

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi Informasi (TI) adalah kebutuhan mendasar untuk mendukung kinerja suatu institusi atau organisasi, dengan adanya teknologi informasi tingkat efisiensi dan efektivitas dari proses bisnis maupun pekerjaan akan meningkat. Salah satu contoh dari berkembangnya TI adalah aplikasi mobile yang memiliki tingkat mobilitas yang tinggi. Pengguna dapat dengan mudah untuk melakukan suatu transaksi maupun memperoleh informasi.

Pada dasarnya online service seperti aplikasi mobile digunakan untuk memudahkan proses pelayanan. Tetapi tidak semua pengguna yang mengoperasikan suatu aplikasi mobile memperoleh kemudahan. Hal yang memengaruhi dalam aplikasi mobile adalah aspek user friendly. Pengguna akan merasakan kemudahan dan kepuasan ketika aspek tersebut ada dalam sebuah layanan. Permasalahan yang sering terjadi dirasakan oleh pengguna adalah minimnya informasi yang diberikan aplikasi, kesulitan dalam menggunakan aplikasi dan sebagainya. Faktor yang menyebabkan terjadinya hal tersebut adalah navigasi yang kurang jelas, fitur yang merepotkan bahkan fitur yang dibutuhkan tidak tersedia pada aplikasi tersebut.

Usability merupakan aspek penting dalam pembuatan aplikasi. Secara umum, *usability* adalah sejauh mana sebuah produk atau layanan dapat digunakan pengguna untuk mencapai tujuannya dan seberapa mudah penggunaan dari antarmuka aplikasi tersebut yang dapat menghasilkan nilai kepuasan pengguna dalam menggunakan produk tersebut. (Nielsen, J., 2012). Salah satu cara agar aplikasi bertahan adalah memperhatikan aspek *usability*. Tingkat *usability* dapat menjadi cerminan apakah aplikasi tersebut dapat bermanfaat bagi pengguna. Semakin baik tingkat *usability* maka aplikasi semakin dapat bertahan dan dapat memiliki banyak pengguna..

PT. Kereta Api Indonesia merupakan salah satu badan usaha milik Negara penyedia jasa transportasi khususnya di sulawesi selatan jalur kereta api Maros-Barru yang memiliki layanan Pengingat jadwal kereta api dengan menggunakan *mobile* aplikasi. Dengan adanya layanan aplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia, diharapkan pengguna dapat memperoleh kemudahan dalam melihat informasi serta melakukan reservasi atau pemesanan tiket. Transportasi kereta api ini menjadi faktor penting dalam bidang sosial dan ekonomi yaitu untuk memperlancar roda perekonomian di Indonesia.

Kereta api merupakan salah satu moda transportasi umum yang banyak digunakan masyarakat. Namun sering kali penumpang mengalami kesulitan dalam mengingat jadwal keberangkatan kereta api, terutama bagi mereka yang memiliki jadwal perjalanan yang padat dan kompleks. Aplikasi pengingat jadwal berbasis multimedia dapat menjadi solusi untuk membantu penumpang mengingat dan mengatur jadwal keberangkatan kereta api mereka dengan lebih efektif.

B. Rumusan Masalah

Berdasarkan latar belakang di atas, Maka dapat diambil suatu rumusan masalah yaitu :

1. Bagaimana Membuat Aplikasi Pengingat Jadwal Keberangkatan Kereta Api Berbasis Multimedia?
2. Bagaimana mengembangkan aplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia yang efektif?
3. Bagaimana pengaruh pengguna aplikasi ini terhadap kemudahan dan kenyamanan pengguna dalam mengingat jadwal keberangkatan kereta api?

C. Batasan Masalah

Agar pembahasan dalam penelitian tidak melebar, maka penulis perlu memberikan batasan-batasan untuk pendekatan permasalahan agar lebih spesifik dalam pelaksanaannya yaitu sebagai berikut :

1. Pembuatan aplikasi ini hanya ditujukan untuk pengguna yang ingin menggunakan transportasi kereta api.
2. Aplikasi ini hanya membahas mengenai jadwal kereta api.
3. Aplikasi ini dibuat menggunakan bahasa pemrograman Java, dan pembuatan database menggunakan MySQL.

D. Tujuan dan Manfaat Penelitian

1. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah dipaparkan di atas, maka tujuan dari penulisan penelitian ini adalah untuk membuat Aplikasi Peningat Jadwal Keberangkatan Kereta Api Berbasis Multimedia.

2. Manfaat Penelitian

a. Bagi Pengguna

Manfaat yang didapatkan oleh pengguna yaitu dapat mempermudah membagi waktu antara menunggu waktu keberangkatan yang telah di atur dan kegiatan lain , serta mengingatkan jadwal bagi pengguna yang telah memesan tiket sehingga pengguna tidak terlambat ataupun lupa terkait jadwal keberangkatan tersebut.

b. Bagi Penulis

Adapun manfaat bagi penulis yaitu untuk menambah wawasan penulis dalam mengembangkan ilmu yang telah didapat selama perkuliahan serta mengimplementasikannya ke dalam aplikasi yang akan dibuat.

c. Bagi Universitas Muhammadiyah Parepare

Untuk digunakan sebagai bahan kajian mahasiswa Universitas Muhammadiyah Parepare dalam mengerjakan tugas akhir mendatang, khususnya mengenai pembuatan aplikasi berbasis multimedia dan juga sebagai bahan kepustakaan bagi kampus Universitas Muhammadiyah Parepare.

E. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan proposal ini adalah sebagai berikut :

BAB I PENDAHULUAN

Dalam bab ini akan diuraikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan beberapa teori-teori yang mendukung dalam pembahasan penyusunan tugas akhir ini serta bahasa pemrograman yang digunakan sehingga memudahkan penulis dalam menyelesaikan masalah.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan mengenai tahapan - tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu tempat penelitian dan waktu penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang atau dibuat.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Dalam bab ini menjelaskan mengenai rancangan sistem yang telah dibuat, metode dan teknik pengujian.

BAB V KESIMPULAN DAN SARAN

Dalam bab ini menjelaskan mengenai kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Penelitian Sebelumnya

Dalam penelitian ini, mengacu pada penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan. Berikut beberapa hasil penelitian yang relevan atau berhubungan dengan penelitian yang dijadikan bahan telaah bagi penelitian yang akan dilakukan, antara lain :

1. Penelitian yang dilakukan oleh: (Faisal & Sunardi, 2020) Jurnal Perkeretaapian Indonesia Volume IV Nomor 1 Maret 2020 yang berjudul “SIMULASI APLIKASI MONITORING KERETA API BERBASIS ANDROID”. Kereta api adalah transportasi masal yang mengutamakan ketepatan waktu serta terjadwal dalam penyelenggaraan perkeretaapian. Penelitian ini memudahkan konsumen dalam melihat realisasi perjalanan kereta api secara digital dengan menggunakan pengembang program Delphi. Input program berasal dari titik koordinat yang dideteksi oleh modul GPS dan selanjutnya dikirim pada database oleh modul *wifi Node MCU*. Database yang digunakan tipe database MySQL. Data primer yang digunakan berupa titik kordinat stasiun yang akan dilewati maupun simulasi titik kordinat yang akan dilalui oleh modul. Penelitian memakai model *Linier Sequential Model* (LSM) dengan 4 tahap yang berulang yaitu tahapan menganalisis, studi literatur, desain atau perancangan, perakitan dengan pengkodean, dan pengujian. Pengujian dilakukan dengan cara simulasi

maupun penerapan langsung pada lapangan. Pemrograman modul nodeMCU dan GPS menggunakan menggunakan bantuan arduino IDE untuk dapat menghubungkan modul dengan database. Selanjutnya database akan dihubungkan pada aplikasi penampil yang akan dibuat dengan pengembang program Delphi.

2. Penelitian oleh (Hannan, 2023) *Applied Business and Administration Journal*, Volume 2, Nomor 1 / 2023 yang berjudul “Perancangan Sistem Informasi Pemesanan Tiket Kereta Api Layanan Lokal Bandung Raya Berbasis Aplikasi KAI Access (Studi Kasus Pada PT KAI)”. Sistem informasi manajemen (SIM) merupakan implementasi penerapan teknologi informasi yang digunakan, tak tekecuali dimanfaatkan pula oleh bidang transportasi. PT KAI sebagai penyedia kereta api menggunakan sistem informasi dalam menunjang aktivitas bisnisnya salah satunya adalah aplikasi pemesanan tiket kereta melalui KAI Access. Keseluruhan sistem dari KAI Access ini sudah berjalan dengan baik, namun terdapat sedikit kekurangan mengenai output yang diharapkan dari pengguna layanan ini guna menjadikan perjalanan lebih efektif dan efisien. Kekurangan output ini berupa masih belum adanya informasi jalur kereta pada Etiket dan pemberitahuan kedatangan kereta secara realtime di Aplikasi. Maka dari itu perlu dirancang suatu sistem informasi dengan pembuatan rancangan proyek yang terstruktur dengan baik agar hasil yang didapatkan layanan pengguna Aplikasi KAI Access ini lebih maksimal. Perancangan dilakukan dengan menggambarkan terlebih dahulu proses bisnis menggunakan BPMN dan dibuat rancangan solusinya menggunakan Data Flow Diagram (DFD)

B. Sistem informasi

Sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen, atau sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen.

Menurut definisi sistem informasi sebelumnya, sistem informasi memiliki hubungan dengan sistem data saat ini di satu sisi dan sistem aktivitas di sisi lain.

Definisi Sistem Informasi Menurut para ahli :

- a. Menurut John F.Nash (1995:8) Sistem informasi adalah kombinasi dari orang-orang, fasilitas atau teknologi, media, prosedur-prosedur, dan pengendalian yang mempunyai maksud untuk menata jaringan komunikasi yang penting, proses-proses dan transaksi-transaksi tertentu secara teratur, membantu manajemen dan para pemakain internal dan eksternal, dan menyediakan suatu dasar pengambilan keputusan yang tepat.
- b. Menurut Robert A. Leitch (1999:11), sistem informasi adalah suatu sistem yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

C. Aplikasi

Secara istilah, aplikasi komputer adalah suatu sub kelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja dan pemutar media. Kumpulan aplikasi komputer yang biasanya disebut paket aplikasi atau application suite contohnya adalah Microsoft Office dan LibreOffice, yang menggabungkan aplikasi pengolah kata, lembar kerja serta beberapa aplikasi lainnya ke dalam satu paket aplikasi. Pada umumnya aplikasi yang terdapat pada application suite memiliki kemampuan untuk saling berinteraksi serta saling terintegrasi sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dimasukkan dalam satu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. Dari pemaparan di atas, dapat disimpulkan bahwa pengertian aplikasi adalah suatu perangkat lunak (software) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu (Hannan, 2023)

D. Peningat

Peningat bisa dikatakan sebagai aplikasi yang berfungsi untuk memberi tahu pada hari/waktu itu ada sebuah kegiatan atau hal yang harus dilakukan. Peningat biasanya berkaitan erat dengan alarm. Alarm pada umumnya untuk memberi peringatan kepada pengguna bahwa ada suatu kegiatan pada waktu yang telah ditentukan sebelum alarm itu berbunyi. Biasanya sebelum mengatur peningat, dirancang dulu sebuah jadwal. Perbedaan antara peningat dengan alarm terletak

pada memo. Pengingat bisa mencantumkan memo atau catatan sekaligus reminder. Sedangkan untuk alarm hanya digunakan sebagai pengingat waktu. Reminder biasanya digunakan sebagai pencatat janji, jadwal keseharian, dan tugas-tugas kuliah. Untuk alarm pada umumnya digunakan sebagai alat bantu untuk membangunkan seseorang jika ingin melakukan kegiatan pada waktu yang telah ditentukan. (Mulyani, S., 2016).

E. Jadwal

Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau sistem kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. (Faisal & Sunardi, 2020).

F. Kereta Api

Kereta api adalah sarana transportasi berupa kendaraan dengan tenaga gerak, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, yang akan ataupun sedang bergerak di rel. Kereta api merupakan alat transportasi massal yang umumnya terdiri dari lokomotif (kendaraan dengan tenaga gerak yang berjalan sendiri) dan rangkaian kereta atau gerbong (dirangkaikan dengan kendaraan lainnya). Rangkaian kereta atau gerbong tersebut berukuran relatif luas sehingga mampu memuat penumpang maupun barang dalam skala besar. Karena sifatnya sebagai angkutan massal efektif, beberapa negara berusaha memanfaatkannya secara maksimal sebagai alat transportasi utama angkutan darat baik di dalam kota, antarkota, maupun antarnegara (Hannan, 2023).

1 Klasifikasi kereta api

Dari segi propulsi atau dari tenaga penggeraknya kereta api dibagi menjadi tiga, yaitu :

- a. kereta api uap,
- b. kereta api diesel,
- c. kereta rel listrik.

Dari segi rel atau tempat kereta berjalan, kereta api dibagi menjadi dua :

- a. kereta api konvensional,
- b. kereta api monorel.

Dari segi letak, kereta api dibagi menjadi tiga, yaitu :

- a. kereta api permukaan (surface),
- b. kereta api layang (elevated),
- c. kereta api bawah tanah (subway).

2 . Kereta rel diesel prameks

Kereta api Prambanan Ekspres merupakan nama bagi layanan transportasi kereta api (KA) yang menghubungkan Kutoarjo, Yogyakarta, dan Surakarta (hingga Stasiun Palur di timur kota). Saat ini beroperasi sepuluh kali pulang pergi dan dikelola oleh PT Kereta Api Daerah Operasi VI Yogyakarta.

Cikal bakal kereta yang melayani rute Yogya-Solo telah beroperasi sejak tahun 1960-an hingga akhir 1970-an. Kereta tersebut bernama "Kuda Putih" (karena ada logo bergambar kuda di bagian atas jendela masinisnya) dan merupakan KRD pertama di Indonesia. Setelah Kuda Putih tidak aktif (dihentikan pada tahun 1980-an), tidak ada lagi kereta api yang menghubungkan kedua kota itu. KA Prameks Solo–Yogyakarta pergi pulang (PP) kali pertama diluncurkan tanggal 20 Mei 1994, dengan hanya menggunakan empat rangkaian kereta kelas bisnis yang ditarik oleh lokomotif diesel dengan tarif 2000 rupiah. Rangkaian ini memakai kereta milik KA Senja Utama Solo yang beroperasi hanya pada malam hari. Awalnya KA itu berjalan hanya dua kali sehari pergi-pulang. Dalam kurun pengoperasiannya, KA Prameks telah mengalami beberapa kali perubahan jadwal pemberangkatan maupun sarana yang dipergunakan. Bahkan untuk periode yang singkat, ke dalam rangkaian ini pernah pula ditambahkan satu gerbong KA eksekutif dengan tarif 5000 rupiah. Setelah pola keberangkatannya diubah sesuai dengan keinginan pelanggan menjadi lima kali PP sehari, maka pada masa angkutan Lebaran 1998, manajemen PT KA (Persero) mengganti rangkaian kereta yang ditarik oleh lokomotif menjadi tiga set rangkaian KRD (kereta rel diesel).



Gambar 2.1 KRDE Prameks pertama.

Namun karena rangkaian KRDE ini dianggap uzur (buatan tahun 1980-an), KA Prameks sering mengalami kerusakan yang mengakibatkan keterlambatan. Akhirnya Ditjen KA Dephub bersama manajemen PT KA menambah satu set armada Prameks berupa KRDE prototipe pertama dari PT Inka Madiun pada 1 Maret 2006. Rangkaian ini adalah yang pertama kali dioperasikan di Indonesia. KRDE ini merupakan modifikasi dari KRL buatan BN/Holec ("Belgien-Nederland-Holland Electric", perusahaan KA Belanda-Belgia), Belgia, yang dimodifikasi oleh PT INKA dengan mengganti sumber daya menggunakan satu mesin diesel. Lima unit kereta Prameks per rangkaian KRDE tersebut terdiri atas satu unit kereta mesin (engine, KDE) diesel, satu unit kereta kotrail, dua unit kabin trailer dan satu unit trailer ditambah kabin masinis. KRDE ini diberi kode KDE-3, yang berarti KRDE kelas ekonomi. Sejak 13 Maret 2006, ditambahlah dua perjalanan KA Prameks ini menjadi tujuh kali PP. Seiring dengan dioperasikannya jalur rel ganda Yogyakarta-Kutoarjo pada 29 September 2007 dan sekaligus untuk memenuhi permintaan masyarakat Kulonprogo dan Purworejo, PT KA Daop VI Yogyakarta sejak 15 Oktober 2007 mulai melakukan uji coba perjalanan KA Prameks Yogyakarta – Kutoarjo - Solo Balapan PP

dengan pola operasi dua kali perjalanan sehari. Dengan bertambahnya dua set KRDE yang diluncurkan oleh Menhub di Stasiun Solo Balapan, 16 Februari 2008, pola operasi KA Prameks Solo-Yogyakarta mengalami peningkatan dari tujuh kali menjadi sepuluh kali PP, sedangkan Solo - Yogyakarta - Kutoarjo menjadi empat kali PP. Kereta api ini sekarang juga berhenti di Stasiun Maguwo (Bandara Adisucipto), sebagai bagian dari sistem terpadu transportasi Yogyakarta yang menghubungkan sarana transportasi umum darat (bus TransJogja dan taksi), kereta api, dan pesawat terbang (Wikipedia, 2019)

G. Multimedia

Menurut buku *Multimedia: Developer's Guide* yang ditulis oleh Perry P, multimedia adalah integrasi dari beberapa komponen seperti teks, audio, video, animasi, dan gambar. Multimedia ini akan diatur melalui software dalam komputer dan terdapat interaksi antara media dan pengguna.

