BAB I

PENDAHULUAN

A. Latar Belakang

Absensi merupakan salah satu bentuk pengawasan atau kontrol terhadap tenaga pendidik maupun pegawai yang dilakukan di dunia pendidikan. Sistem absensi yang umumnya terdapat pada suatu instansi maupun sekolah dengan menggunakan *id card*, *finger print* atau secara manual, dengan *menu*lis nama atau membuat paraf. Metode manual masih memiliki banyak kekurangan seperti pemalsuan paraf, kehilangan id card, biaya yang tidak sedikit dan mengantri yang dapat membuang waktu untuk menyatakan kehadiran.

Pada era modern ini tidak *menu*tup kemungkinan aplikasi pada ponsel cerdas dapat digunakan untuk sistem absensi suatu instansi, dalam hal ini sekolah. Dikarenakan *smartphone* dapat dikatakan sebagai kebutuhan sekunder untuk setiap orang karena fungsinya yang jauh lebih praktis dan efisien serta dengan kemajuan teknologi sudah seharusnya setiap sekolah mempunyai aplikasi tentang informasi kesiswaan khususnya presensi. Salah satu cara dengan *QR Code* bagi siswa yang hadir untuk memindai melalui aplikasi smartphone. Kode bersama dengan identitas siswa yang diambil oleh aplikasi akan mengkonfirmasi kehadiran siswa. Sistem *QR Code* dapat menghemat waktu dan menghemat banyak kertas. Selain itu dosen ketika selesai

mengajar tidak perlu menyimpan catatan apa pun karena data disimpan langsung dalam d*atabase* (Nair, 2019).

Berdasarkan studi pendahuluan yang telah dilakukan oleh peniliti tanggal 28 juli 2024 metode absensi dalam pembelajaran pada SMA Negri 5 Parepare saat ini menggunakan absensi dengan cara manual, diketahui masih menggunakan absensi manual. Metode absensi manual mempunyai kelemahan dengan adanya kecurangan dimana siswa dapat menitipkan absensi yang dilakukan oleh antar siswa dalam kelas. Kedisiplinan siswa tidak terpantau dengan baik saat proses pembelajaran. Selain itu, terjadi penumpukan data arsip absensi karena pendataan dilakukan menggunakan media kertas. Sehingga dari uraian tersebut peneliti tertarik untuk melakukan penelitian tentang sistem informasi absensi siswa menggunakan *QR code* yang dapat memudahkan para staf tata usaha dalam merekap data absensi siswa. Adapun sistem tersebut nantinya melibatkan guru, siswa, dan staf Tata Usaha. Guru dapat melihat informasi kehadiran siswa yang mengikuti peroses pembelajaran. Bagi mahasiswa dapat menginputkan kehadiran menggunakan sistem tersebut. Sedangkan bagi staf tata usaha dapat memudahkan dalam melakukan rekapitulasi kehadiran siswa guna pelaporan.dan memudahkan guru dalam mengolah nilai setiap siswa.

Berdasarkan permasalahan di atas dan semakin berkembangnya teknologi maka akan dibuat suatu aplikasi penerapan *QRCODE* sebagai media pelayanan untuk absensi pada siswa SMA Negeri 5 Parepare berbasis *android*.

B. Rumusan Masalah

Dengan mengacu pada latar belakang yang sudah di jelaskan di atas maka disusunlah rumusan masalah yang akan dibahas dalam penelitian ini adalah bagaimana merancang sebuah aplikasi Absensi Mahasiswa Menggunakan *QR Code* Pada SMA Negri 5 Parepare agar dapat memudahkan siswa dan guru dalam melakukan proses pengelolaan absensi yang *aplikatif*, dan *user friendly*?

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. Aplikasi ini berbasis *android*.
2. Aplikasi ini berjalan online.
3. Aplikasi ini sebagai alternatif solusi untuk absensi.
4. Aplikasi ini sebagai alat bantu dan tidak bersifat mutlak.

D. Tujuan Penelitian

Berdasarkan permasalahan diatas, maka tujuan yang ingin dicapai dalam penelitian ini iyalah membangun dan menerapkan Sistem Informasi Presensi Siswa menggunakan *QR Code* yang memudahkan dalam pengelolaan data presensi dan meminimalisir kecurangan siswa dalam melakukan presensi pada SMA Negri 5 Parepare sehingga tercipta sistem informasi yang *aplikatif, user friendly*.

E. Manfaat Penelitian

Berdasarkan permasalahan diatas, maka manfaat yang ingin dicapai dalam penelitian ini Memudahkan dalam melakukan pengelolaan data presensi siswa dan Memudahkan dalam memperoleh informasi terkait jumlah siswa yang hadir dalam pembelajaraan.

F. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan skripsi ini adalah:

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. Kajian Teori

1. SMA Negri 5 Parepare

SMA Negeri 5 Parepare atau dikenal juga dengan nama Smaeli adalah salah satu dari SMA Negeri di Sulawesi Selatan. Sekolah ini menerapkan sistem Boarding School atau Sekolah Berasrama. SMA Negeri 5 Parepare berada di Kota Parepare, Sulawesi Selatan.

Gambar 2 1 Log SMA Negri 5 Parepare

(Sumber : https://p2k.stekom.ac.id/ensiklopedia/SMA_Negeri_5_Parepare)

a. ***Visi***

Berprestasi Unggul, Beriman, Berkarakter, dan Berbudaya Lingkungan

b. ***Misi***

Untuk mencapai visi tersebut, SMAN 5 Palangka Raya mengemban misi sebagai berikut:

1. Menjalankan ibadah sesuai dengan agama yang dianutnya;
2. Melaksanakan Pembelajaran Paikem;
3. *Menu*mbuhkan semangat berprestasi;
4. Membimbing dan mengembangkan bakat dan minat peserta didik;
5. Melaksanakan Program Ekstrakurikuler yang Berprestasi
6. Menerapkan manajemen berbasis sekolah;
7. Meningkatkan budaya lingkungan.

c. **Tujuan**

Tujuan sekolah sebagai bagian dari tujuan pendidikan nasional adalah meningkatkan kecerdasan, pengetahuan, kepribadian, akhlak mulia, serta keterampilan untuk hidup mandiri dan mengikuti pendidikan lebih lanjut. Tujuan ini sejalan dengan semboyan sekolah yang merupakan *school branding*, yakni Pembelajaran Efektif, Lulusan Berkualitas, Unggul Berkarakter, dan Sekolah Inovasi (PLUS).

Secara rinci tujuan SMA Negeri 5 Palangka Raya adalah:

1. Melaksanakan proses belajar-mengajar secara efektif dan efisien berdasarkan semangat meraih keunggulan;
2. Menyediakan sarana prasarana pendidikan yang memadai;
3. Meningkatkan kinerja komponen sekolah (kepala sekolah, tenaga pendidik, karyawan, peserta didik, dan komite sekolah) untuk bersama-sama melaksanakan kegiatan yang inovatif sesuai dengan tugas pokok dan fungsi (tupoksi) masing-masing;
4. Melaksanakan program ekstrakurikuler bagi seluruh peserta didik sesuai dengan bakat dan minat sebagai salah satu sarana pengembangan diri;
5. Meningkatkan kualitas lulusan yang memiliki sikap, pengetahuan, dan keterampilan yang seimbang;
6. Meningkatkan jumlah lulusan yang melanjutkan ke perguruan tinggi;
7. Menyusun dan melaksanakan tata tertib dan segala ketentuan yang mengatur kegiatan warga sekolah;
8. Meningkatkan kualitas semua sumber daya manusia, baik tenaga pendidik, tenaga kependidikan maupun peserta didik yang kompetitif.

2. Sistem Informasi

Sistem informasi merupakan gabungan dari 2 kata yaitu sistem dan informasi.

a. Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu. (Fatillah, A. 2020) Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

1) Komponen Sistem (*Component*) Komponen-komponen suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

2) Batasan Sistem (*Boundary*) Batas Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3) Subsistem Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing

4) Lingkungan Luar Sistem (*Environment*) Adalah diluar batas dari sistem yang mempengaruhi operasi sistem.

5) Penghubung Sistem (*Interface*) Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

6) Masukan Sistem (*Input*) Adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (*signal input*).

7) Keluaran Sistem (*Output*) Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

8) Pengolah Sistem (*Process)* Suatu sistem dapat mempunyai pengolah atau sistem itu sendiri sebagai pengolahnya.

9) Sasaran Sistem (*Object)* Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem (Yunita & Devitra, 2019).

b. Informasi

Data dan informasi merupakan dua konsep yang amat penting untuk dipahami dan tidak dapat dipisahkan satu sama lain dalam konteks sistem informasi. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah kejadian yang terjadi pada saat tertentu.

Informasi yang berkualitas memiliki 3 kriteria, yaitu :

1) Akurat *(accurate)*

Informasi harus bebas dari kesalahan, tidak bias maupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2) Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan

3) Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. (Yunita & Devitra, 2017)

c. Sistem informasi

Pengertian di atas mengandung arti bahwa, Sistem informasi merupakan satuan komponen yang saling berhubungan yang mengumpulkan (atau mendapatkan kembali), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi. (Yunita & Devitra, 2019)

Manfaat adanya sistem informasi dalam suatu instansi yaitu :

1) Menyajikan suatu pusat informasi guna yaitu mendukung pengambilan suatu keputusan.

2) Menyajikan informasi yang guna mendukung operasi harian.

3) Menyajikan sebuah informasi yang berkenaan dengan kepengurusan.

3. Absensi

Absensi *menu*rut Santoso dan Yulianto (2020:67) Absensi adalah sebuah pembuatan data untuk daftar kehadiran yang biasa digunakan bagi sebuah lembaga atau instansi yang sangat perlu membutuhkan sistem seperti ini. Absensi *menu*aikan sebuah sistem yang harus dipergunakan sebagai konsep sistem absensi, disaat sistem membutuhkan sebuah data maka sistem akan dijadikan sebagai aplikasi yang sanggup menjalankan dan membuat data absensi tersebut.

Secara umum, jenis-jenis absensi *menu*rut cara penggunaannya dapat dikelompokkan menjadi dua, yaitu :

a. Absensi Manual, yang merupakan cara penulisan kehadiran dengan cara menggunakan pena berupa tanda tangan

b. Absensi non manual, yang merupakan cara penulisan kehadiran dengan menggunakan alat yang terkomputerisasi, bisa menggunakan *qr code* kartu RFID ataupun *fingerprint (*Sukowati, A. I., Fauziah, H., & Purwanto, I. 2017).

4. Bar Code

Bar code secara harfiah berasal dari bahasa inggris, bar artinya batang sedangkan *code* artinya kode/sandi. *Bar code* dapt diartikan sebagai sekumpulan *code* yang berbentuk garis-garis dan spasi, dimana masing-masing ketebalan setiap garis dan spasinya berbeda sesuai dengan isi *code* tersebut. Bar*code* adalah informasi terbacakan mesin (*machine readable*) dalam format visual yang tercetak. Bar*code* dibaca dengan menggunakan sebuah alat baca bar*code* atau lebih dikenal dengan Bar*code Scanner*. Merk Bar*code Scanner* yang terkenal diantaranya *DATALOGIC PSC,HHP, CHIPERLAB, ZEBEX,* dan lainlain. (Pulungan, A. 2019.)

5. *Quick Respone Code* kode QR

QR Code adalah bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi. Penggunaan kode QR sudah sangat lazim di Jepang hal ini dikarenakan kemampuannya menyimpan data yang lebih besar daripada kode batang sehingga mampu mengkodekan informasi dalam bahasa Jepang sebab dapat menampung huruf kanji. Kode QR telah mendapatkan standardisasi internasional dan standardisasi dari Jepang berupa ISO/IEC18004 dan JIS-X-0510 dasasan telah digunakan secara luas melalui ponsel di Jepang Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah Kode QR atau biasa dikenal dengan *QR Code* adalah bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi. Penggunaan kode QR sudah sangat lazim di Jepang Hal ini dikarenakan kemampuannya menyimpan data yang lebih besar daripada kode batang sehingga mampu mengkodekan informasi dalam bahasa Jepang sebab dapat menampung huruf

kanji. Kode QR telah mendapatkan standardisasi internasional dan standardisasi dari Jepang berupa ISO/IEC18004 dan JIS-X-0510 dasasan telah digunakan secara luas melalui ponsel di Jepang Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR merupakan singkatan dari quick response atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, kode QR mampu menyimpan informasi secara *horizontal* dan *vertikal,* oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak daripada kode batang (Joseph, A., & Komputer, D. T. 2019.).

6. Android

Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform yang terbuka atau sering juga disebut open source bagi para pengembang (*developer*) untuk membangun aplikasi ciptaan mereka sendiri. Awalnya, Google Inc. membeli *Android* Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan *Android*, dibentuklah Open *Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan

telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm*, *T-Mobile* dan Nvidia.

a. Arsitektur *Android*

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

1) *Applications* dan *Widgets* adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdaat aplikasi inti termasuk *klien email*, program sms, kalender, peta, *browser,* kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java*.
2) *Applications Frameworks* adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *Android,* karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.
3) *Libraries, libraries* ini adalah *layer* di mana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.
4) *AndroidRun Time*, *layer* yang membuat aplikasi *Android* dapat dijalankan di mana prosesnya menggunakan Implementasi *Linux.Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.

5) *Linux Karnel* adalah *layer* di mana inti dari operating sistem dari *Android* itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi *Android* lainnya.

b. Komponen *android*

Ada enam jenis komponen pada aplikasi *Android,* yaitu:

1) *Activities,* suatu *activities* akan menyajikan *user interface (UI)* kepadapengguna, sehingga pengguna dapat melakukan interaksi untuk menjalankan fungsi tertentu. Sebuah aplikasi *Android* bisa jadi hanya memiliki satu *Activity*, tetapi umumnya aplikasi memiliki banyak *Activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.
2) *Service, Service* tidak memiliki *Graphic User Interface (GUI),* tetapi *service* berjalan secara *background* untuk melakukan operasi-operasi yang longrunning (proses yang memakan waktu cukup lama) atau melakukan operasi untuk proses *remote.*
3) *Broadcast Reciever*, *broadcast reciever* berfungsi menerima dan bereaksi untuk menyiapkan notifikasi. *Broadcast Reciever* tidak memiliki *User Interface (UI)* tapi memiliki sebuah *Activity* untuk merespon informasi yang mereka terima atau kepada pengguna. *Broadcast receiver* hanyalah pintu gerbang *menu*ju komponen lain dan memang dirancang untuk hanya melakukan kerja seminimal mungkin.
4) *Content Provider*, *content provider* membuat kumpulan aplikasi data secara *spesifik* sehingga bisa digunakan oleh aplikasi lain. *Content providers* juga

berguna untuk membaca dan *menu*lis data yang berstatus *private* dan tidak dibagikan ke suatu aplikasi. (Safaat, 2018).

c. Kelebihan Sistem Operasi *Android*

Adapun beberapa kelebihan sistem operasi *android* adalah sebagai berikut:

1) Lengkap (Complete Platform)

Android merupakan sistem operasi yang aman dan banyak menyediakan tools yang berguna untuk membangun sebuah aplikasi yang kemudian aplikasi tersebut dapat lebih dikembangkan lagi oleh para developer.

2) Terbuka

Platform *Android* yang bersifat open source menjadikan sistem operasi ini mudah dikembnagkan oleh para developer karena bersifat terbuka.

3) Gratis (Free Platform)

Developer dengan bebas bisa mengembangkan, mendistribusikan dan memperdagangkan sistem operasi *Android* tanpa harus membayar royalty untuk mendapatkan license.

d. Fitur-fitur *Android*

Beberapa fitur-fitur *android* yang paling penting adalah sebagai berikut:

1) *Framework* Aplikasi: memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.
2) Dalvik Virtual Machine: virtual machine yang dioptimalkan untuk Perangkat mobile.
3) Grafik: grafik 2D dan grafik 3D yang didasarkan pada library OpenGL.

4) SQLite: sebagai pengganti dari database pada *android*.

5) Mendukung Media: audio dan video dengan berbagai format.

6) Lingkungan pengembangan yang kaya, termasuk emulator, peralatan debugging, dan plugin untuk Eclipse IDE.

7) Bluetooth, EDGE, 3G, 4G dan WiFi.

8) Mendukung berbagai macam sensor dan GPS.

7. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi *Android*, yang didasarkan pada IntelliJ IDEA . Selainsebagai editor kode dan fitur *developer IntelliJ* yang l, *Android Studio* menawarkanbanyak fitur yang meningkatkan produktivitas dalam membuat aplikasi *Android*, seperti (Muhammad Yusuf Fadillah, 2021**)**:

a) Sistem build berbasis *Gradle* yang *fleksibel*

b) *Emulator* yang cepat dan kaya fitur

Lingkungan terpadu tempat bisa mengembangkan aplikasi untuk semua perangkat *Android*

c) Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi

d) Template kode dan integrasi GitHub untuk membantu membuat fitur aplikasi umum dan mengimpor kode sampel

e) *Framework* dan alat pengujian yang lengkap

f) Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya

g) Dukungan C++ dan NDK

h) Dukungan bawaan untuk *Google Cloud Plat*for*m*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*

8. Famework

Sistem operasi *Android* didukung oleh banyak *framework* untuk mengembangkan aplikasi. Aplikasi *android* secara native dapat dikembangkan dengan bahasa java atau C, sedangkan *framework*-nya menggunakan editor Eclipse dan SDK *Android*. Selain itu, *android* bisa juga dikembangkan menggunakan *framework* lain seperti:

a) Basic4*android*: Bahasa pemrograman Basic.

b) Adobe AIR: Bahasa pemrograman ActionScript.

c) phoneGap: Bahasa Pemrograman HTML5 dan JavaScript.

d) PHP for *Android*: Bahasa pemrograman PHP.

e) AndEngine: *Framework* untuk mengembangkan game *Android*.

9. Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains *Kotlin* merupakan bahasa pemrograman yang pragmatis untuk *android* yang mengkombinasikan *object oriented* (OO) dan *pemrograman fungsional*. *Kotlin* juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java.

Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk backend. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan *Kotlin* untuk mengembangkan aplikasi adalah sebagai berikut:

a) Dapat mengatasi *NullPointerException* yang umumnya terdapat pada Java.

b) Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.

c) Mudah dipelajari.

d) Dukungan IDE untuk mempermudah dalam pemrograman.

10. MySQL

Beberapa pengertian *MySQL menu*rut para ahli:

a) *Menu*rut Raharjo (2018:21), "*MySQL* merupakan RDBMS (atau *server database)* yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*".

b) *Menu*rut Kadir (2020:2), "*MySQL* adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source.* (Adi Nugroho.2020).

11. Metode Pengembangan Perangkat Lunak

a. Metode Pengumpulan Data

Pengumpulan data bertujuan untuk melengkapi dan mendukung materi dalam uraian pembahasan. Pengumpulan data dilakukan guna mendapatkan data yang real sesuai refrensi yang diperlukan. Metode ini meliputi studi pustaka, observasi dan wawancara.

b. *Prototype*

Metode yang digunakan dalam pengembangan sistem ini adalah *prototype*, yaitu metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna. Model *prototype* merupakan salah satu model SDLC yang mempunyai ciri khas sebagai model proses evolusioner. *Prototype* sendiri bertujuan agar pengguna dapat memahami alur proses sistem dengan tampilan dan simulasi yang terlihat siap digunakan. Model *prototype* dapat dilihat pada gambar 2.1 dibawah ini:



Gambar 2 2 *Prototype* Paradigma

Tahap-tahap pada model *prototype* diatas adalah sebagai berikut:

1) Mendengarkan Pelanggan Tahap ini dilakukan pengumpulan data untuk me*menu*hi kebutuhan sistem dengan cara mendengar keluhan dari pelanggan. Analisis kebutuhan pemakai digunakan untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi melalui studi pustaka, observasi, wawancara.
2) Merancang dan Membuat *Prototype* Tahap ini melakukan pembuatan dan perancangan sistem *prototype* serta perancangan antarmuka (*User* Interface). *Prototype* yang akan dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.
3) Uji Coba *Prototype* Tahap ini *prototype* yang telah dirancang akan diuji coba. Kemudian akan dievaluasi kekurangan-kekurangan apa saja yang terdapat pada

sistem. Pengembang kemudian memperbaiki sistem sesuai kekurangan sampai dihasilkan sistem yang benar-benar dikehendaki oleh pengguna (Melisa Adhari, 2021)..

