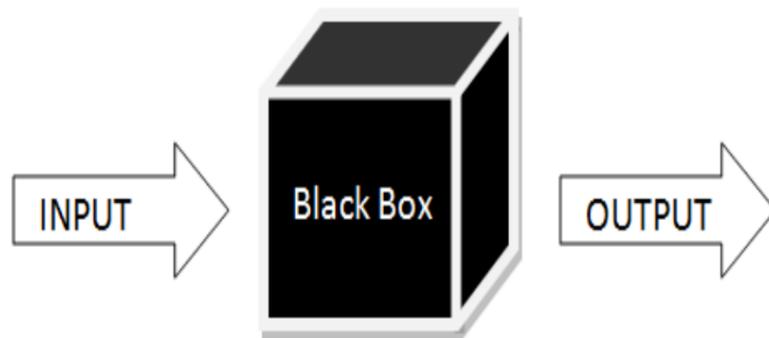
memberikan suatu informasi kepada *user* mengenai kesalahan yang dilakukan. Dalam *white box testing*, penguji memiliki pengetahuan tentang bagaimana sistem bekerja di dalamnya, termasuk kode sumber, alur data, kondisi, dan keputusan logika.

- a. Pengujian Aliran Kontrol (*Control Flow Testing*): Pengujian melibatkan pemeriksaan jalur eksekusi dalam kode. Tujuannya adalah untuk memastikan bahwa setiap jalur logika dalam program telah diuji dan berfungsi dengan benar.
- b. Pengujian Aliran Data (*Data Flow Testing*): Pendekatan ini fokus pada penggunaan variabel dalam program. Penguji mengevaluasi apakah variabel diinisialisasi dengan benar, digunakan dengan benar, dan dimusnahkan pada waktu yang tepat.
- c. Pengujian Kondisi (*Condition Testing*): Dalam pengujian ini, penguji memastikan bahwa kondisi logika dalam program (seperti *if*, *while*, *for*) berfungsi dengan benar dalam semua skenario.
- d. Pengujian Loop (*Loop Testing*): Teknik ini digunakan untuk menguji loop dalam program. Pengujian ini memastikan bahwa loop berfungsi dengan benar dalam berbagai kondisi, seperti loop berjalan 0 kali, 1 kali, atau banyak kali.
- e. Pengujian Fungsi (*Function Testing*): Memfokuskan pada suatu pengujian fungsi atau metode individual untuk memastikan bahwa mereka bekerja.

2. *Black Box Testing*

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti

pengujian *black box testing*, mengevaluasi hanya dari tampilan luarnya (*interfacenya*), fungsionalitas tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*).



Gambar 2.10 Pengujian *Black Box*
(Sumber : <https://www.codester.com/>)

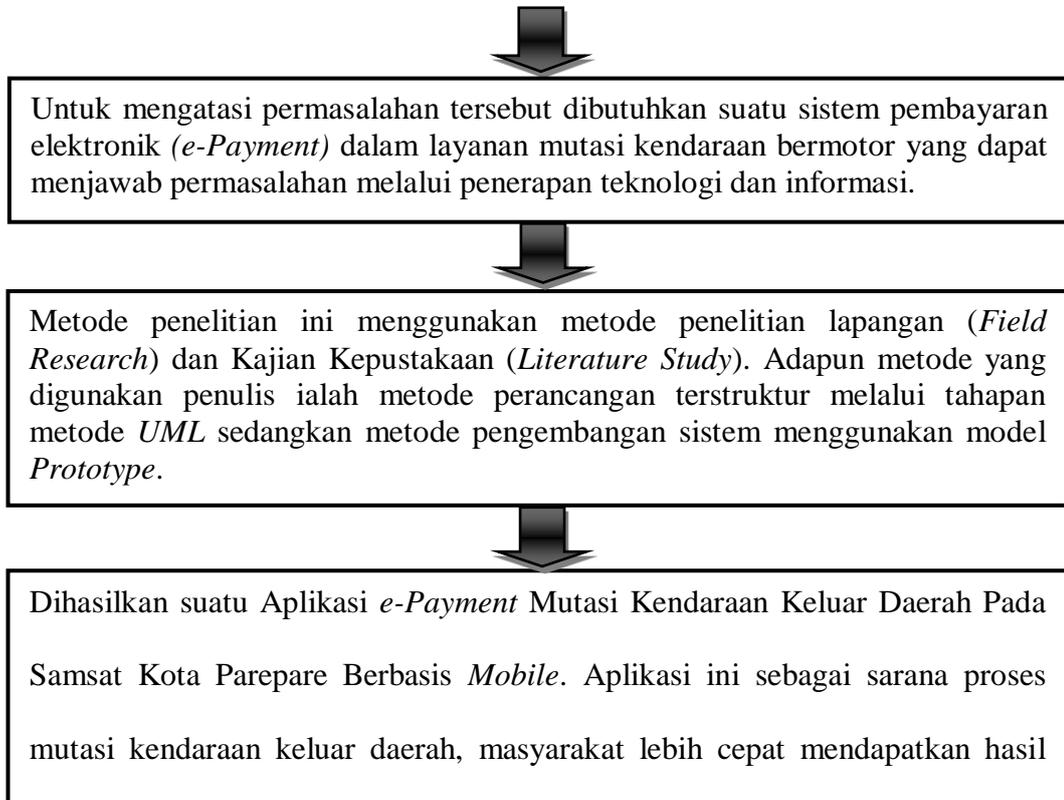
Di sisi lain, dalam *black box testing*, penguji tidak memiliki akses ke kode sumber atau logika internal aplikasi. Penguji hanya melihat aplikasi dari sudut pandang pengguna, dengan fokus pada fungsionalitas dan *output* yang dihasilkan berdasarkan *input* yang diberikan.

- a. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak.
- b. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran (*output*) yang dihasilkan.
- c. Kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

F. Kerangka Pikir

Untuk memperjelas suatu kerangka berpikir yang akan dituliskan dalam penelitian akhir ini. Maka berikut digambarkan kerangka pikir tersebut.

Mengurus mutasi kendaraan identik dengan sesuatu yang ribet, berbelit-belit, buang-buang waktu, birokratis, pungli, calo dan sebagainya terlebih lagi apabila masyarakat baru saja membeli motor baru. Selain itu, hal tersebut juga menjadi salah satu faktor pendukung memasyarakatkan praktik peraculan. Dengan keberadaan calo otomatis



Gambar 2.11 Kerangka Pikir

BAB III

METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

1. Lokasi penelitian dilakukan oleh penulis bertempat di Kantor SAMSAT Kota Parepare, Jl. Jenderal Sudirman No.80, Cappa Galung, Kec. Bacukiki Barat, Kota Parepare, Sulawesi Selatan 91122.
2. Pelaksanaan penelitian selama (\pm) 2 bulan, terhitung pasca seminar proposal.

B. Metode Penelitian

Agar dalam perancangan sistem lebih terarah dan terfokus di dalam mendapatkan data dan informasi yang akurat yang berhubungan dengan pokok pembahasan maka pengumpulan data dilakukan dengan tiga cara adalah metode *literature riview*, wawancara, observasi dan kajian pustaka.

1. Penelitian Kepustakaan (*Literature riview*)

Penulis melakukan studi kepustakaan dengan mencari informasi yang akurat sesuai dengan judul tugas akhir menggunakan beberapa teori, literatur, jurnal, buku dan kepustakaan. Penulis berharap agar program aplikasi yang dibuat nantinya tidak menyimpang dari defenisi-defenisi yang ada sehingga benar-benar memiliki acuan yang sesuai dan lengkap. Studi literatur merupakan salah satu

metode pengumpulan data untuk mendapatkan data-data yang sifatnya teoritis yaitu dengan cara membaca literatur yang relevan dengan pengamatan yang penulis lakukan.

2. Wawancara (*Interview*)

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data terhadap narasumber/sumber data. Wawancara dilakukan untuk mendapatkan informasi yang akurat sesuai dengan data yang dibutuhkan dimana pengumpulan data dilakukan melalui wawancara langsung dengan instansi terkait, yakni dengan cara mengadakan tanya jawab langsung kepada perangkat-perangkat SDM di kantor SAMSAT (Satuan Administrasi Menunggal Satu Atap) Kota Parepare dan masyarakat yang dapat memberikan penjelasan langsung ataupun data sebagai pelengkap penelitian ini.

3. Pengamatan (*Observation*)

Studi lapangan (*observasi*) merupakan teknik pengumpulan data dengan langsung terjun ke lapangan untuk mengamati permasalahan yang terjadi secara langsung di tempat kejadian secara sistematis kejadian-kejadian, perilaku, objek-objek yang dilihat dan hal-hal lain yang diperlukan dalam mendukung penelitian yang sedang berlangsung.

4. Dokumentasi

Dokumentasi berasal dari dokumen yang berarti barang-barang tertulis jadi dokumentasi adalah suatu teknik dimana data diperoleh dari dokumen yang ada pada benda-benda tertulis, buku-buku yang berkaitan dengan objek penelitian. Tujuan metode ini untuk memperoleh data dari hasil laporan dan keterangan-

keterangan secara tertulis, tergambar, terekam ataupun tercetak seperti foto dan dokumen yang berhubungan mengenai dokumen administrasi mutasi kendaraan keluar daerah pada SAMSAT Kota Parepare.

C. Metode Pengumpulan Data

Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan (*observation*) langsung di tempat penelitian. Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan. Adapun uraian dari tahapan adalah sebagai berikut :

1. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Studi *Literature*

Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain.

3. Pengumpulan Data

Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan langsung di tempat penelitian.

4. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

5. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

6. Pengujian

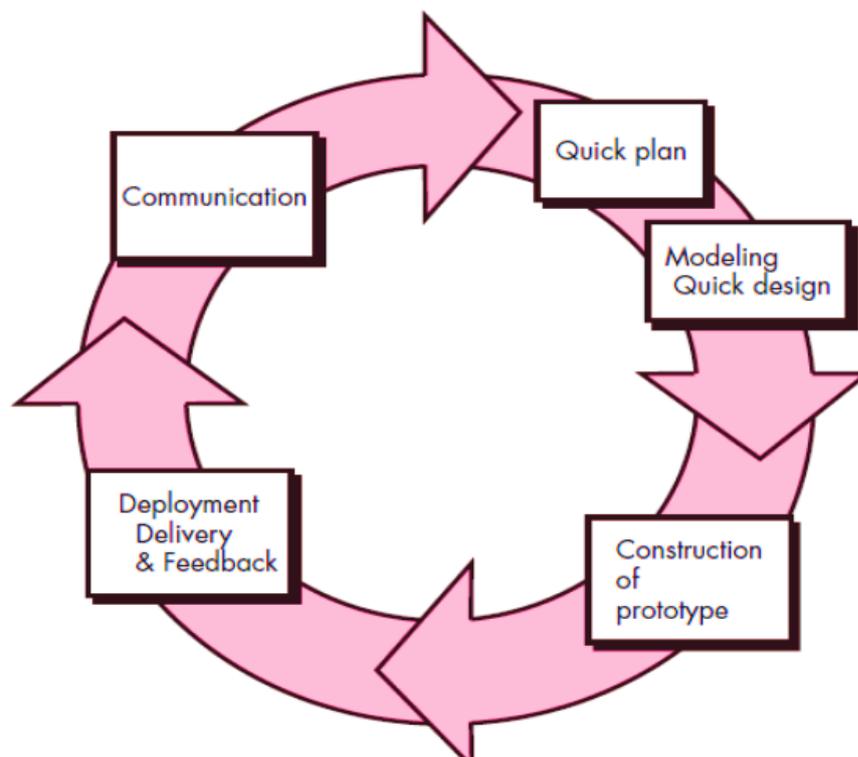
Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

D. Metode Pengembangan Sistem

Metode Pengembangan sistem merupakan suatu kerangka kerja mengenai bagaimana prosedur dan perhitungan dalam mutasi keluar kendaraan bermotor antar daerah, adapun metode pengembangan yang digunakan yaitu *Prototype* hal ini dikarenakan pada kantor SAMSAT (Satuan Administrasi Menunggal Satu Atap) Kota Parepare belum sama sekali memiliki aplikasi yang berkaitan dengan layanan mutase kendaraan dalam bentuk digitalisasi sehingga sangat cocok menggunakan metode *prototype*.

Menurut Pressman (2020), dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan

untuk mengembangkan perangkat lunak yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*. Tahap berikutnya membangun perancangan kilat yang selanjutnya akan di evaluasi kembali sebelum diproduksi secara benar. Metode pengembangan sistem yang digunakan adalah model *prototype* dengan lima tahapan yaitu *communication*, *quick plan*, *modeling quick design*, *construction of prototipe*, *deployment, delivery & feedback*.



Gambar 3.1 Model *Prototype*
(Sumber : <https://www.researchgate.net/>)

Berikut adalah tahapan-tahapan dalam metode *prototype* :

1. *Communication*

Pada tahap ini, antara pelanggan dengan tim pengembang perangkat lunak berkomunikasi mengenai spesifikasi kebutuhan yang diinginkan. Tim pengembang melakukan pengumpulan data yang akan digunakan dalam pembuatan sistem yang dibutuhkan.

2. *Quick Plan*

Tim pengembang memberikan gambaran besar mengenai sistem yang akan dibuat kepada dinas terkait. Dalam hal ini sistem yang akan dibuat adalah Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*, kemudian pengembang merancang penggunaan spesifikasi *hardware* dan *software* yang akan digunakan dalam pembuatan sistem.

3. *Modeling Quick Design*

Pada tahap ini, pengembang membuat perancangan sistem menggunakan UML (*Unified Modeling Language*) sebagai alat bantu untuk pemodelan terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *user interface*.

4. *Construction of Prototipe*

Setelah tahap pemodelan, maka pengembang mulai melakukan pengkodean program. Bahasa pemrograman yang digunakan adalah *Dart* dan *Javascript* menggunakan database *MySQL*.

5. *Deployment, Delivery & Feedback*

Tahap pengkodean program dibarengi oleh tahapan implementasi dan pengujian sistem. Sistem diuji dengan metode *white box* dan *black box testing*

dengan pengujian fungsional. Setelah sistem lulus uji coba, maka sistem yang baru diberikan kepada pengguna serta pengembang dalam mengajari pengguna dalam penggunaan sistem yang baru. Jika ada kekurangan atau penambahan kebutuhan sistem, maka pihak-pihak terkait akan mengkomunikasikan kembali dengan tim pengembang.

E. Sumber Data Penelitian

Adapun jenis dan sumber data yang dibutuhkan dalam penelitian ini adalah sebagai berikut :

1. Data Primer merupakan data yang diperoleh melalui wawancara yang dilakukan langsung dengan responden yang dapat mewakili beberapa sumber data. Didapatkan secara langsung dari informan yang memiliki data dan informasi mengenai penelitian ini oleh peneliti, yaitu oleh SAMSAT (Satuan Administrasi Menunggal Satu Atap) Kota Parepare dan masyarakat melalui wawancara disertai dengan observasi (pengamatan).
2. Data sekunder merupakan data yang dihasilkan oleh peneliti melalui sumber yang sudah tersedia. Data yang diperoleh melalui studi kepustakaan atau dari berbagai literatur dengan menelaah buku-buku dan tulisan-tulisan atau internet, jurnal hukum, serta peraturan perundang-undangan yang relevan dengan permasalahan yang diteliti dan tidak terlepas dari pengamatan objek oleh peneliti.