Namun secara umum, menurut Rosch dalam *Kitab Suci Komputer dan Multimedia*, dapat dikatakan multimedia bila ada gabungan dari minimal tiga elemen. Ketiga elemen tersebut yakni, suara, gambar, dan teks.

Pengkodean dan kompresi multimedia telah digunakan untuk menyandikan data dengan penyimpanan disk yang lebih sedikit. Multimedia bervolume besar dalam metaverse dapat dikompresi menjadi fitur dimensi rendah dengan encoder, dan kemudian direkonstruksi oleh jaringan decoder lain di perangkat lokal perangkat tanpa kehilangan kualitas konten yang signifikan. (Chen, 2022)

Multimedia pada zaman sekarang merujuk kepada penggunaan gabungan beberapa media dalam penyajian pembelajaran melalui komputer. Setelah tahun 1980-an, multimedia didefinisikan sebagai penyampaian informasi secara interaktif dan terintegrasi yang mencakup teks, gambar, suara, video atau animasi. (Sri et al., 2013)

1. Elemen dalam Multimedia

Dalam buku *Information Technology in Business: Principles, Practice, Opportunities* dijelaskan bahwa ada enam elemen penting dalam multimedia.

Keenam elemen tersebut di antaranya

- Teks
- Foto
- Audio (suara)
- Video
- Animasi
- Virtual Reality

2. Jenis Multimedia

Multimedia juga memiliki jenis-jenisnya masing-masing. Berikut rincian dan contohnya:

- Multimedia interaktif artinya pengguna dapat melakukan kontrol dan mengakses berbagai elemen dalam multimedia untuk nantinya akan ditampilkan. Seperti, game, website, hingga multimedia pembelajaran.

- Multimedia hiperaktif maksudnya adalah jenis ini memiliki struktur dari elemen-elemen terkait dengan pengguna yang dapat mengarahkannya. Seperti, world wide web, website, game online.

- Multimedia linier artinya pengguna hanya dapat menjadi penikmat dan penonton

pada produk multimedia. Seperti, televisi, film, koran, dan sejenisnya. Baca artikel di tiku, "Apa Itu Multimedia? Ini Pengertian, Elemen, dan Jenisnya.

H. Android

Android dapat dibagi menjadi analisis statis, dinamis, dan hibrida . Analisis Statis mengekstrak fitur-fitur dari aplikasi Android tanpa menjalankannya ke dalam perangkat atau emulator Android. Oleh karena itu, memantau perilaku yang mencurigakan atau tidak biasa. Hal ini dapat mencapai cakupan fitur yang tinggi tetapi menghadapi banyak kelemahan, yaitu pengaburan kode dan pemuatan kode dinamis. (Elayan & Mustafa, 2021). Adapun definisi Android menurut beberapa para ahli didaftarkan sebagai berikut:

Menurut (Mayrhofer et al., 2021) Android sekarang menjadi antarmuka yang paling umum bagi pengguna global untuk berinteraksi dengan layanan. Di berbagai faktor bentuk yang berbeda (termasuk, misalnya, ponsel, tablet, perangkat yang dapat dikenakan, TV, Internet-of-Things, mobil, dan kategori penggunaan khusus lainnya), terdapat berbagai macam kasus penggunaan dari komunikasi, konsumsi kasus penggunaan dari komunikasi, konsumsi media, dan

hiburan hingga keuangan, kesehatan, dan sensor/aktuator fisik. Banyak dari aplikasi ini yang semakin penting dalam hal keamanan dan privasi, dan Android sebagai OS perlu memberikan jaminan yang memadai dan sesuai kepada pengguna serta dan juga para pengembang.

Menurut (Juansyah, 2015) Android adalah sebuah sistem operasi perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.



Gambar 2.2. Logo Android

Android umumnya digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. Awalnya, Google Inc. Membeli Android Inc. yang merupakan yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance yang terdiri dari 34 perusahaan perangkat keras, perangkat lunak dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia. Pada saat perilisan perdana Android pada tanggal 5 November 2007, Android bersama Open Handset Alliance mendukung pengembangan open source

pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android yang pertama yang mendapatkan dukungan penuh dari Google atau Google Mail Service (GMS) dan yang kedua adalah yang benar-benar bebas distribusikan tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution. Perkembangan sistem operasi Android yang sekarang menjadi sangat populer karena bersifat open source menjadikannya sebagai sebuah sistem operasi yang diminati oleh banyak pengguna. Adapun beberapa kelebihan dari sistem operasi Android adalah sebagai berikut:

1. Dukungan Produsen Hardware

Smartphone dengan sistem operasi Android kini menjadi sangat populer dan digemari. Hal ini tak lepas dari dukungan produsen hardware, seperti Sony dengan Xperia-nya, Samsung dengan Galaxy-nya, HTC, LG, Sharp, Fujitsu, Infinix, Xiaomi, Oppo , Vivo, Realme dan berbagai merek lainnya.

2. Praktis dan Banyak Aplikasi Tersedia

Android sangat nyaman digunakan untuk browsing, download, bermain game, belajar dan lainnya hanya dengan berlangganan paket internet atau WiFi. Selain itu Android dapat digunakan sebagai modem atau hotspot.

a. Versi Android

Perkembangan sistem operasi Android dari awal pertama dipakai hingga saat ini terdapat berbagai versi Android yang telah diliris. Adapun versi-versi Android yang telah diliris adalah sebagai berikut :

- (1) Android versi 1.1
- (2) Android versi 1.5 (Cupcake)
- (3) Android versi 1.6 (Donut)
- (4) Android versi 2.0/2.1 (Eclair)
- (5) Android versi 2.2 (Froyo : Frozen Yoghurt)
- (6) Android versi 2.3 (Gingerbread)
- (7) Android versi 3.0/3.1 (Honeycomb)
- (8) Android versi 3.0/3.1 (ICS : Ice Cream Sandwich)
- (9) Android Versi 4.1/4.3 (Jelly Bean)
- (10) Android versi 4.4 (KitKat)
- (11) Android versi 5.0/5.1 (Lollipop)
- (12) Android versi 6.0 (Marshmallow)
- (13) Android versi 7.0/7.1 (Nougat)
- (14) Android versi 8.0/8.1 (Oreo)
- (15) Android versi 9.0 (Pie)

(16) Android versi 10.0 (Q)

(17) Android versi 11.0 (R)

b. Struktur Android

Sistem Operasi Android memiliki arsitektur yang terdiri dari Linux Kernel, Hardware Abstraction Layer (HAL), Android Runtime, Libraries, Framework, System Applications.



Gambar 2.3 Arsitektur Android

Penjelasan dari tiap-tiap komponen pada arsitektur Android adalah sebagai berikut :

1. Linux Kernel Kernel merupakan lapisan terdalam yang berada pada sistem operasi baik itu Linux maupun Android. Lapisan ini tidak benar — benar berinteraksi dengan pengguna maupun developer, tapi lapisan ini merupakan

jantung dari seluruh sistem di Android karena lapisan inilah yang memberikan fungsi-fungsi berikut pada sistem Android:

- a. Abstraksi Hardware
- b. Program Manajemen Memory
- c. Pengaturan Keamanan
- d. Manajemen Energi Software (Baterai)
- e. Driver
- f. Network Stack

2. Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer (HAL) memberikan antarmuka standar yang mengungkapkan kemampuan perangkat keras perangkat ke kerangka kerja API Java yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing menerapkan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Ketika API framework melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul library untuk komponen perangkat keras tersebut.

3. Android Runtime (ART)

Terletak pada level yang sama dengan lapisan Library juga terdapat Lapisan Android Runtime dan juga sekumpulan Library Java yang dikhususkan untuk

Android. Programmer Aplikasi Android membuat aplikasinya menggunakan bahasa pemrograman Java.

4. Library

Library membawa sekumpulan instruksi untuk mengarahkan perangkat Android kita dalam menangani berbagai tipe data. Contohnya, perekam dari berbagai macam format Video dan Audio ditangani oleh Media Framework Library.

Berikut adalah beberapa kegunaan Library:

- (a) Surface Manager: Mengolah tampilan Windows Pada Layar
- (b) SGL: Grafik 2 Dimensi
- (c) Open GL|ES: Grafik 3 Dimensi maupun 2 Dimensi
- (d) Media Framework: Menunjang perekaman dari berbagai macam format audio, video, dan gambar
- (e) Free Type: Penerjemah Font
- (f) WebKit: Mesin Browser
- (g) libc (System C libraries)
- (h) MySQL: Database
- (i) Open SSL: Keamanan Jaringan

I. Algoritma

Ditinjau dari asal-usul katanya, Algoritma sendiri mempunyai sejarah yang aneh. Orang hanya menemukan kata algorism yang berarti proses menghitung dengan angka arab. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dari algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam beberapa konteks, algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegus apapun algoritma, kalau memeberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik.

Pertimabangan kedua yang harus diperhatikan adalah kita harus mengetahui seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini penting terutama pada algoritma untuk menyelesaikan masalah yang memerlukan aproksimasi hasil (hasil yang hanya berupa pendekatan). Algoritma yang baik harus mampu memberikan hasil yang sedekat mungkin dengan dengan nilai yang sebenarnya. Ketiga adalah efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari dua hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar (paling mendekati), tetapi jika kita harus menunggu berjam-jam untuk mendapatkan keluarannya, algoritma tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin

besar memori yang terpakai maka semakin buruklah algoritma tersebut. Dalam kenyataannya, setiap orang bisa membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, walaupun terjadi perbedaan dalam menyusun algoritma, tentunya kita mengharapkan keluaran yang sama. Jika terjadi demikian carilah algoritma yang paling efisien dan cepat.

Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang mengarah dalam terminologi algoritma. Namun, jangan beranggapan algoritma selalu identik dengan ilmu komputer saja. Dalam kehidupan sehari-hari pun banyak terdapat proses yang dinyatakan dalam suatu algoritma.

Struktur Dasar Algoritma berisi langkah-langkah penyelesaian suatu masalah. Langkah-langkah tersebut bisa berupa runtunan aksi (*sequence*), pemilihan aksi (*selection*), perulangan aksi (*iteration*) atau kombinasi dari ketiganya (Ade Hastuty Hasyim, 2021).

Jadi struktur dasar pembangunan algoritma ada tiga, yaitu:

- a. Struktur runtunan digunakan untuk program yang pernyataannya sequential atau urutan.
- b. Struktur pemilihan digunakan untuk program yang menggunakan pemilihan atau penyeleksian kondisi.
- c. Struktur perulangan digunakan untuk program yang pernyataannya akan dieksekusi berulang-ulang.

J. Dasar Pemrograman

1. Element Pemrograman Pascal/Delphi

Pascal adalah bahasa pemrograman yang pertama kali dibuat oleh Profesor Niklaus Wirth, seorang anggota *international federation of Processing* (IFIP) pada tahun 1971. Dengan mengambil nama dari matematikawan Perancis, Blaise Pascal, yang pertama kali menciptakan mesin perhitungan, Profesor Niklaus Wirth membuat bahasa Pascal ini sebagai alat bantu untuk mengajarkan konsep pemrograman komputer kepada mahasiswanya. Selain itu Profesor Niklaus Wirth membuat Pascal juga untuk melengkapi kekurangan-kekurangan bahasa pemrograman yang ada pada saat itu.

Sedangkan Borland Delphi dipersiapkan perusahaan Borland untuk menggantikan Turbo Pascal, karna semakin ramainya pemrograman berorientasi obyek (*Object Oriented Programing*) atau Visual. Borland Delphi adalah sebuah alat pengembangan aplikasi untuk sistem operasi Microsoft Windows. Delphi sangat berguna dan mudah digunakan untuk membuat suatu program berbasis GUI (*Graphical User Interface*) atau *console* (mode teks).

Borland delphi atau bisa disebut delphi saja, merupakan sarana pemrograman aplikasi visual. Delphi merupakan generasi penerus turbo pascal. Turbo pascal yang digunakan pada tahun 1983 dirancang untuk dijalankan pada sistem operasi DOS (yang merupakan sistem operasi yang paling banyak digunakan pada saat itu). Sedangkan delphi yang diluncurkan untuk beroperasi dibawah sistem operasi windows (Ade Hastuty Hasyim, 2021).

2. Struktur Pemrograman Pascal/Delphi

Elemen-elemen dalam program harus sesuai dengan urutannya, beberapa elemen bisa dihilangkan bila tidak diperlukan. Komentar dapat disertakan dalam penulisan kode. Komentar tidak akan disertakan dalam kompilasi (compile) atau saat program dijalankan (execute). Penanda komentar adalah (* dan diakhiri dengan *). atau dapat pula dengan tanda { dengan akhiran}. Pemakaian komentar dalam bahasa pascal tidak boleh salah, karena akan menimbulkan masalah. Penulisan komentar yang salah seperti :({disini komentar}]

```
PROGRAM
NamaProgram (FileList);
CONST
(*Deklarasi Konstanta*)
TYPE
(*Deklarasi Type*)
VAR
(*Deklarasi Variabel*)(*Definisi
SubProgram*)BEGIN
```

Pada saat code dicompile, akan memberikan pesan kesalahan karena compiler akan melihat tanda { yang pertama dan tanda } yang pertama pula, sehingga tanda } yang kedua akan dianggap kesalahan. Berikut beberapa contoh penulisan komentar yang benar:

```
({ disini komentar]
(disini komentar ) writeln('test komentar'); { komentar lagi]
(* disini komentar *)
```

Pemberian komentar akan mempermudah dalam memahami suatu kode program (source code). Bila kita menulis program tanpa memberikankomentar, saat kita membuka kembali kode yang kita tulis dalam jangka waktu berselang lama. Akan mempersulit kita memahami Program yang kita buat sebelumnya (bila program sangat rumit) (Ade Hastuty Hasyim, 2021).

K. Flowchart

Flowchart dalam Bahasa Indonesia diterjemahkan sebagai Diagram Alir. Dari dua kata ini, maka dapat kita bayangkan bahwa *flowchart* itu berbentuk diagram yang bentuknya dapat mengalirkan sesuatu. Hal ini memang benar, *flowchart* memang melukiskan suatu aliran kegiatan dari awal hingga akhir mengenai suatu langkah-langkah dalam penyelesaian suatu masalah. Masalah yang kita pelajari tentu saja masalah pemrograman dengan menggunakan komputer.

Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan penerimaan *input*, pemrosesan *input*, dan diakhiri dengan penampilan *output*. Dengan kata lain *flowchart* merupakan suatu gambar yang menjelaskan urutan pembaca data, pemrosesan data, pengambilan keputusan terhadap data, penyajian hasil pemrosesan data.

Ada dua macam *flowchart* yang menggambarkan proses dengan komputer, yaitu:

a. *Flowchart* sistem

Flowchart sistem yaitu bagan dengan simbol-simbol tertentu yang menggambarkan urutan prosedur dan proses suatu file dalam suatu media menjadi file didalam media lain, dalam suatu sistem pengolahan data.

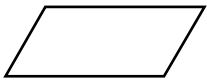
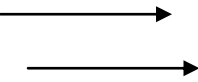
b. *Flowchart* program

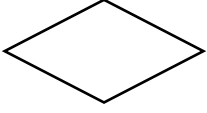
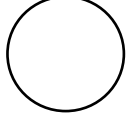
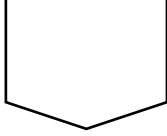
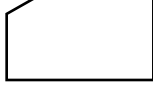
Flowchart program yaitu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses dan hubungan antar proses secara mendetail didalam suatu program.

Dalam pembuatan *flowchart* program tidak ada rumus atau patokan yang bersifat mutlak. Karena *Flowchart* merupakan gambaran hasil pemikiran dalam menganalisis suatu masalah dengan komputer. Sehingga flowchart yang dihasilkan dapat bervariasi antara satu pemrograman dengan yang lainnya. Namun secara garis besar setiap pengolahan selalu terdiri atas tiga bagian utama, yaitu *Input*, Proses pengolahan dan *Output* (Ade Hastuty Hasyim, 2021).

Berikut simbol-simbol beserta fungsi *flowchart*:

Tabel 2.1. Simbol Flowchart

No	Simbol	Nama	Fungsi
1		Terminal	Awal atau akhir program
2		<i>Input/Output</i>	Menunjukkan proses input atau output tanpa mempertimbangkan jenis peralatan.
3		<i>Process</i>	Menunjukkan tindakan yang dilakukan komputer.
4		<i>Flow</i>	Menunjukkan cara suatu proses bekerja.

No	Simbol	Nama	Fungsi
5		<i>Decision</i>	Menampilkan kondisi tertentu dengan opsi ya/tidak.
6		<i>Connector</i>	Menunjukkan bagaimana proses terhubung di halaman yang sama.
7		<i>Offline Connector</i>	Menunjukkan bagaimana berbagai halaman prosedur berhubungan satu sama lain.
8		<i>Prefefned Process</i>	Menunjukkan lokasi pengolahan harga awal.
9		<i>Punched card</i>	Mengirimkan input atau output ke kartu.
10		<i>Document</i>	Output dicetak menggunakan printer.