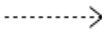
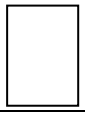
c. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk *visualisasi,* merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap seb uah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

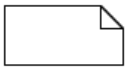
d. *Use Case Diagram*

Diagram Use Case menjelaskan kapabilitas sistem yang diharapkan. Penekanannya adalah pada "apa" yang dilakukan sistem daripada "bagaimana". Sebuah *Use Case* menggambarkan interaksi aktor dan sistem. Kasus penggunaan adalah pekerjaan tertentu, seperti masuk ke sistem atau membuat daftar belanja. Aktor adalah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan tugas tertentu. Di antara simbol-simbol *Use Case Diagram* adalah:

Tabel 2. 1 Simbol *Usecase Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna *Main*kan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

Lanjutan Tabel 2. 1

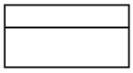
NO	GAMBAR	NAMA	KETERANGAN
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

e. *Class Diagram*

class diagram adalah jenis *Diagram* struktur statis dalam UML yang menggambarkan struktur sistem dengan *menu*njukkan sistem *class*, atributnya, metode, dan hubungan antar objek. *Class diagram* disebut jenis *Diagram* struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen.

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2. 2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

Lanjutan Tabel 2. 2

NO	GAMBAR	NAMA	KETERANGAN
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

f. *Activity* ***Diagram***

Menyatakan bahwa *Diagram* aktivitas atau *Activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak.

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2 3 Simbol *Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		For*k Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

g. *Sequence* **Diagram**

Menyatakan bahwa *Sequence Diagram* adalah suatu *Diagram* yang menggambarkan interaksi antar obyek dan mengindikasi komunikasi diantara obyek-obyek tersebut. *Diagram* ini juga *menu*njukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu obyek-obyek tersebut kemudian diurutkan dari kiri kekanan, aktor yang menginisialisasi interaksi biasanya ditaruh di paling kiri dari *Diagram*.

Adapun simbol-simbol *Sequence Diagram* antara lain :

Tabel 2. 4 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

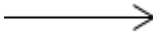
h. ***StateChartDiagram***

*Statechar*t *Diagram* merupakan model perilaku yang dinamis dari class secara individual maupun beberapa bentuk dari obyek. Semua itu *menu*njukkan deretan dari *state* yang dilakukan obyek melalui event yang menyebabkan sebuah transisi dari satu

ke aktifitas yang lain, dan beberapa aksi yang menghasilkan dari satu *state* atau aktifitas yang berubah. *State*chart *Diagram* fokus pada *state* dan secara bentuk digunakan untuk memodelkan tahapan yang nampak terpotong-potong dari sebuah aktivitas yang kontinyu dari obyek. *Diagram* ini memiliki hubungaan yang tetrtutup dengan *Activity Diagram*. *Activity Diagram* fokus pada aktifitas dan pada pemodelan deretan aktivitas dari proses.

Adapun simbol-simbol *StateChart Diagram* antara lain :

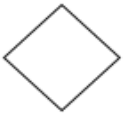
Tabel 2. 5 Simbol *StateChart Diagram*

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

i. *Flowchart*

Awaluddin. (2022). *Flowchart* (*Diagram* alir) dapat digunakan sebagai alternative untuk menyajikan algoritma. *Flowchart* adalah bentuk penyajian grafis yang menggambarkan solusi langkah demi langkah terhadap suatu masalah.

Tabel 2. 6 Simbol Standar *Flowchart*

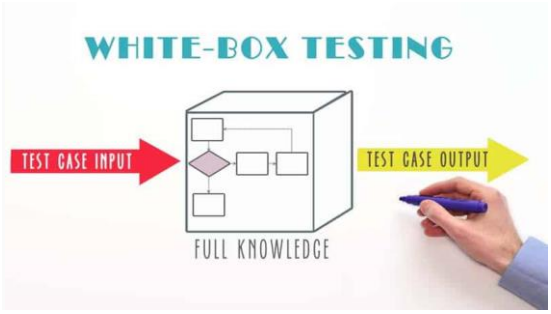
Simbol	Nama dan Funsi
	***Terminator***(Tanda mulai atau selesai), menyatakan titik awal dan titik akhir *Diagram* alir.
	**Proses** (Tanda komputasi), menyatakan sebarang proses.
	**Keputusan** (Pengambilan keputusan), digunakan untuk pengambilan keputusan. Dalam hal ini yang ada didalam symbol ini berupa suatu pertanyaan yang jawabannya berupa Ya atau Tidak.
	**Proses Terdefinisi** (Tanda prosedur atau fungsi subalgoritma), menyatakan prosedur lain yang telah di*Diagram*alirkan.
	**Konektor** (Tanda penghubung), digunakan untuk menghubungkan ke berbagai bagian dalam alir *Diagram*.

10. Metode Pengujian

a. *White Box*

White box testing adalah suatu pengujian oleh *software* atau aplikasi dengan melihat modul untuk menganalisis kode program ada yang salah atau tidak. Bila suatu modul yang diproduksi tidak me*menu*hi syarat, kode akan dikompilasi ulang dan periksa lagi hingga mencapai apa yang diharapkan dalam arti bahwa *white box testing*

menguji dengan cara melihat pure *code* dari suatu perangkat lunak atau aplikasi yang diuji tanpa mempedulikan tampilan atau UI dari aplikasi tersebut.



Gambar 2 3 *White Box Testing*

1) Teknik

Pada *white box testing* ada beberapa teknik yang digunakan. Berikut akan kami sajikan :

a) Basis *path testing* adalah metode yang memungkinkan perancang *tast case* untuk membuat pengukuran kompleksitas logika dari rancangan prosedural dan menggunakan pengukuran untuk mendefinisikan himpunan basis dari jalur eksekusi.

b) *Flow graph* merupakan notasi sederhana guna mempresentasi *control flow*.

c) *Cyclomatic complexity* yang digunakan guna mengetahui jumlah *path* yang perlu dicari. Di mana *cyclomatic complexity* merupakan metric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika program.

d) *Branch coverage* merupakan pengujian yang didesain agar masing-masing *branch code* yang paling tidak diuji satu kali.

e) *Loop testing* merupakan *testing* pada berbagai *looping*/perulangan yang terdapat pada program, seperti for, *while*, dan do-*while*. Dengan pengujian ini Anda bisa mengecek kondisi perulangan, sudahkah berjalan dengan benar ataupun tidak.

f) *Condition coverage* merupakan teknik pengujian yang bertujuan guna melakukan uji semua kode untuk menghasilkan *value* FALSE atau TRUE. *Test*er bisa memastikan *software* dalam beroperasi dan menghasilkan output sesuai input *user*.

g) *Multiple condition coverage* merupakan teknik yang digunakan guna menguji semua kombinasi beragam kode yang digunakan dalam berbagi kondisi.

h) *Statement coverage* merupakan teknik yang minimal 1 kali dilakukan guna menguji masing-masing *statement* pada *software*. Anda bisa mengetahui beberapa *code* yang *error* dengan pengujian ini sehingga bisa segera memperbaikinya.

2) Trik Pengujian

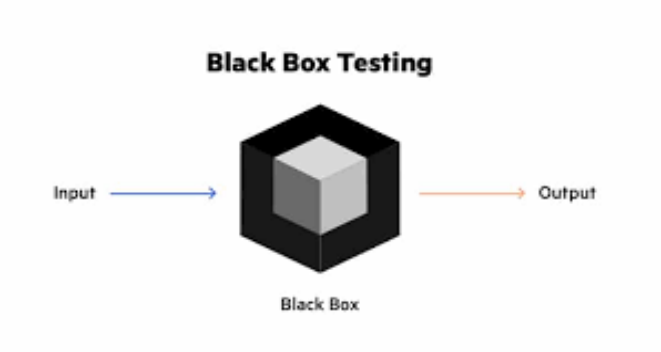
White box testing merupakan pengujian dengan dasar pada pengecekan di dalam detail desain dari suatu rancangan pemrograman untuk membagi pengujian. Dalam membuat pengujian *white box* maka *test*er perlu memiliki pemahaman terkait pemrograman. Sebab untuk memahami kode sumber yang dipelajari dan harus memiliki pandangan global tentang fungsi aplikasi serta elemen yang dirancangnya. Metode *white test* ini dapat diterapkan untuk tes integrasi, tes unit dan tes sistem.

3) Contoh *White Box*

Contoh *white box testing* menggunakan teknik *cyclomatic*, di mana pada dasarnya dengan cara menghitung jumlah siklomatik yaitu menghitung jumlah *condition*al I predikat dan tambahkan 1. Misalnya terdapat lima *conditional* maka *cyclomaticnya* adalah 6. Berikut cara menghitungnya :

a) 1-2-3-4-5-10 yaitu suatu properti yang dimiliki orang lain, tidak mempunyai uang untuk sewa
b) 1-2-3-4-6-10 suatu properti yang dimiliki orang lain dan membayar sewa
c) 1-2-3-10 yaitu properti yang dimiliki oleh pemain
d) 1-2-7-10 yaitu properti yang tersedia dan tidak memiliki cukup uang
e) 1-2-7-8-10 yaitu properti yang tersedia, memiliki uang dan tidak ingin membelinya
f) 1-2-7-8-9-10 yaitu properti yang tersedia, memiliki uang dan membelinya. (Sumber : https://it*box*.id/blog/*white-box-testing*-adalah/)

b. *Black Box*



Gambar 2 4 *Black Box Testing*

Pengujian *black box* adalah pengujian yang dilakukan semata-mata dengan memantau hasil eksekusi melalui data pengujian dan memvalidasi fungsionalitas perangkat lunak. Jadi, mirip dengan melihat kotak hitam, kita hanya bisa melihat tampilan luarnya saja dan tidak tahu apa yang ada di balik bungkus hitamnya. Mirip dengan pengujian *black box*, hanya mengevaluasi tampilan luar (antarmuka) dan fungsionalitasnya. Tanpa mengetahui apa yang terjadi dalam proses detail (hanya mengetahui input dan output).

1) Jenis Pengujian

Pada suatu *black box testing* terdapat tiga jenis pengujian, berikut akan kami jelaskan :

a) *Functional testing*

Merupakan suatu proses pengujian pada fitur spesifik atau fungsi pada suatu *software*. Pengujian ini bisa dilakukan guna memastikan bahwa *user* tidak dapat masuk pada aplikasi yang diuji tanpa informasi sehingga program tersebut terjaga keamanannya.

b) Non *functional testing*

Memiliki fungsi yang tidak kalah dari *functional testing*, di mana Non *functional testing* merupakan suatu pengujian pada beberapa aspek tambahan non fungsional. Bila *functional testing* menguji *software* menjalankan suatu tugas atau peringkat.

c) *Regression testing*

Jenis ini merupakan pengujian yang dilakukan pada *software* untuk mengamati apakah terjadinya kemunduran atau regresi ketika *software* diperbaharui. Dalam *regression test* dilakukan pengecekan aspek non fungsional maupun fungsional.

2) Teknik Menguji *Black Box* Dengan *Software*

Tulisan di atas sudah menjelaskan apa itu *black box testing* dan contohnya. Namun *black box testing* memiliki beberapa teknik dalam menguji suatu *software*. Berikut akan kami jelaskan :

a) *Equivalence partitioning* merupakan cara kerja dengan melakukan pembagian atau partition menjadi beberapa partisi dari input data.

b) *Fuzzing* merupakan teknik untuk mencari gangguan atau bug dari *software* dengan cara injeksi data yang terbilang sesi semi otomatis atau cacat.

c) *Boundary value analysis* merupakan teknik yang lebih fokus pada *boundary* di mana jika ada *error* dari sisi atau luar baik minimum maupun maksimum pada pada *software* yang didapat.

d) *Orthogonal array testing* merupakan teknik yang digunakan jika input *domain* yang relatif ukurannya kecil tetapi cukup berat jika digunakan dalam skala besar.

e) Ca*use effect graph* merupakan teknik yang digunakan di mana *graph*ic sebagai pacuannya. Dalam teknik ini di mana *graph*ic menggambarkan relasi antara penyebab dan efek dari *error* tersebut.

f) *State transition* merupakan *testing* yang digunakan untuk mengecek apakah kondisi dari navigasi dan mesin dalam bentuk grafik.

g) *All pair testing*, pada teknik ini semua pasangan dari *test case* didesain agar bisa dieksekusi semua kemungkinan kombinasi diskrit berdasarkan input parameternya. Dengan teknik ini memiliki tujuan agar semua memiliki pasangan *test case* (Sumber : https://it*box*.id/blog/*black-box-testing*-adalah/).

B. 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: Agus Irawan dan Tajudin Noor (2020) Jurnal INTEKNA, Tahun XIV, No. 2, Nopember 2020 : 102 – 209 yang berjudul "SISTEM INFORMASI ABSENSI MAHASISWA (STUDI KASUS JURUSAN *ADMIN*ISTRASI BISNIS POLIBAN)". Hasil dari penelitian tersebut adalah Sistem informasi yang dibangun menggunakan *arsitektur client/server* dengan menggunakan arsitektur 2-tier, sehingga dapat dimanfaatkan dengan menggunakan aplikasi desktop maupun web. Bahasa pemrograman yang digunakan adalah PHP dan *database* server *Mysql*. Untuk menjalankan aplikasi ini diperguna- kan sebuah komputer server dengan nomor IP yang disetup dengan IP Private sehingga hanya dapat dipergunakan dalam jaringan local/intranet di lingkungan Jurusan

*Admin*istrasi Bisnis Politeknik Negeri Banjarmasin. Dalam implementasinya sistem ini akan melibatkan beberapa bagian, diantaranya bagian *Admin*istrasi, dan Dosen yang bersinggungan langsung dengan pengisian data absensi mahasiswa.

2. Penelitian oleh Rendy Rian Chrisna Putra , Fransiskus Panca Juniawan (2017) Jurnal Telematika Vol. 10 No. 1 Februari 2017 146 ISSN : 1979 – 925X e-ISSN : 2442 – 4528 yang berjudul "PENERAPAN ALGORITMA FISHERFACES UNTUK PENGENALAN WAJAH PADA SISTEM KEHADIRAN MAHASISWA BERBASIS *ANDROID*". Teknologi yang dapat membantu aktivitas manual menjadi terkomputerisasi. Kehadiran mahasiswa yang tadinya secara manual dapat diganti dengan sistem terkomputerisasi yaitu dengan *biometrik* pengenalan wajah dengan menggunakan *smartphone* berbasis *android*. Pengenalan wajah bertujuan untuk meminimalkan tindakan kecurangan dan menghindari kesalahan *entry* ke dalam *database.* Algoritma yang digunakan yaitu *local binary pattern* (LBP) cascade untuk mendeteksi wajah dan algoritma f*isherfaces* untuk pengenalan wajah. Perangkat lunak yang digunakan untuk penerapan algoritma *fisherfaces* pada sistem pengenalan wajah ini menggunakan sistem operasi *Android* serta bahasa pemrograman Java.

3. Penelitian oleh Antonius Irianto Sukowati, Helmi Fauziah Yulianti dan Imam Purwanto (2017) Jurnal Ilmiah KOMPUTASI, Volume 16 No :2, September 2017, p-ISSN 1412-9434/e-ISSN 2549-7227 yang berjudul "Rancang Bangun Sistem Absensi Mahasiswa Sekolah Tinggi Teknik. Cendekia (STTC) Berbasis *Radio*

Frequency Identication (RFID) menggunakan Arduino UNO R3". Penelitian tersebut menghasilkan sistem informasi kehadiran mahasiswa elektronik yang datanya diintegrasikan dengan Sistem Informasi Akademis (SIA). Alat ini dirancang dengan memadukan kerja mikrokontroler dengan Radio *Frequency Identification* (RFID) kedalam sebuah sistem. RFID Tag yang mempunyai nomor unik sebagai data mahasiswa digunakan sebagai kartu penanda kehadiran dengan cara menempelkannya pada alat pencatat kehadiran. Led berwarna hijau akan menyala apabila data mahasiswa sesuai, sebaliknya led merah akan menyala apabila data mahasiswa tidak sesuai. Data kehadiran dicatat dalam database yang dibuat dengan menggunakan *Mysql*. Untuk memudahkan pengguna, dibuat antarmuka dengan menggunakan *visual basic.* Data ini selanjutnaya akan dimasukan kedalam Sistem Informasi Akademik yang sudah ada untuk keperluan lebih lanjut. Berdasarkan penelitian yang relevan diatas dapat disimpulkan bahwa ketiga penelitian tersebut membahas terkait sistem absensi yang masih manual. Adapun ketiga penelitian diatas memiliki perbedaan dalam penyelesaian masalahnya. Mengacu pada penelitian diatas, penelitian yang akan dilakukan dalam proses penyelesaian masalahnya menggunakan *QR Code.* Penelitian ini juga difokuskan untuk memudahkan staf tata usaha dalam rekapitulasi data absensi mahasiswa. Dengan begitu sistem ini dapat memberikan informasi presensi lebih *up to date* dan efisien.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kualitatif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan Skripsi ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat. Penelitian ini dilakukan dengan cara mengumpulkan data secara langsung dari objek penelitian melalui pengamatan secara langsung yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

B. Waktu dan Tempat Penelitian

Adapun waktu dan tempat penelitian yang dilaksanakan pada bulan Juli tahun 2024, di SMA Negri 5 Parepare, dimana proses pengambilan data yang akan dimasukkan dalam aplikasi nantinya.

Tabel 3. 1 Jadwal Penelitian

No	UraianKegiatan	Bulan 2024	
		Juli	Agustus
1	*Studi Literatur*		
2	Perancangan aplikasi		
3	Pembuatan aplikasi		
4	Pengujian		

C. Alat dan Bahan Penelitian

Dalam pelaksanaan penelitian ini digunakan beberapa alat dan bahan yang terdiri dari Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*).Perangkat keras yang digunakan terdiri dari Laptop ASUS. Perangkat lunak yang digunakan adalah *Windows 10 Home Single Language*, *Android Studio, Kotlin, dan MySQL.* Sedangkan bahan penelitian berupa data-data yang berasal dari hasil pengumpulan data yang telah dilakukan oleh peneliti yang nantinya akan dianalisis lebih lanjut sebagai landasan untuk merancang sebuah sistem.

D. Tahap Penelitian

Tahap penelitian dilakukan dengan cara :

1. Persiapan Penelitian

Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini penelitian melakukan observasi dengan peninjauan dan juga studi literatur.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem dan juga data yang didapatkan saat melakukan pengumpulan data, kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi e-marketplace wedding organizer yang ingin dibuat berdasarakan alternatif pemecahan masalah.