F. Alat dan Bahan

Tahap desain dilakukan untuk menentukan ruang lingkup pembuatan aplikasi yang ditangani, yang selanjutnya dijabarkan dalam bentuk diagram alir. Dalam pembuatan program aplikasi tersebut penulis membutuhkan perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Adapun *Hardware* dan *Software* yang

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 (64 bit)</i>
Bahasa Pemrograman	<i>Flutter, Javascript</i>
Database	<i>MySQL</i>
<i>Tools</i>	<i>Visual Studio Code</i>
<i>Interface</i>	<i>Microsoft Visio</i>

digunakan adalah :

1. Perangkat keras (*hardware*)

Perangkat keras yang digunakan untuk membuat aplikasi menggunakan PC (*Personal Komputer*) dengan spesifikasi yang dapat dilihat pada tabel 3.1 berikut :

Table 3.1 Spesifikasi Perangkat Keras

2. Perangkat Lunak (*software*)

Perangkat lunak yang digunakan untuk merancang dan membangun aplikasi ini dapat dilihat pada tabel 3.2 berikut :

Table 3.2 Spesifikasi Perangkat Lunak

Jenis	Spesifikasi
Laptop	<i>Lenovo</i>
<i>Processor</i>	<i>Prosesor Intel(R) Core (TM) i5-3210M</i>
<i>Ram</i>	4 GB DDR 4
<i>Hardisk</i>	500 GB
<i>Smartphone</i>	<i>Android versi 11 Red Velvet Cake</i>

3. Bahan Penelitian

Berupa data-data yang dikumpulkan peneliti dari berbagai laporan-laporan atau dokumen-dokumen yang bersifat informasi tertulis yang digunakan dalam penelitian yang diperoleh langsung dari kantor SAMSAT (Satuan Administrasi Menunggal Satu Atap) Kota Parepare.

BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem

1. Analisis Sistem Berjalan

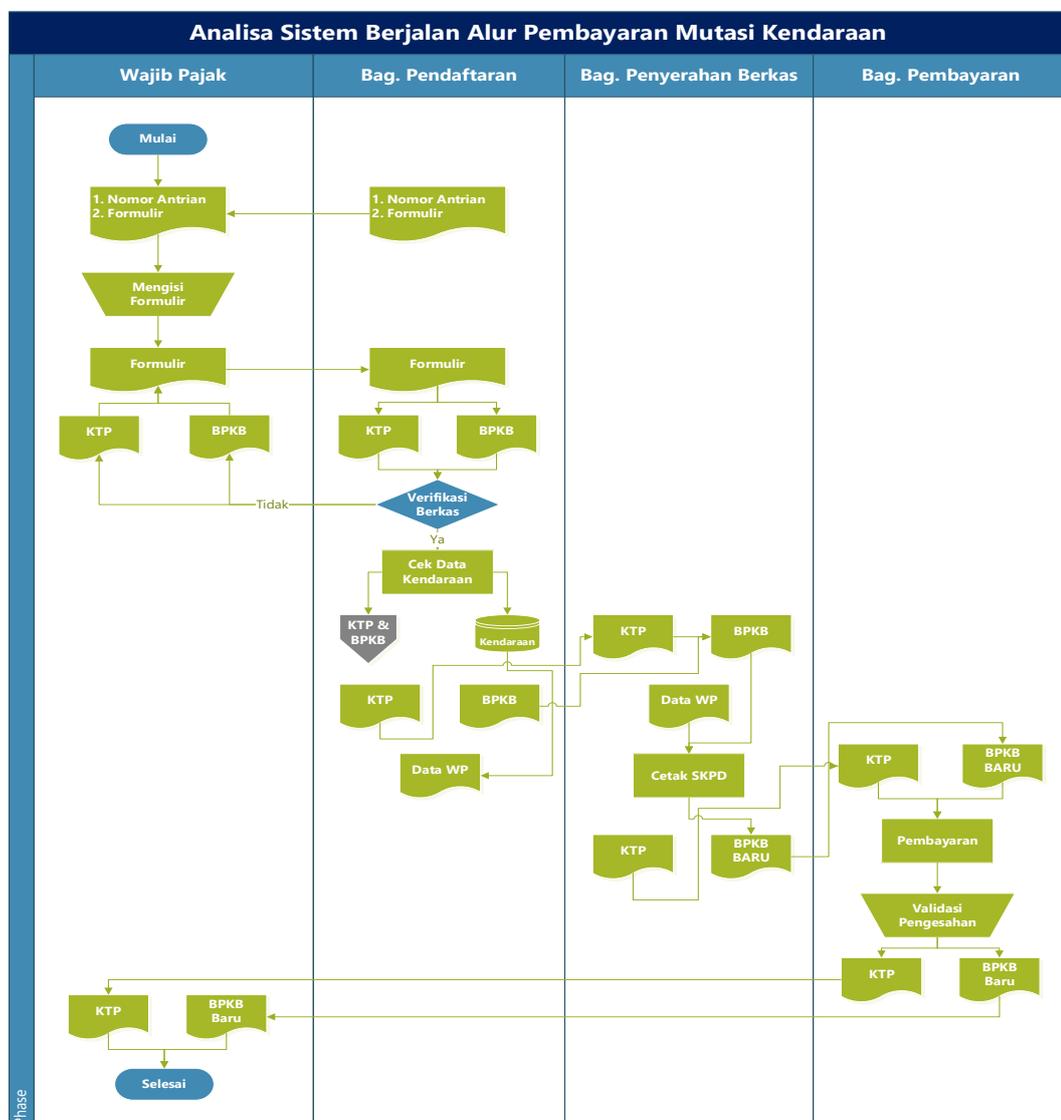
Samsat Kota Parepare merupakan suatu sistem administrasi pelayanan publik yang diselenggarakan oleh tiga instansi pemerintahan dalam satu gedung, sistem ini dengan maksud guna mempercepat suatu kegiatan pelayanan publik maka diadakan pelayanan dalam satu gedung. Kerjasama yang dilakukan dalam satu atap ini meliputi beberapa instansi pemerintah diantaranya Polri, Dinas Pendapatan Daerah, dan PT Jasa Raharja (Persero).

Mutasi kendaraan merupakan hal yang biasa terjadi. Mutasi selain fisik kendaraan yang biasanya berpindah lokasi sekaligus merupakan perpindahan lokasi pembayaran mutasi kendaraan bermotor namun masih dalam wilayah hukum Negara Kesatuan Republik Indonesia (NKRI) antar provinsi ataupun kabupaten/kota. Mengurus mutasi kendaraan identik dengan sesuatu yang ribet, berbelit-belit, buang-buang waktu, birokratis, pungli, calo dan sebagainya terlebih lagi apabila masyarakat baru saja membeli motor baru. Sehingga banyak orang yang menyerahkan pengurusan dokumen kendaraan tersebut kepada biro jasa.

Permasalahan yang sering terjadi mengenai administrasi mutasi kendaraan keluar ini identik sangat tidak efektif, menggunakan waktu yang sangat lama, menghidari terjadinya kecurangan dalam administrasi sehingga masyarakat sering menyerahkan kepengurusannya melalui suatu jalur yang tidak resmi. Dari petugas

bagian kepengurusan administrasi sering mengalami lama dalam memproses administrasi mutasi kendaraan. Berdasarkan suatu pengamatan dan observasi yang dilakukan oleh penulis, adapun gambaran alur kerja sistem yang sedang berjalan dapat digambarkan dalam bentuk *flowmap diagram* seperti di bawah ini :

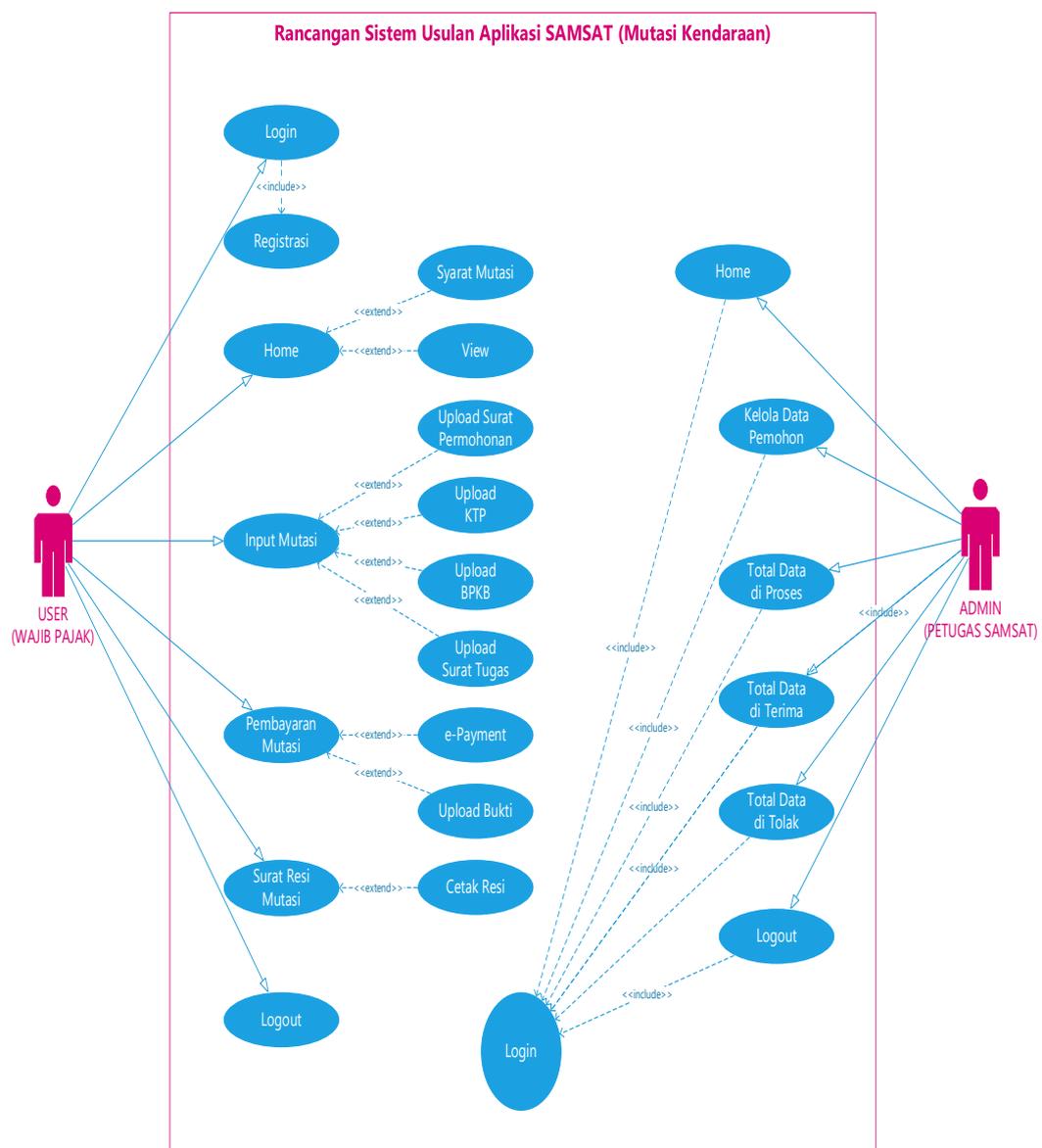
Berdasarkan pengamatan dan observasi yang dilakukan oleh penulis, adapun gambaran alur kerja sistem yang sedang berjalan dapat digambarkan dalam bentuk *use case diagram* seperti di bawah ini.



Gambar 4.1 Use Case Diagram Sistem Berjalan

2. Analisis Sistem Diusulkan

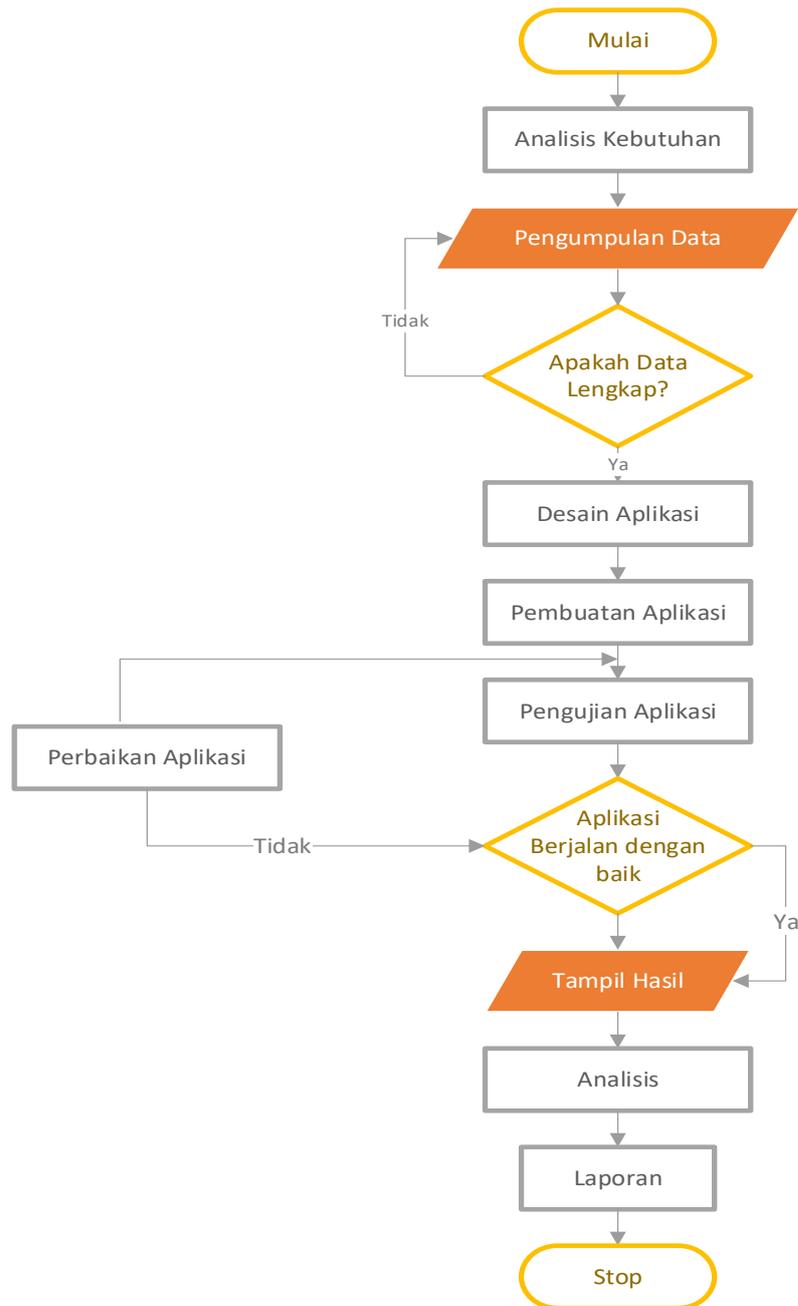
Analisis sistem merupakan penguraian dari suatu sistem yang utuh kedalam suatu bagian komponennya untuk mengidentifikasi dan mengevaluasi permasalahan. Bagian analisis terdiri dari analisis masalah dan analisis kebutuhan. Sistem yang diusulkan menggunakan *Use Case Diagram*. Adapun sistem yang penulis usulkan dapat dilihat pada gambar berikut.



Gambar 4.2 *Use Case Diagram* Sistem Diusulkan

3. Flowchart

Flowchart sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.



Gambar 4.3 *Flowchart* Alur Penelitian

Alur penelitian yang digambarkan dalam *flowchart* ini menunjukkan proses pengembangan aplikasi berbasis tahapan analisis dan evaluasi. Berikut adalah deskripsi detail dari setiap langkah :

a. Mulai (*Start*)

Tahapan pertama dalam alur ini adalah memulai proses penelitian atau proyek pengembangan aplikasi. Ini adalah langkah awal yang menginisiasi seluruh proses kerja.

b. Analisis Kebutuhan (*Needs Analysis*)

Pada tahap ini, dilakukan analisis kebutuhan yang mendetail. Tujuannya adalah untuk memahami secara mendalam apa yang dibutuhkan oleh pengguna atau *stakeholder* dari aplikasi yang akan dikembangkan. Tahapan ini mencakup wawancara dengan pengguna, diskusi dengan tim, dan pengumpulan informasi untuk merumuskan spesifikasi aplikasi.

c. Pengumpulan Data (*Data Collection*)

Setelah kebutuhan dianalisis, data yang diperlukan untuk pengembangan aplikasi mulai dikumpulkan. Data ini bisa berupa informasi teknis, preferensi pengguna, serta data operasional lain yang mendukung aplikasi. Data ini menjadi dasar untuk desain dan pengembangan aplikasi.

d. Apakah Data Lengkap? (*Is the Data Complete?*)

Setelah pengumpulan data, dilakukan evaluasi terhadap kelengkapan data yang sudah dikumpulkan. Jika data tidak lengkap, proses ini akan kembali ke tahap Pengumpulan Data untuk melengkapi informasi yang diperlukan.