L. UML (Unified Modelling Language)

UML merupakan salah satu standart bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis & desain, sert amenggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan,

menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak. Pendekatan analisa & rancangan dengan menggunakan model *object oriented* mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi *software* sudah meningkat dan mulai kompleks. (Fitria et al., n.d.)

Tujuan dibuatnya UML adalah sebagai berikut:

1. Dapat memberikan sebuah bahasa pemodelan yang bebas dari bahasa pemrograman dan proses rekayasa.
2. Memberi model yang siap dipakai. Ini disebabkan karena UML adalah sebuah bahasa pemodelan yang visual dan ekspresif untuk dikembangkan dan saling menukar model.
3. Menyatukan praktek terbaik yang ada pada proses pemodelan
4. Dapat berfungsi sebagai dasar (*blueprint*) sebuah aplikasi karena bersifat sangat lengkap dan detail. Dengan adanya cetak biru atau blueprint akan dapat dilakukan *reverse engineering* pada aplikasi. (Chandra et al., n.d.)

M. Human Computer Interaction (HCI)

Interaksi Manusia Komputer (IMK), yang sering disebut sebagai Human Computer Interaction (HCI), merujuk pada cabang ilmu yang memperdalam aspek desain dan penggunaan teknologi komputer.

HCI menitikberatkan pada pemahaman antarmuka antara individu dan komputer, serta bagaimana merancang, mengevaluasi, dan menerapkan sistem

komputer interaktif yang memuaskan pengguna. Ini adalah tentang bagaimana manusia dan sistem komputer, secara luas, berkolaborasi untuk mencapai tujuan bersama. Bidang Interaksi Manusia Komputer telah mengalami pertumbuhan pesat sejak awal kemunculannya. Saat komputer pribadi pertama kali dikenal sebagai *personal computer* (PC), pengguna awal harus menjalani pelatihan yang intensif agar dapat menggunakan perangkat ini dengan benar.

Pentingnya HCI muncul ketika pengalaman pengguna dengan komputer pada masa awal tersebut penuh dengan tantangan. Tugas sederhana seperti menggerakkan kursor atau menghapus teks menjadi hal yang rumit dan sulit dilakukan oleh pengguna.

Sementara membawa manusia dan non-manusia ke tingkat yang sama dalam ontologi relasional masih jauh dari tidak kontroversial, saya berpendapat bahwa ini mungkin merupakan cara yang produktif untuk HCI untuk berevolusi dalam menanggapi dunia yang terus berubah. Siklus paradigma mengidentifikasi di mana kerangka pemikiran yang ada secara sistematis berjuang untuk mengikuti apa yang diamati secara empiris-krisis model (Frauenberger, 2019)

N. MySQL

Beberapa pengertian MySQL menurut para ahli:

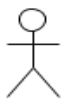
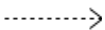
1. Menurut Raharjo (2011:21), “MySQL merupakan RDBMS (atau *server database*) yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*”.

2. Menurut Kadir (2008:2), “MySQL adalah sebuah *software open source* yang digunakan untuk membuat sebuah *database*.” Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*

O. Use case

Use case merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang akan diproses oleh sistem dan komponen – komponennya. *Use case* bekerja dengan menggunakan *scenario* yang merupakan deskripsi dari urutan atau langkah – langkah yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem maupun sebaliknya .(Setiyani, 2021). Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu:

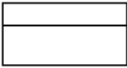
Tabel 2.2. Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</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>).

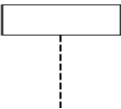
NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</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 aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2.3 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
----	--------	------	------------

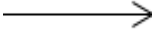
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>NaryAssociation</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2.4. Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

NO	GAMBAR	NAMA	KETERANGAN
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

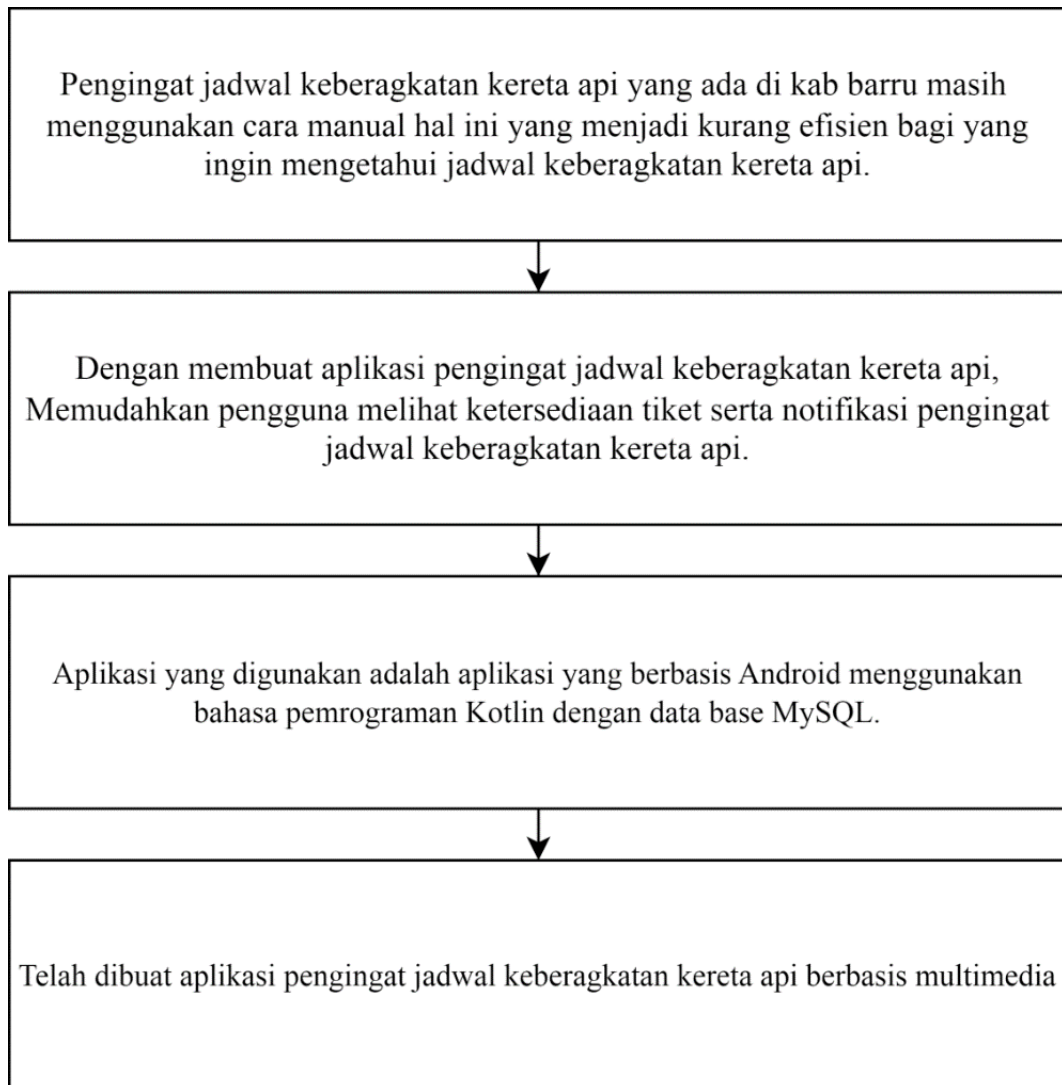
Tabel 2.5. Simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.6. Simbol *Actifity* Diagram

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

P. Kerangka Fikir



BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Tempat penelitian dilakukan di PT. Kereta Api yang beralamat lengkap Manuba, Kec. Mallusetasi, Kabupaten Barru, Sulawesi Selatan, dimana proses pengambilan data yang akan dimasukkan dalam aplikasi nantinya

NO	UraianKegiatan	Bulan 2023				
		Mei	Mei	Juni	Juni	Juli
1	StudiLiteratur					
2	Perancanganaplikasi					
3	Pembuatanaplikasi					
4	Pengujian					

B. Jenis Penelitian

1. Penelitian Kepustakaan

Yaitu pengumpulan data dengancara membaca buku mengenai literature dan buku lain yang bersifat ilmiah yang berkaitan dengan materi pembahasan.

2. Penelitian Lapangan

Yaitu kegiatan yang dilakukan dengan cara mengumpulkan data secara langsung dari objek penelitian melalui pengamatan secara langsung.

C. Metode Pengumpulan Data

Dalam menunjang penelitian ini diperlukan data untuk menentukan pemecahan masalah terhadap permasalahan yang dibahas.

Metode pengumpulan data yang dilakukan pada penelitian ini antara lain:

1. Observasi

Observasi yang dilakukan yaitu dengan melakukan pengumpulan data secara langsung ke lapangan untuk mengetahui permasalahan yang terjadi secara langsung berdasarkan sistem yang berjalan dan untuk memperoleh gambaran yang jelas mengenai permasalahan yang diteliti.

2. Wawancara

Metode pengumpulan data melalui wawancara dilakukan untuk mendapatkan informasi secara langsung dengan tatap muka dan tanya jawab langsung.

3. Studi Literatur

Studi Literatur dilakukan dengan menelusuri sumber-sumber tulisan yang pernah dibuat berdasarkan pengamatan dilakukan yang kemudian dapat dijadikan acuan ataupun referensi dalam melakukan penelitian.

D. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan penulis menggunakan pendekatan metode SDLC (System Development Life Cycle) Waterfall Model, sering juga disebut model sekuensial linier (sequential linier). Waterfall Model adalah suatu model yang menyediakan pendekatan alur pengembangan perangkat lunak secara terurut yang dimulai dari tahapan Requirement Analysis and

Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing, Operation and Maintenance. Adapun penjelasan dari langkah – langkah yang dilakukan dalam membangun sistem dengan

menggunakan Waterfall Model yaitu sebagai berikut :

1. Requirement Analysis and Definition,

Tahapan ini guna menggali semua informasi yang menganalisis bagaimana pengguna Sistem Informasi mengatur pengingat jadwal yang mereka gunakan untuk Pengingat Jadwal Kereta Api. Untuk mengidentifikasi dan memberikan solusi terhadap permasalahan yang ada. Penulis akan melakukan analisis menggunakan metode 4 (empat) tahapan yaitu Identify, Understand, Analyze, dan Report.

2. System and Software Design,

Tahapan ini adalah tahapan yang fokus pada perancangan aplikasi yang dibuat. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan serta rumusan masalah diterjemahkan ke dalam bentuk desain sistem, database, use case diagram, activity diagram, sequence diagram dan class diagram. Adapun program yang digunakan oleh penulis untuk pembuatan diagram yang telah disebutkan sebelumnya menggunakan program StarUML.

3. Implementation and Unit Testing,

Tahapan ini adalah tahapan untuk mengubah desain ke dalam bentuk nyata. Hasil dari tahap ini adalah aplikasi sesuai dengan “blueprint” yang telah dibuat pada tahap desain.

4. Integration and System Testing,

Tahapan ini fokus pada pengujian perangkat lunak dari segi logika dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian yang dilakukan berdasarkan metode black box testing untuk mengetahui tingkat keberhasilan dari sistem yang diuji.

E. Analisis

1. Analisis Sistem Yang Sedang Berjalan

Sistem pengingat jadwal kereta api yang sedang berjalan masih menggunakan sistem konvensional, sehingga pengguna harus menanyakan langsung jadwal pemberangkatan kereta api, mendatangi stasiun, lalu mengecek jadwal atau melihat jadwal pemberangkatan stasiun kereta api baru-makassar di brosur yang ada di papan informasi yang dimana untuk datang ke lokasi tersebut cukup memakan waktu sehingga dinilai tidak efisien.

2. Analisis Kelemahan Sistem Yang Berjalan

Dalam penyusunan tugas akhir ini, penulis memaparkan analisis sistem melalui 4 (empat) tahapan antara lain Identify, Understand, Analyze dan Report sebagai metode untuk mengidentifikasi, menganalisis serta memecahkan

permasalahan yang nantinya dapat digunakan sebagai referensi untuk pengembangan sistem untuk masa mendatang. Berikut ini hasil analisis sistem dari pengingat jadwal kereta api yang digunakan oleh pengguna Sistem Informasi:

a. Identify Identify,

yaitu mengidentifikasi masalah. Pada penelitian ini masalah yang dihadapi pada bagian pengingat jadwal yang digunakan pengguna adalah masih menggunakan cara konvensional dengan cara melihat jadwal keberangkatan pada papan informasi di lokasi stasiun atau menanyakan jadwal keberangkatan pada orang lain yang dimana untuk menavigasikannya cukup memakan waktu sehingga dinilai tidak efisien.

b. Understand Understand,

yaitu memahami kerja dari sistem yang ada. Sistem yang berjalan saat ini adalah pengingat dan penjadwalan kereta api yang digunakan pengguna adalah masih menanyakan secara repetitif dan sering kali ditemui banyak kekurangan-kekurangan yang harus dihadapi seperti salah informasi jadwal, jam keberangkatan yang diampun berubah, kesalahan pengguna dalam menavigasikan pada web, serta resetnya sesi pada web. Hal ini mengakibatkan waktu yang dibutuhkan untuk melihat jadwal menjadi lebih memakan banyak waktu.

c. Analyze Analyze,

yaitu menganalisis sistem. Dari penjelasan pada poin sebelumnya dapat diketahui bahwa proses pengguna melihat jadwal dinilai kurang efektif serta memakan banyak waktu karena pengguna masih menggunakan cara konvensional.

Dari pembahasan di atas, maka dari itu diperlukan suatu aplikasi yang dapat memudahkan pengguna dalam mengingat jadwal keberangkatan kereta api. Diharapkan aplikasi dapat membantu meningkatkan efektivitas serta efisiensi pengguna dalam menjadwalkan keberagkaatan yang ditempuh.

d. Report Report,

yaitu membuat laporan akhir analisis. Dari hasil analisis di atas maka akan disimpulkan bahwa aplikasi yang dibuat dapat menangani pengingat penjadwalan dan catatan keberangkatan serta mengingatkan pengguna terhadap kemudahan karna adanya aplikasi pengingat jadwal kereta api yang akan ditempuhnya.

3. Analisis Kebutuhan Sistem

a. Kebutuhan Informasi

Kebutuhan informasi yang diperlukan dalam pembuatan aplikasi ini adalah data-data yang terdiri dari :

- 1) Data nama mata kuliah
- 2) Data nama dosen pengampu mata kuliah
- 3) Data hari dan waktu perkuliahan
- 4) Data kelas mata kuliah

b. Kebutuhan Perangkat Keras

1. Perangkat Keras

Perangkat keras yang digunakan adalah sebuah laptop dengan spesifikasi yang dapat dilihat dari tabel berikut :

Tabel 3. 1. Perangkat Keras

Jenis	Spesifikasi
Laptop	Asus
Processor	Intel(R) Celeron(R) N4000 CPU @1.10GHz 1.10 GHz
Memory	4,00 GB
Hardisk	500GB

2. Perangkat Lunak

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

Tabel 3. 2 Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	java
Tools	XAMPP
	Sublime Text
	Adobe XD ,StarUML dan Browser Google

	<i>Chrome</i>
--	---------------

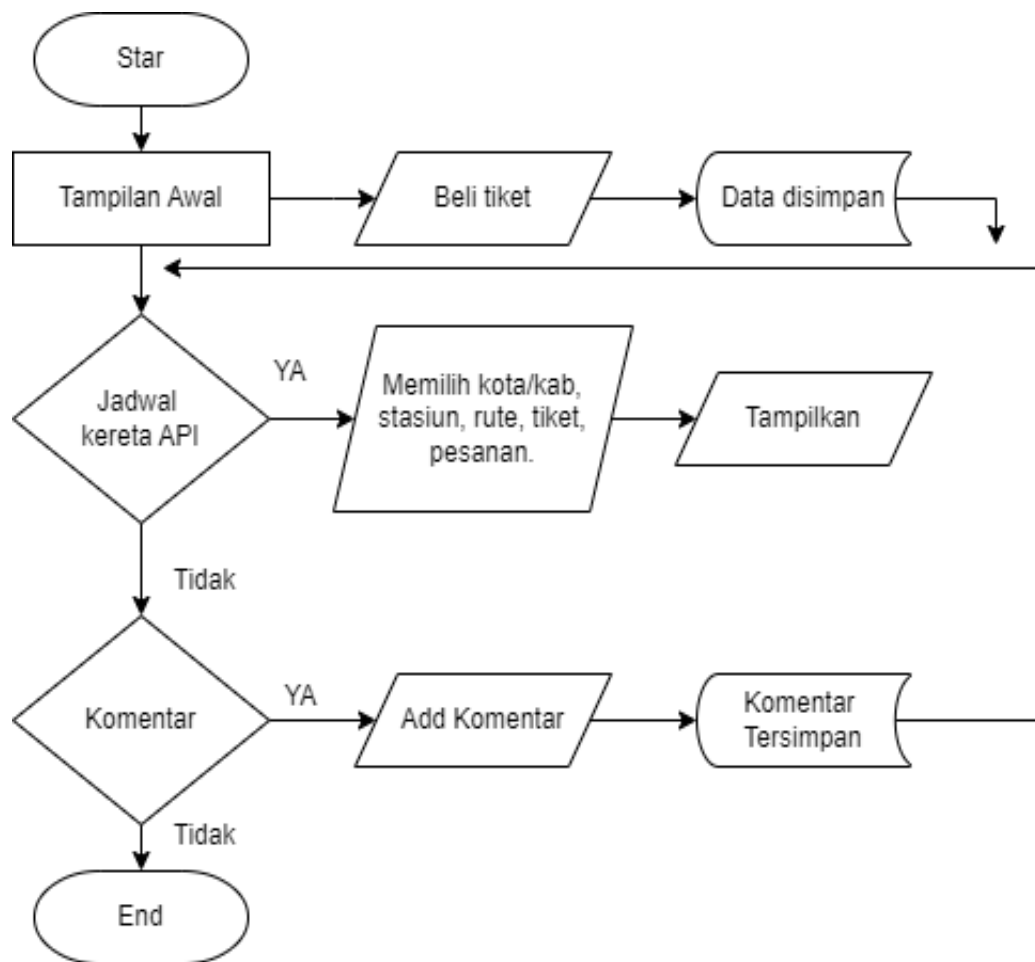
f. Pengguna Sistem

Adapun pengguna yang akan berinteraksi dengan aplikasi ini yaitu masyarakat yang ingin menggunakan aplikasi pengingat jadwal kereta api sebagai Sistem Informasi yang disebut user. User adalah orang yang mempunyai hak penuh terhadap aplikasi yang ada atau orang yang sepenuhnya memiliki hak untuk mengolah data dan informasi yang ada pada aplikasi.