5. Pengujian

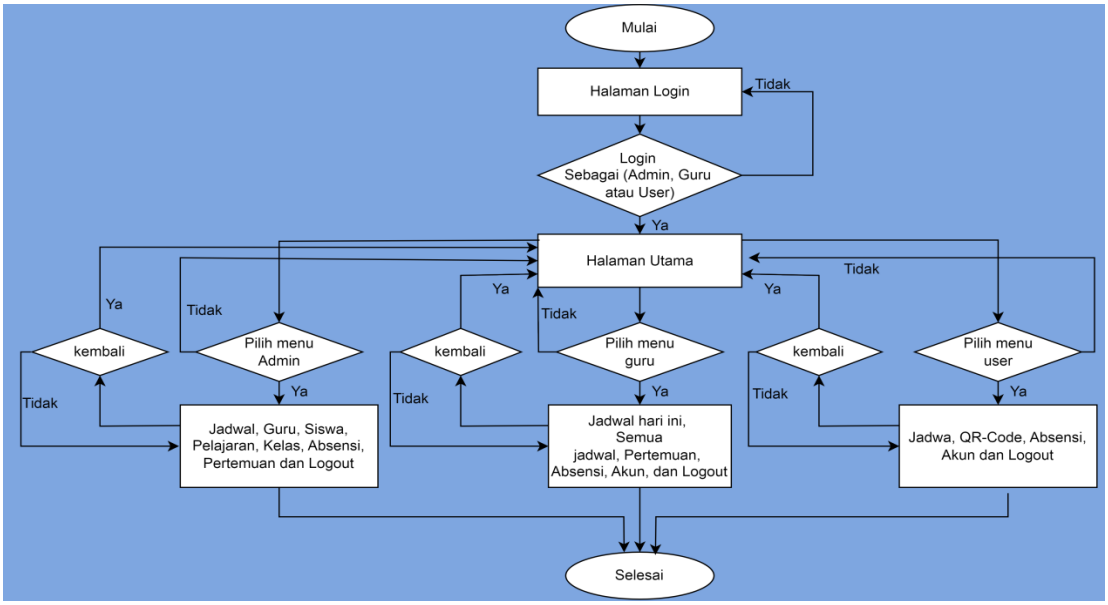
Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan aplikasi.Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan.

E. Analisis Sistem

1. *Flowchart* Aplikasi

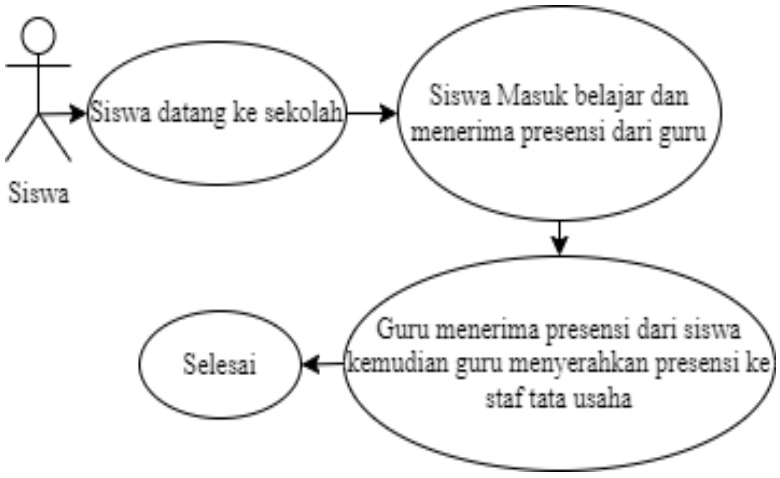


Gambar 3 1 *Flowchart* Aplikasi

Pada gambar 3.1 yaitu *flowchart* sistem dimana pada suatu *flowchart* pastinya diawali dengan mulai atau start. Kemudian ada beberaapa *menu* yaitu *menu* utama yaitu *menu admin*, *menu* guru dan *menu user*, kemudian selesai

2. Analisis Sistem yang Berjalan

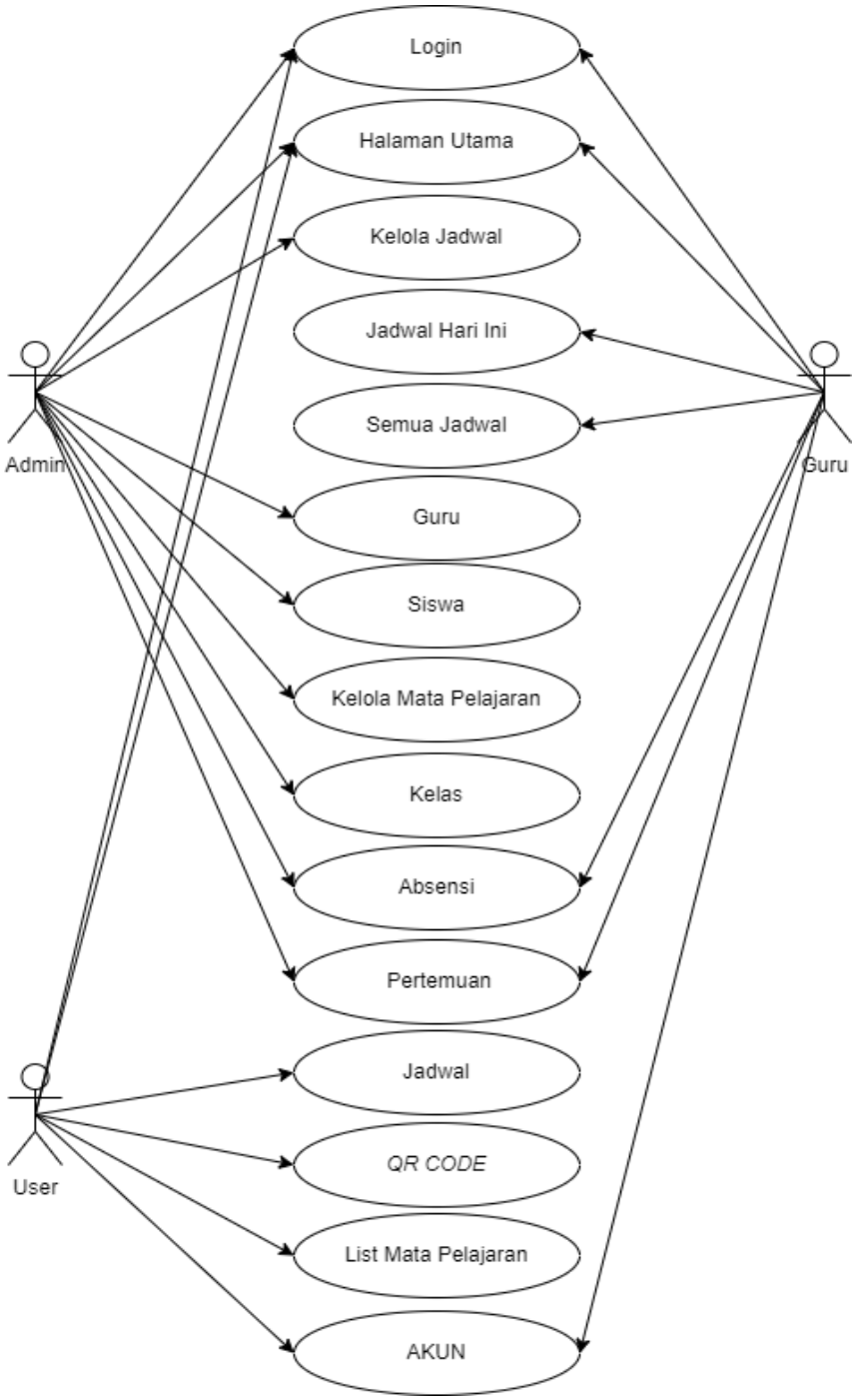
Sebelum dilakukan perancangan sistem yang baru, terlebih dahulu dilakukan analisa terhadap sistem yang berjalan untuk membandingkan kinerja sistem yang telah ada dengan sistem yang akan diusulkan. Adapun berikut merupakan sistem yang berjalan



Gambar 3 2 Sistem Yang Berjalan

Berdasarkan gambar diatas diketahui bagaimana proses sistem berjalan di mana siswa datang ke sekolah lalu siswa masuk belajar dan menerima presensi dari guru, setelah melakukan presensi guru menerima presensi dari siswa kemudian guru menyerahkan presensi ke sta tata usaha selesai.

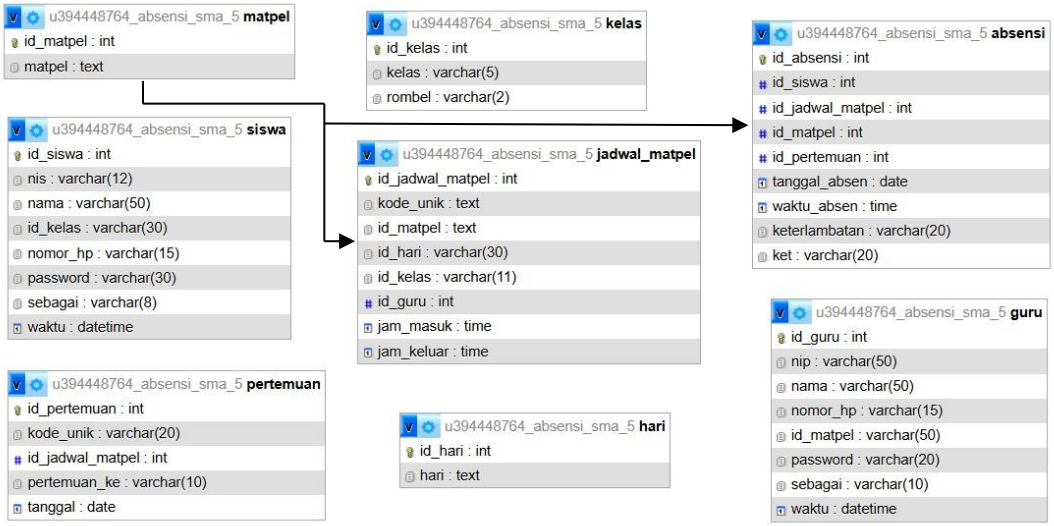
3. Analisis Sistem yang diusulkan



Gambar 3 3 Analisa sistem yang diusulkan

Dari gambar diatas menjelaskan mengenai sistem yang diusulkan mempunyai tiga aktor *admin*, guru dan siswa, *Admin* dapat mengakses *menu* Kelola jadwal, Guru, Siswa, Kelola materi pelajaran, Kelas, Absensi dan Pertemuan, sedangkan guru dapaat mengakses Jadwal hari ini, Semua jadwal, Pertemuan, Absensi dan Akun, kemudian siswa Jadwal, *QR Code,* Pertemuan, Absensi dan Akun.

4. Relasi *Database*



Gambar 3 4 Relasi *Database*

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Aplikasi *QR-Code* sebagai media pelayanan untuk absensi untuk siswa SMA Negri 5 Parepre berbasis *Android*, menggunakan pemrograman *Kotlin*, sedangkan untuk pengelolaan basis data menggunakan *Mysql*. Metode yang digunakan dalam perancangan aplikasi ini adalah UML (Unified Modelling Langguage).

B. Pembahasan

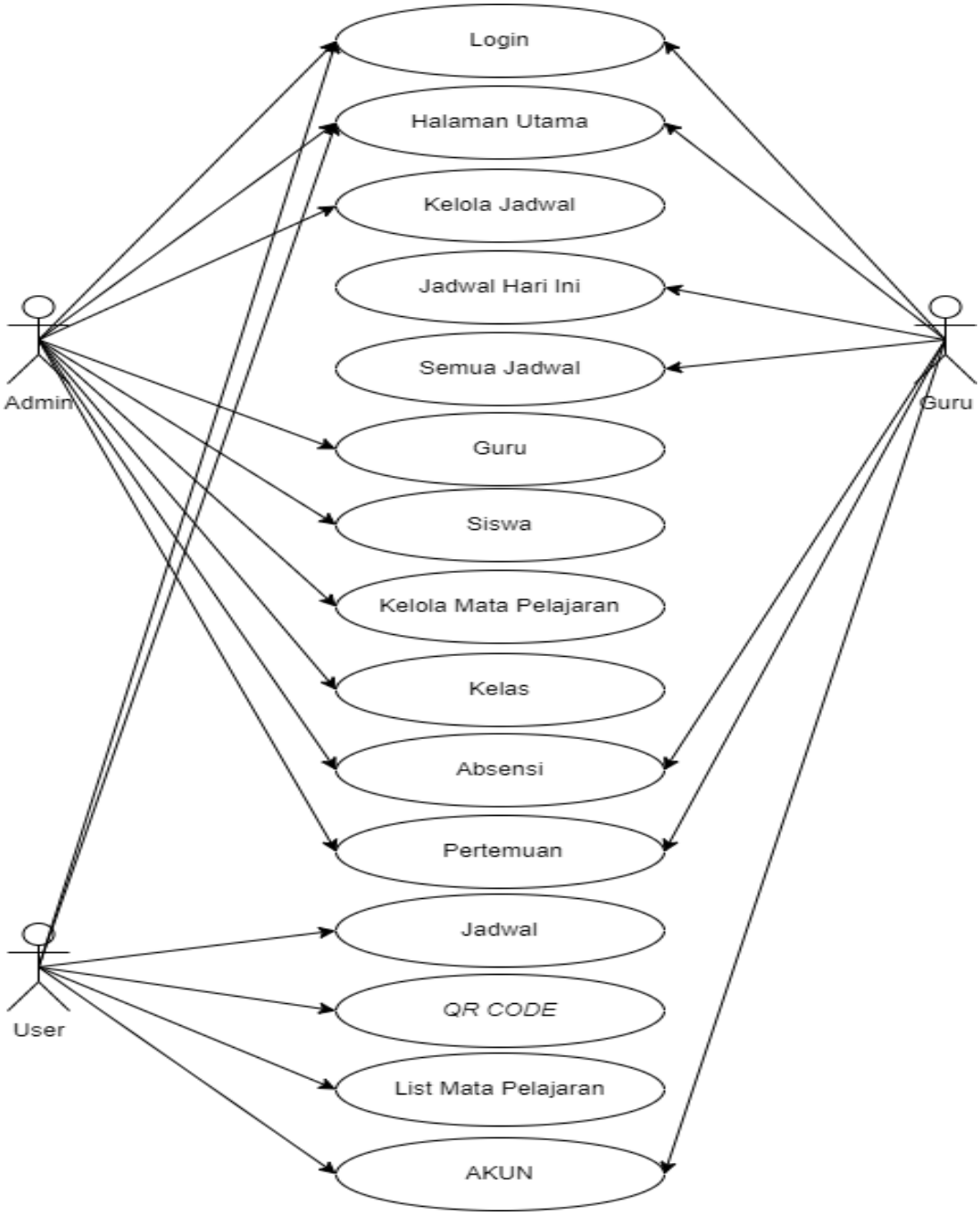
1. Analisis Aliran Data UML

b) *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 penerapan *qr-code* sebagai media

pelayanan untuk absensi pada siswa SMA Negri 5 Parepare berbasis *android* pada penelitian ini digambarkan seperti berikut ini:



Gambar 4 1 *Use Case Diagram*.

c) Tabel *Use Case*

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

Tabel 4.1. Definisi Aktor dalam Sistem.

No	Aktor	Keterangan
1	*User*	*User* merupakan aktor yang berperan sebagai pengguna sistem informasi akuntansi pembelian dan penjualan.
2	*Admin*	*Admin* merupakan aktor yang memiliki hak akses lebih tinggi dibanding *User* dan Guru.
3	Guru	Guru merupakan aktor yang memiliki hak akses lebih tinggi dari *user.*

Sedangkan untuk penjelasan definisi setiap *use case* sebagai berikut:

Tabel 4.2. Definisi *Use Case* Sistem

No	*Use Case*	Keterangan
1	*Login*	Melakukan *Login* akun.
2	*Home*	Menampilkan halaman utama aplikasi
3	Jadwal	Menampilkan halaman jadwal
4	Jadwal Hari ini	Menampilkan halaman jadwal hari ini.
5	Semua Jadwal	Menampilkan halaman semua jadwal
6	Guru	Manampilkan halaman guru
7	Siswa	Menampilkan halaman siswa
8	Mata Pelajaran	Menampilkan halaman mata pelajaran
9	Kelas	Menampilkan halaman kelas

No	*Use Case*	Keterangan
10	Absensi	Menampilkan halaman absensi
11	Pertemuan	Menampilkan halaman pertemuan
12	*Qr-Code*	Menampilkan halaman *qr-code*
13	Akun	Menampilkan halaman akun

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*:

1) *Use Case Login*

Aktor : *Admin* dan *User*

Deskripsi : Sebuah kegiatan untuk Masuk ke aplikasi

Pre-*condition* : -

Post-*condition* : - Data baru sudah tersimpan.

Tabel 4.3. .Skenario *Login*.

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan *User* name dan password.	
2. Menekan tombol *Login*.	3. Mengecek validitas isian data.

Aksi Aktor	Reaksi Sistem
	4. Jika data yang dimasukkan valid, maka lansung masuk kehalaman utama aplikasi.
Jika data yang dimasukkan tidak valid, maka akan muncul pesan “Silahkan masukkan *user name* dan *password* 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 4.4. 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* Jadwal

Aktor : *User* dan *Admin*.

Deskripsi : Sebuah kegiatan untuk menampilkan jadwal.

Pre-*condition* : - .*User* dan *Admin* berada di halaman jadwal.

Post-*condition* : -

Tabel 4.5. Skenario *Use Case* Jadwal.

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* jadwal	
	2. Menampilkan form jadwal

4) Skenario *Use Case* Jadwal Hari Ini

Aktor : Guru

Deskripsi : Sebuah kegiatan untuk melihat Jadwal Hari Ini

Pre-*condition* :- *Admin* berada di halaman Jadwal Hari Ini

Post-*condition* : -

Tabel 4.6. Skenario *Use Case* Jadwal Hari Ini

Aksi Aktor	Reaksi Sistem
1. Menekan *menu* Jadwal Hari Ini	
	2. Menampilkan data yang dipilih (Jadwal Hari Ini.) yang dimaksud dalam sistem.
	3. Menampilkan data yang dipilih sesuai data yang ada.

5) *Use Case* Semua Jadwal

Aktor : Guru

Deskripsi : Sebuah kegiatan untuk melihat Semua Jadwal

Pre-*condition* : - *User* berada di halaman Semua Jadwal

Post-*condition* : -

Tabel 4.7. Skenario *Use Case* Semua Jadwal

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* Semua Jadwal	
	2.Menampilkan form Semua Jadwal

6) Skenario *Use Case* Guru

Aktor : *Admin*

Deskripsi : Sebuah kegiatan untuk melihat guru

Pre-*condition* :- *Admin* berada di halaman guru

Post-*condition* : -

Tabel 4.8. Skenario *Use Case* guru

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* guru	
	2.Menampilkan form guru

7) *Use Case* Siswa

Aktor : *Admin*

Deskripsi : Sebuah kegiatan untuk melihat Siswa

Pre-*condition* : - *Admin* berada di halaman Siswa

Post-*condition* : - Data dalam sistem akan diperbarui.

Tabel 4.9. Skenario *Use Case* Siswa

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* Siswa	
	2.Menampilkan form Siswa

8) *Use Case* Mata Pelajaran

Aktor : *Admin*

Deskripsi : Sebuah kegiatan untuk melihat mata pelajaran.

Pre-*condition* : - *User* berada di halaman mata pelajaran.

Post-*condition* : -

Tabel 4.10. Skenario *Use Case* mata pelajaran.

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* mata pelajaran.	
	2.Menampilkan form mata pelajaran.

9) Skenario *Use Case* Kelas

Aktor : *Admin*

Deskripsi : Sebuah kegiatan untuk melihat kelas.

Pre-*condition* :- *Admin* berada di halaman kelas.

Post-*condition* : -

Tabel 4.11. Skenario *Use Case* kelas.

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* kelas.	
	2.Menampilkan form kelas.