- 1) Jika Tidak (*No*) → Proses kembali ke Pengumpulan Data untuk melengkapi data.
- 2) Jika Ya (*Yes*) → Proses berlanjut ke tahap berikutnya.

e. Desain Aplikasi (*Application Design*)

Pada tahap ini, aplikasi mulai dirancang. Desain mencakup spesifikasi teknis, antarmuka pengguna (*UI*), dan struktur kerja aplikasi. Desain ini menjadi cetak biru bagi tim pengembang aplikasi untuk memastikan aplikasi sesuai dengan kebutuhan dan data yang telah dikumpulkan.

f. Pembuatan Aplikasi (*Application Development*)

Setelah desain selesai, aplikasi mulai dibangun oleh tim pengembang. Tahapan ini melibatkan pemrograman, penyiapan database, integrasi sistem, dan pembuatan fitur sesuai desain yang telah disusun.

g. Pengujian Aplikasi (*Application Testing*)

Setelah aplikasi dibuat, perlu dilakukan pengujian untuk memastikan bahwa aplikasi berfungsi dengan baik. Pengujian ini bisa melibatkan beberapa tahap, seperti uji coba internal oleh tim pengembang dan uji coba eksternal oleh pengguna atau *stakeholder*.

h. Apakah Aplikasi Berjalan dengan Baik? (*Is the Application Running Well?*)

Setelah pengujian, evaluasi dilakukan untuk mengetahui apakah aplikasi berjalan dengan baik. Jika ditemukan bug atau masalah lain.

- 1) Jika Tidak (*No*) → Proses lanjut ke Perbaikan Aplikasi (*Application Improvement*). Setelah diperbaiki, aplikasi kembali diuji.
- 2) Jika Ya (*Yes*) → Proses berlanjut ke Tampil Hasil (*Show Results*).

i. Perbaiki Aplikasi (*Application Improvement*)

Jika aplikasi tidak berfungsi dengan baik, dilakukan proses perbaikan untuk memperbaiki bug atau meningkatkan performa aplikasi. Setelah perbaikan selesai, aplikasi diuji ulang untuk memastikan semua masalah telah diatasi.

j. Tampil Hasil (*Show Results*)

Jika aplikasi telah berjalan dengan baik, hasil dari aplikasi atau fungsionalitasnya ditampilkan kepada pengguna atau stakeholder. Tahap ini bertujuan untuk menunjukkan bahwa aplikasi bekerja sesuai dengan spesifikasi dan kebutuhan yang telah ditentukan.

k. Analisis (*Analysis*)

Setelah aplikasi dijalankan dan hasilnya ditampilkan, dilakukan analisis terhadap hasil yang diperoleh. Tahapan ini melibatkan evaluasi terhadap data atau hasil yang dihasilkan oleh aplikasi untuk memastikan kualitas, efisiensi, dan kegunaannya.

l. Laporan (*Report*)

Setelah analisis selesai, laporan disusun berdasarkan hasil dan temuan selama pengembangan dan pengujian aplikasi. Laporan ini dapat digunakan untuk dokumentasi proyek, evaluasi, dan referensi untuk pengembangan lebih lanjut.

m. Stop (*End*)

Setelah seluruh proses selesai dan laporan telah disusun, proyek atau penelitian dihentikan. Tahap ini menandai berakhirnya seluruh rangkaian proses pengembangan aplikasi.

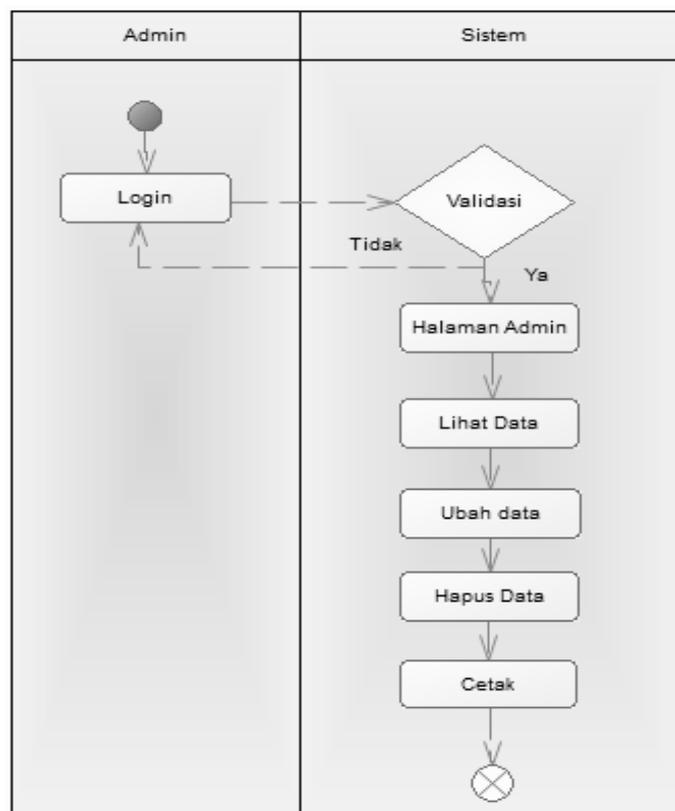
B. Analisis Aliran Data UML

Analisis aliran data yang diusulkan ini bertujuan mengetahui proses informasi, dalam analisis sistem penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Sequence diagram*.

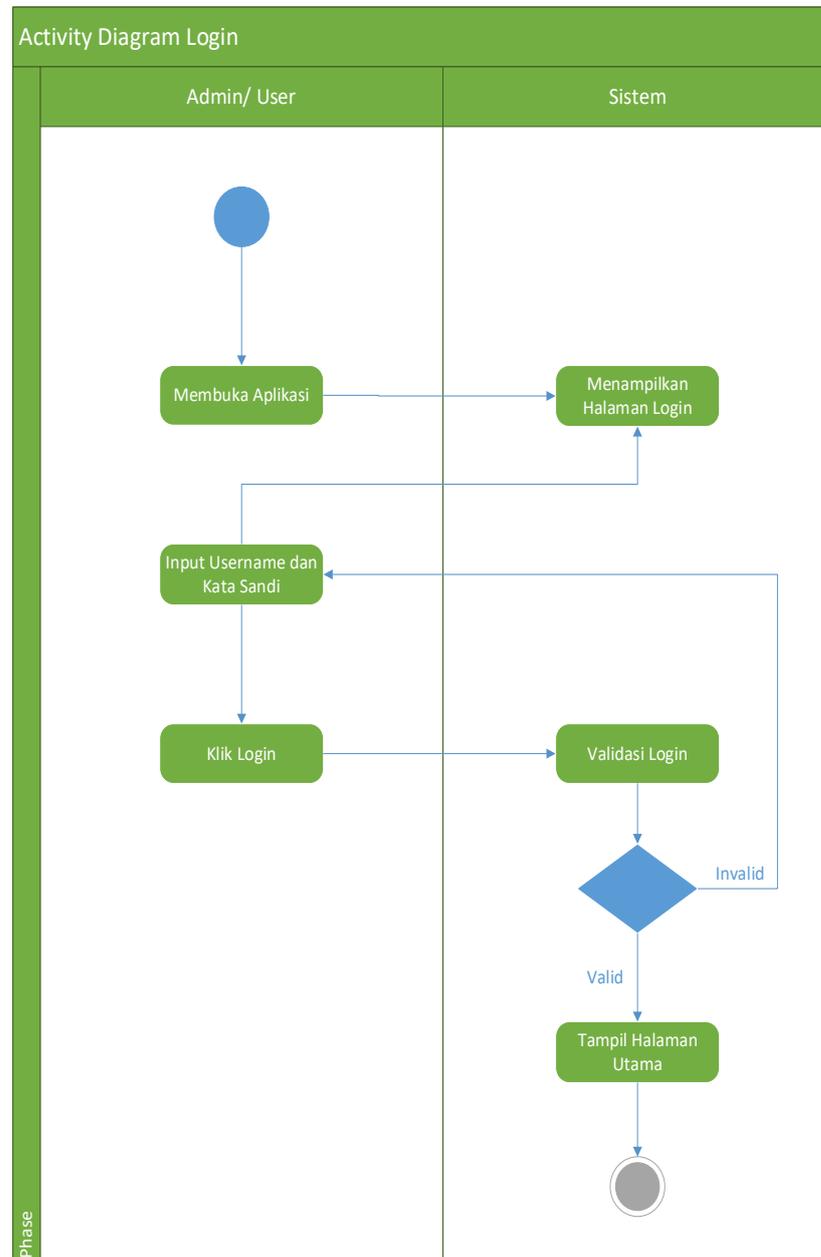
1. Activity Diagram

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah *system*.

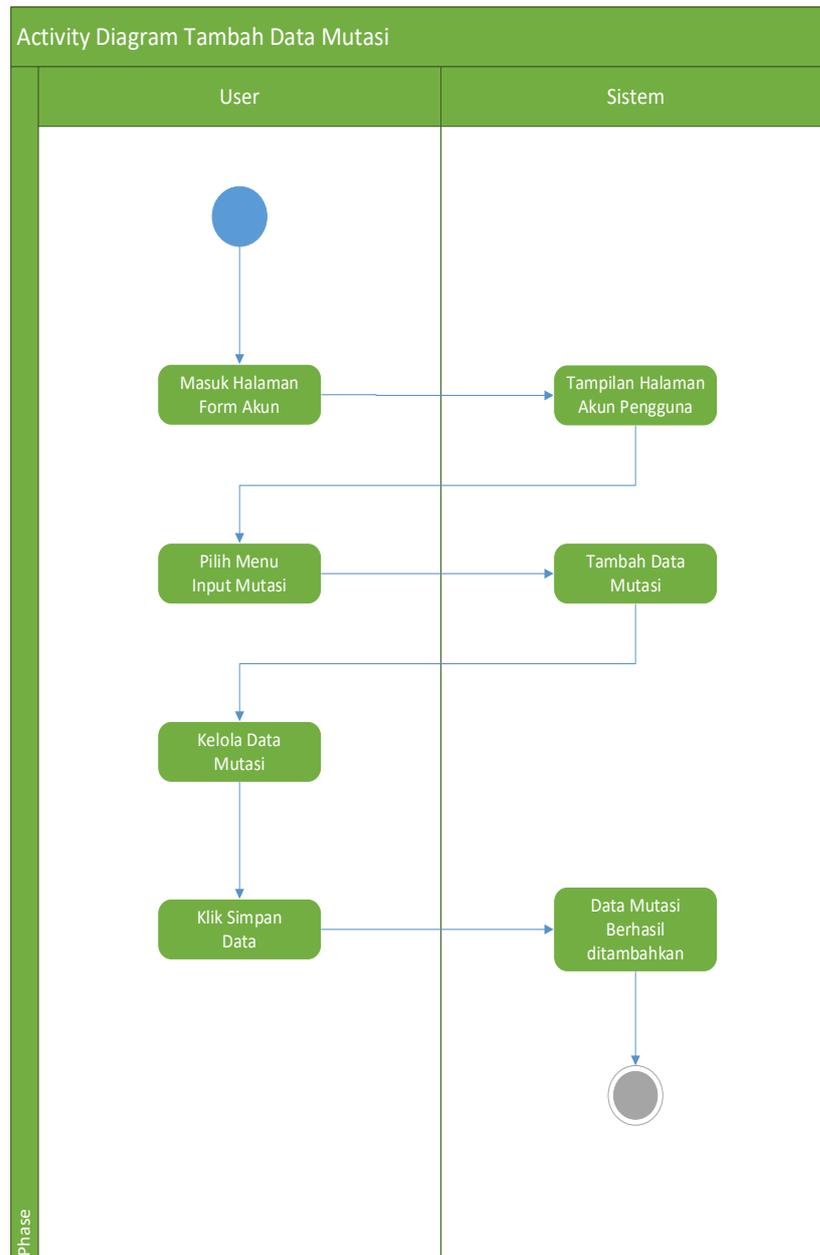
a. Activity Diagram Admin



Gambar 4.4 Activity Diagram Admin

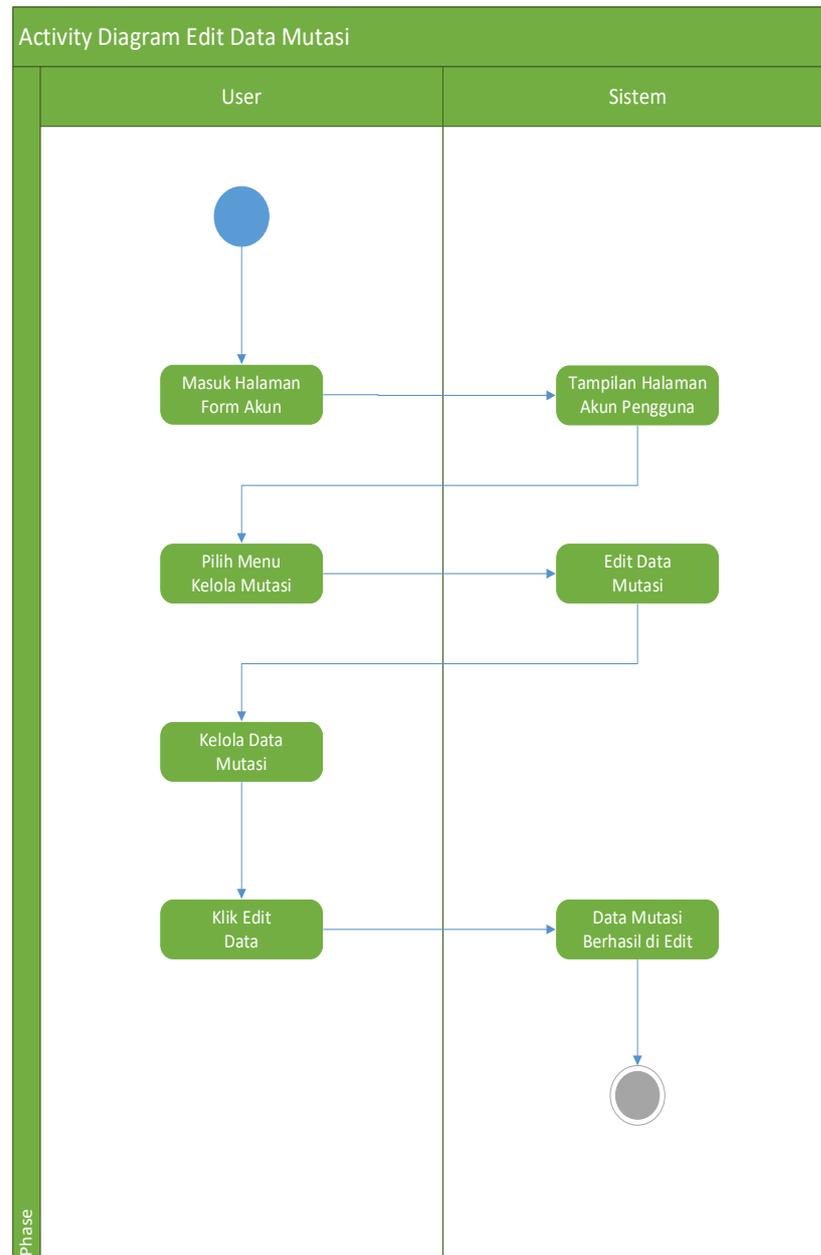
b. *Activity Diagram Login*Gambar 4.5 *Activity Diagram Login*