F. Desain Sistem

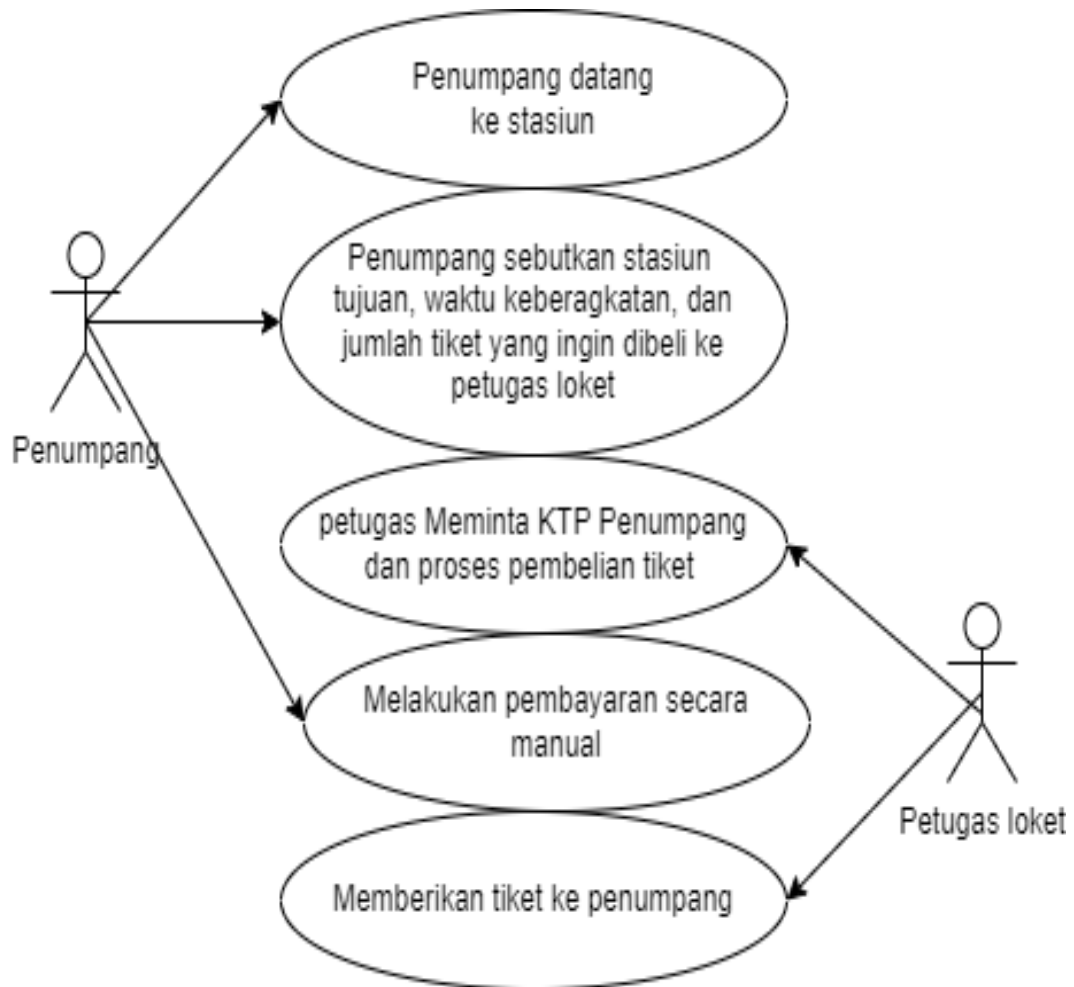
1. Flowchart Aplikasi

Pada tahap perancangan aplikasi jadwal kereta api membutuhkan penjelasan pada setiap halaman aplikasi secara bertahap, mulai dari proses awal ketika memulai aplikasi, memasuki menu dan sub menu pada aplikasi sampai dengan mengakhiri aplikasi, *Flowchart* Aplikasi jadwal kereta api dapat dilihat pada gambar 3.1.



Gambar 3.1 *Flowchart* Aplikasi

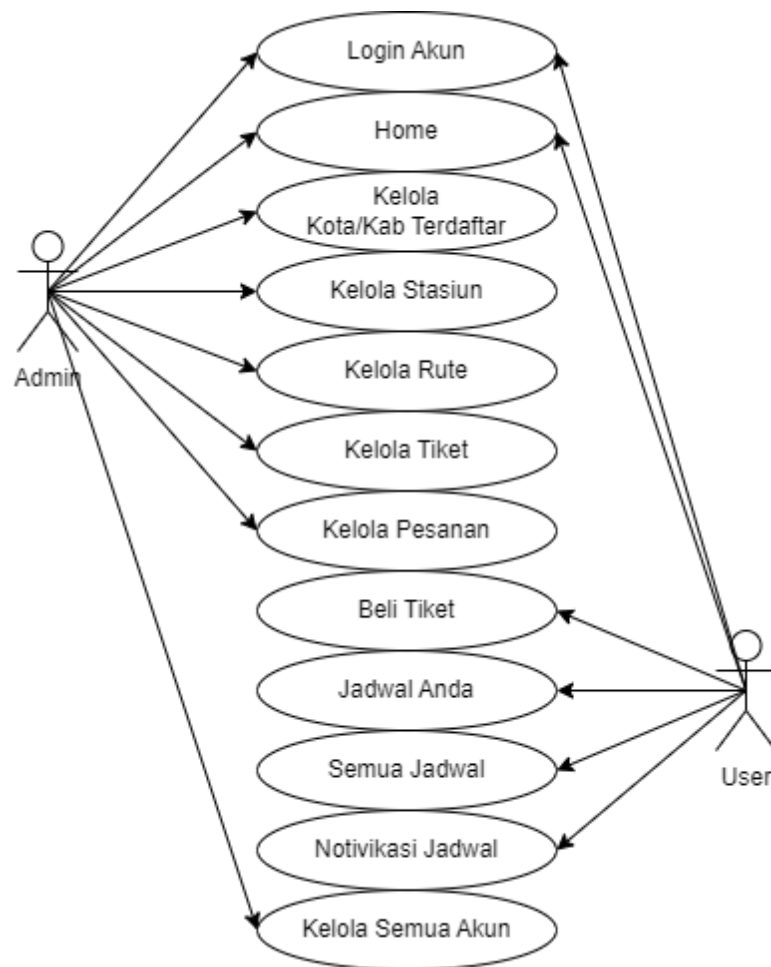
2. Sistem berjalan



Gambar 3.2 Sistem Berjalan

Sistem yang berjalan di atas menjelaskan dimana penumpang datang ke stasiun kemudian menanyakan ke petugas loket, kemudian petugas tentunya meminta kartu identitas KTP untuk menindak lanjuti pembelian tiket, penumpang melakukan pembayaran secara manual kemudian petugas memberikan tiket.

3. Sistem yang diusulkan



Gambar 3.3 Sistem yang diusulkan

Sistem yang diusulkan tentunya memiliki dua aktor admin dan user. Admin dapat mengelola kota/kab terdaftar, stasiun, tiket, pesanan serta kelola semua akun, sedangkan user dapat mengakses beli tiket, jadwal anda, notifikasi jadwal.

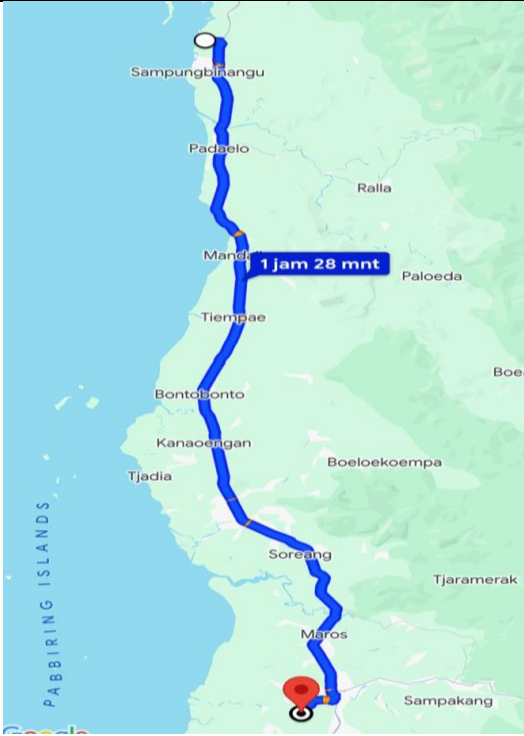
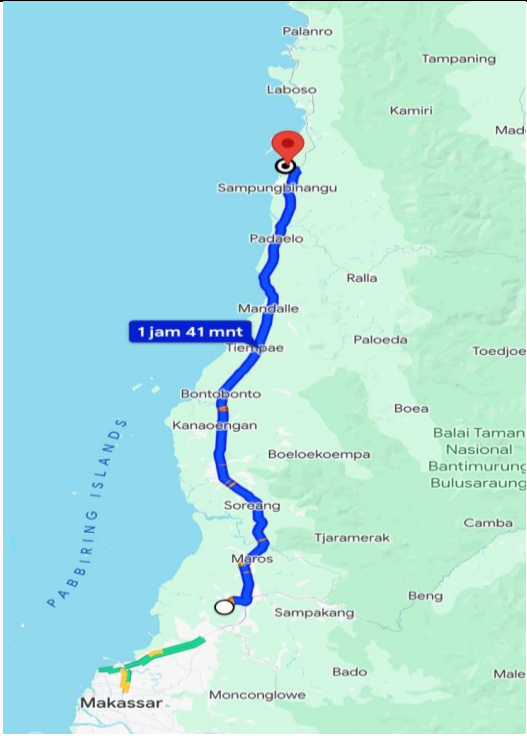
BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Selama melakukan penelitian di PT. Kereta Api yang beralamat lengkap di Manuba, Kec. Mallusetasi, Kabupaten Barru, Sulawesi Selatan, dalam membangun aplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia, didapatkanlah hasil penelitian antara lain. Jadwal Stasiun maros – Stasiun garogkong, Jadwal Stasiun garogkong – Stasiun maros, Jarak antara stasiun, dan Daftar harga tiket asal – tujuan kereta api sulawesi selatan.

Tabel 4.1 Maps Stasiun

Stasiun maros – Stasiun garogkong	Stasiun garogkong – Stasiun maros
	

Tabel 4.2 Jadwal Stasiun maros – Stasiun garogkong

Stasiun	Kode 1	Kode 3	KAIS 5
Maros	08.00	14.00	10.35
Rammang-Rammang	08.47	14.17	Ls
Pagkajene	08.57	14.27	Ls
Labakkang	09.12	14.42	11.33
Ma’rang	09.25	14.55	Ls
Mandalle	09.34	15.04	Ls
Tanete Rilau	09.50	15.20	Ls
Barru	10.03	15.33	Ls
Garongkong	10.15	15.45	12.24

Tabel 4.3 Jadwal Stasiun garogkong – Stasiun maros

Stasiun	Kode 2	Kode 4	KAIS 6
Garongkong	10.45	16.15	12.40
Barru	10.59	16.29	Ls
Tanete Rilau	11.12	16.42	Ls
Labakkang	11.28	16.58	Ls
Mandalle	11.36	17.06	Ls
Ma’rang	11.51	17.21	13.38
Pagkajene	12.05	17.35	Ls
Rammang-Rammang	12.15	17.45	Ls
Maros	12.30	18.00	14.35

Tabel 4.4. Jarak antara stasiun

Jarak Antara Stasiun		
Stasiun Asal	Stasiun Tujuan	Jarak
STA. Garongkong	STA.Barru	4,7 Km
STA.Barru	STA.Tanete Rilau	9 Km
STA.Tanete Rilau	STA.Mandalle	12,7 Km
STA.Mandalle	STA.Ma’rang	7,8 Km
STA.Ma’rang	STA.Labakkang	9 Km
STA.Labakkang	STA.Pagkajene	11 Km
STA.Pagkajene	STA.Rammang-Rammang	8 Km
STA.Rammang-Rammang	STA.Maros	12 Km

Tabel 4.5 Daftar harga tiket asal – tiket tujuan kereta api sulawesi selatan.

Stasiun Asal										
Stasiun Tujuan		Maros	Rammang2	Pagkajene	Labakkang	Ma'rang	Mandalle	Tanete Riau	Barru	Garongkong
	Maros	5k	5k	5k	5k	10k	10k	10k	10k	10k
	Rammang2	5k	5k	5k	5k	5k	10k	10k	10k	10k
	Pagkajene	5k	5k	5k	5k	5k	5k	10k	10k	10k
	Labakkang	5k	5k	5k	5k	5k	5k	5k	5k	5k
	Ma'rang	10k	5k	5k	5k	5k	5k	5k	5k	5k
	Mandalle	10k	10k	5k	5k	5k	5k	5k	5k	5k
	Tanete Riau	10k	10k	10k	5k	5k	5k	5k	5k	5k
	Barru	10k	10k	10k	5k	5k	5k	5k	5k	5k
	Garongkong	10k	10k	10k	5k	5k	5k	5k	5k	5k

Dari data yang didapatkan hasil dari wawancara hanya terdapat 3 poin penting diatas dari jadwal stasiun, Jarak antara stasiun, dan Harga tiket asal dan tiket tujuan. Data diatas digunakan untuk diinput diaplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia.

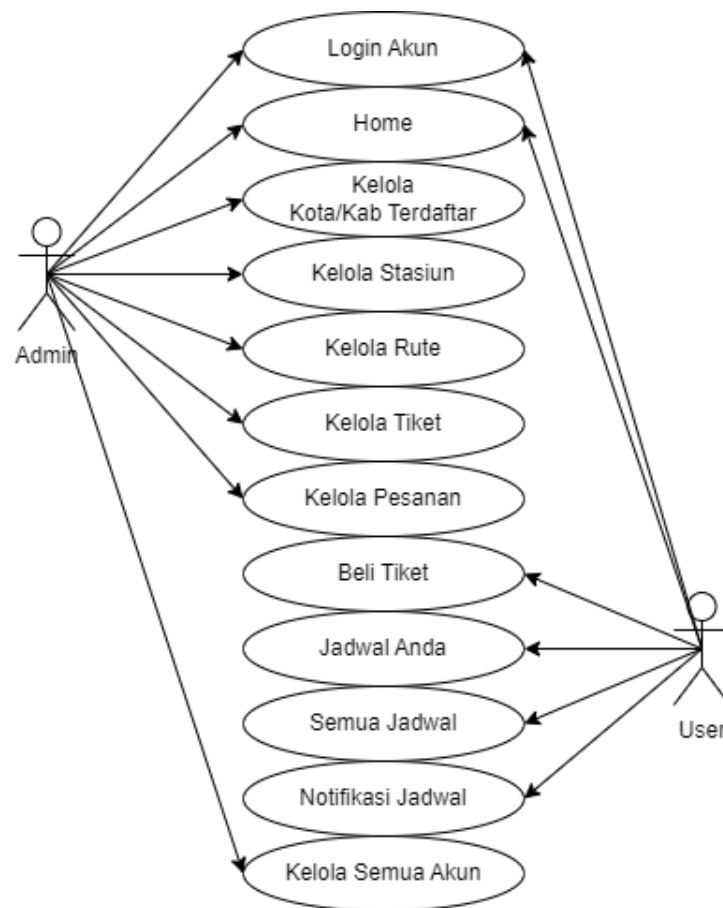
B. Pembahasan

1. Analisis Aliran Data UML

1.1. Use Case Diagram

Use Case Diagram atau diagram *use case* merupakan pemodelan untuk memodelkan kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi. *Use case* merupakan suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem.