10) *Use Case* Absensi

Aktor : *Admin*, Guru, dan *User*

Deskripsi : Sebuah kegiatan untuk melihat Absensi

Pre-*condition* : - *Admin* berada di halaman Absensi

Post-*condition* : - Data dalam sistem akan diperbarui.

Tabel 4.12. Skenario *Use Case* Absensi

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* Absensi	
	2.Menampilkan form Absensi

11) *Use Case* Pertemuan

Aktor : *Admin* dan Guru.

Deskripsi : Sebuah kegiatan untuk melihat Pertemuan.

Pre-*condition* : - *User* berada di halaman Pertemuan.

Post-*condition* : -

Tabel 4.13. Skenario *Use Case* Pertemuan.

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* Pertemuan	
	2.Menampilkan form Pertemuan

12) Skenario *Use Case Qr Code*

Aktor : *User*

Deskripsi : Sebuah kegiatan untuk melihat *Qr Code*

Pre-*condition* :- *Admin* berada di halaman *Qr Code*

Post-*condition* : -

Tabel 4.14. Skenario *Use Case Qr Code*

Aksi Aktor	Reaksi Sistem
1. Memilih *menu Qr Code*.	
	2.Menampilkan form *Qr Code*.

13) *Use Case* Akun

Aktor : *Admin*, Guru, dan *User*.

Deskripsi : Sebuah kegiatan untuk melihat akun.

Pre-*condition* : - *Admin* berada di halaman akun.

Post-*condition* : -.

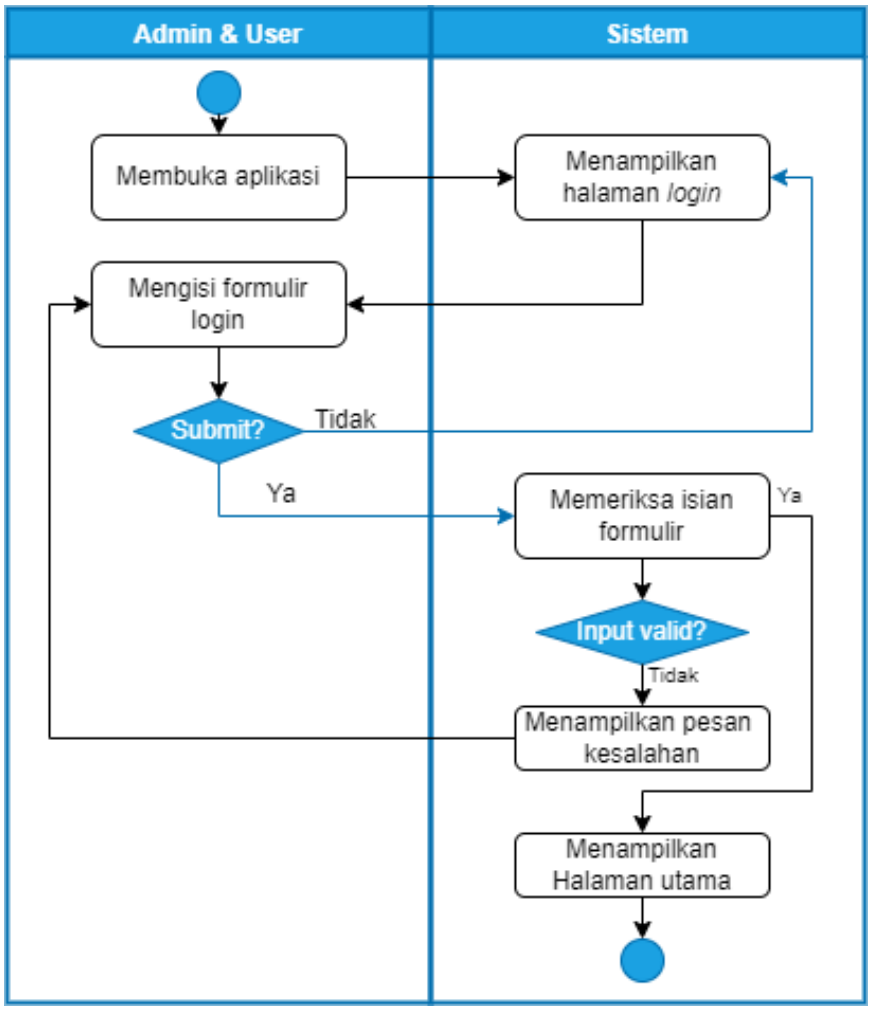
Tabel 4.15. Skenario *Use Case* akun.

Aksi Aktor	Reaksi Sistem
1. Memilih *menu* akun.	
	2.Menampilkan form akun.

d) *Activity Diagram*

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

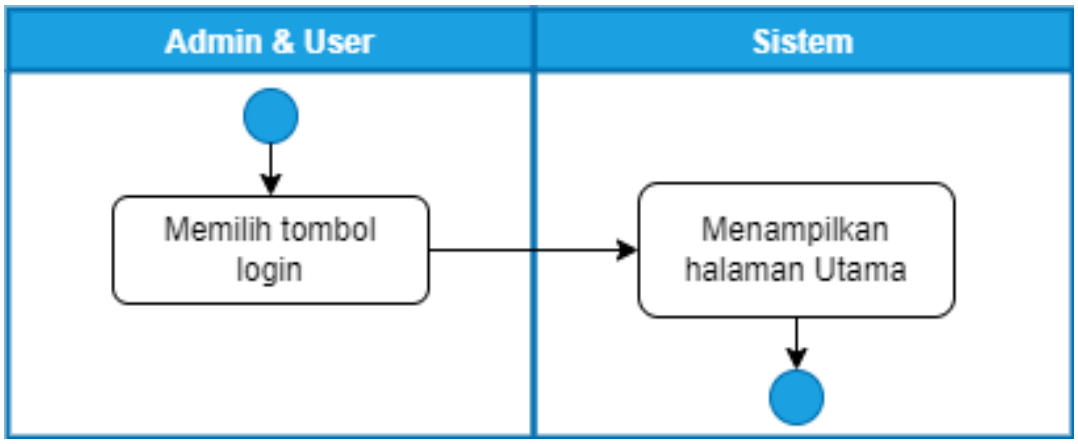
1) *Activity Diagram Login*



Gambar 4 2 *Activity Diagram Login*

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

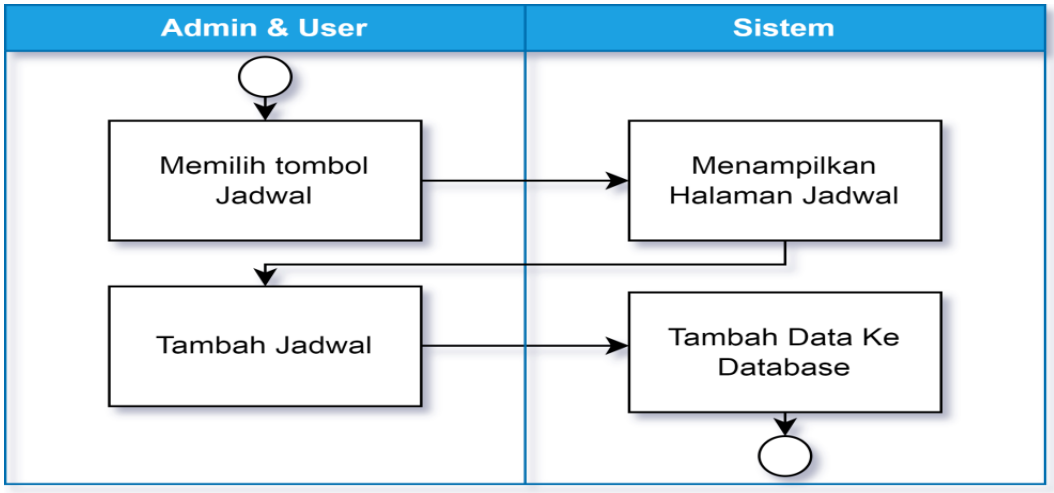
2) *Activity Diagram Home*



Gambar 4 3 *Activity Diagram Home*

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

3) *Activity Diagram* Jadwal

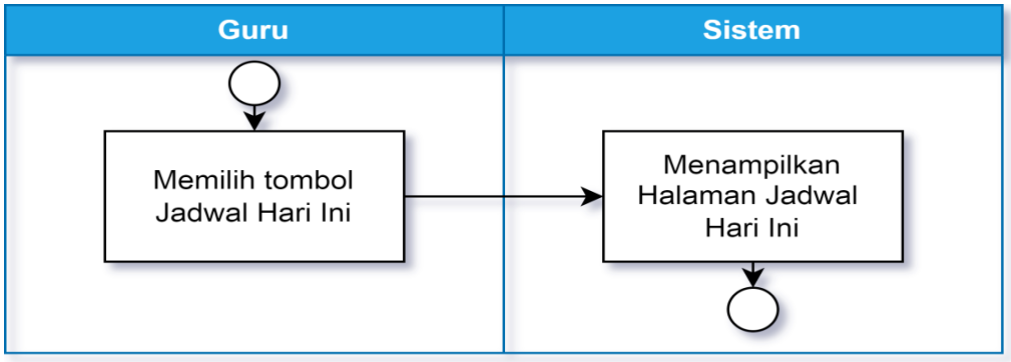


Gambar 4 4 *Activity Diagram* Jadwal

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas jadwal Proses dimulai dengan *admin* dan *user* memilih tombol jadwal, kemudian sistem

menampilkan halaman jadwal Jika ingin tambah data, maka dapat memilih tambah data maka sistem akan menambahkan data tersebut ke database.

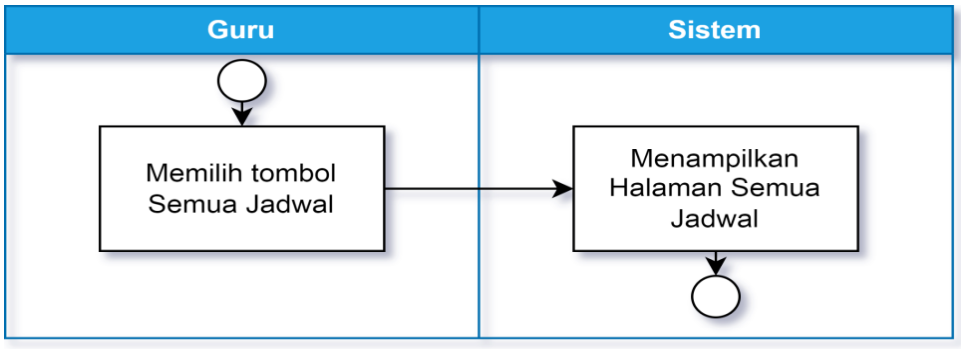
4) *Activity Diagram* Jadwal Hari Ini



Gambar 4 5 *Activity Diagram* Jadwal Hari Ini

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas jadwal hari ini Proses dimulai dengan *admin* memilih tombol aktivitas jadwal hari ini, kemudian sistem menampilkan halaman aktivitas jadwal hari ini.

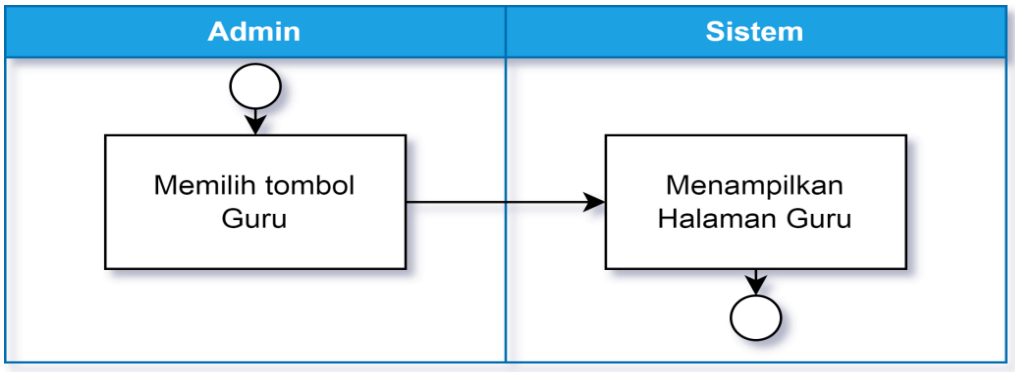
5) *Activity Diagram* Semua Jadwal



Gambar 4 6 *Activity Diagram* Semua Jadwal

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas semua jadwal. Proses dimulai dengan guru memilih semua jadwal, kemudian sistem menampilkan halaman semua jadwal.

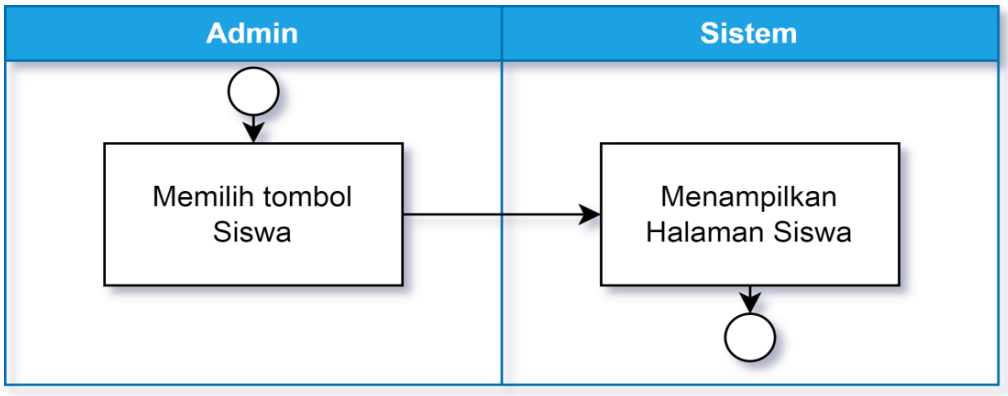
6) *Activity Diagram* Guru



Gambar 4 7 *Activity Diagram* Guru

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas guru. Proses dimulai dengan *admin* memilih guru, kemudian sistem menampilkan halaman guru.

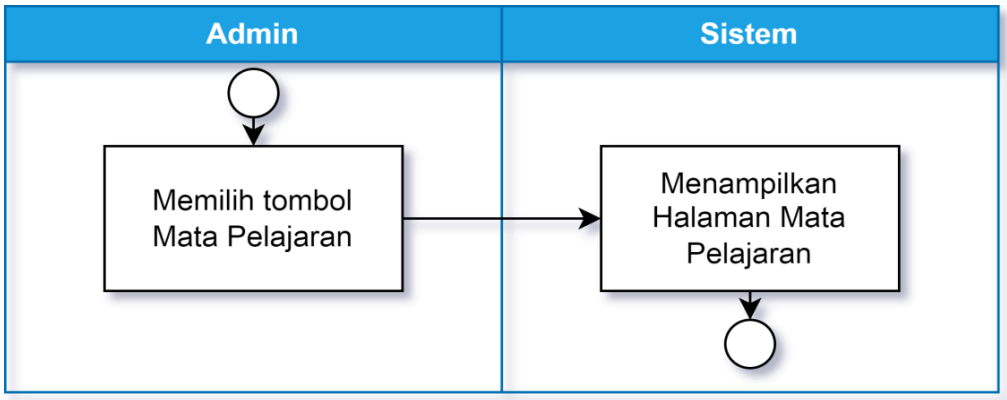
7) *Activity Diagram* Siswa



Gambar 4 8 *Activity Diagram* Siswa

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas siswa. Proses dimulai dengan *admin* memilih siswa, kemudian sistem menampilkan halaman siswa.

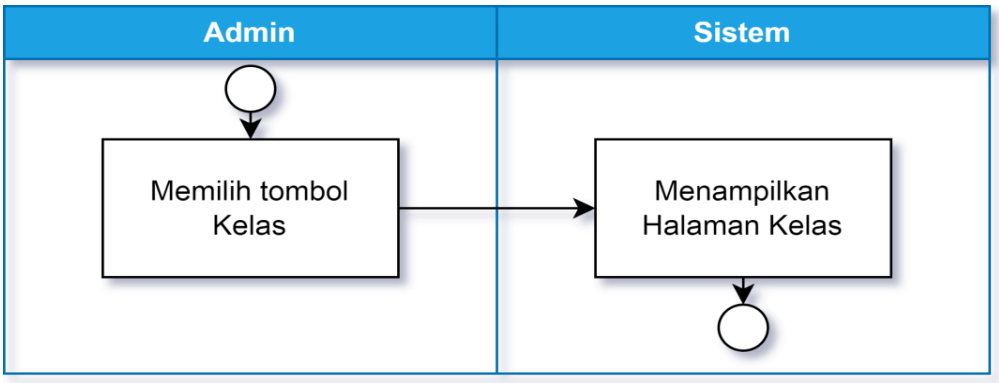
8) *Activity Diagram* Mata Pelajaran



Gambar 4 9 *Activity Diagram* Mata Pelajaran

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas mata pelajaran Proses dimulai dengan *admin* memilih mata pelajaran, kemudian sistem menampilkan halaman mata pelajaran.

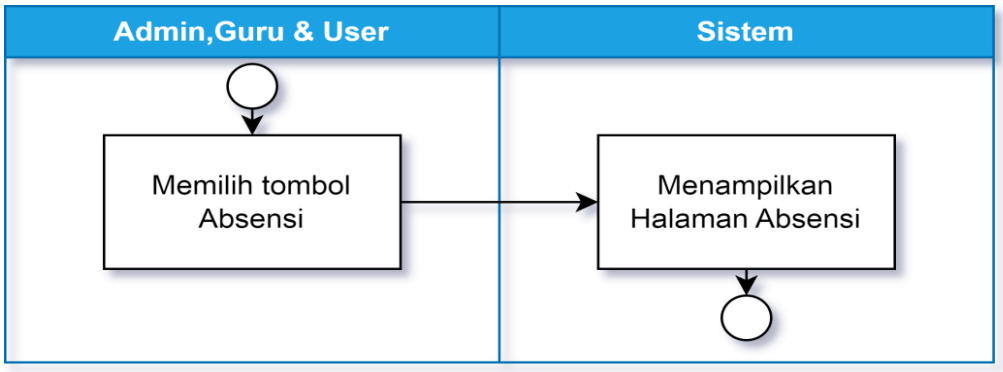
9) *Activity Diagram* Kelas



Gambar 4 10 *Activity Diagram* Kelas

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas kelas Proses dimulai dengan *admin* memilih kelas, kemudian sistem menampilkan halaman kelas.

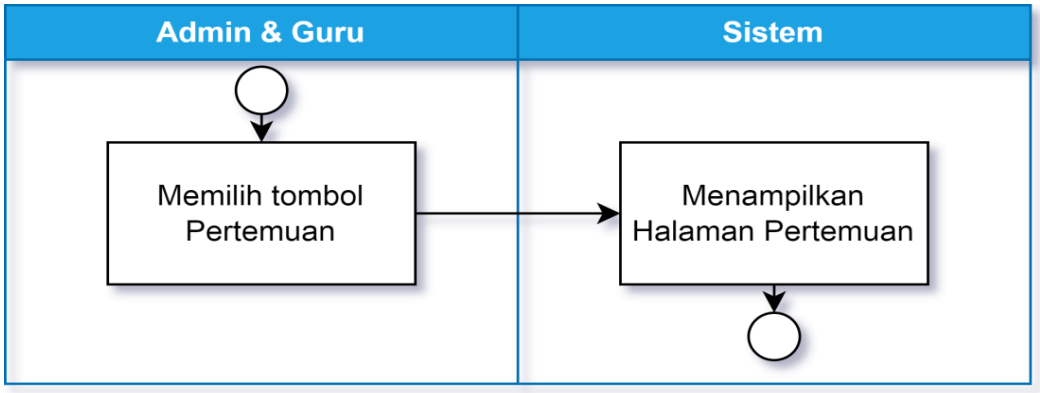
10) *Activity Diagram* Absensi



Gambar 4.7. *Activity Diagram* Absensi

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas absensi. Proses dimulai dengan *admin*, guru dan *user* memilih absensi, kemudian sistem menampilkan halaman absensi.