c. *Activity Diagram* Tambah Data Mutasi

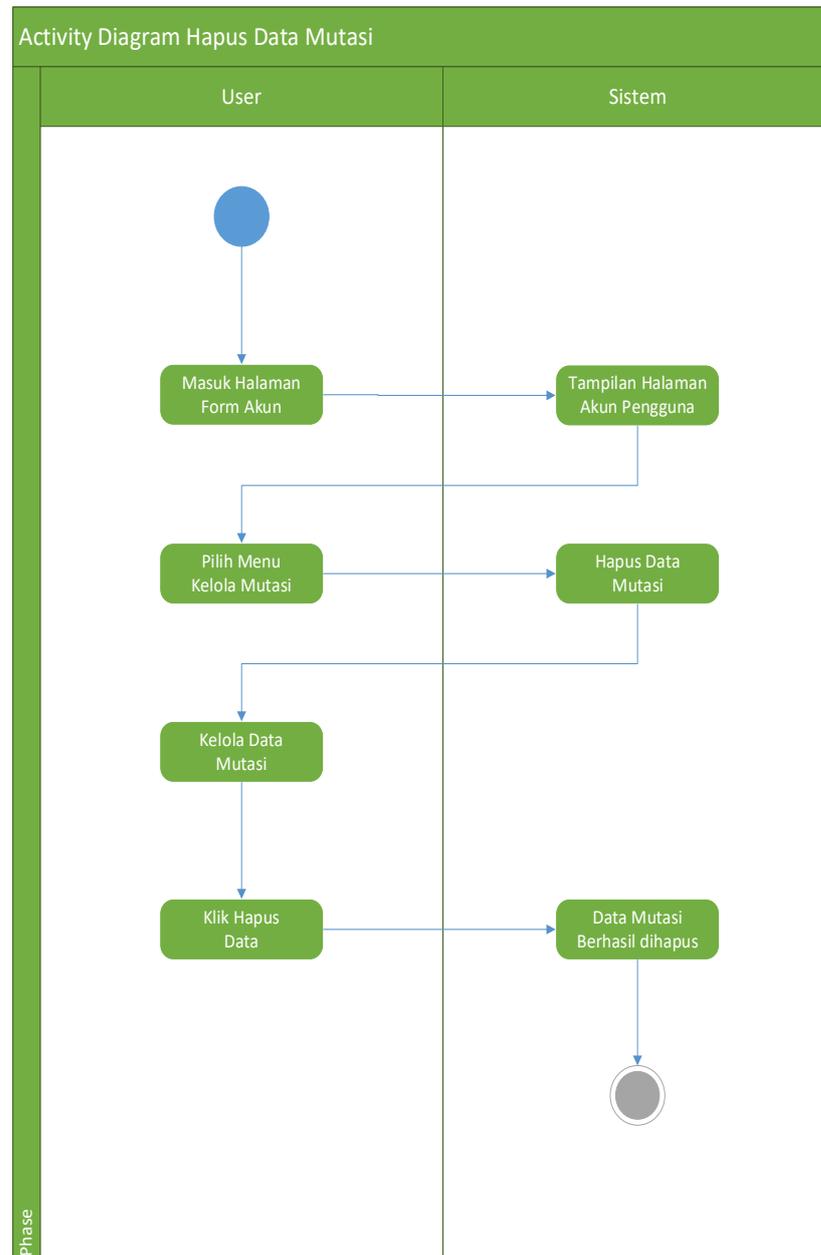


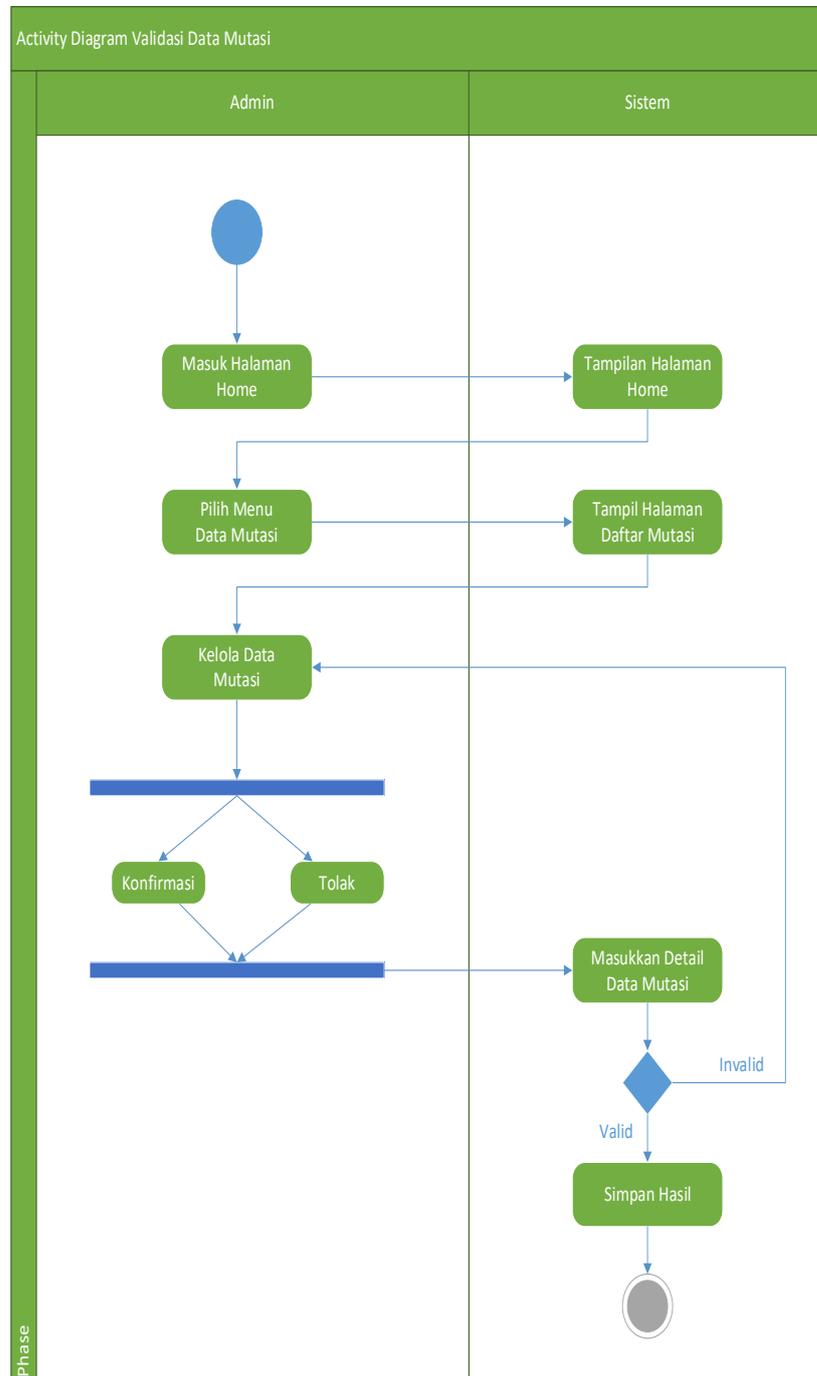
Gambar 4.6 *Activity Diagram* Tambah Data Mutasi

d. *Activity Diagram Edit Data Mutasi*



Gambar 4.7 *Activity Diagram Edit Data Mutasi*

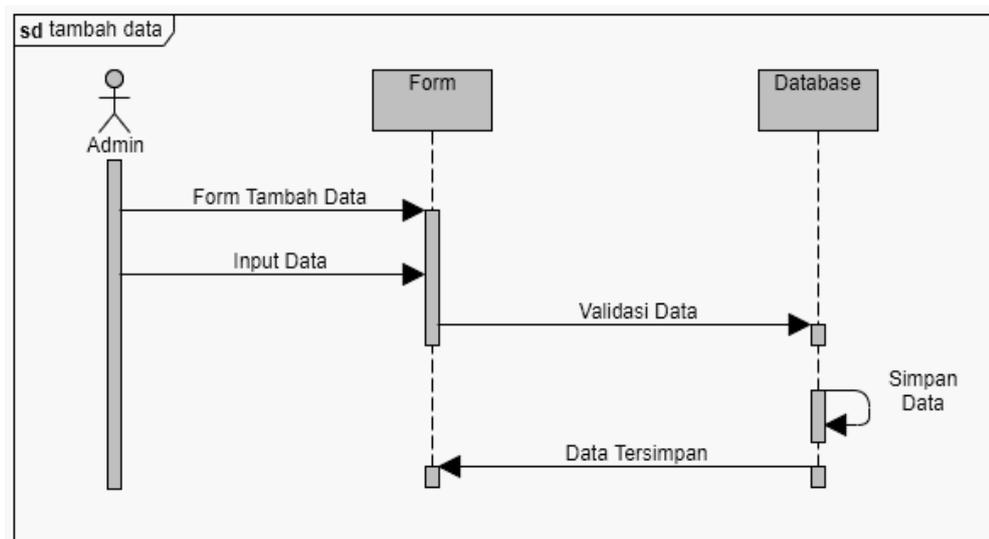
e. *Activity Diagram* Hapus Data MutasiGambar 4.8 *Activity Diagram* Hapus Data Mutasi

f. *Activity Diagram* Validasi Data MutasiGambar 4.9 *Activity Diagram* Validasi Data Mutasi

2. Sequence Diagram

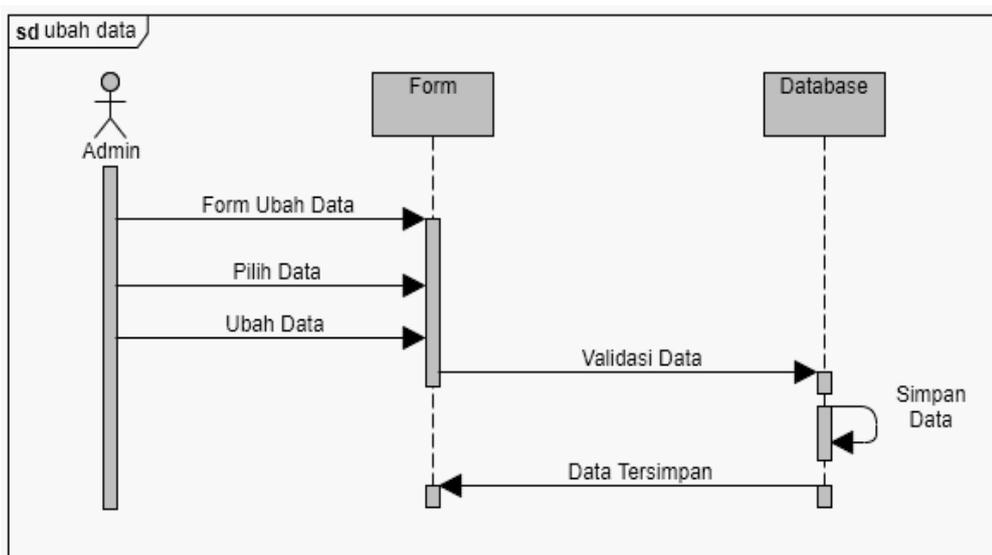
Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. Sequence Diagram Tambah Data Admin



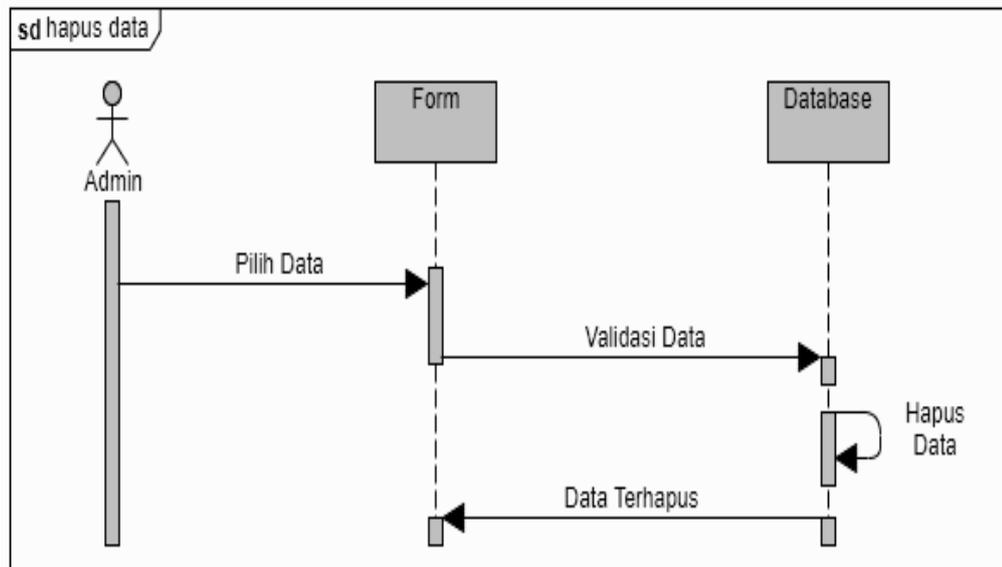
Gambar 4.10 *Sequence Diagram* Tambah Data

b. Sequence Diagram Ubah Data Admin



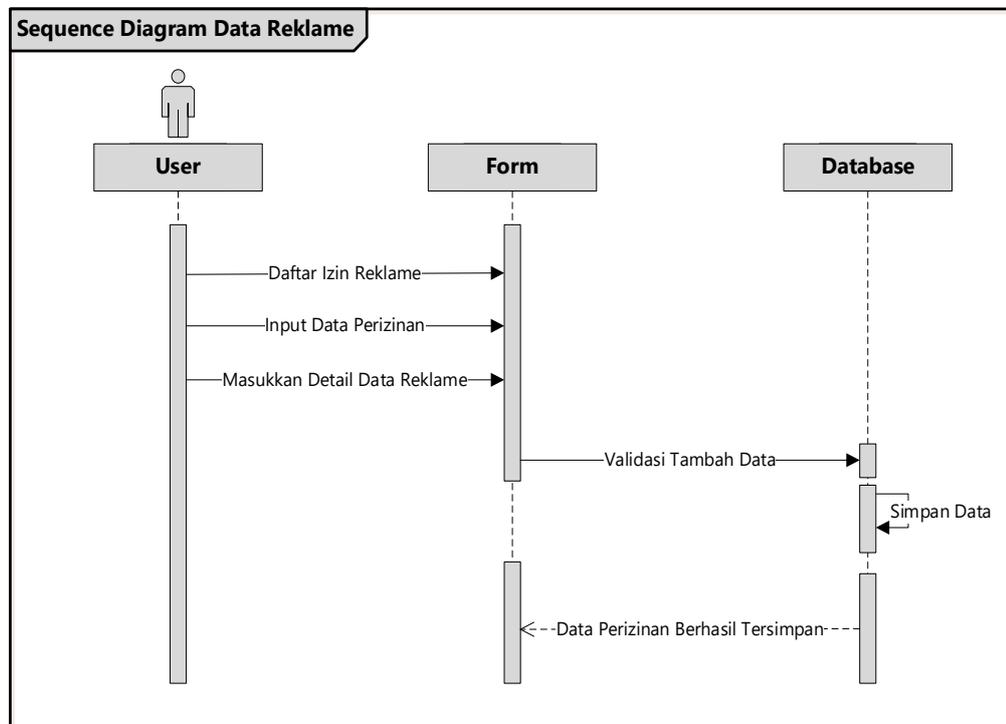
Gambar 4.11 *Sequence Diagram* Ubah Data