Use case juga dapat digunakan untuk merepresentasikan interaksi yang terjadi antara aktor dengan proses sistem yang dibuat. Pada perancangan *use case* juga terdapat skenario, yaitu langkah yang menerangkan urutan kejadian antara pengguna dengan sistem. Diagram *use case* pada aplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia pada penelitian ini digambarkan seperti berikut ini:



Gambar 4.1. *Use Case Diagram.*

1.2. Tabel Use Case

Penjabaran use case diagram di atas didefinisikan atas definisi aktor dan definisi use case. Definisi aktor sebagai berikut :

Tabel 4.6. Definisi Aktor dalam Sistem.

No	Aktor	Keterangan
1	User	User merupakan aktor yang berperan sebagai pengguna pengingat jadwal kereta api berbasis multimedia .
2	Admin	Admin merupakan aktor yang memiliki hak akses lebih tinggi dibanding User karena Admin dapat mengelola Kota/kab terdaftar, stasiun, rute, tiket, pesanan, serta semua akun dan User setelah beli tiket dapat melihat jadwal anda dan semua jadwal.

Sedangkan untuk penjelasan definisi setiap use case sebagai berikut:

Tabel 4.7. Definisi Use Case Sistem

No	Use Case	Keterangan
1	Login	Melakukan Login akun.
2	Home	Menampilkan halaman utama aplikasi
3	Kota/kab terdaftar	Menampilkan Kota serta kabupaten terdaftar
4	Stasiun	Menampilkan Stasiun-stasiun kereta api yang ada di sulawesi selatan.

5	Rute	Menampilkan rute kereta api dari stasiun awal ke stasiun tujuan.
6	Tiket	Manampilkan harga tiket, serta stok tiket.
No	Use Case	Keterangan
7	Pesanan	Manampilkan pesanan tiket yang dibuat user.
8	Beli Tiket	Menampilkan beli tiket.
9	Jadwal anda	Menampilkan jadwal anda hari ini.
10	Semua jadwal	Menampilkan semua jadwal yang ada di stasiun.
11	Notifikasi jadwal	Menampilkan notifikasi di smartphone 30 menit sebelum jadal keberangkatan .
12	Semua akun	Menampilkan semua akun yang menggunakan aplikasi ini.

Dalam use case yang telah didefinisikan di atas, terdapat beberapa use case yang memiliki alur proses yang hampir sama, sehingga dalam penjabaran selanjutnya akan didefinisikan secara ringkas untuk memudahkan pemahaman proses tiap use case. Berikut adalah skenario use case yang telah didefinisikan sebagai berikut :

1) *Use Case Login*

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk Masuk ke aplikasi

Pre-condition : -

Post-condition : - Data baru sudah tersimpan.

Tabel 4.8. .Skenario Login.

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan User name dan password.	
2. Menekan tombol Login.	3. Mengecek validitas isian data.
	4. Jika data yang dimasukkan valid, maka langsung masuk kehalaman utama aplikasi.
Jika data yang dimasukkan tidak valid, maka akan muncul pesan “Silahkan masukkan <i>user name</i> dan <i>password</i> dengan benar”.	

2) Use Case Home.

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk Menampilkan menu-menu yang ada pada aplikasi.

Pre-condition : - . Admin dan user berada di halaman utama.

Post-condition : - .

Tabel 4.9. Skenario Use Case Home.

Aksi Aktor	Reaksi Sistem
1. Masuk kehalaman utama aplikasi	
	2. Menampilkan menu-menu aplikasi.
3. Memilih salah satu menu.	
	4. Jika menu yang dipilih valid, maka menu tersebut akan ditampilkan di form baru.
Jika menu yang diakses tidak valid, maka menu tidak dapat diakses.	

3) Use Case Kota/kab terdaftar

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk menampilkan kota/kab terdaftar.

Pre-condition : - .Admin berada di halaman Kota/kab terdaftar.

Post-condition : - Data dalam sistem akan diperbarui.

Tabel 4.10. Skenario Use Case Kota/kab terdaftar

Aksi Aktor	Reaksi Sistem
1. Memilih menu Kota/kab terdaftar	
	2. Menampilkan form Kota/kab terdaftar

4) Skenario Use Case Stasiun, rute, tiket dan pesanan.

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk melihat stasiun, rute, tiket, serta pesanan.

Pre-condition :- Admin berada di halaman Stasiun, rute tiket, serta pesanan .

Post-condition : -

Tabel 4.11. Skenario Use Case stasiun, rute, tiket dan pesanan.

Aksi Aktor	Reaksi Sistem
1. Menekan menu “stasiun, rute, tiket, atau pesanan”	

Aksi Aktor	Reaksi Sistem
	2. Menampilkan data yang dipilih (stasiun, rute, tiket atau pesanan) yang dimaksud dalam sistem.
	3. Menampilkan data yang dipilih sesuai data yang ada.

5) Use Case Beli tiket

Aktor : User

Deskripsi : Sebuah kegiatan membeli tiket.

Pre-condition : - Karyawan berada di halaman beli tiket .

Post-condition : - Data dalam sistem akan diperbarui.

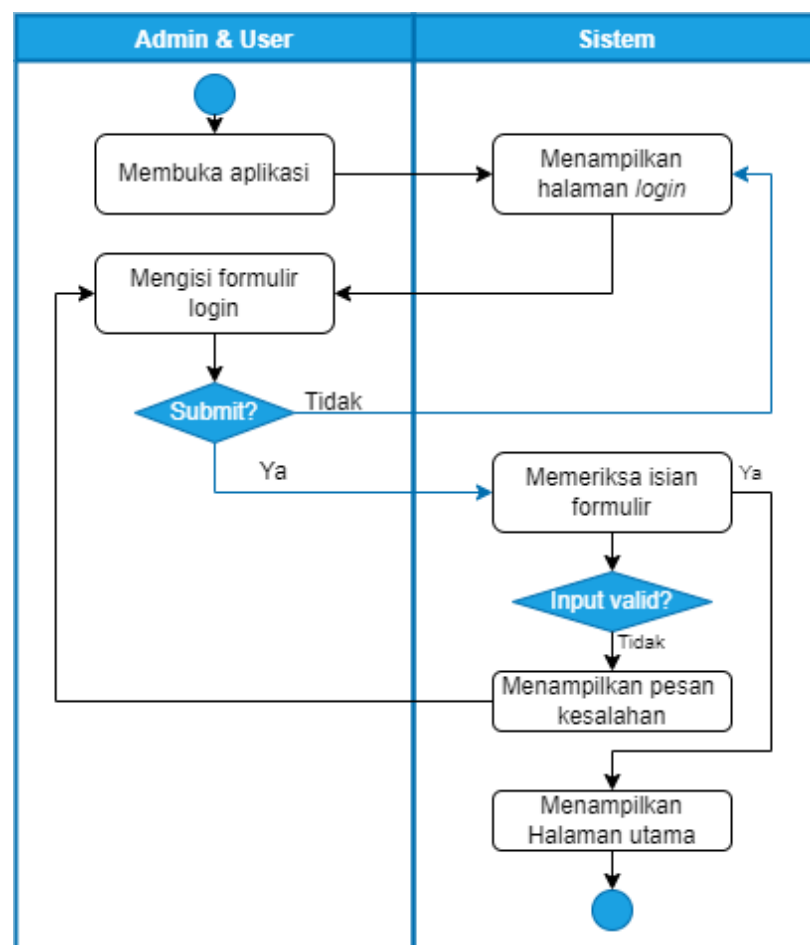
Tabel 4.12. Skenario Use Case Beli tiket

Aksi Aktor	Reaksi Sistem
1. Menekan Menu beli tiket	
2. Memilih tiket yang ingin dibeli	3. Memproses pembelian tiket.
4. Menekan tombol “Bayar dan Save Tiket” untuk menyimpan file tiket tersebut.	

1.3. 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 Login

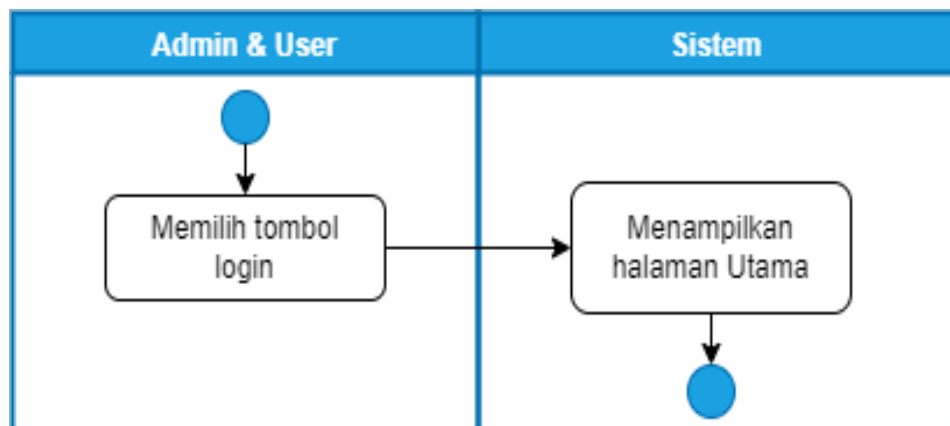


Gambar 4.2. Activity Diagram Login

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas login. Proses dimulai dengan admin atau user memilih tombol login, kemudian sistem menampilkan form isian yang kemudian admin atau user akan

mengisi form isian. Ketika sudah disimpan, sistem akan memperbarui data di database yang akan mengecek data yang dikirim, jika terdapat kesalahan, akan menampilkan pesan kesalahan, jika tidak, sistem menampilkan halaman utama.

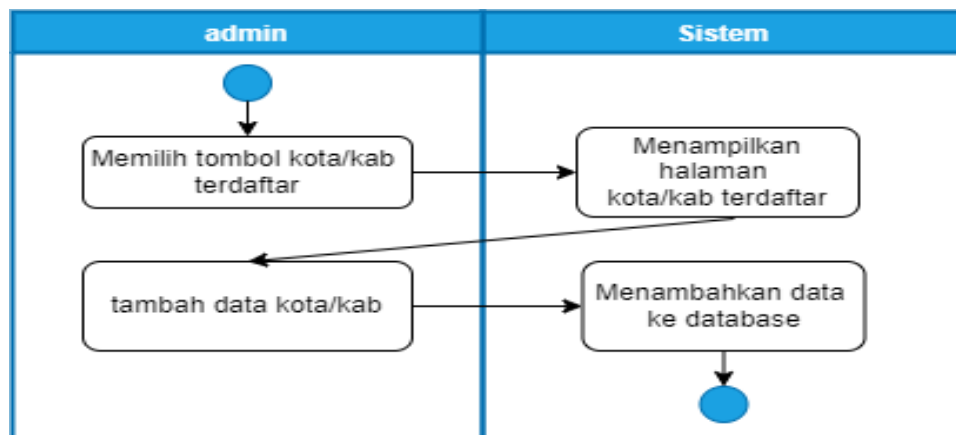
b) *Activity Diagram Home*



Gambar 4.3. *Activity Diagram Home*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas Home. Proses dimulai dengan admin atau karyawan memilih tombol login, kemudian system berhasil menampilkan halaman utama aplikasi.

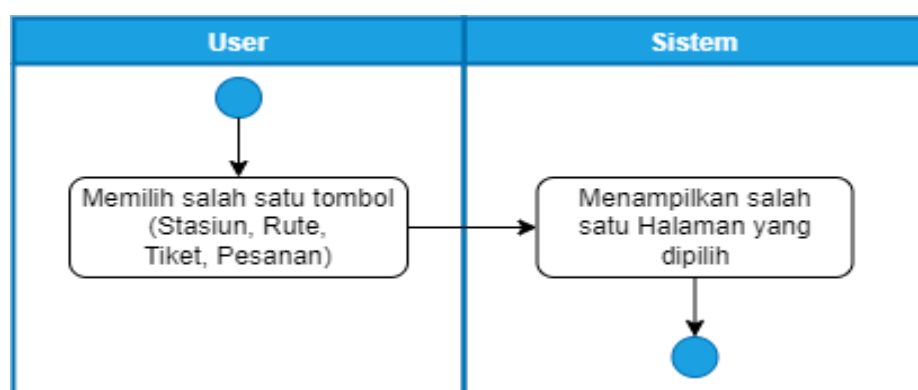
c) *Activity Diagram kota/kab terdaftar*



Gambar 4.4. *Activity Diagram kota/kab terdaftar*

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas kota/kab Proses dimulai dengan admin memilih tombol Kota/kab terdaftar, kemudian sistem menampilkan halaman kota/kab terdaftar Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

d) *Activity Diagram view (stasiun, rute, tiket dan pesanan)*

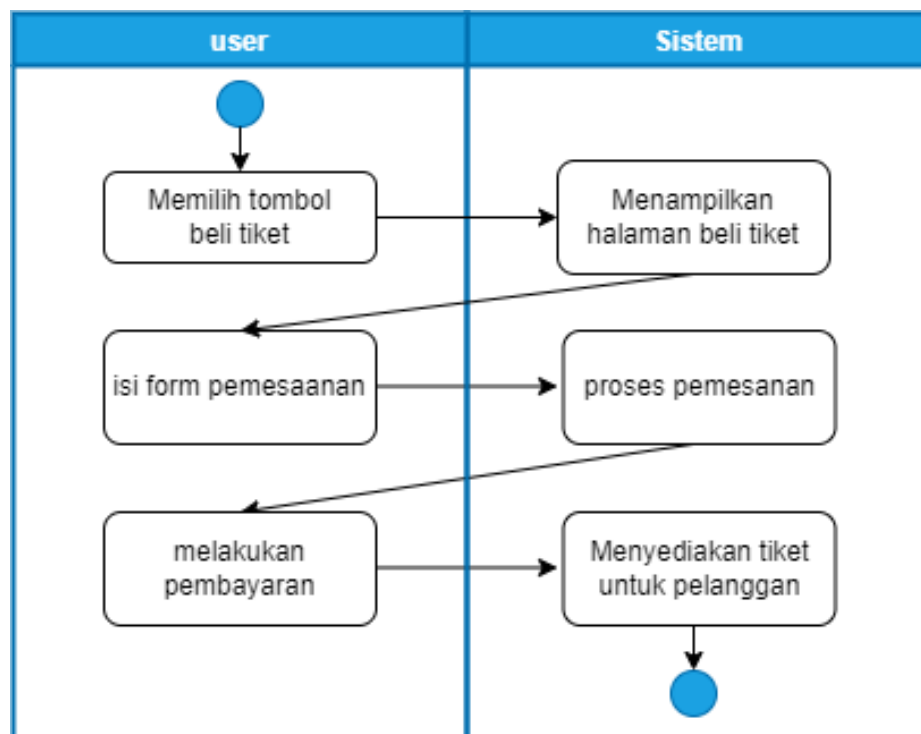


Gambar 4.5. *Activity Diagram view*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas view (stasiun, rute, tiket dan pesanan). Proses dimulai dengan user memilih salah

satu tombol (stasiun, rute, tiket dan pesanan), kemudian sistem menampilkan salah satu halaman yang dipilih.

e) *Activity Diagram* beli tiket



Gambar 4.6. *Activity Diagram* beli tiket

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas beli tiket. Proses dimulai dengan user memilih menu beli tiket kemudian sistem menampilkan halaman beli tiket lalu pelanggan mengisi isian pesanan, sistem proses pemesanan, pelanggan melakukan pembayaran kemudian sistem menyediakan tiket untuk pelanggan.

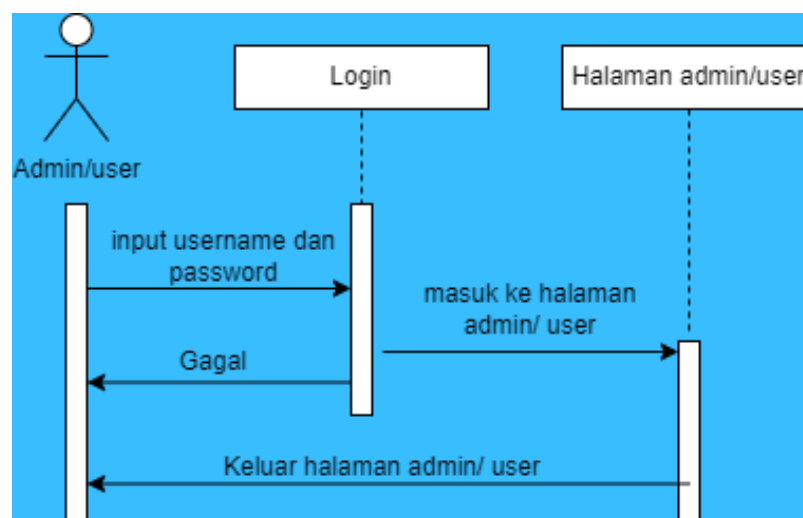
1.4. Sequence Diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang

dikirim dan diterima antar objek. Proses menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta method yang dimiliki kelas yang diinisialisasi menjadi objek yang sudah tergambar dalam class diagram.

Berdasarkan desain use case, terdapat beberapa use case yang prosesnya hampir sama satu sama lain. Untuk mempermudah pembahasan proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada sistem informasi yang dikembangkan:

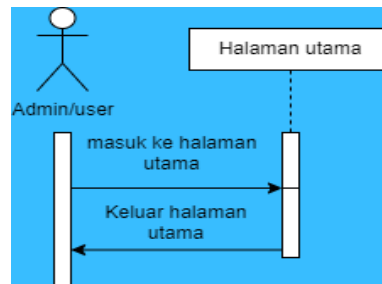
1) *Sequence Diagram: Login*



Gambar 4.7. *Sequence Diagram login*

Diagram Sequence diagram login Admin dan user merupakan penggambaran proses login yang dilakukan oleh admin dan user. Bila proses login benar admin dan user bisa masuk ke dalam halaman utama sebaliknya bila proses login gagal akan kembali ke menu login.