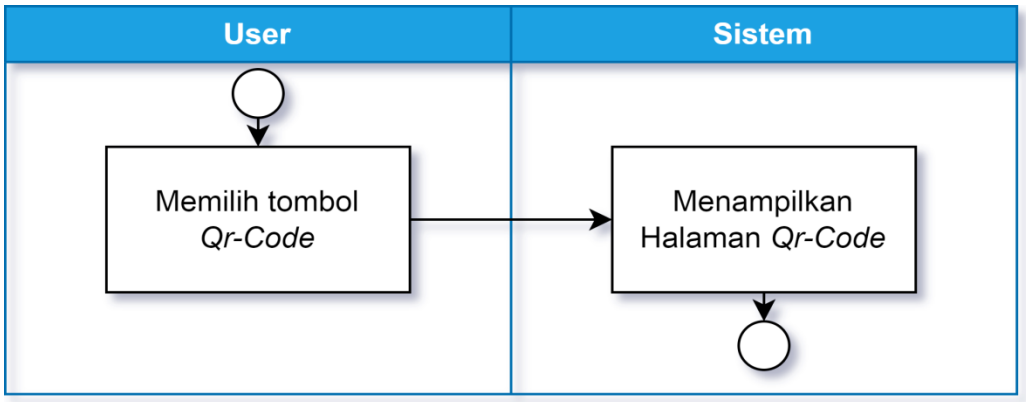
11) *Activity Diagram* Pertemuan



Gambar 4.8. *Activity Diagram* Pertemuan

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas pertemuan. Proses dimulai dengan *admin* dan guru memilih pertemuan, kemudian sistem menampilkan halaman pertemuan.

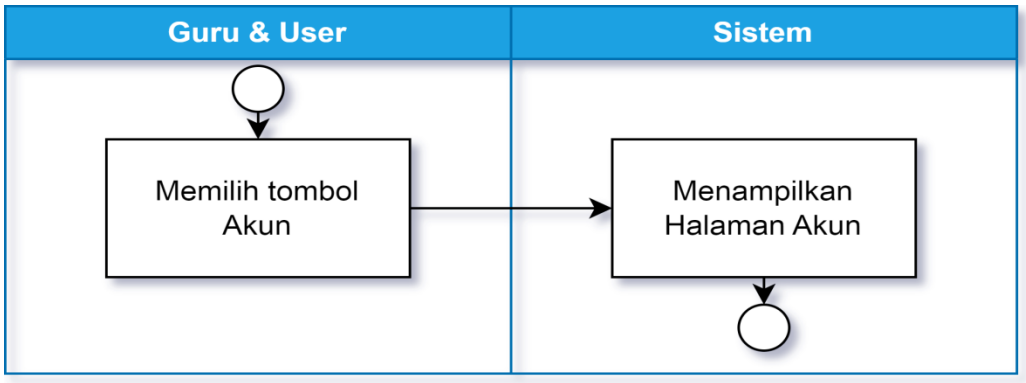
12) *Activity Diagram Qr Code*



Gambar 4.9. *Activity Diagram Qr Code*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas mata pelajaran Proses dimulai dengan *user* memilih *qr code*, kemudian sistem menampilkan halaman *qr code*.

13) *Activity Diagram* Akun



Gambar 4.10. *Activity Diagram* Akun

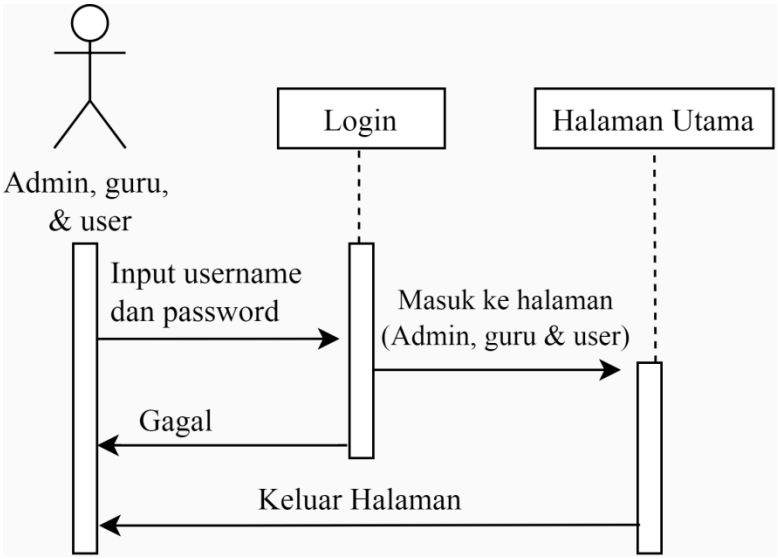
Diagram aktivitas di atas menerangkan alur proses untuk aktivitas kelas Proses dimulai dengan Guru dan *user* memilih akun, kemudian sistem menampilkan halaman akun.

d. *Sequence Diagram*

Sequence Diagram atau *Diagram* sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripisikan 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:

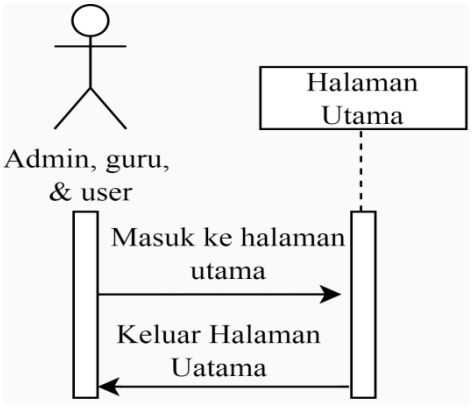
1) *Sequence Diagram*: *Login*



Gambar 4 11 *Sequence Diagram login*

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

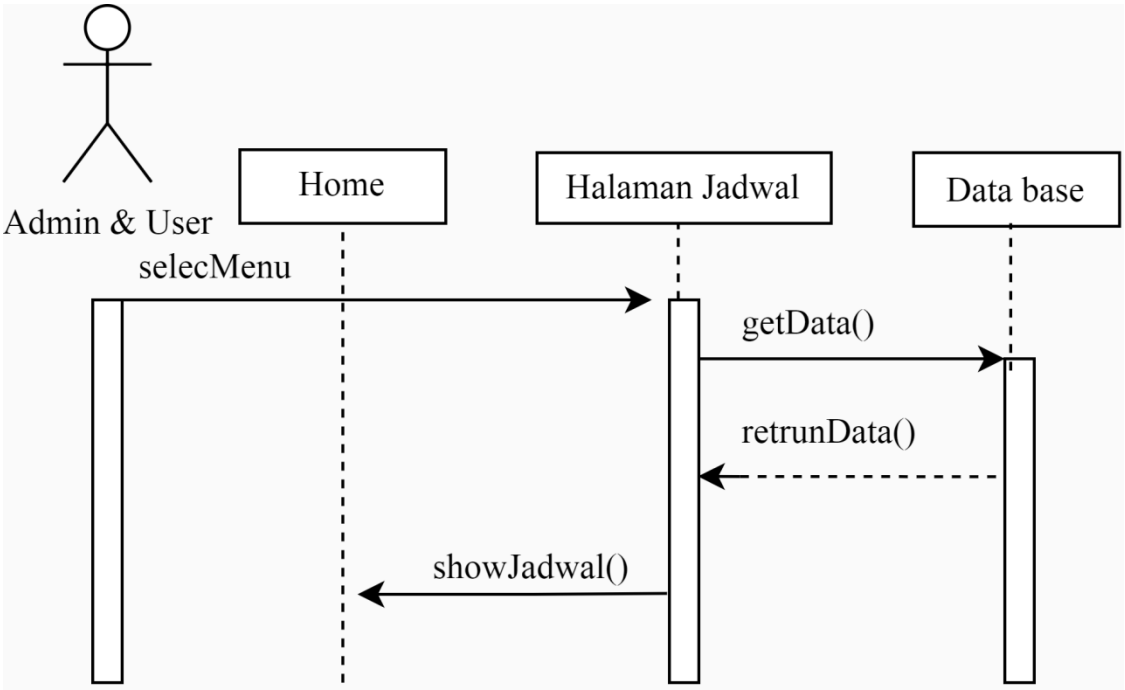
2) *Sequence Diagram*: *Home*



Gambar 4 12 *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.

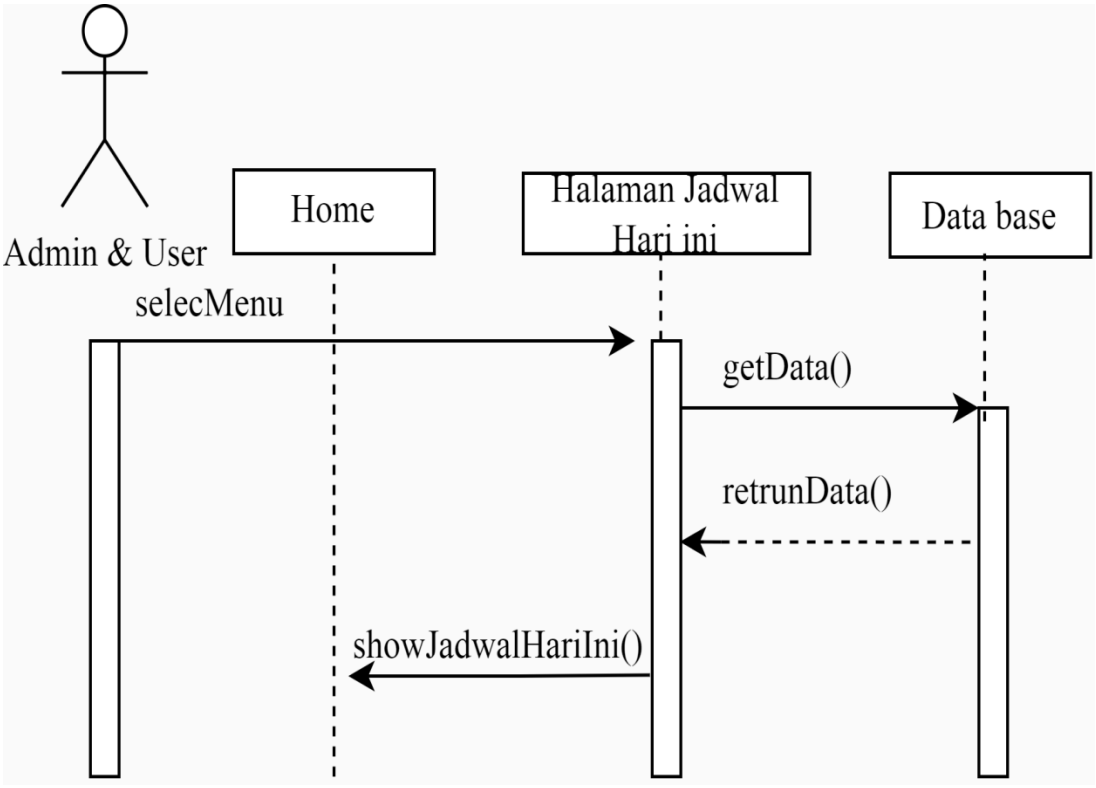
3) *Sequence Diagram*: Jadwal



Gambar 4 13 *Sequence Diagram* Jadwal

Diagram sekuen di atas menerangkan alur proses untuk aktivitas jadwal *User* memilih data yang akan ditampilkan pada halaman jadwal dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowJadwal*() akan menampilkan jadwal.

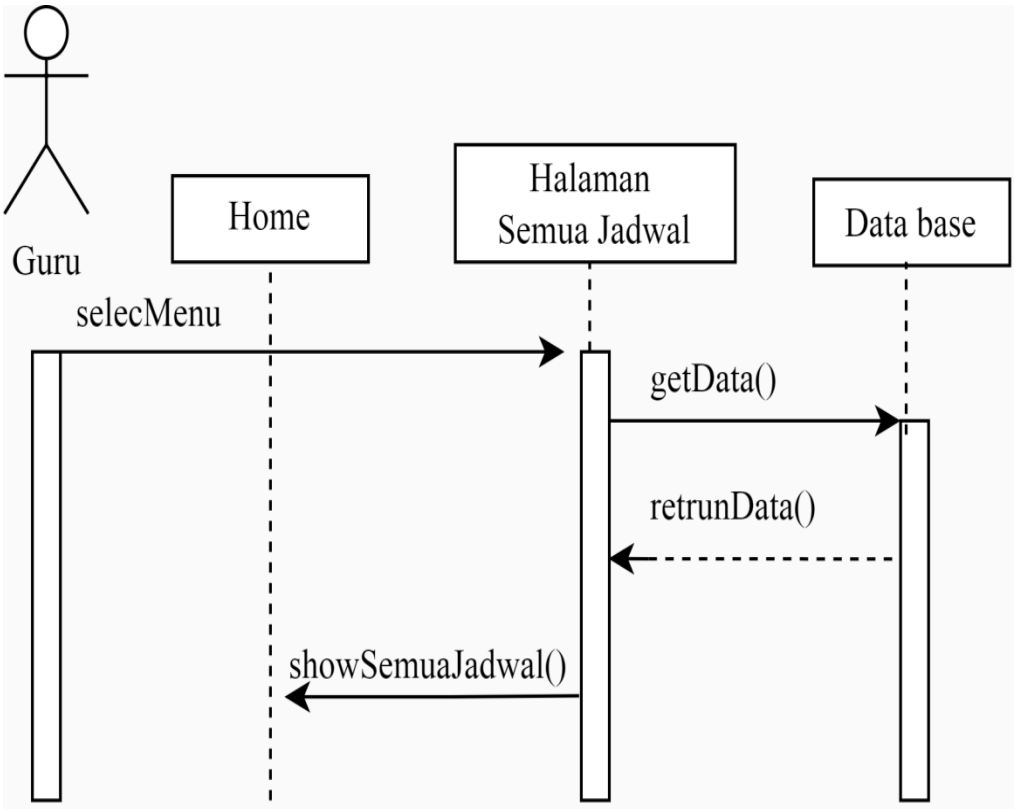
4) *Sequence Diagram*: Jadwal Hari ini



Gambar 4 14 *Sequence Diagram* Jadwal Hari Ini

Diagram sekuen di atas menerangkan alur proses untuk aktivitas jadwal hari ini *Admin* da *user* memilih data yang akan ditampilkan pada halaman jadwal hari ini, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowJadwalHariIni*() akan menampilkan jadwal hari ini.

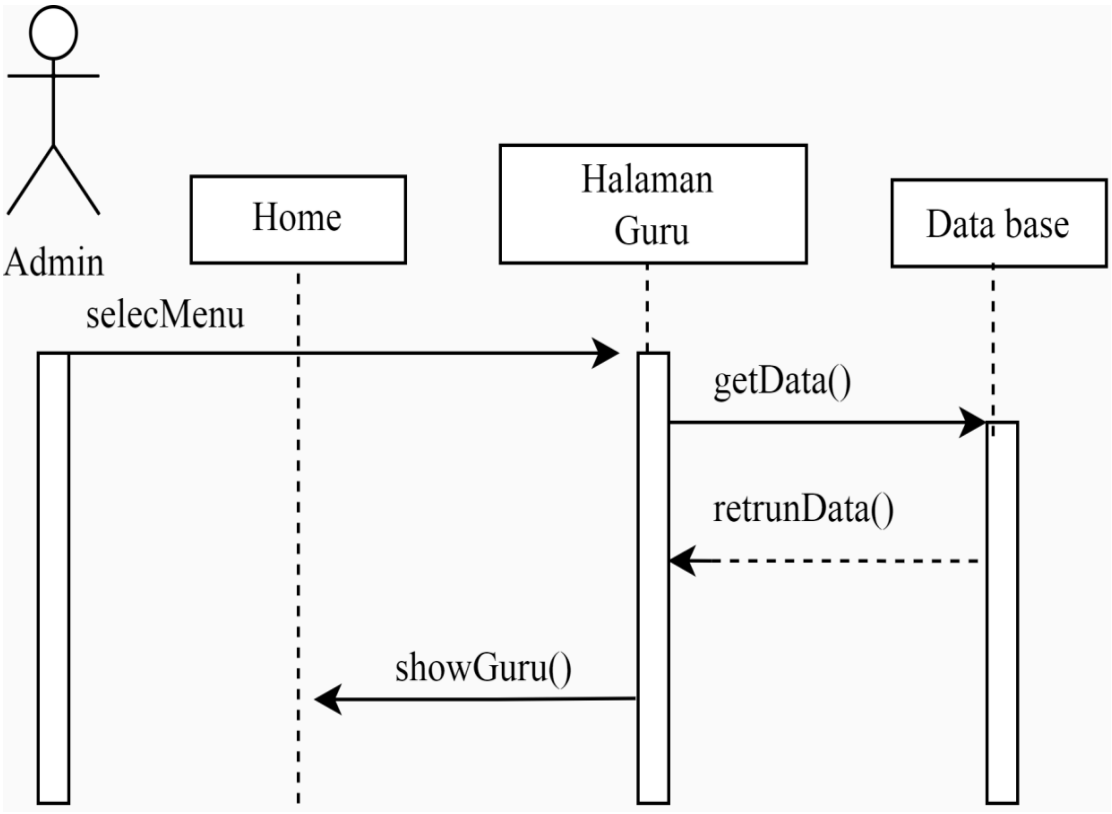
5) *Sequence Diagram*: Semua Jadwal



Gambar 4 15 *Sequence Diagram* Semua Jadwal

Diagram sekuen di atas menerangkan alur proses untuk aktivitas semua jadwal guru memilih data yang akan ditampilkan pada halaman semua jadwal, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowSemuaJadwal*() akan menampilkan Semua Jadwal.

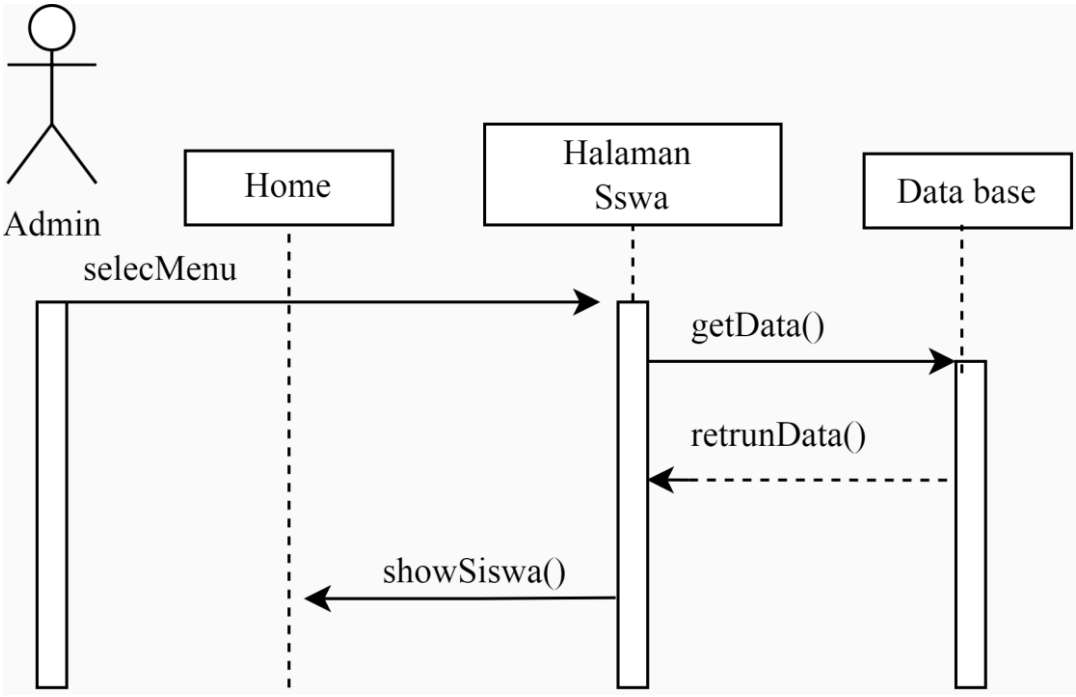
6) *Sequence Diagram*: Guru



Gambar 4 16 *Sequence Diagram* Guru

Diagram sekuen di atas menerangkan alur proses untuk aktivitas guru *admin* memilih data yang akan ditampilkan pada halaman guru, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowGuru*() akan menampilkan guru.