c. *Sequence Diagram Hapus Data Admin*



Gambar 4.12 *Sequence Diagram Hapus Data*

d. *Sequence Diagram Data Mutasi*

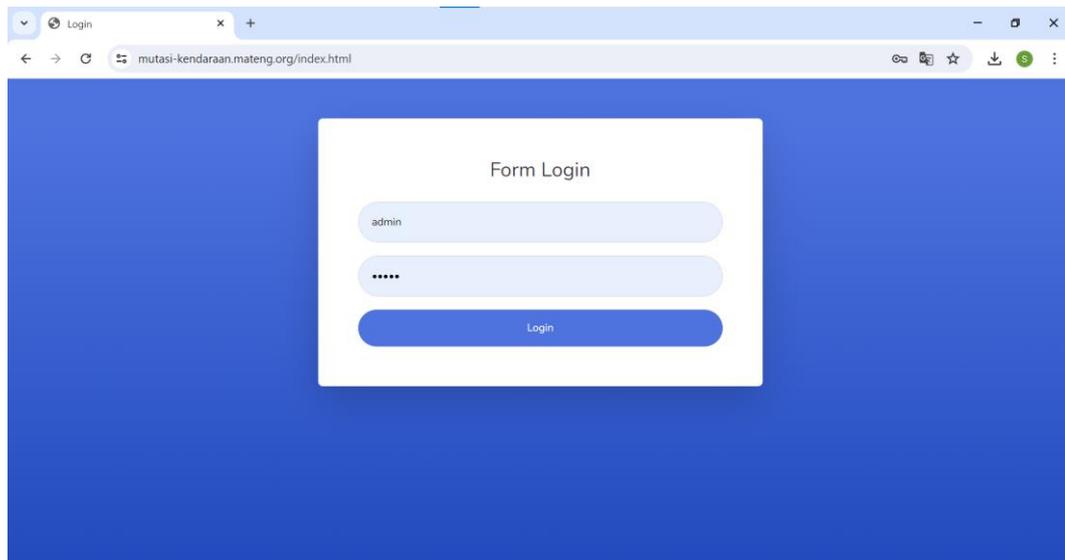


Gambar 4.13 *Sequence Diagram Data Mutasi*

C. Detail Aplikasi

1. Admin

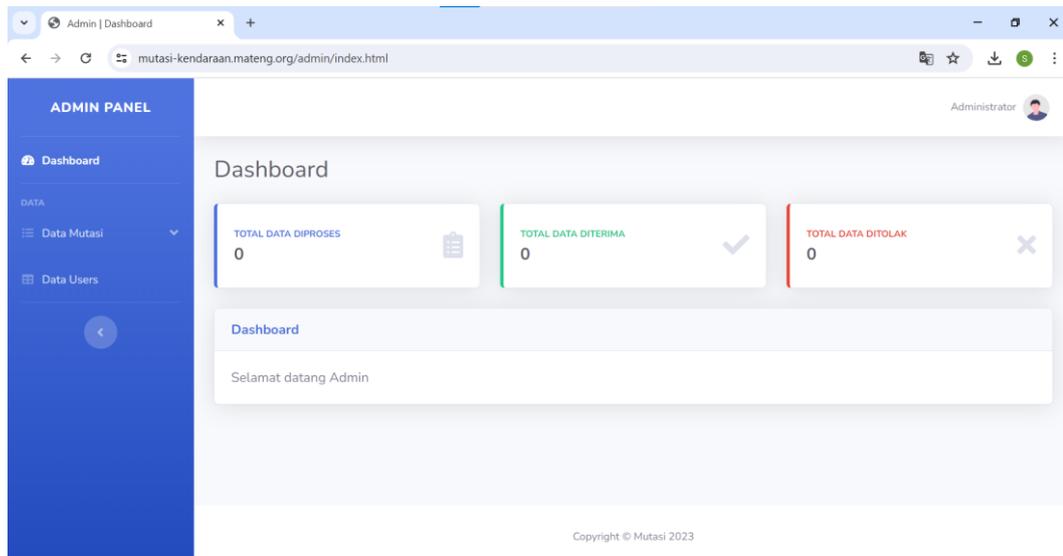
a. Halaman *Login Admin*



Gambar 4.14 Tampilan Halaman *Login Admin*

Pada menu halaman *form login* terdapat pengaksesan untuk pengguna yakni *admin*. Pada halaman ini merupakan menu *form* yang berada di halaman awal. *Login* merupakan tampilan awal yang ada pada sistem, yang digunakan pengguna untuk masuk ke halaman selanjutnya. *Form Login* digunakan untuk melindungi *system* dari hal-hal yang dapat merusak data dan sebagai *security* sistem.

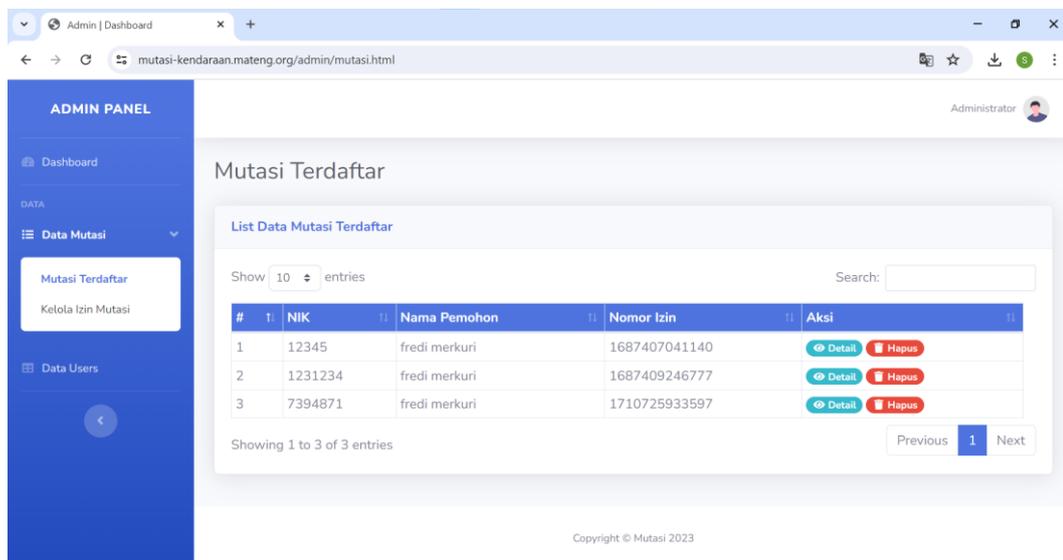
b. Halaman Utama *Admin*



Gambar 4.15 Halaman Utama *Admin*

Halaman utama merupakan halaman *home admin*, halaman ini digunakan *admin* untuk mengelola data-data mutasi. Pada halaman ini juga menampilkan jumlah data izin diproses, data izin diterima dan data ditolak.

c. Halaman *Form* Data Mutasi

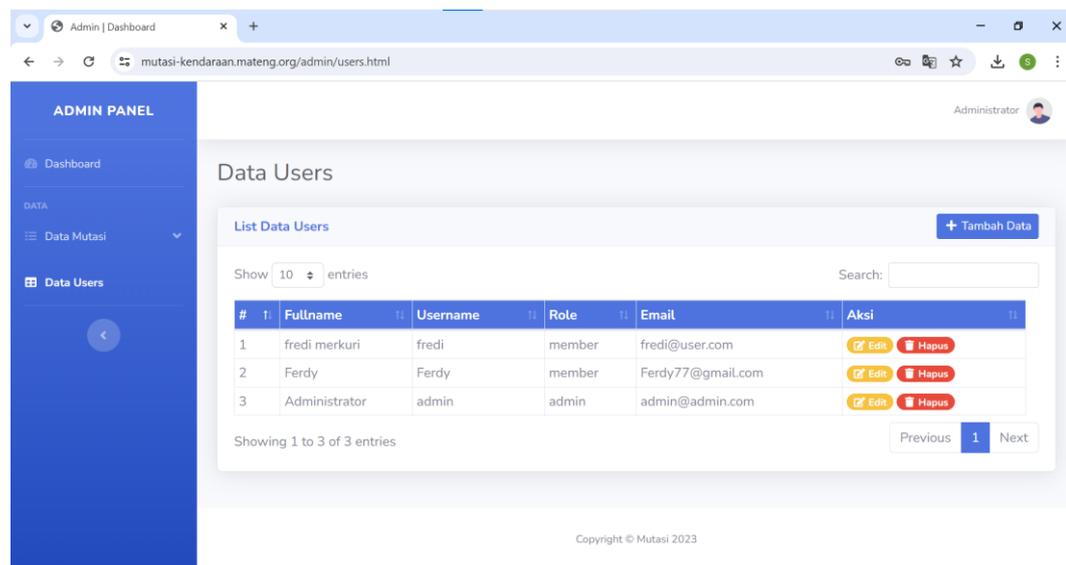


Gambar 4.16 Halaman *Form* Data Mutasi

Pada menu halaman *form* data mutase terdapat pengaksesan untuk *admin*. Halaman ini menampilkan daftar mutasi yang berhasil dimasukkan ke dalam

aplikasi. Pada halaman ini juga sebagai manajemen data mutasi serta terdapat fitur detail dan hapus laporan data mutasi.

d. Halaman *Form Data User*



Gambar 4.17 Halaman *Form Data User*

Pada menu halaman *form* data usulan terdapat pengaksesan untuk *admin*. Halaman ini menampilkan daftar *user* yang telah berhasil dimasukkan ke dalam aplikasi. Pada halaman ini juga sebagai manajemen data *user* serta terdapat fitur *edit* dan hapus data *user*.

e. Halaman *Form Mutasi Terdaftar*

ADMIN PANEL

Dashboard

DATA

Data Mutasi

Mutasi Terdaftar

Kelola Izin Mutasi

Data Users

Mutasi Terdaftar

List Data Mutasi Terdaftar

Show 10 entries

Search:

#	NIK	Nama Pemohon	Nomor Izin	Aksi
1	12345	fredi merkuri	1687407041140	Detail Hapus
2	1231234	fredi merkuri	1687409246777	Detail Hapus
3	7394871	fredi merkuri	1710725933597	Detail Hapus

Showing 1 to 3 of 3 entries

Previous 1 Next

Copyright © Mutasi 2023

Gambar 4.18 Halaman *Form* Mutasi Terdaftar

Pada menu halaman *form* mutasi terdaftar terdapat pengaksesan untuk *admin*. Halaman ini menampilkan daftar mutasi yang berhasil terdaftar ke dalam aplikasi. Pada halaman ini juga sebagai manajemen data mutasi serta terdapat fitur lihat detail dan hapus data mutasi.

f. Halaman *Form* Kelola Izin Mutasi

ADMIN PANEL

Dashboard

DATA

Data Mutasi

Mutasi Terdaftar

Kelola Izin Mutasi

Data Users

Kelola Izin Mutasi

List Data Diproses

Cari: Masukkan Nomor Mutasi

Result:

Data Tidak Ditemukan

#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
1		fredi merkuri	1713840836052	pending	Detail Verifikasi Tolak
2		fredi merkuri	1713840917198	pending	Detail Verifikasi Tolak
3		fredi merkuri	1716482536746	pending	Detail Verifikasi Tolak
4		fredi merkuri	1716485677519	pending	Detail Verifikasi Tolak

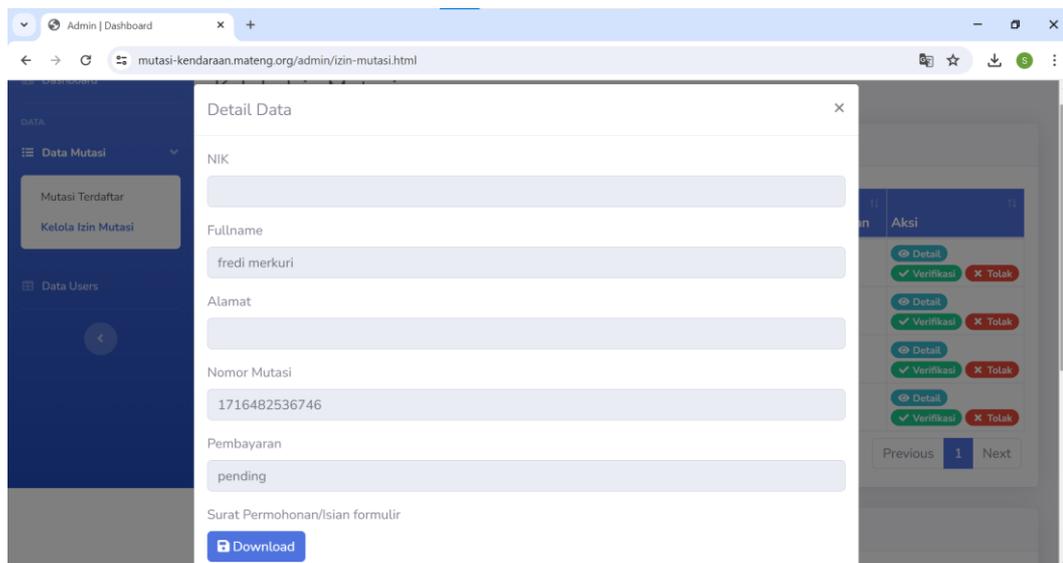
Showing 1 to 4 of 4 entries

Previous 1 Next

Gambar 4.19 Halaman *Form* Kelola Izin Mutasi

Pada menu halaman *form* data mutasi terdapat pengaksesan untuk *admin*. Halaman ini menampilkan daftar mutasi yang berhasil registrasi ke dalam aplikasi. Pada halaman ini juga sebagai manajemen data mutasi serta terdapat fitur lihat detail, verifikasi dan tolak data mutasi.

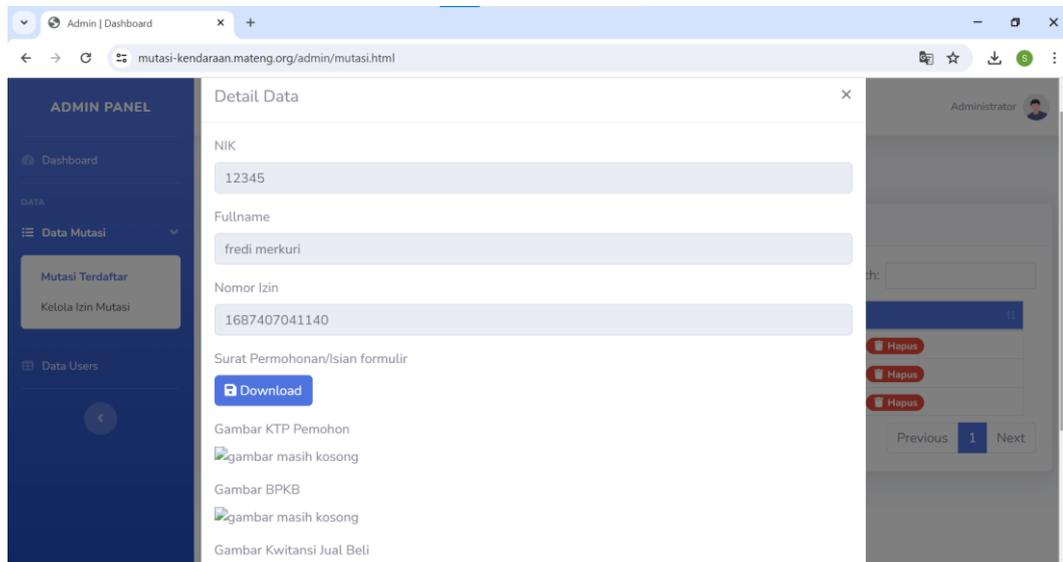
g. Halaman Menu Lihat Detail Data Mutasi



Gambar 4.20 Halaman Menu Lihat Detail Data Mutasi

Pada menu halaman lihat detail data mutasi terdapat pengaksesan untuk *admin*. Halaman ini menampilkan detail informasi daftar pengusulan data mutasi serta terdapat fitur *download* laporan data mutasi.