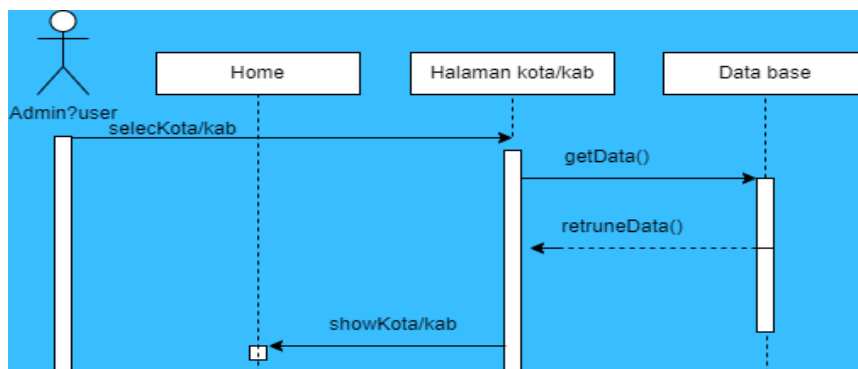
2) *Sequence Diagram: Home*



Gambar 4.8. *Sequence Diagram Home*

Diagram Sequence diagram home Admin dan user merupakan penggambaran proses masuk ke halaman utama. Bila proses login benar admin dan user bisa masuk ke dalam halaman utama dan menampilkan menu-menu yang ada pada aplikasi.

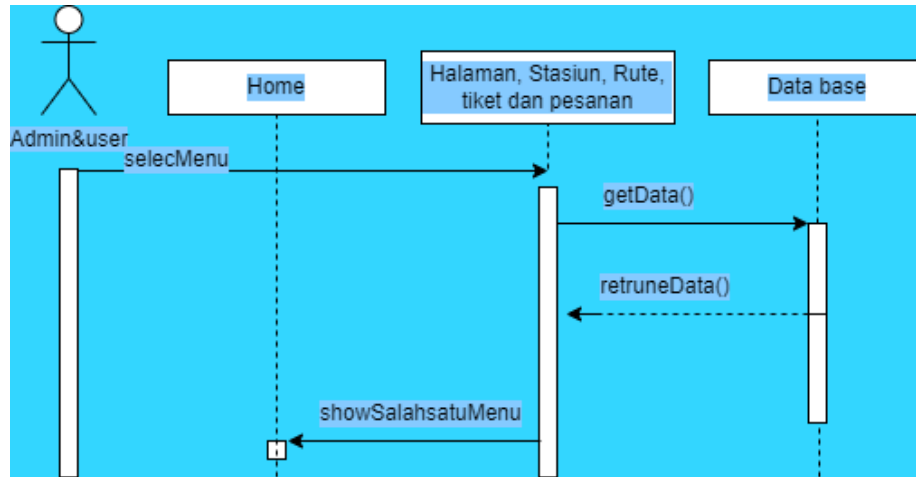
3) *Sequence Diagram*: Kota/kab terdaftar



Gambar 4.9. *Sequence Diagram Kota/kab terdaftar*

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Kota/kab. Admin dan user memilih data yang akan ditampilkan pada halaman kota/kab, dengan method selecKab/kota(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showKota/kab() akan menampilkan Kota/kab terdaftar.

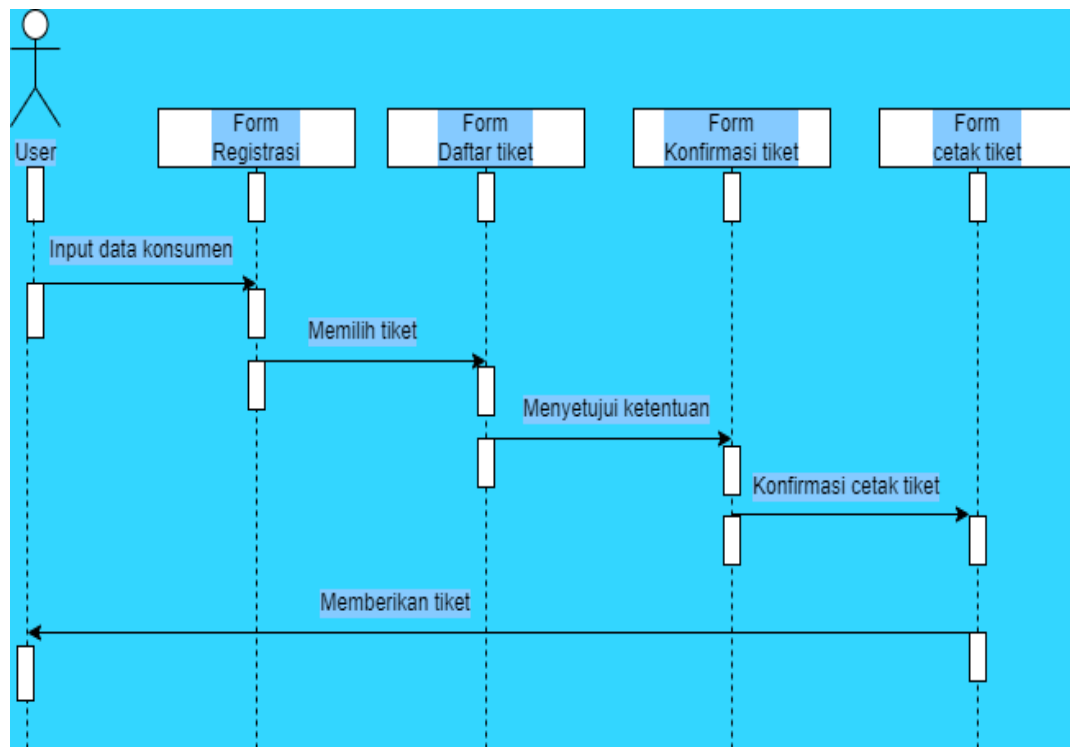
4) Sequence Diagram: View



Gambar 4.10. Sequence Diagram View

Diagram sekuen di atas menerangkan alur proses untuk aktivitas View (Stasiun, Rute, Tiket, dan Pesanan). Admin dan user memilih data yang akan ditampilkan pada halaman View (Stasiun, Rute, Tiket, dan Pesanan), dengan method `selecMenu ()`, kemudian method `getData()` digunakan untuk mengambil data yang akan ditampilkan ke database. Database akan mengembalikan data, kemudian method `showSalahSatuMenu()` akan menampilkan menu yang dipilih.

5) Sequence Diagram: Beli tiket



Gambar 4.11. *Sequence Diagram* Beli tiket

Diagram sekuen berikut menerangkan alur proses untuk aktivitas beli tiket. User sebagai aktor, input data konsumen lalu memilih tiket, kemudian menyetujui ketentuan lalu user melakukan konfirmasi cetak tiket, kemudian memberikan tiket.

1.5. Tabel Database

Adapun rancangan *database* dalam pembangunan sistem ini adalah terdiri dari beberapa tabel diantaranya yaitu :

- a. Tabel User. Nama *table*: user, Adapun struktur *table* user sebagai berikut.

Tabel 4.13. Tabel User

No	Nama Field	Tipe Data	Ukuran
1	id_user	Varchar	12
2	Userid	Varchar	12
3	Passid	Varchar	8
4	Nama	Varchar	20
5	Alamat	Varchar	50

b. Tabel Pembelian. Nama *table*: pembelian, Adapun struktur *table* pembelian sebagai berikut.

Tabel 4.14. Tabel Pembelian

No	Nama Field	Tipe Data	Ukuran
1	id_pembelian	Varchar	10
2	Id_Tiket	Varchar	10
3	Id_pelanggan	Varchar	10
4	Jumlah	Varchar	20
5	tanggal_pembelian	Date	-

c. Tabel *Supplier*. Nama *table*: *supplier*, Adapun struktur *table supplier* sebagai berikut.

Tabel 4.15. Tabel pelanggan

No	Nama Field	Tipe Data	Ukuran
1	Id_pelanggan	Varchar	10
2	Nama	Varchar	20
3	Alamat	Varchar	50
4	kode_pos	Varchar	20

d. Tabel Barang. Nama *table*: Tiket, Adapun struktur *table* tiket sebagai berikut.

Tabel 4.16. Tabel Tiket

No	Nama Field	Tipe Data	Ukuran
1	id_Tiket	Varchar	10
2	id_jenis	Varchar	10
3	nama_tiket	Varchar	20
4	harga_tiket	Varchar	20
5	stok_tiket	Varchar	20

e. Tabel Pesanan. Nama table: pesanan, Adapun struktur table pesanan sebagai berikut.

Tabel 4.17. Tabel Pesanan

No	Nama Field	Tipe Data	Ukuran
1	id_pesanan	Varchar	10
2	id_pelanggan	Varchar	10
3	id_tiket	Varchar	10
4	Jumlah	Varchar	20
5	tanggal_pesan	Date	-

f. Tabel Pembeli. Nama table: pembeli, Adapun struktur table pembeli sebagai berikut.

Tabel 4.18. Tabel Pembeli

No	Nama Field	Tipe Data	Ukuran
1	id_pembeli	Varchar	10
2	Nama	Varchar	20
3	Alamat	Varchar	50
4	NIK	Varchar	10

2. Perancangan Sistem

2.1. Halaman *Login*

1. Halaman *Login* Akun

Berikut adalah tampilan login akun pada pengingat jadwal kereta api. Halaman menu login merupakan halaman *login user*. *Login* merupakan tampilan yang ada pada sistem, yang digunakan user untuk masuk ke halaman selanjutnya.



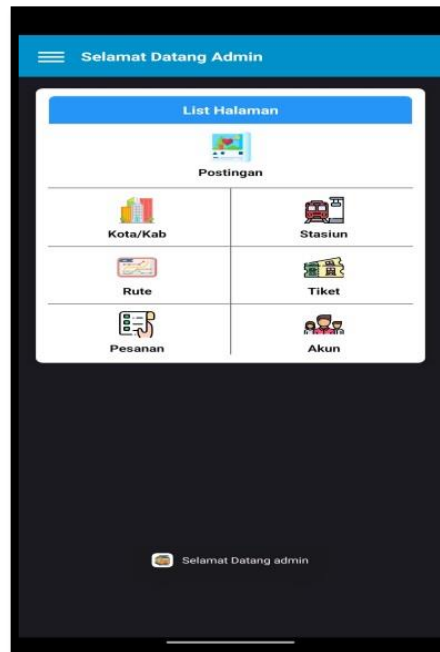
Gambar 4.12. Halaman *login*

2.2. Halaman Admin

1. Halaman *List* Admin

Berikut adalah tampilan list halaman admin pada pengingat jadwal kereta api. Halaman list admin merupakan halaman yang memiliki beberapa menu

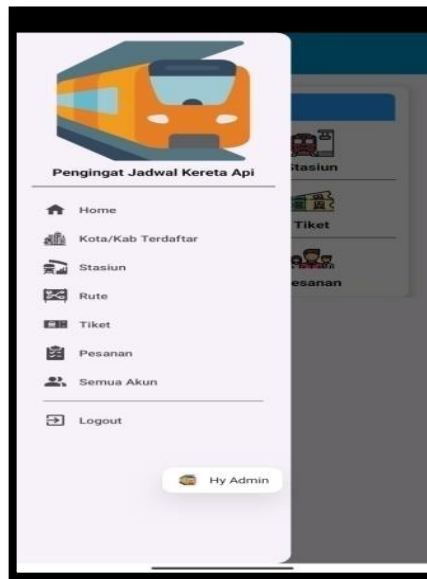
seperti, Kota/kab, Stasiun, Rute, tiket dan pesanan dan dilengkapi postingan penting terkait informasi kereta api.



Gambar 4.13. Halaman *List Admin*

2. Halaman Navigasi

Berikut adalah tampilan halaman navigasi pada pengingat jadwal kereta api. Halaman navigasi ini menampilkan Home, Kota,kab terdaftar, Stasiun, Rute, Tiket, Pesanan, Semua akun, dan *Logout*.



Gambar 4.14. Halaman Navigasi

3. Halaman Kota/kab terdaftar

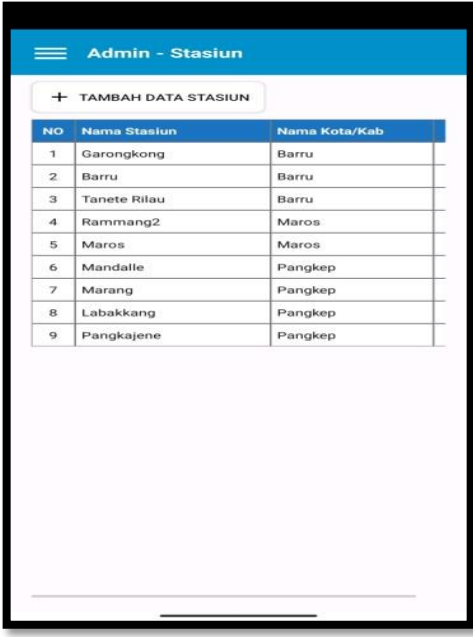
Berikut adalah tampilan halaman Kota/kab pada pengingat jadwal kereta api. Halaman navigasi ini menampilkan data tentang nama-nama kota/kab terdaftar.



Gambar 4.15. Halaman Kota/kab terdaftar

4. Halaman Stasiun

Berikut adalah tampilan halaman Stasiun pada pengingat jadwal kereta api. Halaman stasiun ini menampilkan nama stasiun serta nama kota dan kabupaten.



The screenshot shows a mobile application interface titled 'Admin - Stasiun'. Below the title is a button labeled '+ TAMBAH DATA STASIUN'. Below this is a table with three columns: 'NO', 'Nama Stasiun', and 'Nama Kota/Kab'. The table contains 9 rows of data.

NO	Nama Stasiun	Nama Kota/Kab
1	Garongkong	Barru
2	Barru	Barru
3	Tanete Rilau	Barru
4	Rammang2	Maros
5	Maros	Maros
6	Mandalle	Pangkep
7	Marang	Pangkep
8	Labakkang	Pangkep
9	Pangkajene	Pangkep

Gambar 4.16. Halaman Stasiun

5. Halaman Rute

Berikut adalah tampilan halaman rute pada pengingat jadwal kereta api. Halaman rute ini menampilkan Berangkat jam, Sampai jam, Dari stasiun, Sampai kota/kab, Sampai stasiun, serta harga tiket.



NO	Berangkat Jam	Sampai Jam	Dari Kota/Kab	Dari Stasiun	Sampai Kota/Kab	Sampai Stasiun	Harga	Tiket
1	12:00 Wita	14:00 Wita	Barru	Garongkong	Barru	Garongkong	Rp. 5.500	26
2	12:00 Wita	13:00 Wita	Barru	Tanete Rilau	Barru	Barru	Rp. 5.000	20
3	12:00 Wita	12:00 Wita	Barru	Tanete Rilau	Barru	Tanete Rilau	Rp. 5.000	20
4	17:00 Wita	18:00 Wita	Barru	Tanete Rilau	Pangkep	Mandalle	Rp. 5.000	20
5	17:00 Wita	19:00 Wita	Barru	Tanete Rilau	Pangkep	Marang	Rp. 5.000	20
6	17:00 Wita	20:00 Wita	Barru	Tanete Rilau	Pangkep	Labakkang	Rp. 5.000	20

Gambar 4.17. Halaman Rute

6. Halaman Tiket

Berikut adalah tampilan halaman tiket pada pengingat jadwal kereta api.

Halaman tiket ini menampilkan Halaman tiket, tambah tiket serta tambah tiket perminggunya.



Gambar 4.18. Halaman Tiket

7. Halaman Pesanan

Berikut adalah tampilan halaman pesanan pada pengingat jadwal kereta api. Halaman pesanan ini menampilkan Halaman pesanan secara detail.



NO	Waktu Pembelian	Tanggal	Waktu	Nama User	Nomor HP	Dari Kota/Kab
1	21 Agust 2023 09:02 Wita	22:40 Wita	22:40:00	Pelanggan Setia2	082345678912	Maros
2	21 Agust 2023 07:52 Wita	23:00 Wita	23:00:00	Pelanggan Setia2	082345678912	Maros
3	12 Sep 2023 00:17 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
4	12 Sep 2023 00:19 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
5	12 Sep 2023 00:19 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
6	12 Sep 2023 00:20 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
7	12 Sep 2023 00:23 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
8	12 Sep 2023 00:25 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros

Gambar 4.19. Halaman Pesanan

8. Halaman Akun

Berikut adalah tampilan halaman akun pada pengingat jadwal kereta api. Halaman akun ini menampilkan Halaman akun serta tambah akun.



Nama User	Alamat	Nomor HP	Username
Pelanggan Setia2	Jalan Ahmad Yani, Parepare	082345678912	user
Pelanggan Setia Dari Parepare	Jln. Ahmad Yani, UMPAR, Parepare	08276571536	user2
Pelanggan Tidak setia	Parepare	0894531243	pelanggan
muh arya	barru	085294445194	arya
artod	artod	852	artod

Gambar 4.20. Halaman Akun

2.3. Halaman User

1. Halaman utama *user*

Berikut adalah tampilan *user* pada pengingat jadwal kereta api. Halaman utama *user* merupakan halaman yang memiliki beberapa menu seperti, Jadwal anda, Semua jadwal serta notifikasi jadwal.