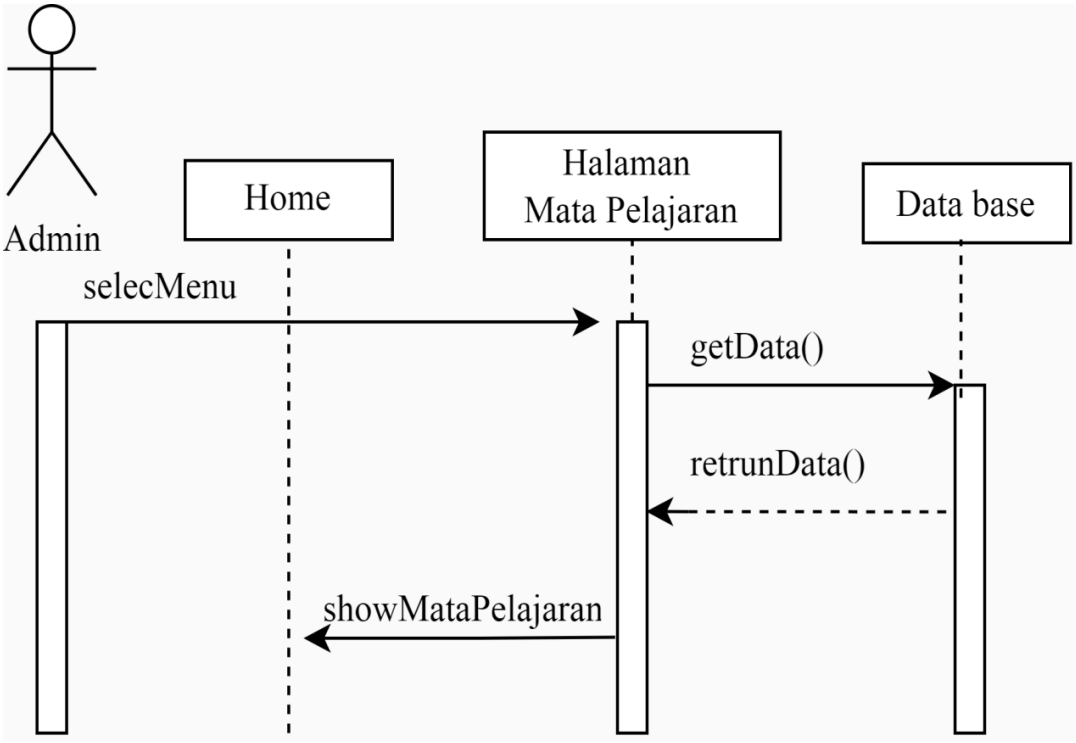
7) *Sequence Diagram*: Siswa



Gambar 4 17 *Sequence Diagram* Siswa

Diagram sekuen di atas menerangkan alur proses untuk aktivitas siswa *Admin* memilih data yang akan ditampilkan pada halaman siswa, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowSiswa*() akan menampilkan siswa.

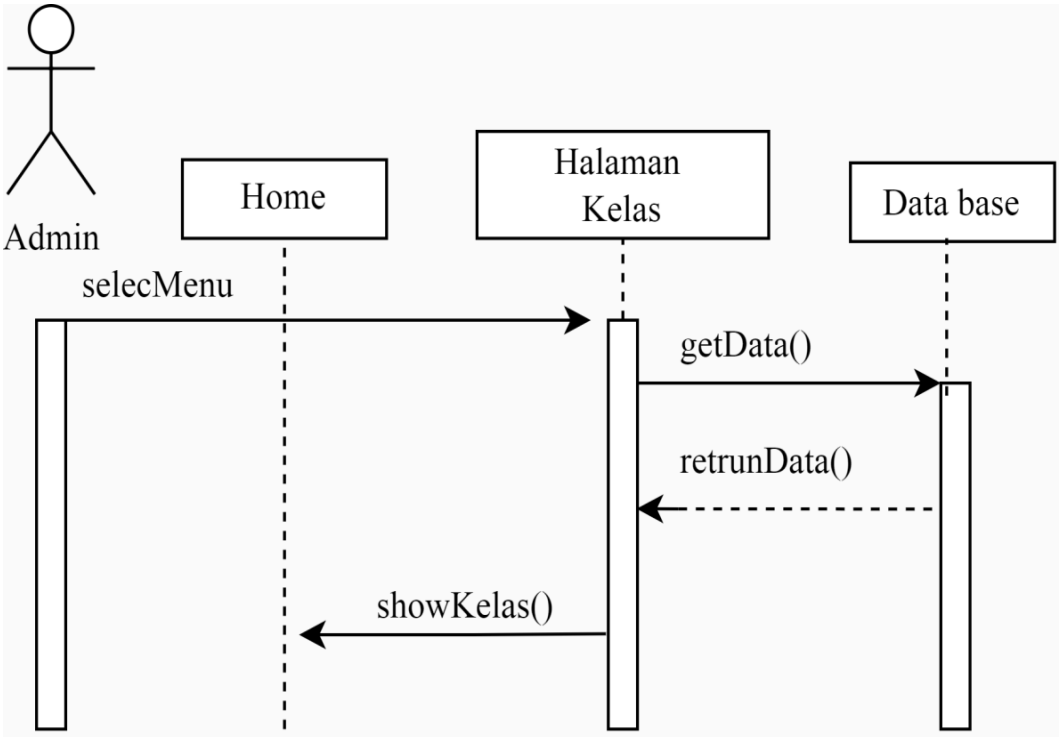
8) *Sequence Diagram*: Mata Pelajaran



Gambar 4 18 *Sequence Diagram* Mata Pelajaran

Diagram sekuen di atas menerangkan alur proses untuk aktivitas mata pelajaran Pengguna *Admin* memilih data yang akan ditampilkan pada halaman mata pelajaran, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowMataPelajaran*() akan menampilkan mata pelajaran.

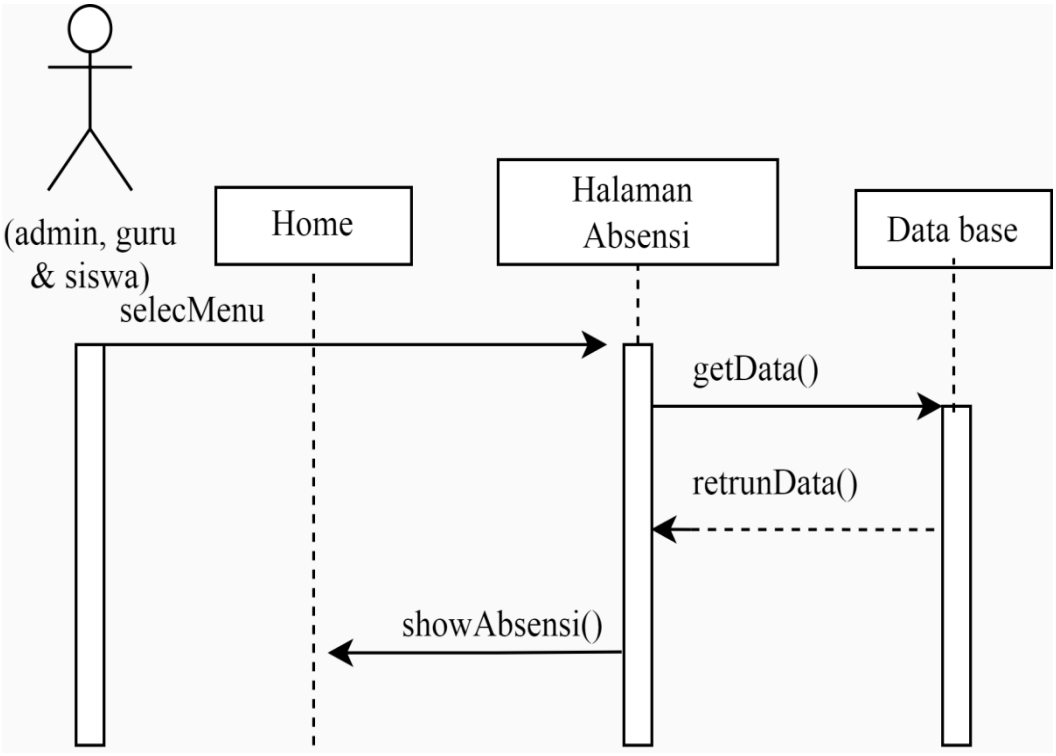
9) *Sequence Diagram*: Kelas



Gambar 4 19 *Sequence Diagram* Kelas

Diagram sekuen di atas menerangkan alur proses untuk aktivitas kelas *Admin* memilih data yang akan ditampilkan pada halaman kelas, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowKelas*() akan menampilkan Kelas.

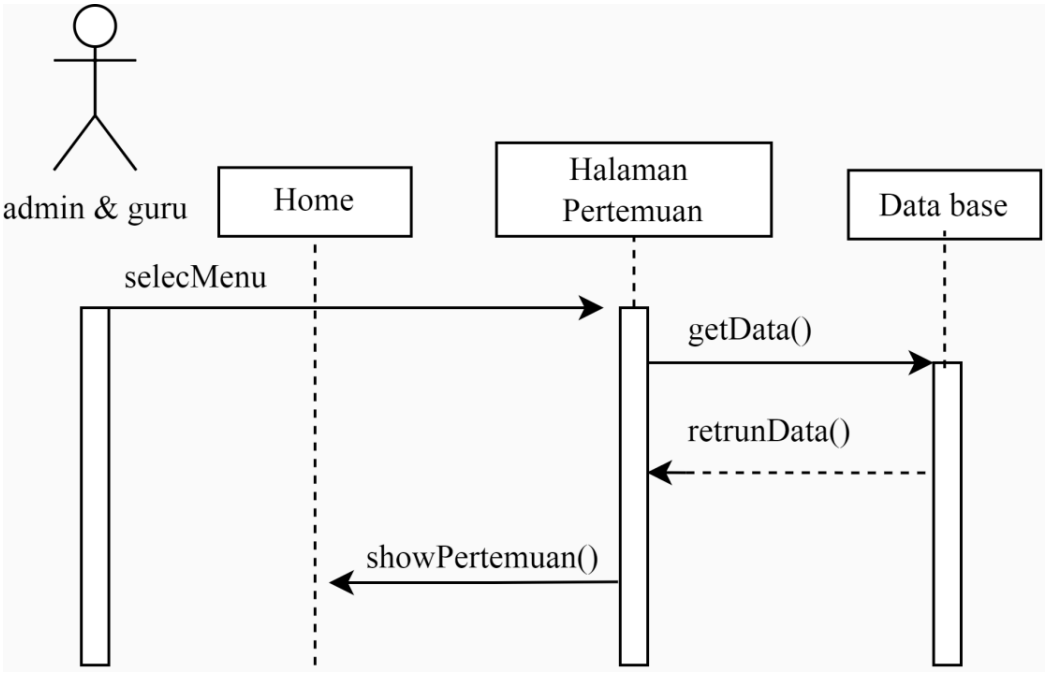
10) *Sequence Diagram*: Absensi



Gambar 4 20 *Sequence Diagram* Absensi

Diagram sekuen di atas menerangkan alur proses untuk aktivitas *admin*, guru dan *user* memilih data yang akan ditampilkan pada halaman absensi, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowAbsensi*() akan menampilkan absensi.

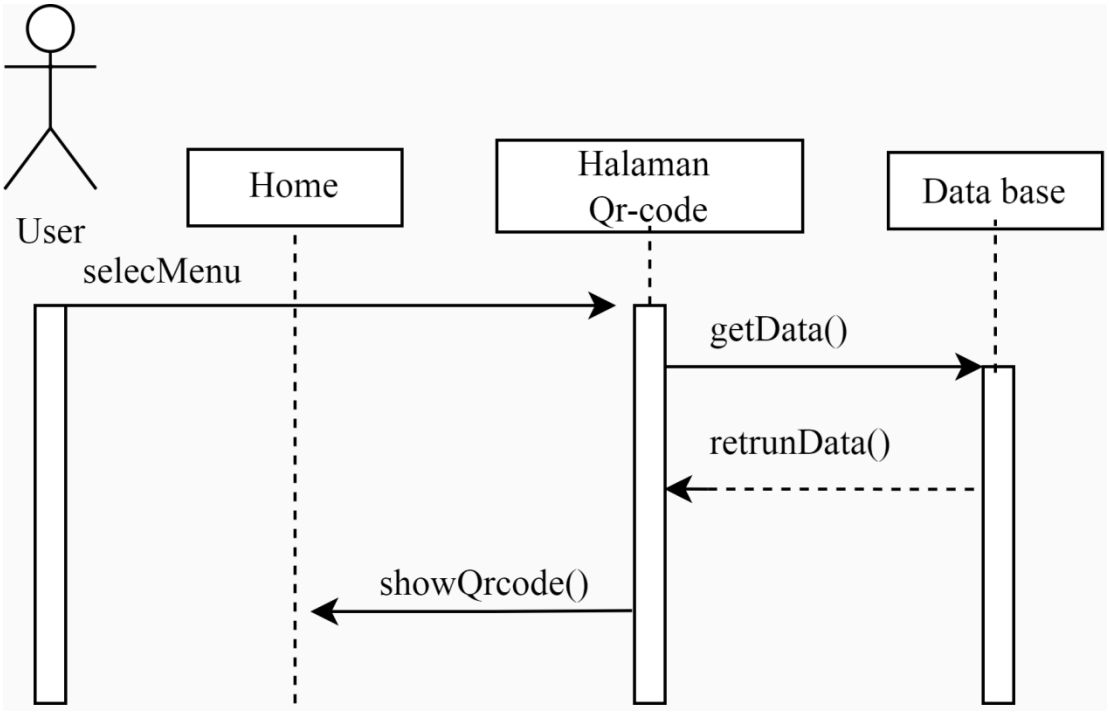
11) *Sequence Diagram*: Pertemuan



Gambar 4 21 *Sequence Diagram* Pertemuan

Diagram sekuen di atas menerangkan alur proses untuk aktivitas siswa *Admin* dan guru memilih data yang akan ditampilkan pada halaman Pertemuan, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowPertemuan*() akan menampilkan Pertemuan.

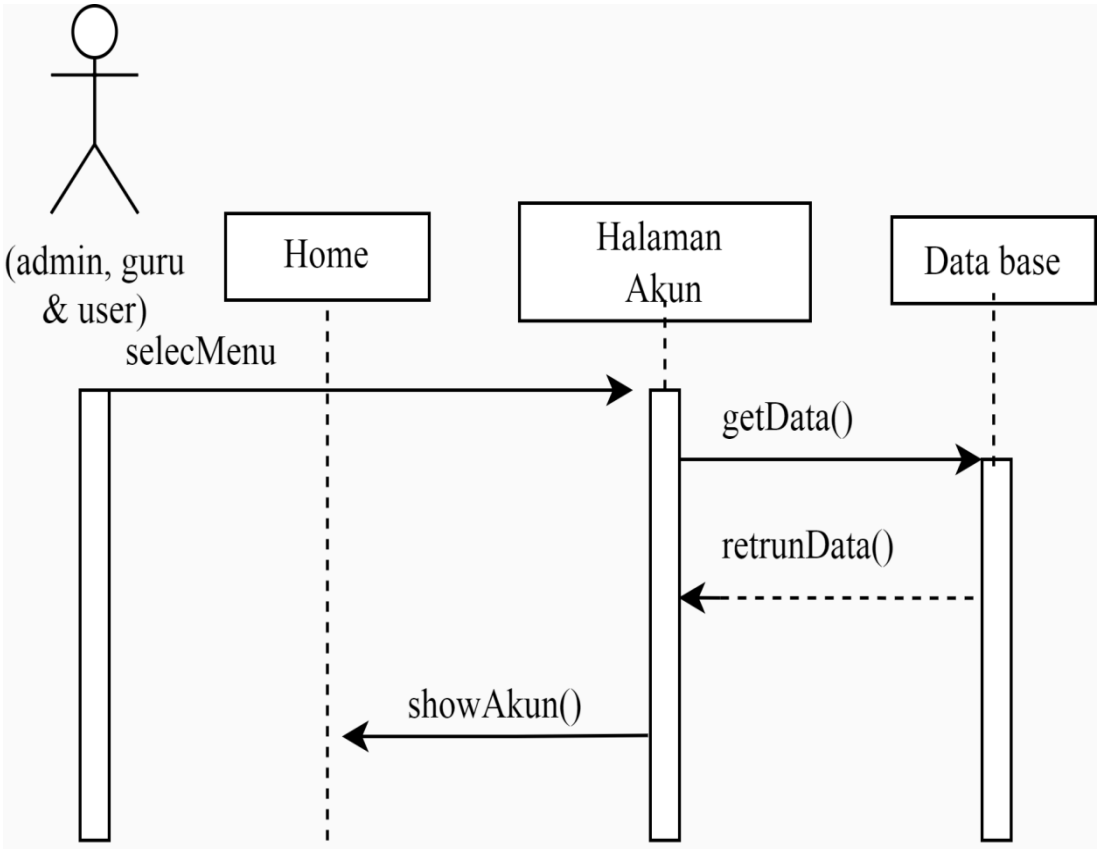
8) *Sequence Diagram*: *Qr Code*



Gambar 4 22 *Sequence Diagram* Qr *Code*

Diagram sekuen di atas menerangkan alur proses untuk aktivitas *qr-code* Pengguna *user* memilih data yang akan ditampilkan pada halaman *qr-code*, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *Showqrcode*() akan menampilkan *qr-code.*

9) *Sequence Diagram*: Akun



Gambar 4 23 *Sequence Diagram* Akun

Diagram sekuen di atas menerangkan alur proses untuk aktivitas akun *Admin*, guru, dan *user* memilih data yang akan ditampilkan pada halaman akun, dengan method *selecMenu*(), kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowAkun*() akan menampilkan akun.

e. 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 4.10. Tabel *User*

No	Nama Field	Tipe Data	Ukuran
1	id_*user*	Varchar	12
2	*User*id	Varchar	12
3	Passid	Varchar	8
4	nama	Varchar	20
5	alamat	Varchar	50

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

Tabel 4.11. Tabel Jadwal.

No	Nama Field	Tipe Data	Ukuran
1	id_Jadwal	Varchar	10
2	Id_Hari	Varchar	10
3	Id_Mata_pelajaran	Varchar	10
4	Id_Ruangan	Varchar	20

c. Tabel Mata_pelajaran Nama table: Mata_pelajaran , Adapun struktur table Mata_pelajaran sebagai berikut.

Tabel 4.12. Tabel Mata_pelajaran.

No	Nama Field	Tipe Data	Ukuran
1	Id_ Mata_pelajaran	Varchar	10
2	Nama_ Mata_pelajaran	Varchar	20
3	kode_ Mata_pelajaran	Varchar	20

d. Tabel Kelas Nama *table:* Kelas Adapun struktur *table* kelas sebagai berikut.