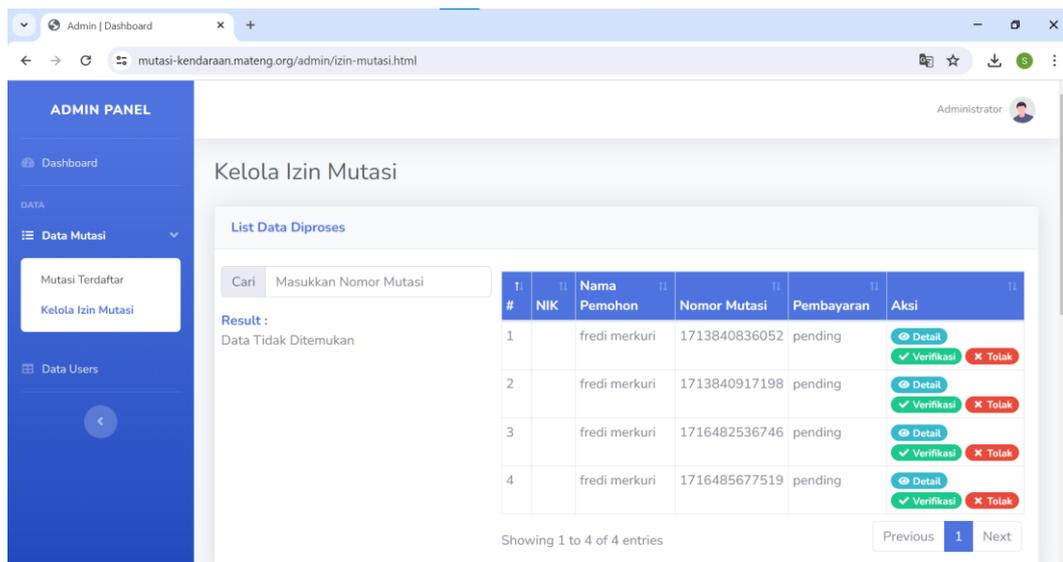
h. Halaman Menu Lihat Detail Mutasi Terdaftar



Gambar 4.21 Halaman Menu Lihat Detail Mutasi Terdaftar

Pada menu halaman lihat detail mutasi terdaftar terdapat pengaksesan untuk *admin*. Halaman ini menampilkan detail informasi daftar mutasi terdaftar serta terdapat fitur *download* laporan izin mutasi.

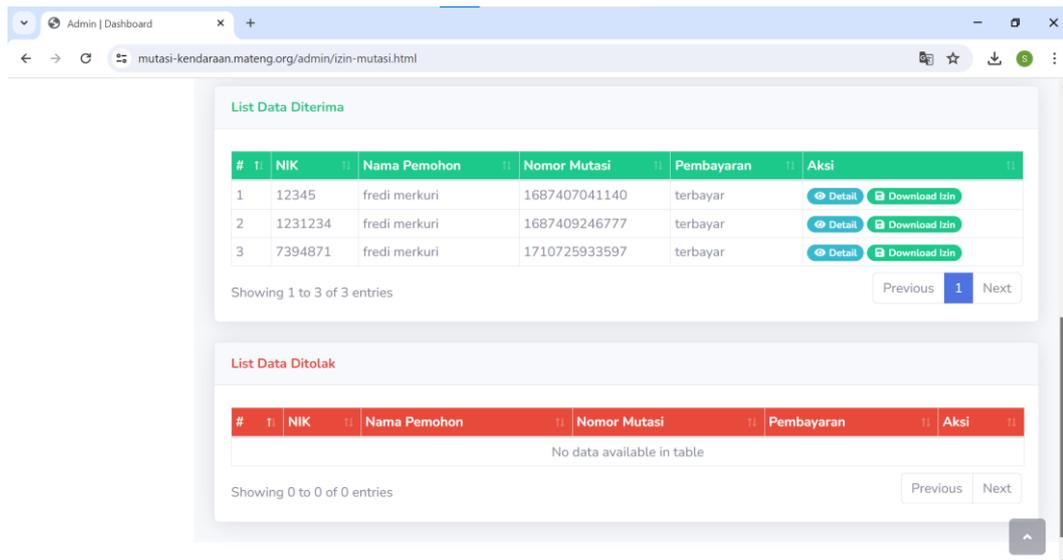
i. Halaman Menu Lihat Data Usulan



Gambar 4.22 Halaman Menu Lihat Data Usulan

Pada menu halaman lihat detail data usulan mutasi terdapat pengaksesan untuk *admin*. Halaman ini menampilkan detail informasi data usulan mutasi serta terdapat fitur *download* laporan data usulan.

j. Halaman Menu Lihat Usulan Diterima



The screenshot shows a web browser window with the URL `mutasi-kendaraan.mateng.org/admin/izin-mutasi.html`. The page title is "List Data Diterima". It contains a table with the following data:

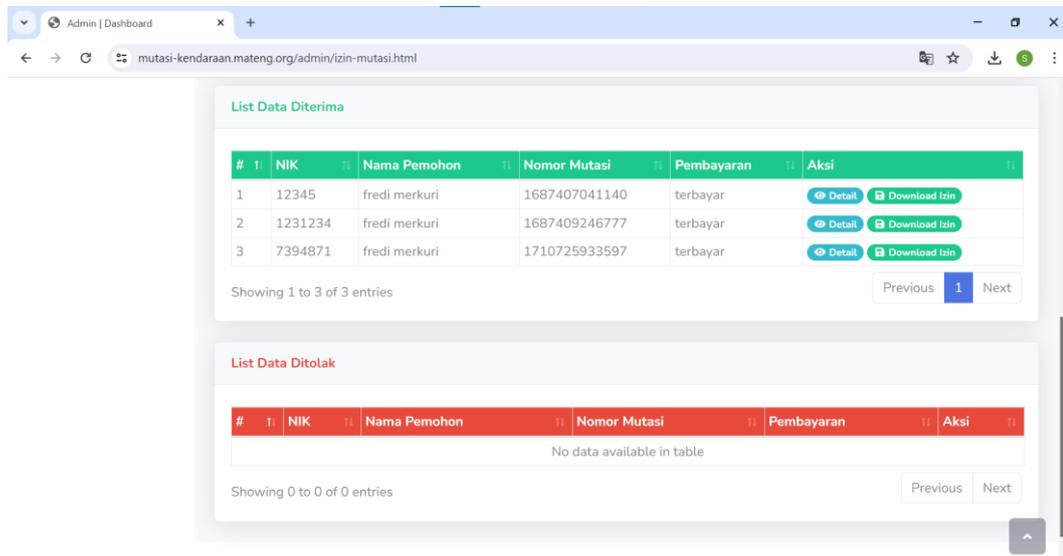
#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
1	12345	fredi merkuri	1687407041140	terbayar	Detail Download Izin
2	1231234	fredi merkuri	1687409246777	terbayar	Detail Download Izin
3	7394871	fredi merkuri	1710725933597	terbayar	Detail Download Izin

Below the table, it says "Showing 1 to 3 of 3 entries" with "Previous" and "Next" buttons. Below that is a section titled "List Data Ditolak" which is currently empty, with the message "No data available in table" and "Showing 0 to 0 of 0 entries" with "Previous" and "Next" buttons.

Gambar 4.23 Halaman Menu Lihat Usulan Diterima

Pada menu halaman lihat detail data usulan diterima terdapat pengaksesan untuk *admin*. Halaman ini menampilkan detail informasi data usulan mutasi diterima serta terdapat fitur *download* laporan data.

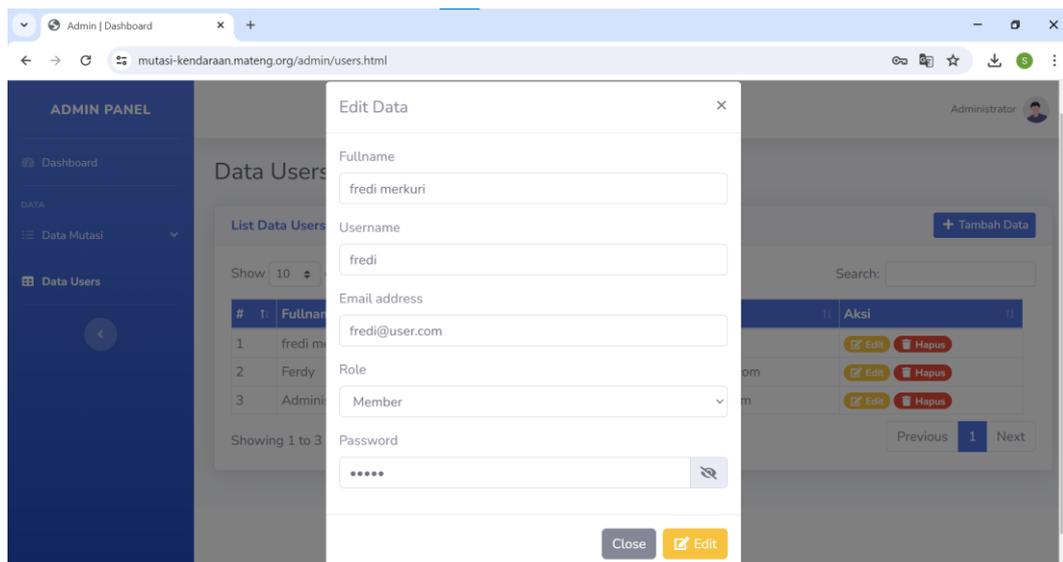
k. Halaman Menu Lihat Usulan Ditolak



Gambar 4.24 Halaman Menu Lihat Usulan Ditolak

Pada menu halaman lihat detail data usulan ditolak terdapat pengaksesan untuk *admin*. Halaman ini menampilkan detail informasi data-data usulan mutasi ditolak serta terdapat fitur lihat detail informasi.

l. Halaman Menu *Edit* Data Akun Pengguna



Gambar 4.25 Halaman Menu *Edit* Data Akun Pengguna

2. User

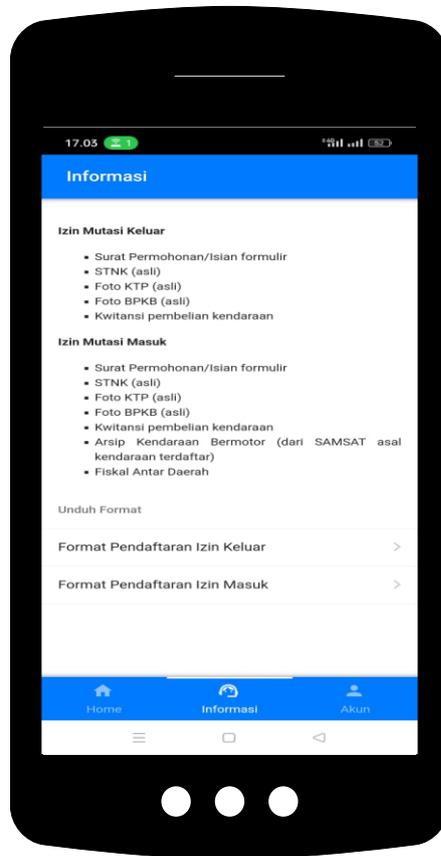
a. Halaman *Home*



Gambar 4.26 Halaman *Home*

Pada halaman *home* terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman awal. Halaman ini menampilkan gambaran umum dinas terkait.

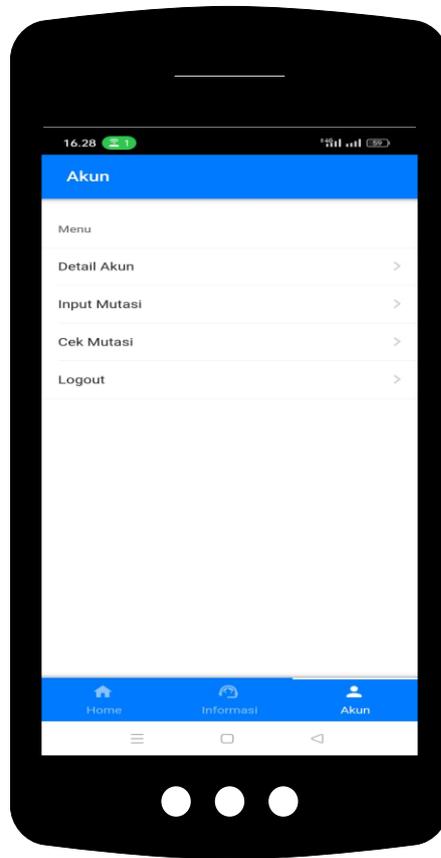
b. Halaman *Form* Informasi



Gambar 4.27 Halaman *Form* Informasi

Pada halaman *form* informasi terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman awal. Halaman ini menampilkan informasi persyaratan mutasi keluar daerah.

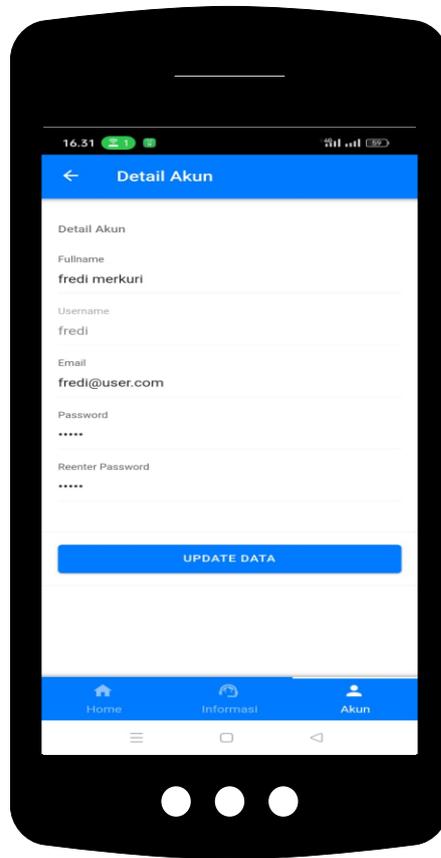
c. Halaman *Form* Akun



Gambar 4.28 Halaman *Form* Akun

Pada halaman *form* akun terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman awal. Halaman ini sebagai manajemen data akun pengguna serta terdapat menu detail akun, input mutasi, cek mutasi dan *logout*.

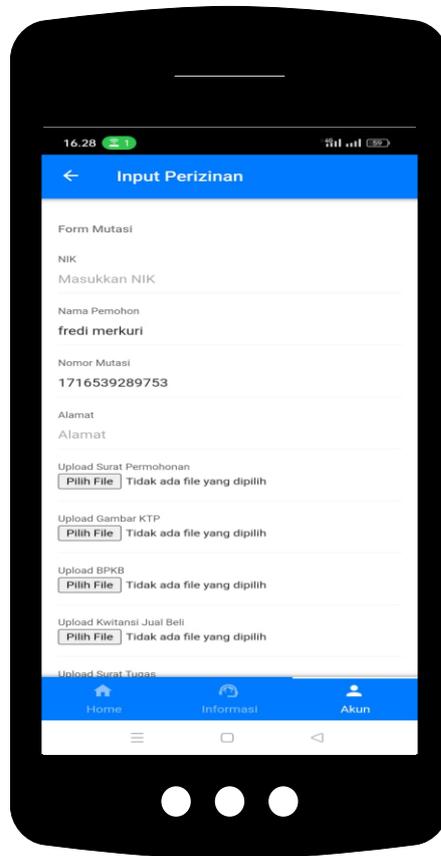
d. Halaman Menu Detail Akun



Gambar 4.29 Halaman Menu Detail Akun

Pada menu halaman detail akun terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman akun. Halaman ini sebagai manajemen data akun pengguna serta terdapat menu *update* data akun pengguna.