Gambar 4.21. Halaman utama *user*

2. Halaman Jadwal Anda

Berikut adalah tampilan jadwal anda pada pengingat jadwal kereta api. Halaman jadwal anda merupakan halaman yang memiliki rincian jadwal yang telah diatur dalam aplikasi pengingat jadwal.



Gambar 4.22. Halaman Jadwal Anda

3. Halaman Semua Jadwal

Berikut adalah tampilan semua jadwal pada pengingat jadwal kereta api.

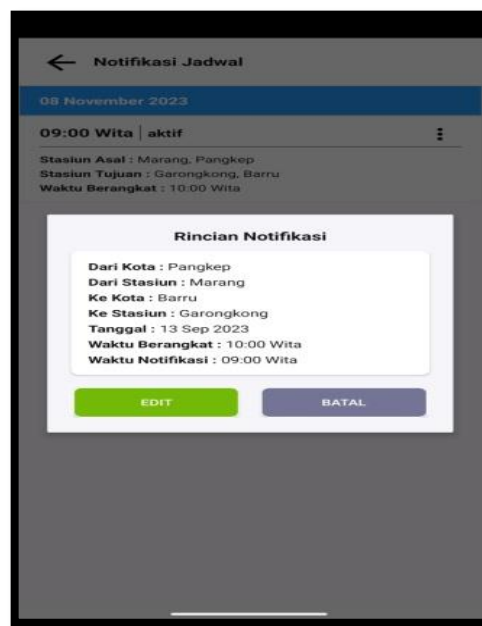
Halaman semua jadwal merupakan halaman yang menampilkan semua halaman jadwal.



Gambar 4.23. Halaman Semua Jadwal

4. Halaman Rincian Notifikasi

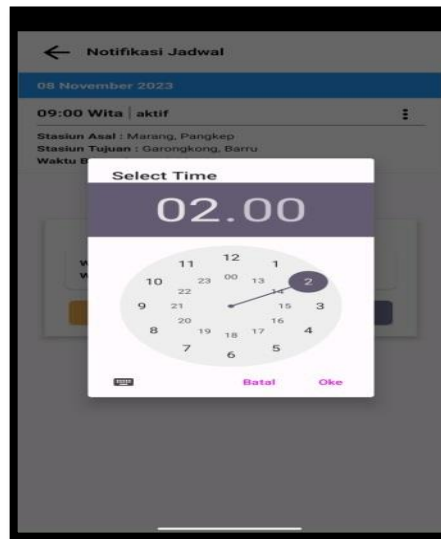
Berikut adalah tampilan rincian Notifikasi pada pengingat jadwal kereta api. Halaman rincian Notifikasi merupakan halaman yang menampilkan semua rincian notifikasi yang suda dibuat.



Gambar 4.24. Halaman Rincian Notifikasi

5. Halaman *Select Time*

Berikut adalah tampilan rincian *select time* pada pengingat jadwal kereta api. Halaman *select time* merupakan halaman yang menampilkan *select time* untuk mengatur waktu Notifikasi.



Gambar 4.25. Halaman *Select Time*

6. Halaman Notifikasi Jadwal

Berikut adalah tampilan rincian Notifikasi jadwal pada pengingat jadwal kereta api. Halaman Notifikasi jadwal merupakan halaman yang menampilkan status Notifikasi.



Gambar 4.26. Halaman Notifikasi Jadwal

7. Halaman Hasil Notifikasi

Berikut adalah tampilan hasil Notifikasi jadwal pada pengingat jadwal kereta api. Halaman hasil Notifikasi jadwal merupakan halaman yang menampilkan Notifikasi berupa pesan serta suara.



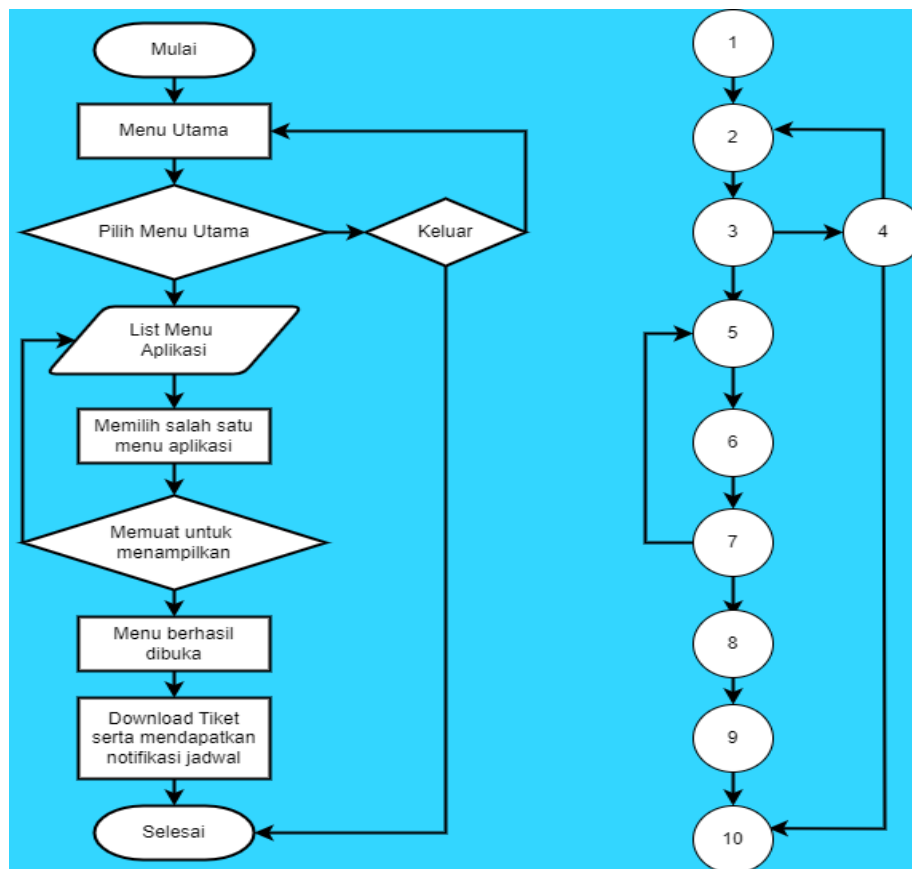
Gambar 4.27. Halaman Hasil Notifikasi

3. Pengujian Sistem

3.1 White Box.

White box testing atau pengujian kotak putih adalah salah satu metode pengujian perangkat lunak. Metode pengujian ini juga terkadang disebut juga *glass box* testing atau pengujian kotak kaca. Berbeda dengan *black box testing*, white box testing ini justru menguji perangkat lunak dari sisi internal tanpa memperhatikan fungsional seperti antarmuka perangkat lunak itu sendiri.

a) *White box* menu aplikasi pengingat jadwal kereta api.



Gambar 4.28. *White box* menu aplikasi pengingat jadwal kereta api

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 10 \quad E = 12 \quad R = 2$$

Penyelesaian :

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

$$= (12 - 10) + 2$$

$$= 4$$

Independent Path :

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10

Path 2 = 1 – 2 – 3 – 4 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10

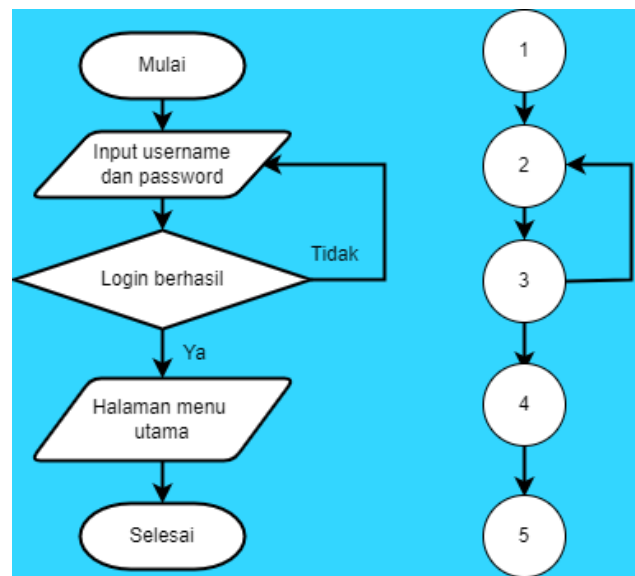
Path 3 = 1 – 2 – 3 – 4 – 10

Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 5 – 6 – 7 – 8 – 9 – 10

Tabel 4.19. *White Box* Perhitungan menu aplikasi pengingat jadwal kereta api

	1	2	3	4	5	6	7	8	9	10	E-1
1		1									1-1=0
2			1								1-1=0
3				1							1-1=0
4		1			1					1	3-1=2
5						1					1-1=0
6							1				1-1=0
7					1			1			2-1=1
8									1		1-1=0
9										1	1-1=0
10											0
SUM(E+1)											3+1=4

b) *White box* pada menu *login*



Gambar 4.29. *White box* pada menu *login*

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 5 \qquad E = 5 \qquad R = 2$$

Penyelesaian :

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

$$= (5 - 5) + 2$$

$$= 2$$

Independent Path :

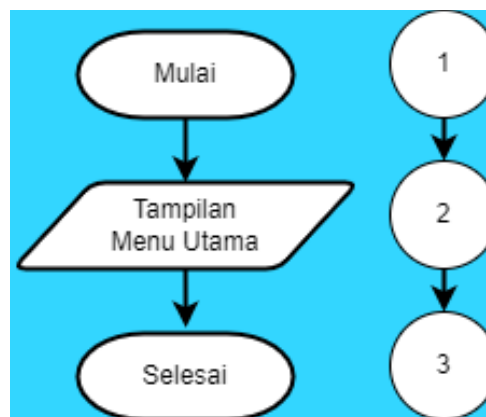
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 – 5

Tabel 4.20. *White Box* Perhitungan menu login

	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0
3		1		1		2-1=1
4					1	1-1=0
5						1-1=0
SUM(E+1)						1+1=2

c) *White box* pada menu utama



Gambar 4.30. *White box* pada menu utama

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan

berikut :

Diketahui :

$$N = 3 \qquad E = 3 \qquad R = 2$$

Penyelesaian :

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

$$= (3 - 3) + 2$$

$$= 2$$

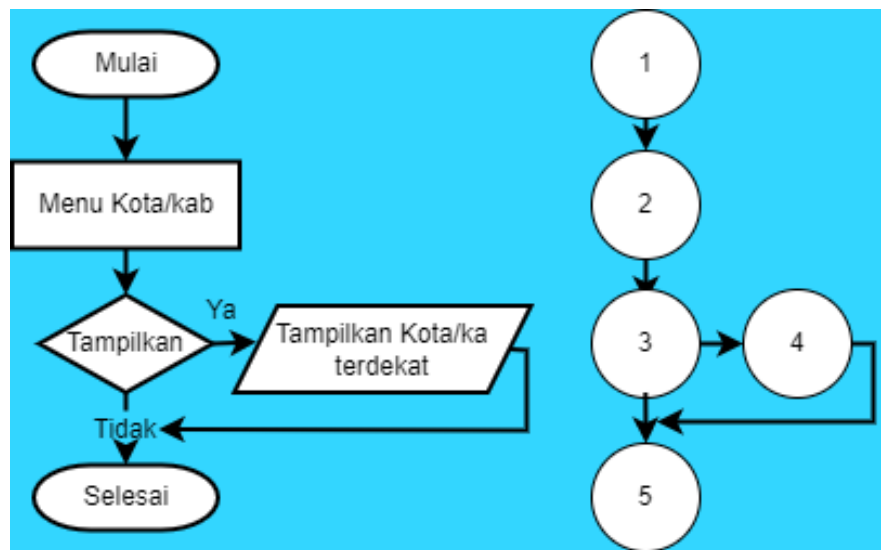
Independent Path :

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

Tabel 4.21. *White Box* Perhitungan menu *utama*

	1	2	3	E-1
1		1		1-1=0
2			1	1-1=0
3		1		1-1=0
SUM(E+1)				0+1=1

d) *White box* pada menu kota/kab



Gambar 4.31. *White box* pada menu Kota/kab

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 5 \qquad E = 5 \qquad R = 2$$

Penyelesaian :

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

$$= (5 - 5) + 2$$

$$= 2$$

Independent Path :

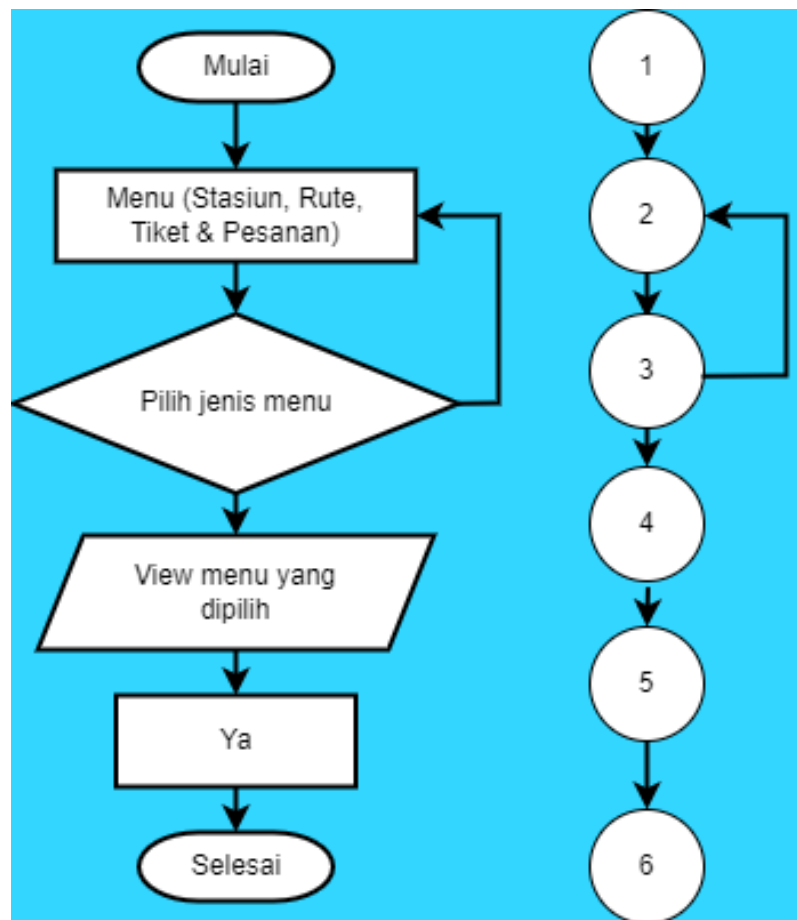
$$\text{Path 1} = 1 - 2 - 3 - 5$$

Path 2 = 1 – 2 – 3 – 4 – 5

Tabel 4.22. *White Box* Perhitungan menu Kota/kab

	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0
3		1		1		2-1=1
4					1	1-1=0
5						0
SUM(E+1)						1+1=2

e) *White box* pada menu view.



Gambar 4.32. White box pada menu view.

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6 \qquad E = 6 \qquad R = 2$$

Penyelesaian :

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

$$= (6 - 6) + 2$$

$$= 2$$

Independent Path :

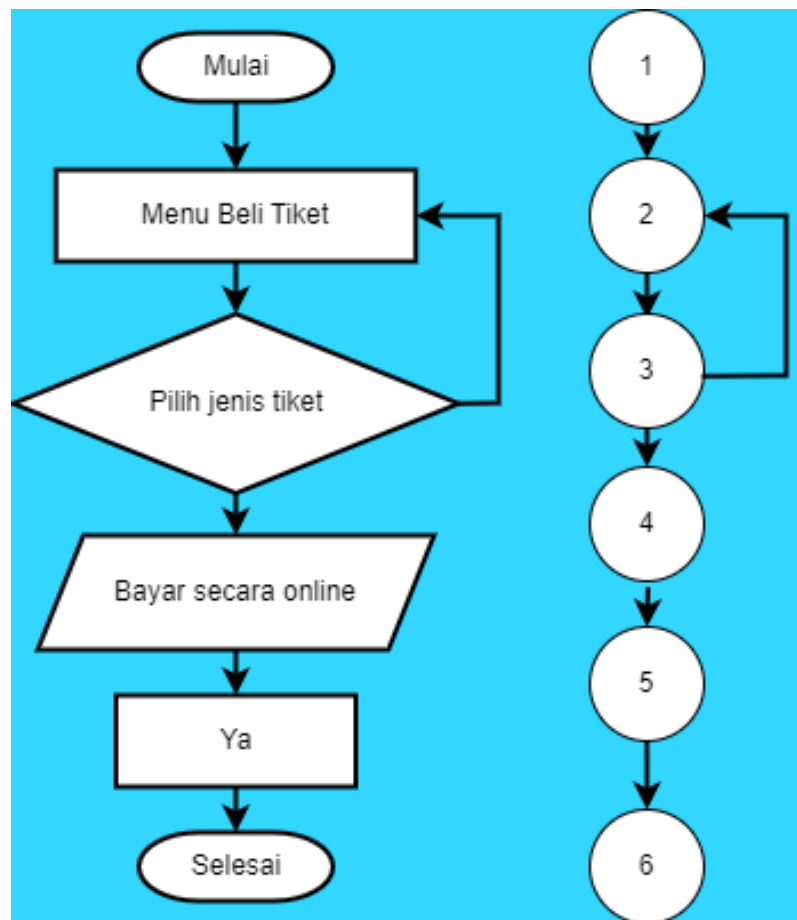
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 – 5 – 6

Tabel 4.23. *White Box* Perhitungan menu view.