Tabel 4.13. Tabel kelas

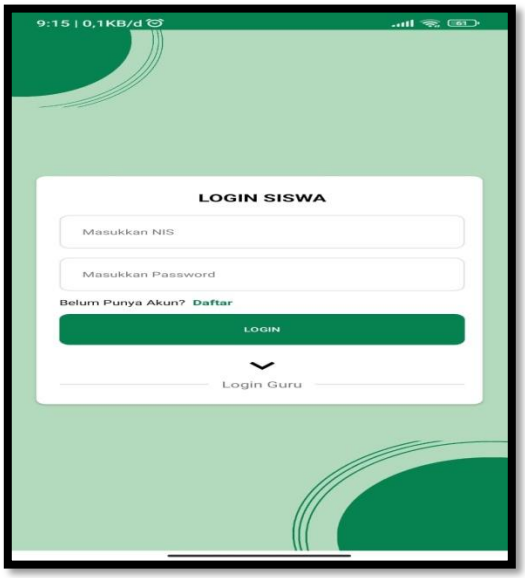
No	Nama Field	Tipe Data	Ukuran
1	id_kelas	Varchar	10
2	kd_matakuliah	Varchar	10
3	Id_ThAkademik	Varchar	20
4	Kapasitas	Varchar	20

2. Perancangan Sistem

a. Halaman *User*

1) Halaman *Login* Akun

Berikut adalah tampilan *login* akun pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman *menu login* merupakan halaman *login user. Login* merupakan tampilan yang ada pada sistem, yang digunakan *user* untuk masuk ke halaman selanjutnya.



Gambar 4 24 *Login* akun

2) Halaman Utama

Berikut adalah tampilan halaman utama pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman ini merupakan halaman awal pelanggan yang lansung menampilkan *menu-menu* yang ada pada halaman ini.



Gambar 4 25 Halaman Utama

3) Halaman Jadwal

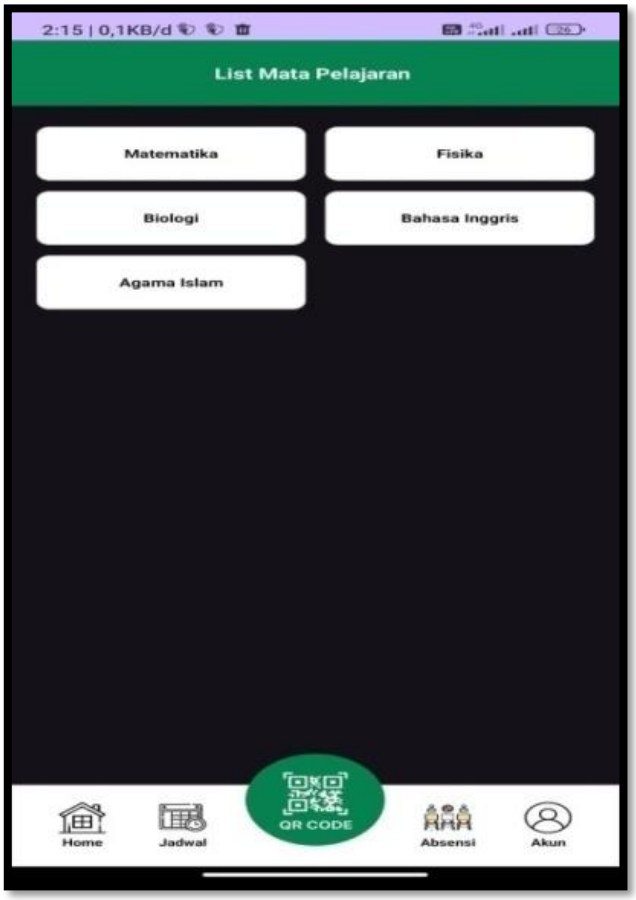
Berikut adalah tampilan jadwal pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman jadwal ini menampilkan semua jadwal mata pelajaran



Gambar 4 26 Halaman Jadwal

4) Halaman Absensi

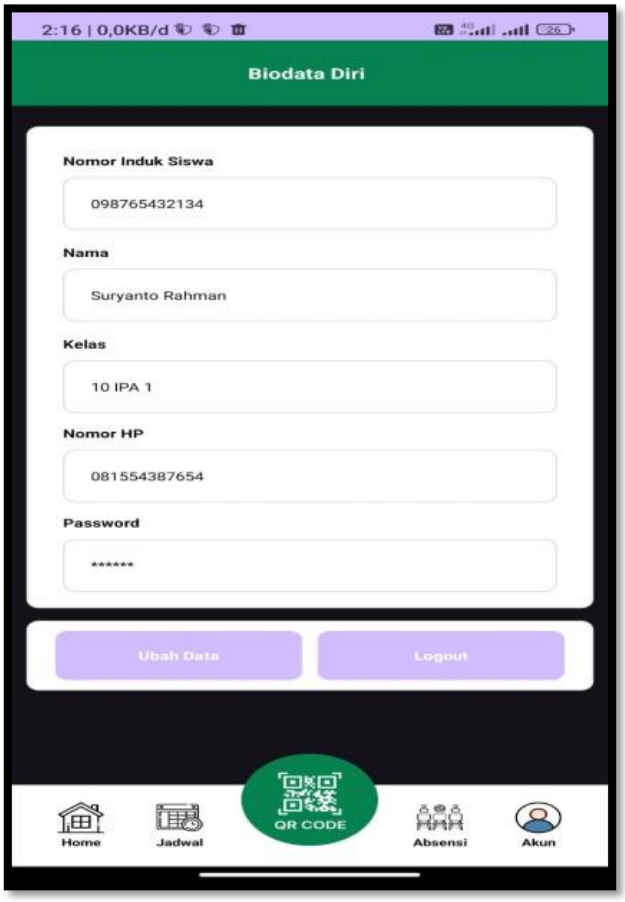
Berikut adalah tampilan absensi pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman absensi menampilkan list mata pelajaran dan kemudian bisa melakukan absensi di mata pelajaran yang dipilih..



Gambar 4 27 Halaman Absensi

5) Halaman Akun

Berikut adalah tampilan akun pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman akun terbaru menampilkan biodata diri siswa..

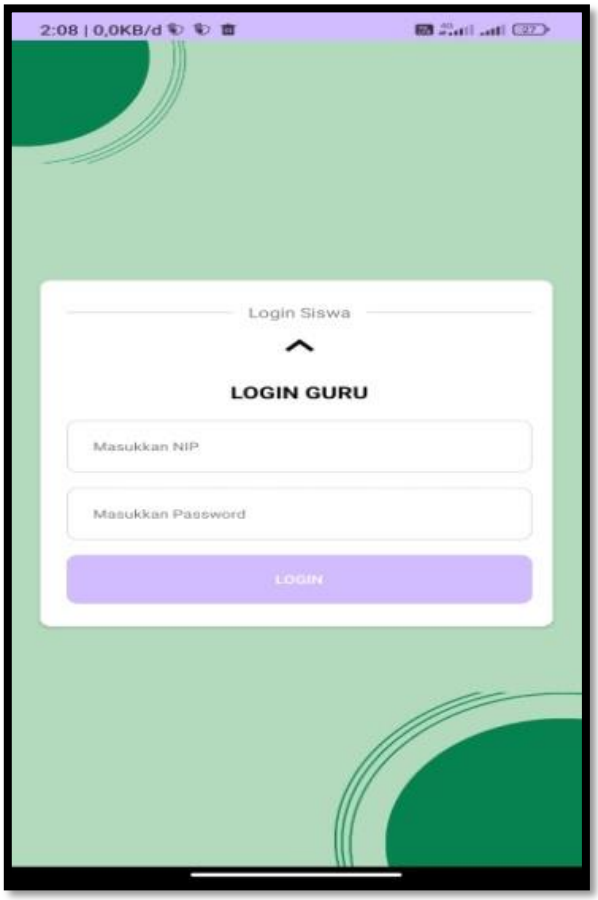


Gambar 4 28 Halaman Akun

b. Halaman Guru

1) Halaman *Login* Akun Guru

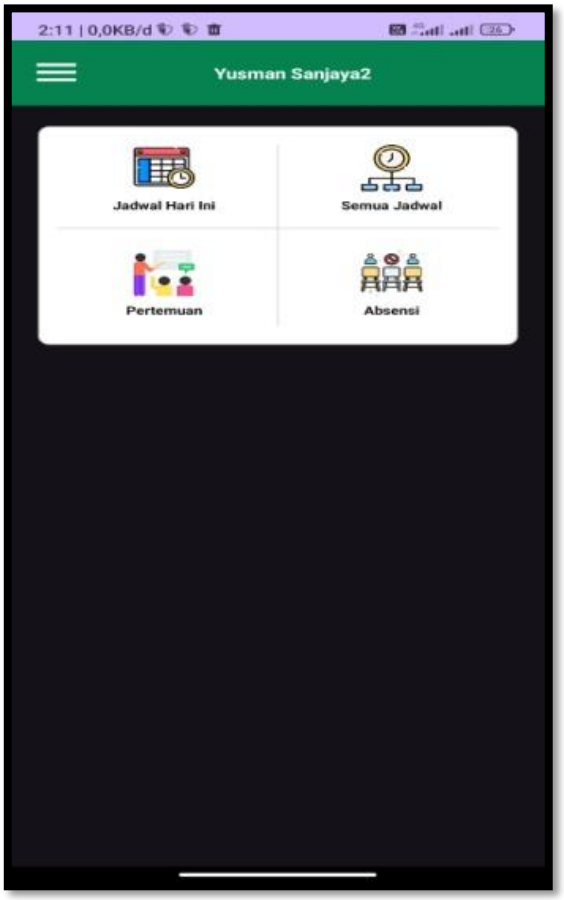
Berikut adalah tampilan *login* akun pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman *menu login* merupakan halaman *login guru. Login* merupakan tampilan yang ada pada sistem, yang digunakan *user* untuk masuk ke halaman selanjutnya.



Gambar 4 29 *Login* akun Guru

2) Halaman Utama

Berikut adalah tampilan halaman utama pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman ini merupakan halaman awal pelanggan yang lansung menampilkan *menu-menu* yang ada pada halaman ini.



Gambar 4 30 *Halaman Utama*

3) Halaman Semua Jadwal

Berikut adalah tampilan semua jadwal pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman semua jadwal menampilkan halaman semua jadwal mengajar.



Gambar 4 31 Halaman Semua Jadwal

4) Halaman Absensi

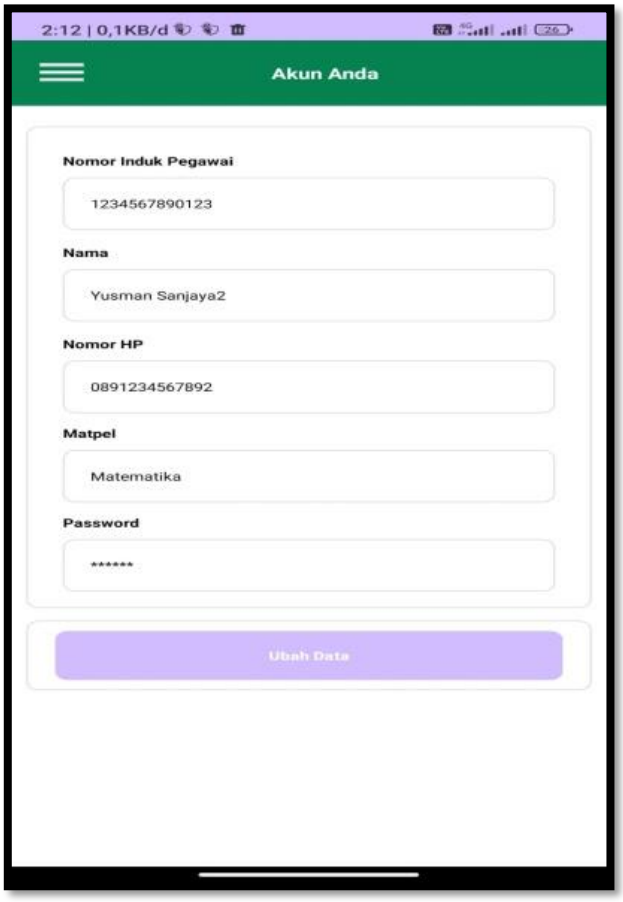
Berikut adalah tampilan absensi pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman absensi menampilkan list mata pelajaran dan kemudian bisa melakukan absensi di mata pelajaran yang dipilih..



Gambar 4 32 Halaman Absensi

5) Halaman Akun Guru

Berikut adalah tampilan akun pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman akun terbaru menampilkan biodata diri siswa..

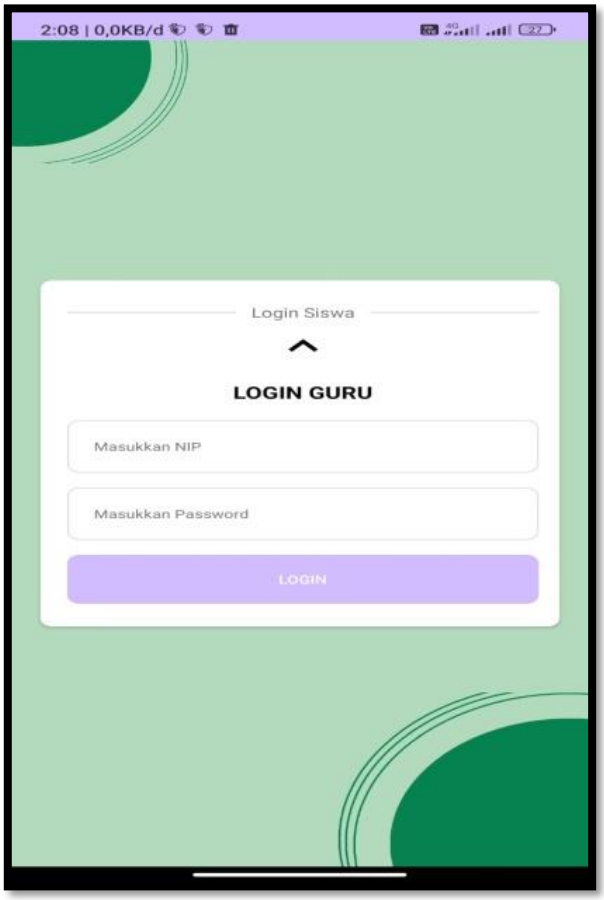


Gambar 4.24. Halaman Akun Guru

c. Halaman Admin

1) Halaman *Login* Akun *Admin*

Berikut adalah tampilan *login* akun pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman *menu login* merupakan halaman *login admin. Login* merupakan tampilan yang ada pada sistem, yang digunakan *user* untuk masuk ke halaman selanjutnya.



Gambar 4 33 *Login* Akun *Admin*

2) Halaman Utama

Berikut adalah tampilan halaman utama pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman ini merupakan halaman awal pelanggan yang lansung menampilkan *menu-menu* yang ada pada halaman ini.



Gambar 4 34 Halaman Utama

3) Halaman Jadwal

Berikut adalah tampilan jadwal pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman jadwal menampilkan halaman semua jadwal mengajar .



Gambar 4 35 Halaman Jadwal

4) Halaman Guru

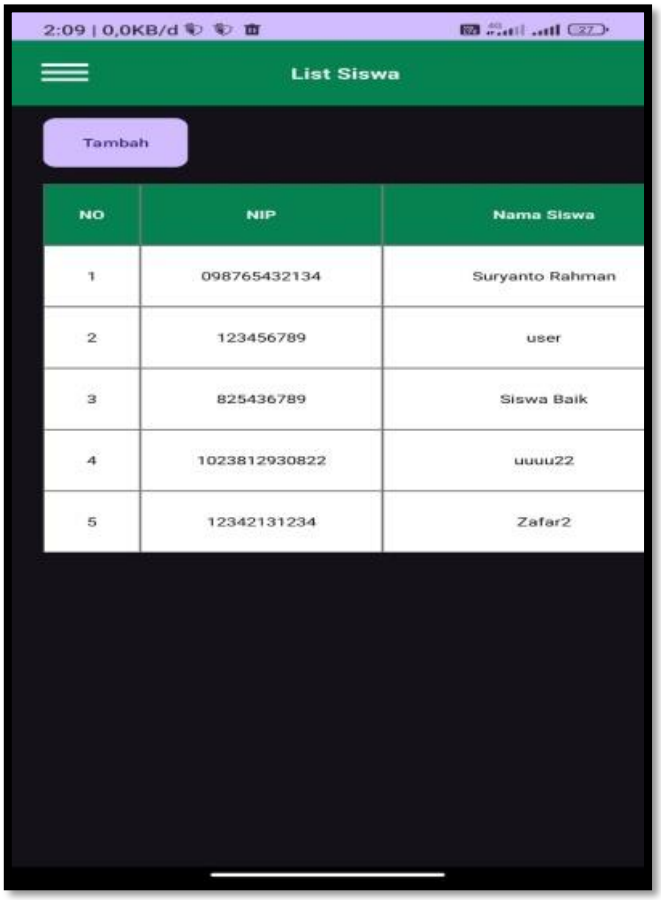
Berikut adalah tampilan guru pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman guru menampilkan halaman list gurulegkap dengan NIP dan Nama Guru.



Gambar 4 36 Halaman Guru

5) Halaman Siswa

Berikut adalah tampilan siswa pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman siswa menampilkan biodata diri siswa legkap degan NIS dan Nama Siswa.



Gambar 4 37 Halaman Siswa

6) Halaman Kelas

Berikut adalah tampilan halaman kelas pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman ini merupakan halaman list kelas menampilkan semua kelas beserta jurusannya..



Gambar 4 38 Halaman Kelas

7) Halaman Absensi

Berikut adalah tampilan absensi pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman absensi menampilkan halaman list absensi .



Gambar 4 39 Halaman Absensi

8) Halaman Pertemuan

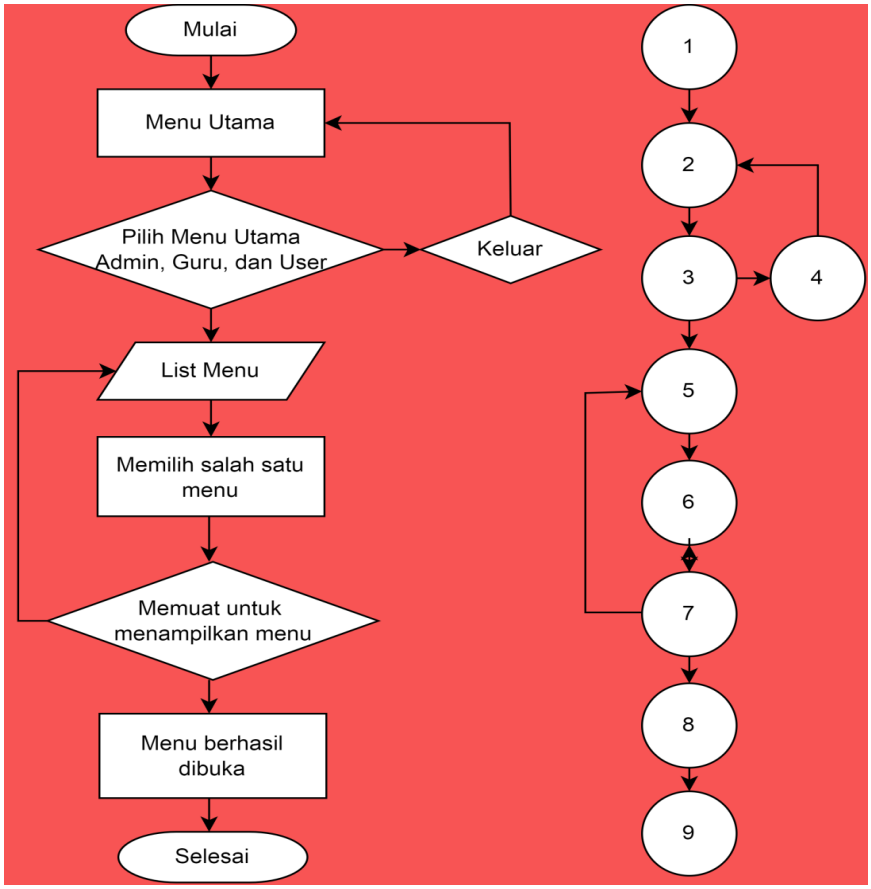
Berikut adalah tampilan pertemuan pada aplikasi penerapan qr*code* sebagai media pelayanan untuk absensi pada siswa sma negeri 5 parepare. Halaman pertemuan menampilkan halaman pertemuan.