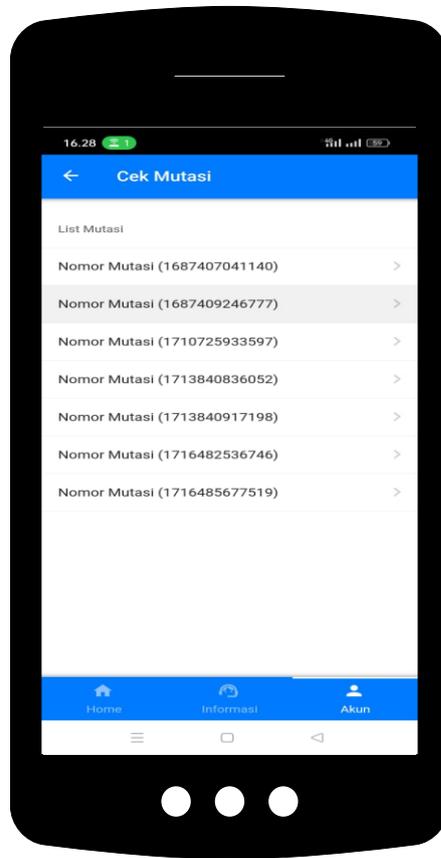
e. Halaman Menu *Input Mutasi*



Gambar 4.30 Halaman Menu *Input Mutasi*

Pada menu halaman *input* mutasi terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman akun. Halaman ini sebagai menu inputan data mutasi dan perpanjangan mutasi serta terdapat menu *upload* data Permohonan, KTP, BPKB, Kwitansi Jual Beli dan Surat Tugas.

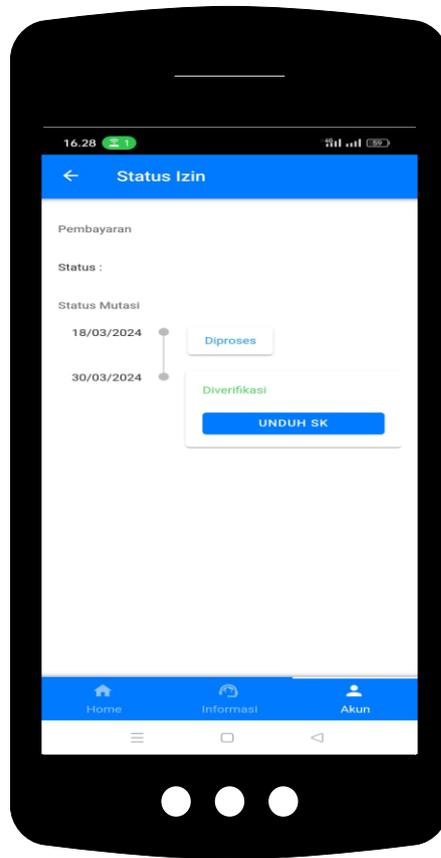
f. Halaman Menu Cek Mutasi



Gambar 4.31 Halaman Menu Cek Mutasi

Pada menu halaman cek izin terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman akun. Halaman ini menampilkan data mutasi yang telah berhasil dimasukkan serta terdapat menu *visual progress* data permohonan mutasi.

g. Halaman Menu Status Mutasi



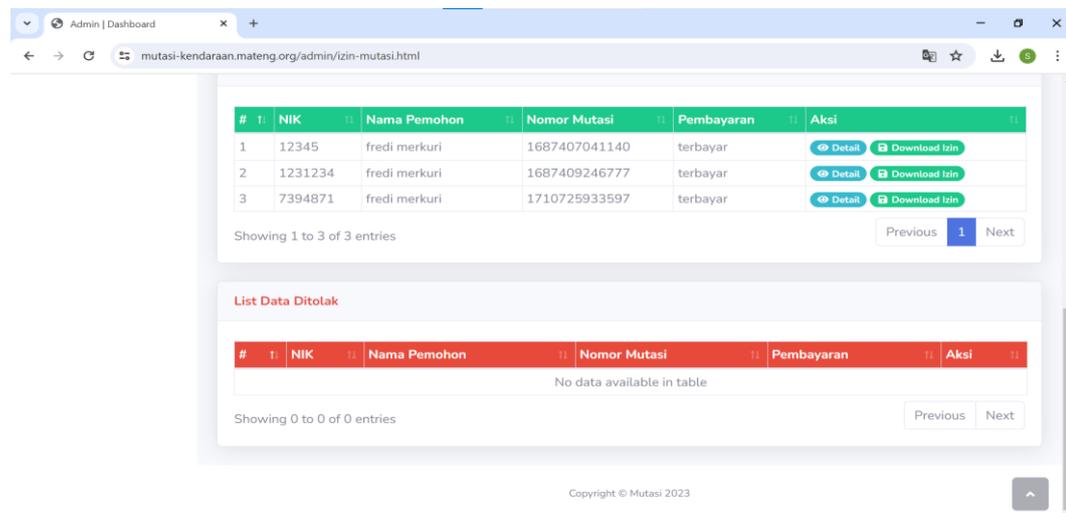
Gambar 4.32 Halaman Menu Status Mutasi

Pada menu halaman status perizinan terdapat pengaksesan untuk pengguna yakni *user*. Pada halaman ini merupakan menu *form* yang ada di halaman akun. Halaman ini menampilkan data *visual progress* data mutasi yang telah berhasil diverifikasi.

D. Rancangan *Output* Program

Perancangan *output* program merupakan perancangan yang dilakukan untuk mengetahui informasi berupa Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*.

a. Data Permohonan Mutasi



#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
1	12345	fredi merkuri	1687407041140	terbayar	Detail Download Izin
2	1231234	fredi merkuri	1687409246777	terbayar	Detail Download Izin
3	7394871	fredi merkuri	1710725933597	terbayar	Detail Download Izin

Showing 1 to 3 of 3 entries

Previous 1 Next

List Data Ditolak

#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
No data available in table					

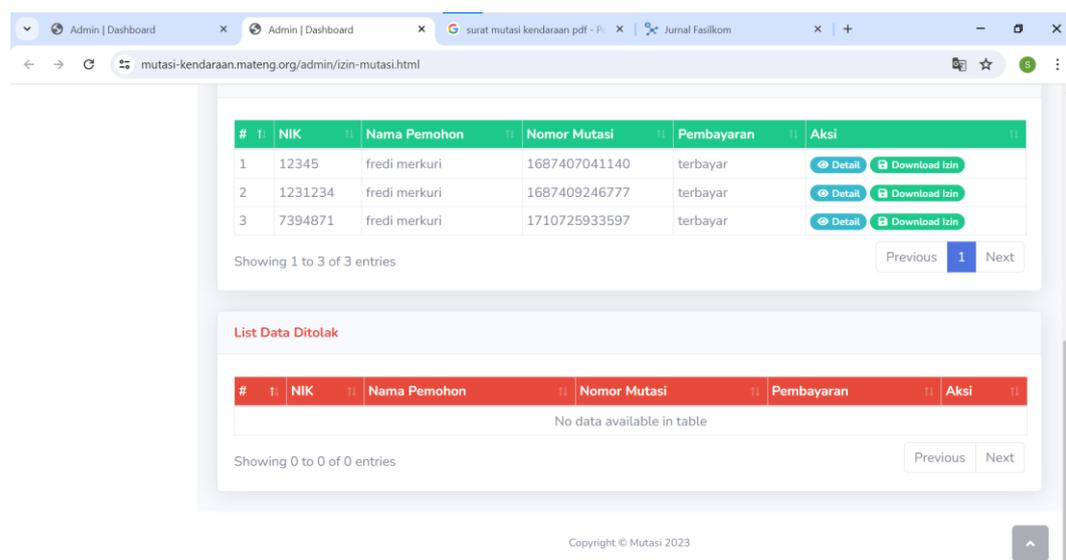
Showing 0 to 0 of 0 entries

Previous Next

Copyright © Mutasi 2023

Gambar 4.33 Halaman Data Mutasi

b. Laporan Data Mutasi



#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
1	12345	fredi merkuri	1687407041140	terbayar	Detail Download Izin
2	1231234	fredi merkuri	1687409246777	terbayar	Detail Download Izin
3	7394871	fredi merkuri	1710725933597	terbayar	Detail Download Izin

Showing 1 to 3 of 3 entries

Previous 1 Next

List Data Ditolak

#	NIK	Nama Pemohon	Nomor Mutasi	Pembayaran	Aksi
No data available in table					

Showing 0 to 0 of 0 entries

Previous Next

Copyright © Mutasi 2023

Gambar 4.34 Halaman Data Resi Mutasi

E. Implementasi

Adapun kebutuhan yang digunakan untuk membangun sebuah sistem dengan judul suatu “Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*”.

1. Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi *hardware* pengguna aplikasi ini adalah sebagai berikut :

Tabel 4.1 Spesifikasi *Hardware* Komputer

Jenis	Spesifikasi
<i>Laptop</i>	<i>Lenovo</i>
<i>Processor</i>	<i>Prosesor Intel(R) Core (TM) i5-3210M</i>
<i>RAM</i>	4 GB DDR 4
<i>Hardisk</i>	500 GB

2. Kebutuhan Perangkat Lunak (*Software*)

Spesifikasi *software* pengguna aplikasi ini sebagai berikut :

Tabel 4.2 Spesifikasi *Software* Komputer

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 (64 Bit)</i>
Bahasa Pemrograman	<i>Flutter, Javascript</i>
Database	<i>MySQL</i>
<i>Tools</i>	<i>Visual Studio Code</i>
<i>Interface</i>	<i>Microsoft Visio</i>

F. Metode pengujian

Pengujian sistem merupakan proses eksekusi suatu program atau sistem secara keseluruhan. Pengujian menunjukkan bahwa fungsi dari perangkat lunak bekerja sesuai dengan spesifikasi dan bahwa persyaratan kinerja telah dipenuhi.

Metode pengujian yang digunakan ada *black-box*. Pengujian *blackbox* perangkat lunak dilakukan untuk memperlihatkan bahwa masing-masing fungsi telah beroperasi dan secara bersamaan pula mencari kesalahan pada setiap fungsi.

G. Teknik Pengujian

Pengujian sistem dilakukan untuk memeriksa kekompakan antara komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengetesan sistem merupakan pengetesan program secara keseluruhan, adapun tahap yang dilakukan sebagai berikut :

1. Mekanisme pengujian :

Pengetesan program suatu aplikasi dengan cara menjalankan aplikasi, pengetesan pemasukan, perubahan dan penghapusan serta pengetesan terhadap fungsi tombol yang terdapat pada halaman apakah berfungsi dengan baik.

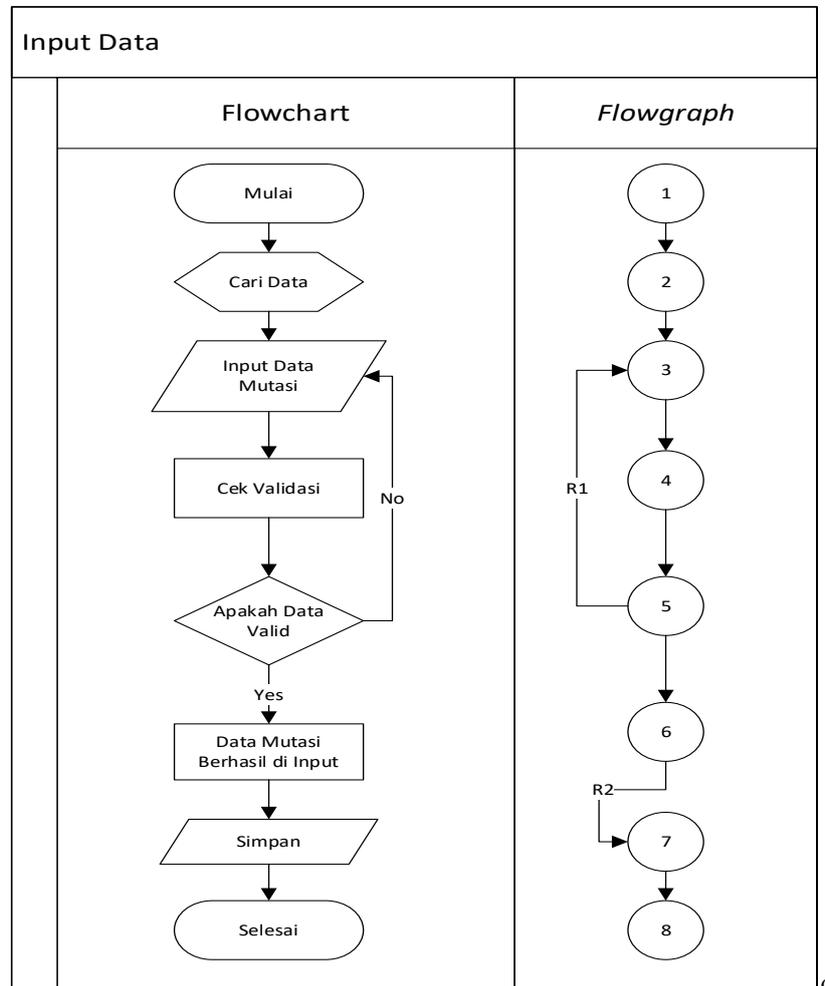
2. Hasil yang diperoleh :

Penelitian dengan judul Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile* sesuai usulan dapat berjalan dengan lancar sesuai dengan yang diharapkan.

3. Pengujian *White Box*

a. *Input Data*

Tabel 4.3 Pengujian *Input Data* Mutasi



Dari gambar *flowgraph* *Input* data mutasi diatas dapat dilakukan proses perhitungan sebagai berikut :

- 1) Menghitung *Cyclomatic Complexity* $V(G)$ dari Edge dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

Dengan E (edge) = 7

N (node) = 7

Predikat *Node* (P) = 1

Penyelesaian : $V(G) = E - N + 2$

$$= 7 - 7 + 2 = 2$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 1 + 1 = 2 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki

$$\text{Region} = 2$$

3) Independent *Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

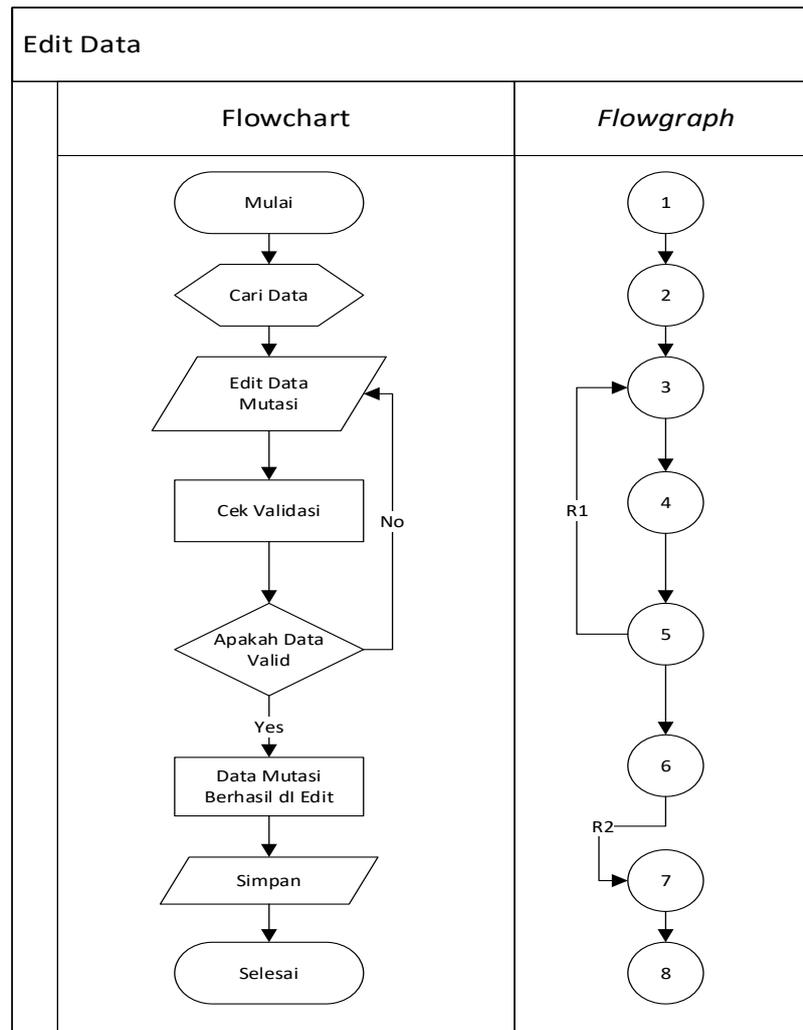
4) Grafik *Matriks*

Tabel 4.4 Grafik *Matriks* pada *Flowgraph Input Data Mutasi*

	1	2	3	4	5	6	7	$E - 1 = 0$
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4		1			1			$2 - 1 = 1$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
Sum (E) + 1								$1 + 1 = 2$

b. *Edit Data*

Tabel 4.5 Pengujian *Edit Data Mutasi*



Dari gambar *flowgraph* edit data mutasi diatas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity* $V(G)$ dari Edge dan Node:

Dengan Rumus : $V(G) = E - N + 2$

Dengan E (edge) = 8

N (node) = 8

Predikat Node (P) = 1

Penyelesaian : $V(G) = E - N + 2$

$$= 8 - 8 + 2 = 2$$

Predikat (P) = P + 1

$$= 1 + 1 = 2$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki

$$Region = 2$$

3) Independent *Path* pada *flowgraph* diatas adalah :

$$Path 1 = 1 - 2 - 3 - 4 - 5 - 3$$

$$Path 2 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

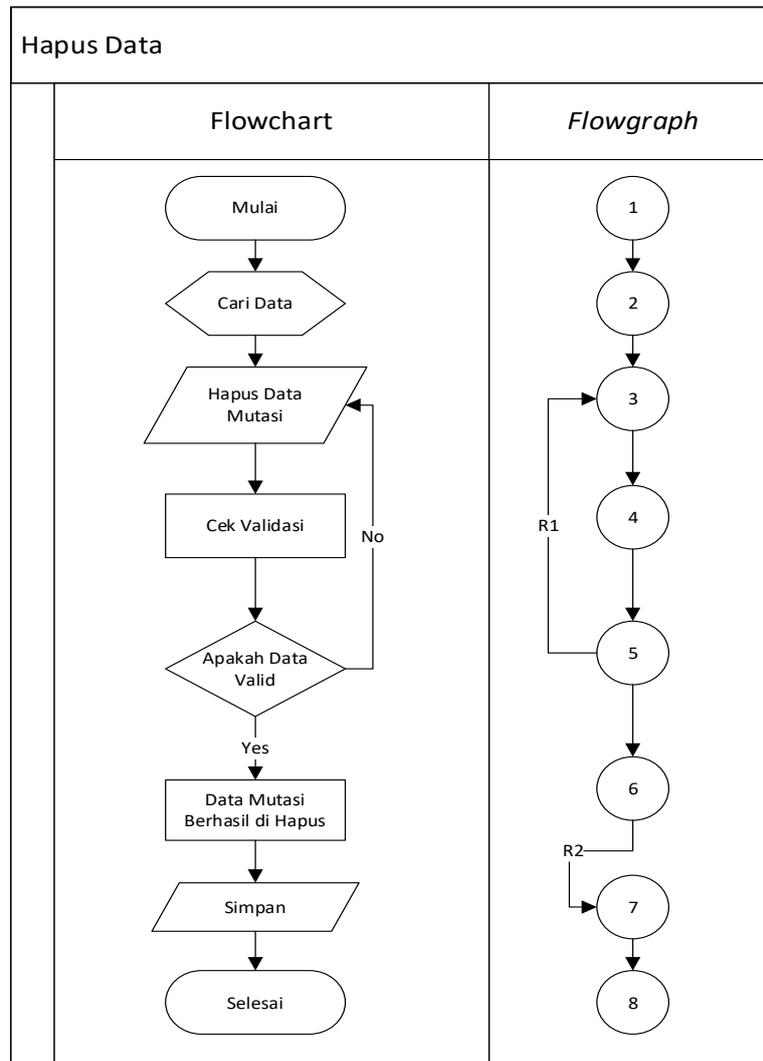
4) Grafik *Matriks*

Tabel 4.6 Grafik *Matriks* pada *Flowgraph Edit Data Mutasi*

	1	2	3	4	5	6	7	8	$E - 1 = 0$
1		1							$1 - 1 = 0$
2			1						$1 - 1 = 0$
3				1					$1 - 1 = 0$
4					1				$1 - 1 = 0$
5			1			1			$2 - 1 = 1$
6							1		$1 - 1 = 0$
7								1	$1 - 1 = 0$
8									0
Sum (E) + 1									$1 + 1 = 2$

c. Hapus Data

Tabel 4.7 Pengujian Hapus Data Mutasi



Dari gambar *flowgraph* hapus data mutasi diatas dapat dilakukan proses perhitungan sebagai berikut :

- 1) Menghitung *Cyclomatic Complexity* $V(G)$ dari Edge dan Node:

Dengan Rumus : $V(G) = E - N + 2$

Dengan E (edge) = 8

N (node) = 8

Predikat Node (P) = 1

Penyelesaian : $V(G) = E - N + 2$

$$= 8 - 8 + 2 = 2$$

Predikat (P) = P + 1

$$= 1 + 1 = 2$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki

$$Region = 2$$

3) Independent *Path* pada *flowgraph* diatas adalah :

$$Path 1 = 1 - 2 - 3 - 4 - 5 - 3$$

$$Path 2 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

4) Grafik *Matriks*

Tabel 4.8 Grafik *Matriks* pada *Flowgraph* Hapus Data Mutasi

	1	2	3	4	5	6	7	8	$E - 1 = 0$
1		1							$1 - 1 = 0$
2			1						$1 - 1 = 0$
3				1					$1 - 1 = 0$
4					1				$1 - 1 = 0$
5			1			1			$2 - 1 = 1$
6							1		$1 - 1 = 0$
7								1	$1 - 1 = 0$
8									0
Sum (E) + 1									$1 + 1 = 2$

4. Pengujian *Black Box*

Pada *fase* ini dilakukan pengujian terhadap sistem dengan menggunakan kotak hitam (*black-box testing*). Pengujian sistem ini bertujuan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari sistem telah sesuai dengan spesifikasi yang dibutuhkan. Cara pengujian *black-box testing* dilakukan dengan menjalankan sistem dan melakukan *input* data

serta melihat inputnya apakah sudah sesuai yang diharapkan. Berdasarkan rencana hasil pengujian, maka dapat dilakukan pengujian sebagai berikut.

a. Skenario *Admin*

Tabel 4.9 Hasil Pengujian *BlackBox Form Admin*

Kasus dan Hasil Uji		
Aksi/ data masukan	Yang diharapkan	Kesimpulan
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>Admin</i> dapat mengakses halaman <i>home</i> dan semua fitur yang ada di halaman utama berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Data diproses	<i>Admin</i> dapat mengakses halaman <i>form</i> data permohonan diproses dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman Menu Data diterima	<i>Admin</i> dapat mengakses halaman <i>form</i> data permohonan diterima dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman Menu Data ditolak	<i>Admin</i> dapat mengakses halaman <i>form</i> data permohonan ditolak dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Akun	<i>Admin</i> dapat mengakses halaman <i>form</i> akun dan semua fitur berfungsi dengan baik	Berhasil ✓
<i>Logout</i>	<i>Admin</i> keluar dari halaman sistem	Berhasil ✓

b. Skenario *User*

Tabel 4.10 Hasil Pengujian *BlackBox Form User*

Kasus dan Hasil Uji		
Aksi/ data masukan	Yang diharapkan	Kesimpulan
<i>Activity Splash Screen</i>	<i>User</i> dapat mengakses halaman utama dan menampilkan <i>form login</i>	Berhasil ✓
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>User</i> dapat mengakses halaman utama dan menampilkan gambaran informasi dinas terkait	Berhasil ✓
Halaman <i>Form</i> Informasi	<i>User</i> dapat mengakses <i>form</i> informasi tanpa ada <i>error</i> dan menampilkan data persyaratan mutasi	Berhasil ✓

Tambah Data Mutasi	<i>User</i> dapat melakukan penambahan data mutasi dengan menekan <i>button</i> “Tambah Data”	Berhasil ✓
<i>Edit</i> Data Mutasi	<i>User</i> dapat melakukan perubahan data mutasi dengan menekan <i>button</i> “ <i>Edit</i> ”	Berhasil ✓
Hapus Data Mutasi	<i>User</i> dapat melakukan menghapus data mutasi dengan menekan <i>button</i> “Hapus Data”	Berhasil ✓
<i>Logout</i>	<i>User</i> keluar dari halaman sistem	Berhasil ✓

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkannya suatu Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile*. Berdasarkan implementasi dan hasil pengujian dengan dua teknik pengujian yaitu pengujian *Black-Box* dan pengujian *White-Box* dapat disimpulkan

bahwa aplikasi ini berjalan baik sesuai dengan fungsinya. Penulis mendapatkan data yang kemudian di analisis dan diuraikan, maka penulis mengambil kesimpulan sebagai berikut :

1. Aplikasi ini menampilkan suatu panduan lengkap mengenai prosedur dan kelengkapan dokumen yang diperlukan. Fitur-fitur aplikasi ini membantu pemohon mengelola data-data mutasi dengan mudah, termasuk *input* data kendaraan, nomor polisi serta pengajuan dokumen secara digitalisasi. Selain itu, aplikasi ini memiliki suatu fitur pengecekan status pengajuan dan fitur pembayaran digital juga disediakan, memungkinkan pemohon membayar biaya mutasi melalui *QRIS* sehingga mempermudah proses administrasi tanpa perlu datang langsung ke kantor Samsat untuk sebagian besar prosedur.
2. Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile* telah berhasil dibuat dengan menggunakan Bahasa pemrograman *Flutter*, *Javascript* dan *Visual Studio Code* untuk penulisan *script* serta *MySQL* untuk pengelolaan basis data.

B. Saran-saran

Aplikasi *e-Payment* Mutasi Kendaraan Keluar Daerah Pada Samsat Kota Parepare Berbasis *Mobile* masih terdapat beberapa kekurangan. Untuk itu perlu dilakukan pengembangan dan penyempurnaan agar dapat lebih baik. Apabila ada yang berniat untuk mengembangkan sistem ini, Adapun saran agar aplikasi ini bisa berjalan dengan lebih optimal dan lebih menarik sebagai berikut :

1. Aplikasi *e-Payment* mutasi kendaraan ini hanya dapat berjalan pada satu *platform* yaitu *Android*. Kelemahan ini akan menjadi acuan untuk dapat dikembangkan lagi agar dapat digunakan di beberapa *platform*.
2. Hendaknya sistem yang penulis rancang dapat dikembangkan sehingga dalam pengolahan datanya dapat lebih dijelaskan secara mendetail, agar pengolahan data dan pembuatan laporan dapat dilakukan lebih efektif dan efisien.

3. Sistem aplikasi yang telah dikembangkan penulis sebaiknya dipelihara dengan sebaik-baiknya dan di perbaharui sesuai dengan kebutuhan yang ada serta melakukan *backup* data pada jangka waktu tertentu untuk menghindari kemungkinan data hilang atau rusak.

Demikian saran yang dapat penulis berikan, semoga saran tersebut bisa dijadikan sebagai bahan masukan yang dapat bermanfaat bagi penulis khususnya dan bagi pengembang pada umumnya.

DAFTAR PUSTAKA

- Ade Hastuty Hasyim. 2021. *Dasar Pemrograman* (Cetakan ke-2). Makassar: CV. Bangun Bumitama.
- Ali Ikhwan, (2022). “*Implementasi Metode Geofence Pada Aplikasi Reminder Berbasis Android (Studi Kasus Jamaah Majelis Taklim Al-Markas)*”.
- Anike, M., & Melo, F. C. (2019). *Application Emergency Panic Button (AEPB) Berbasis Android (Studi Kasus : Rumah Sakit ST. Carolus Boromeus Bello)*. Sistem Informasi, Volume 8, Nomor 3, 366 – 376.
- A.S Rosa, dan M. Shalahuddin. 2018. *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek*. Bandung : Informatika.
- Adi Nugroho, 2019. *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.
- Darius Antoni, (2021). “*Sistem Presensi Karyawan Menggunakan Metode Geofence dan Face Capture Push Notification*”.
- Garude & Haldikar, N. (2019). *Real Time Position Tracking System Using Google Maps API V3*. *International Journal of Scientific and Research Publications*, 1–4.
- Hasan, 2020. “*Efektivitas Pelayanan Samsat dalam Pengelolaan Pajak Kendaraan Bermotor.*” *Jurnal Administrasi Publik*. 5 (3): 45-60.

- Hartono, (2020). *Konsep dan Aplikasi Structural Equation Modeling Berbasis Varian dalam Penelitian Bisnis*. (Ed. 2). UPP STIM YKPN, Yogyakarta.
- Hermawan S, Stephanus. 2019. "*Mudah Membuat Aplikasi Android*". Yogyakarta : Andi Offset.
- Jogiyanto, 2015. *Analisis & Desain Sistem Informasi : Pendekatan terstruktur teori dan praktik aplikasi bisnis*. Andi Offset. Yogyakarta.
- KBBI, 2015. *Kamus Besar Bahasa Indonesia (KBBI)*. [Online] Available at: <http://kbbi.web.id/di> [Diakses 10 Agustus 2022].
- Lardinois, F. (2021). *Microsoft Launches Visual Studio Code, a Free CrossPlatform Code Editor for OS X, Linux and Windows*.
- Munawar. (2015), *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta.
- Nugroho, (2019). "*Implementasi Geofencing Pada Layanan Pemantau Anak Berbasis Lokasi*". Seminar Nasional IPTEK Terapan (SENIT), vol. II, no. 1, pp. 63-66, 2019.
- Putra, Budi. 2020. *Revolusi Digital: QRIS dan Inovasi Pembayaran di Indonesia*. Jakarta: Elex Media Komputindo.
- Patton, 2021. "*Black Box Testing: Techniques for Functional Testing of Software and Systems*." *Journal of Software Testing and Verification*. 15(4): 289-292.
- Ramzi, (2017). *Rancang Bangun Aplikasi Penjadwalan Mata Pelajaran Berbasis Web Pada SMK Negeri 1 Cerme*. Undergraduate thesis, STIKOM. Surabaya.
- Rosa, Shalahuddin. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Penerbit Modula, 2018.
- Sanjaya (2015). *Media Komunikasi Pembelajaran*. Jakarta : Kencana Prenada Media Group.
- Safaat, 2020, *Rancang Bangun Aplikasi Multiplatform*, Informatika, Bandung.
- Sunyoto. 2019. *Integrasi modul GPS Receiver dan GPRS untuk penentuan posisi dan jalur pergerakan obyek*. Yogyakarta: Stimik Amikom.
- Software Testing Help. 2023. *White Box Testing: What Means, Why & How to Do White-Box and Black-Box Testing*. Diakses tanggal 10 Maret 2024, dari <https://www.codester.com/>.
- Smyth, N. (2015). *Android Studio Development Essential (Vol. II)*. e-Book Frenzy.
- Setiabudi, Rostianingsih, (2015). *Pemanfaatan Google Maps API pada Sistem Informasi Geografis Direktori Perguruan Tinggi Kota Bengkulu*. Meda Infotama.
- Sukamto, & Shalahuddin. (2019). *Analisa dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.

Sharma, (2020). *Software Engineering*. New Delhi: APH Publising Corporation.

Supardi. (2017) *Aplikasi Statistika dalam Penelitian Konsep Statistika yang Lebih Komprehensif*. Jakarta: Change Publication.

Sholiq. 2016. *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*. Graha Ilmu. Yogyakarta.

Suharsimi, Arikunto. 2021. *Prosedur Penelitian: Suatu Pendekatan Praktik*. Jakarta : Rineka Cipta.

Tia Monica Andriani, (2020). “*Perancangan Aplikasi Pemantauan Browser Anak Melalui SMS*”.

Universitas Muhammadiyah Parepare. (2021). *Panduan Akademik 2020-2021*, Parepare : UMPAR Press.