	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3		1		1			2-1=1
4					1		1-1=0
5						1	1-1=0
6							0
SUM(E+1)							1+1=2

f) *White box* pada menu Beli tiket



Gambar 4.33. *White box* pada menu beli tiket

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan,. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6 \qquad E = 6 \qquad R = 2$$

Penyelesaian :

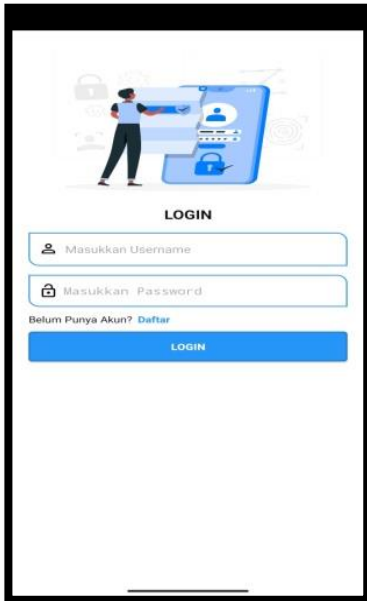
$$V(G) = (E - N) + 2$$

$$= (6 - 6) + 2$$

3.2 Black Box

Pengujian *Black Box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau *user*, pengujian ini tidak melihat dan menguji Logika program.

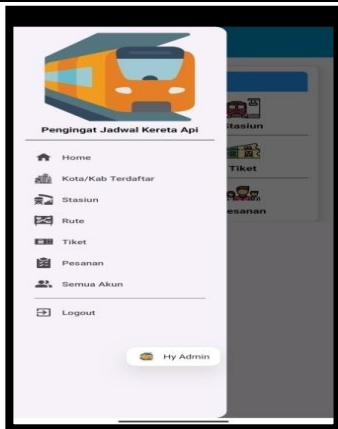
Tabel 4.25. Tabel Pengujian Login

Test Faktor	Hasil	Kesimpulan
Login	✓	Sukses, karena ketika <i>users</i> memasukkan <i>user ID</i> dan <i>Password</i> , akan masuk di halaman Utama
Screen Shot		
		

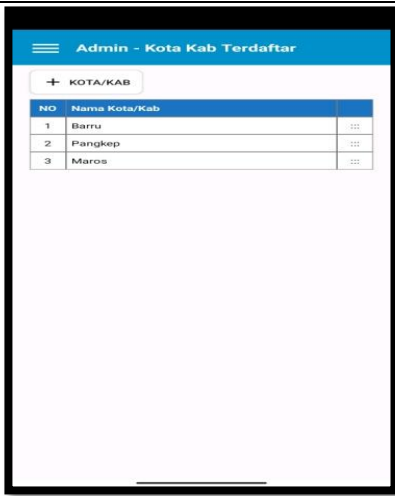
Tabel 4.26. Tabel Pengujian Halaman Utama Admin

Test Faktor	Hasil	Kesimpulan
Halaman Utama Admin	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka langsung masuk ke halaman utama admin
Screen Shot		
		

Tabel 4.27. Tabel Pengujian Halaman Navigasi

Test Faktor	Hasil	Kesimpulan
Halaman Navigasi	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman navigasi
Screen Shot		
		

Tabel 4.28. Tabel Pengujian Halaman Kota/kab

Test Faktor	Hasil	Kesimpulan
Halaman Kota/kab	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman kota/kab
Screen Shot		
		

Tabel 4.29. Tabel Pengujian Halaman Stasiun

Test Faktor	Hasil	Kesimpulan
Halaman Stasiun	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman stasiun
Screen Shot		



Tabel 4.30. Tabel Pengujian Halaman Rute

Test Faktor		Hasil	Kesimpulan
Halaman Rute		✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman rute

Screen Shot

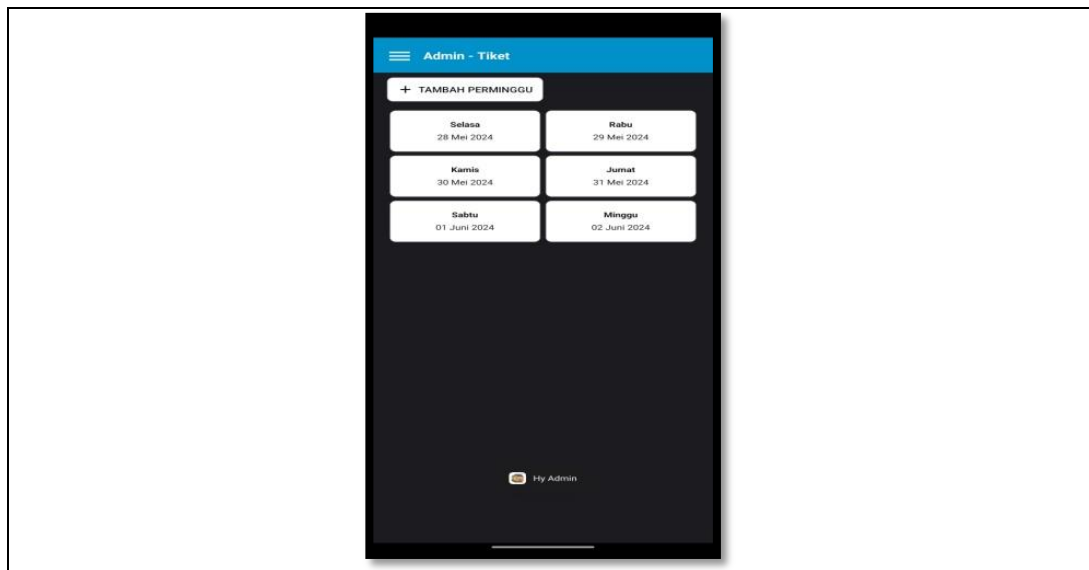
☰Admin - Rute

+ TAMBAH

NO	Berangkat Jam	Sampai Jam	Dari Kota/Kab	Dari Stasiun	Sampai Kota/Kab	Sampai Stasiun	Harga	Tkt
1	12:00 Wita	14:00 Wita	Barru	Garongkong	Barru	Garongkong	Rp. 5.500	26
2	12:00 Wita	13:00 Wita	Barru	Tanete Rilau	Barru	Barru	Rp. 5.000	20
3	12:00 Wita	12:00 Wita	Barru	Tanete Rilau	Barru	Tanete Rilau	Rp. 5.000	20
4	17:00 Wita	18:00 Wita	Barru	Tanete Rilau	Pangkep	Mandalle	Rp. 5.000	20
5	17:00 Wita	19:00 Wita	Barru	Tanete Rilau	Pangkep	Marang	Rp. 5.000	20
6	17:00 Wita	20:00 Wita	Barru	Tanete Rilau	Pangkep	Labakkang	Rp. 5.000	20

Tabel 4.31. Tabel Pengujian Halaman Tiket

Test Faktor	Hasil	Kesimpulan
Halaman Tiket	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman tiket
Screen Shot		



Tabel 4.32. Tabel Pengujian Halaman Pesanan

Test Faktor	Hasil	Kesimpulan
Halaman Pesanan	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman pesanan
Screen Shot		

NO	Waktu Pembelian	Tanggal	Waktu	Nama User	Nomor HP	Dari Kota/Kab
1	21 Agust 2023 09:02 Wita	22:40 Wita	22:40:00	Pelanggan Setia2	082345678912	Maros
2	21 Agust 2023 07:52 Wita	23:00 Wita	23:00:00	Pelanggan Setia2	082345678912	Maros
3	12 Sep 2023 00:17 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
4	12 Sep 2023 00:19 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
5	12 Sep 2023 00:19 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
6	12 Sep 2023 00:20 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
7	12 Sep 2023 00:23 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros
8	12 Sep 2023 00:25 Wita	06:00 Wita	06:00:00	Pelanggan Setia2	082345678912	Maros

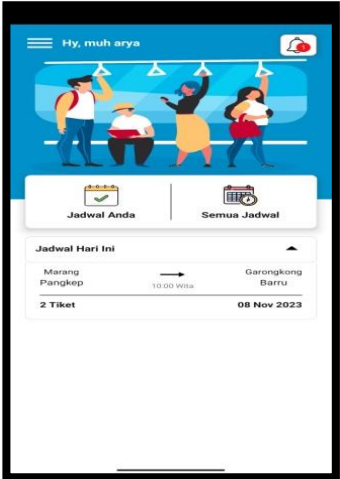
Tabel 4.33 Tabel Pengujian Halaman Akun

Test Faktor	Hasil	Kesimpulan
Halaman Akun	✓	Sukses, karena ketika <i>admin</i> berhasil <i>login</i> sebagai admin maka bisa akses halaman akun
Screen Shot		



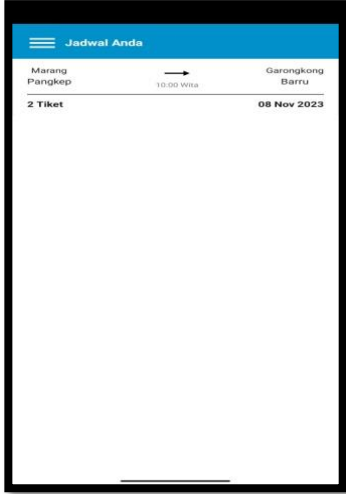
Admin - Semua Akun			
+ TAMBAH			
Nama User	Alamat	Nomor HP	Username
Pelanggan Setia2	Jalan Ahmad Yani, Parepare	082345678912	user
Pelanggan Setia Dari Parepare	Jln. Ahmad Yani, UMPAR, Parepare	08276571536	user2
Pelanggan Tidak setia	Parepare	0894531243	pelanggan
muh arya	barru	085294445194	arya
artod	artod	852	artod

Tabel 4.34. Tabel Pengujian Halaman Utama User

Test Faktor	Hasil	Kesimpulan
Utama User	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman utama user
Screen Shot		
		

Tabel 4.35 Tabel Pengujian Halaman Jadwal anda

Test Faktor	Hasil	Kesimpulan
Halaman Jadwal anda	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman

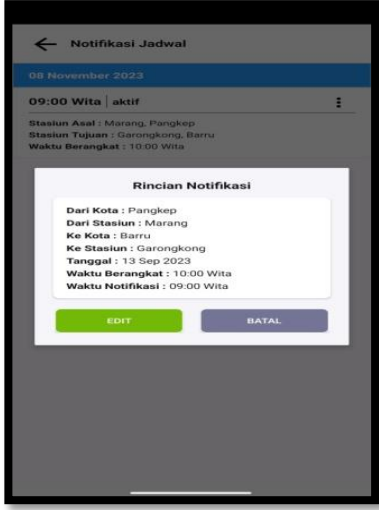
	jadwal anda
Screen Shot	
	

Tabel 4.36. Tabel Pengujian Halaman Semua Jadwal

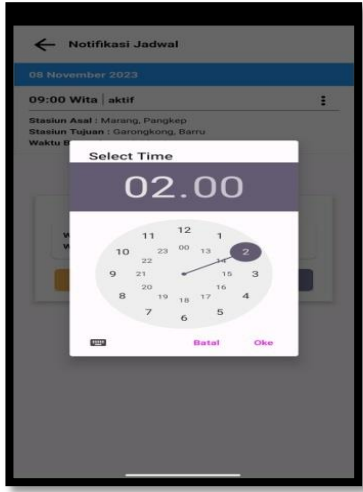
Test Faktor	Hasil	Kesimpulan
Semua Jadwal	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman semua jadwal
Screen Shot		
		

Tabel 4.37 Tabel Pengujian Halaman Rincian Notifikasi

Test Faktor	Hasil	Kesimpulan
Halaman Rincian Notifikasi	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman

	Rincian Notifikasi
Screen Shot	
	

Tabel 4.38 Tabel Pengujian Halaman *Select Time*

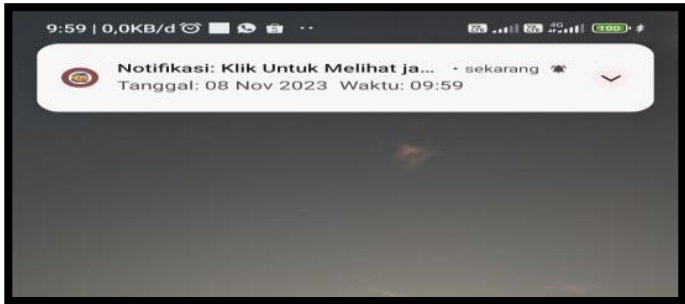
Test Faktor	Hasil	Kesimpulan
<i>Select Time</i>	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman <i>Select Time</i>
Screen Shot		
		

Tabel 4.39 Tabel Pengujian Halaman Notifikasi Jadwal

Test Faktor	Hasil	Kesimpulan
Halaman Notifikasi Jadwal	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman

	Notifikasi Jadwal
Screen Shot	
	

Tabel 4.40 Tabel Pengujian Halaman Hasil Notifikasi

Test Faktor	Hasil	Kesimpulan
Hasil Notifikasi	✓	Sukses, karena ketika <i>user</i> berhasil <i>login</i> sebagai user maka bisa akses halaman Hasil Notifikasi
Screen Shot		
		

4. Maintenance

Pada Tahap ini sistem yang telah dibangun sudah berhasil untuk digunakan dan dilakukan pemeliharaan sistem termasuk memperbaiki perubahan sistem kembali atau update sistem sesuai dengan kebutuhan perusahaan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dan pada pembahasan bab sebelumnya, maka dapat diperoleh kesimpulan bahwa aplikasi pengingat jadwal keberangkatan kereta api berbasis multimedia dibuat menggunakan bahasa pemrograman kotlin *tools* Android Studio dan *database MySql* dengan metode *push* notifikasi yang merupakan penyampaian informasi dari sebuah aplikasi *software*. Pengembangan aplikasi yang efektif memerlukan fokus pada pengalaman pengguna, personalisasi, dan integrasi multimedia yang relevan, serta fitur yang memungkinkan pengguna untuk menyesuaikan pengingat dan notifikasi sesuai kebutuhan mereka. Pengguna juga akan mengalami peningkatan kemudahan dan kenyamanan dalam mengingat jadwal keberangkatan. Aplikasi menyediakan akses informasi yang cepat dan akurat, notifikasi tepat waktu, sekaligus dapat melakukan pembelian tiket, serta fitur multimedia yang mendukung pemahaman informasi tambahan.

B. Saran

Penelitian selanjutnya diharapkan dapat mengembangkan lagi sistem yang berbasis android dengan menambahkan fitur-fitur terbaru. Diharapkan kepada peneliti selanjutnya menciptakan sebuah sistem agar pelanggan dapat dengan mudah melihat informasi dalam bentuk video maupun animasi agar lebih menarik.

DAFTAR PUSTAKA

- Ade Hastuty Hasyim. (2021). *Dasar Pemrograman* . (M. Basri, Penyunt.) Makassar, Sulawesi Selatan: CV. Bangun Bumitama .
- Chen, S. C. (2022). Multimedia Research Toward the Metaverse. *IEEE Multimedia*, 29(1), 125–127. <https://doi.org/10.1109/MMUL.2022.3156185>
- Elayan, O. N., & Mustafa, A. M. (2021). Android malware detection using deep learning. *Procedia Computer Science*, 184, 847–852. <https://doi.org/10.1016/j.procs.2021.03.106>
- Faisal, N., & Sunardi, S. (2020). Simulasi Aplikasi Monitoring Kereta Api Berbasis Android. *Jurnal Perkeretaapian Indonesia (Indonesian Railway Journal)*, 4(1). <https://doi.org/10.37367/jpi.v4i1.112>
- Fitria, O., Hasanah, N., Pd, M., & Untari, R. S. (n.d.). *Buku ajar rekayasa perangkat lunak Diterbitkan oleh umsida press universitas muhammadiyah sidoarjo 2020*.
- Frauenberger, C. (2019). Entanglement HCI the next wave? *ACM Transactions on Computer-Human Interaction*, 27(1). <https://doi.org/10.1145/3364998>
- Hannan, A. A. (2023). Perancangan Sistem Informasi Pemesanan Tiket Kereta Api Layanan. *Applied Business and Adminisration Journal*, 2(1), 36–54.
- Juansyah, A. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System (A-GPS) Dengan Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 1(1), 1–8.
- Mayrhofer, R., Stoep, J. Vander, Brubaker, C., & Kravovich, N. (2021). The Android Platform Security Model. *ACM Transactions on Privacy and Security*, 24(3). <https://doi.org/10.1145/3448609>
- Setiyani, L. (2021). *Implementasi Cybersecurity pada Operasional Organisasi*. <https://e-journal.rosma.ac.id/index.php/inotek/article/view/183>
- Sri, A., Dosen, L., Tarbiyah, J., Sultan, S., & Kendari, Q. (2013). *Pembelajaran multimedia* (Vol. 6, Issue 2).