3. Pengujian Sistem

a) *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. Berlainan dengan *black box testing*, *white box testing* ini justru menguji perangkat lunak dari sisi internal tanpa memperhatikan fungsional seperti antarmuka perangkat lunak itu sendiri.

1) *White box menu* aplikasi.



Gambar 4 40 *White box menu* aplikasi.

.

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 = 9 E = 11 R = 2

Penyelasaian :

V(G) = (E- N) + 2 = (11 – 9) + 2 = 4

Independent Path :

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

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

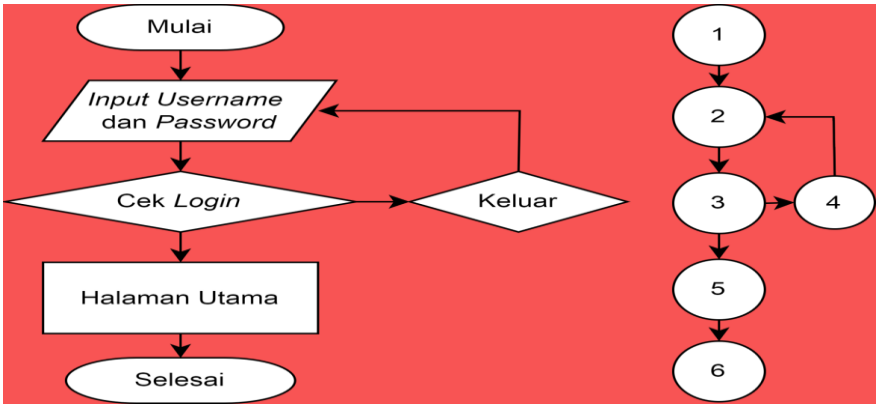
Path 3 = 1 – 2 – 3 – 4 – 9

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

Tabel 4.15. *White Box* Menu Aplikasi

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

2) *White box* pada *menu login*



Gambar 4 41 *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 = 6 E = 6 R = 2

Penyelasaian :

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

= (6 – 6) + 2

= 2

Independent Path :

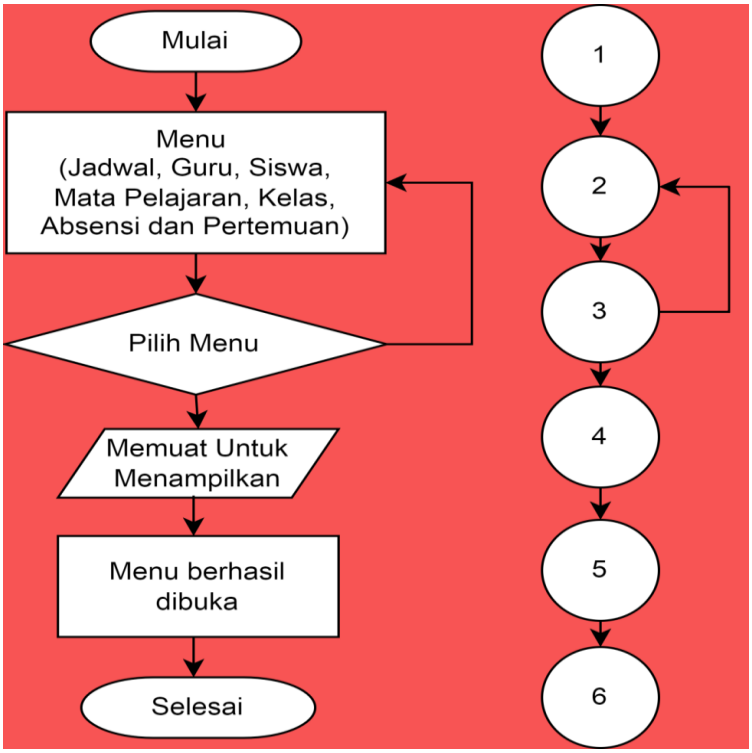
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 –5 – 6

Tabel 4.16. *White Box*Perhituangan menu login

	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

3) *White box* Pada *Menu Admin.*



Gambar 4 42 *White box* Pada *Meenu Admin*

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 E = 6 R = 2

Penyelasaian :

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

= (6 – 6) + 2

= 2

Independent Path :

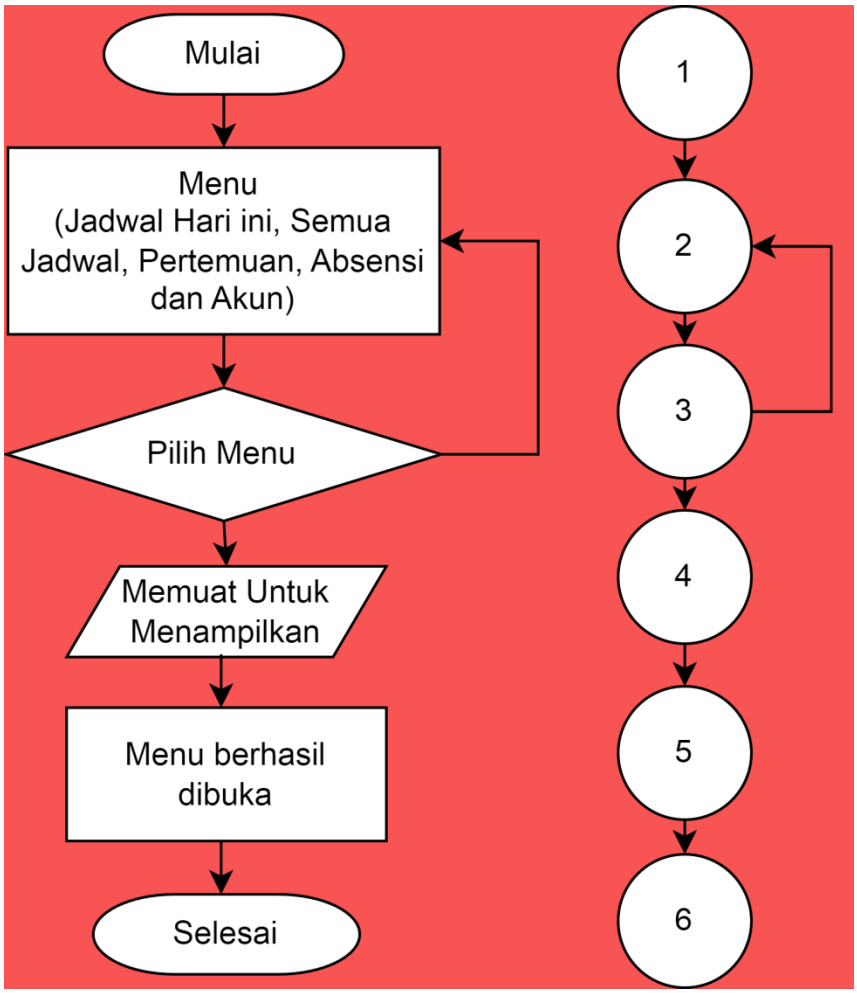
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 –5 – 6

Tabel 4.17. *White Box*Perhituangan Menu Admin

	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

4) *White box* Pada Menu Guru



Gambar 4 43 *White box* Pada *Meenu* Guru

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 E = 6 R = 2

Penyelasaian :

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

= (6 – 6) + 2

= 2

Independent Path :

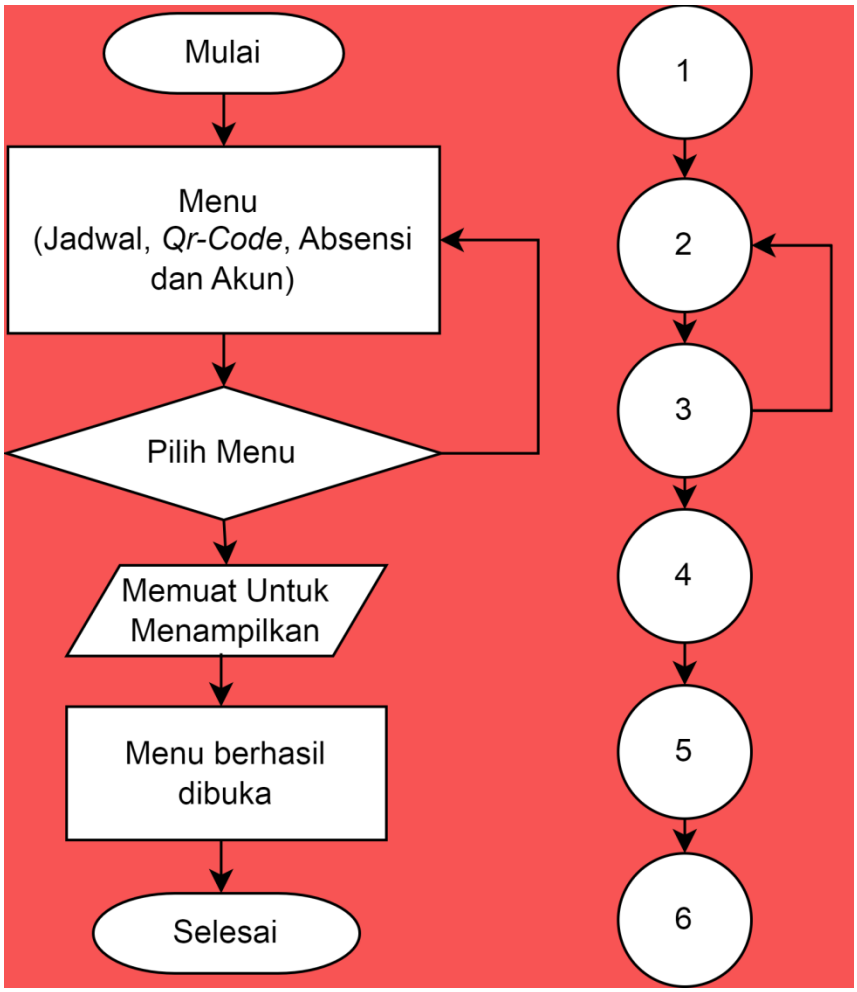
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 –5 – 6

Tabel 4.17. *White Box*Perhituangan Menu Guru

	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

5) *White box* Pada *Menu User*



Gambar 4 44 *White box* Pada *Meenu User*

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 E = 6 R = 2

Penyelasaian :

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 4.17. *White Box*Perhituangan Menu User

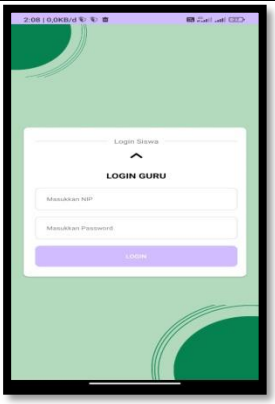
	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3		1		1			2-1=1
4					1		1-1=0
5						1	1-1=0
6							0
SUM(E+1)							1+1=2

b) *Black Box*

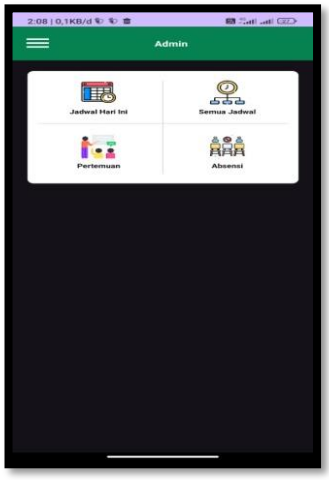
1) *Black Box Admin*

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 4.21. Tabel Pengujian *Login*

Test Faktor	Hasil	Kesimpulan
Login	✔	Sukses, karena ketika *users* memasukkan *user* ID dan *Password*, akan masuk di halaman Utama
Screen Shot		
		

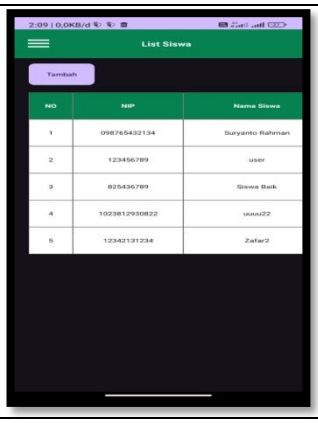
Tabel 4.22. Tabel Pengujian Halaman Utama

Test Faktor	Hasil	Kesimpulan
Halaman Utama	✔	Sukses, karena ketika *users* sudah *login* sebagai *admin*maka lansung masuk ke halaman utama *admin*
Screen Shot		
		

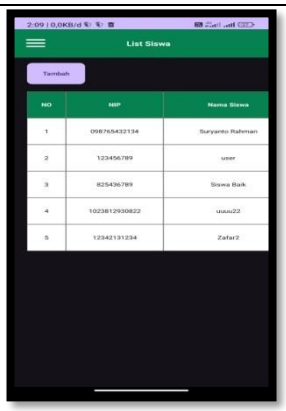
Tabel 4.23. Tabel Pengujian Jadwal

Test Faktor	Hasil	Kesimpulan
Jadwal	✔	Sukses, karena ketika *user*s menekan *icon* jadwal maka lansung menampilkan jadwal pada *menu admin*
Screen Shot		
		

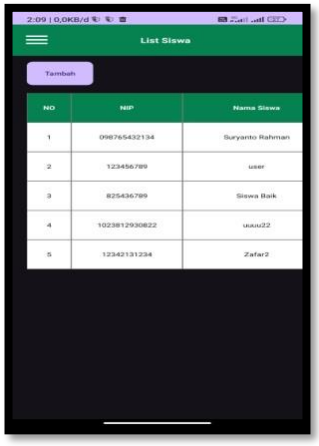
Tabel 4.24. Tabel Pengujian Guru

Test Faktor	Hasil	Kesimpulan
Guru	✔	Sukses, karena ketika *users* sudah *login* sebagai *admin* dan memilih *menu* guru maka lansung masuk ke halaman guru
Screen Shot		
		

Tabel 4.25. Tabel Pengujian Siswa

Test Faktor	Hasil	Kesimpulan
Siswa	✔	Sukses, karena ketika *user*s menekan *icon* siswa maka lansung menampilkan siswa pada *menu admin*
Screen Shot		
		

Tabel 4.26. Tabel Pengujian Mata Pelajaran

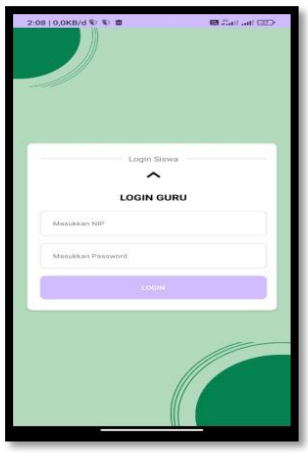
Test Faktor	Hasil	Kesimpulan
Mata Pelajaran	✔	Sukses, karena ketika *users* sudah *login* sebagai *admin* dan memilih *menu* mata pelajaran maka lansung masuk ke halaman tersebut.
Screen Shot		
		

Tabel 4.27. Tabel Pengujian Kelas

Test Faktor	Hasil	Kesimpulan
Kelas	✔	Sukses, karena ketika *user*s menekan *icon* kelas maka lansung menampilkan kelas pada *menu admin*
Screen Shot		
		

2) *Black Box* Guru

Tabel 4.28. Tabel Pengujian *Login*

Test Faktor	Hasil	Kesimpulan
Login	✔	Sukses, karena ketika *users* memasukkan *user* ID dan *Password*, akan masuk di halaman Utama
Screen Shot		
		

Tabel 4.29. Tabel Pengujian Halaman Utama

Test Faktor	Hasil	Kesimpulan
Halaman Utama	✔	Sukses, karena ketika *users* sudah *login* sebagai guru maka lansung masuk ke halaman utama *admin*
Screen Shot		
		

Tabel 4.30. Tabel Pengujian Jadwal Hari ini

Test Faktor	Hasil	Kesimpulan
Jadwal hari ini	✔	Sukses, karena ketika *user*s menekan *icon* jadwal hari ini maka lansung menampilkan *menu* tersebut
Screen Shot		
		

Tabel 4.31. Tabel Pengujian Semua Jadwal

Test Faktor	Hasil	Kesimpulan
Semua Jadwal	✔	Sukses, karena ketika *users* sudah *login* sebagai *admin* dan memilih *menu* semua jadwal maka lansung masuk ke halaman tersebut.
Screen Shot		
		

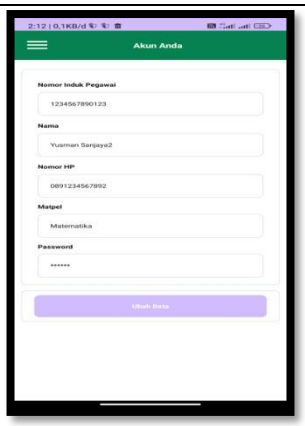
Tabel 4.32. Tabel Pengujian Pertemuan

Test Faktor	Hasil	Kesimpulan
Pertemuan	✔	Sukses, karena ketika *user*s menekan *icon* pertemuan maka lansung menampilkan halaman tersebut pada *menu admin*
Screen Shot		
		

Tabel 4.33. Tabel Pengujian Absensi

Test Faktor	Hasil	Kesimpulan
Absensi	✔	Sukses, karena ketika *users* sudah *login* sebagai *admin* dan memilih *menu* absensi maka lansung masuk ke halaman tersebut.
Screen Shot		
		

Tabel 4.34. Tabel Pengujian Akun

Test Faktor	Hasil	Kesimpulan
Akun	✔	Sukses, karena ketika *user*s menekan *icon* akun maka lansung menampilkan halaman tersebut pada *menu admin*
Screen Shot		
		

3) *Black Box User*

Tabel 4.35. Tabel Pengujian Halaman Utama

Test Faktor	Hasil	Kesimpulan
Halaman Utama	✔	Sukses, karena ketika *users* sudah *login* sebagai *user* maka lansung masuk ke halaman utama *admin*
Screen Shot		
		

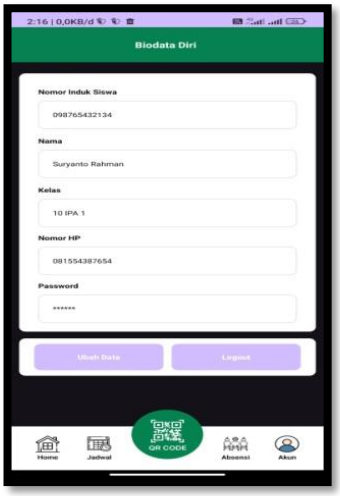
Tabel 4.36. Tabel Pengujian Jadwal

Test Faktor	Hasil	Kesimpulan
Jadwal	✔	Sukses, karena ketika *user*s menekan *icon* jadwal maka lansung menampilkan jadwal pada *menu admin*
Screen Shot		
		

Tabel 4.37. Tabel Pengujian Absensi

Test Faktor	Hasil	Kesimpulan
Absensi	✔	Sukses, karena ketika *users* sudah *login* sebagai *user*/siswa dan memilih *menu* absensi maka lansung masuk ke halaman absensi
Screen Shot		
		

Tabel 4.38. Tabel Pengujian Akun

Test Faktor	Hasil	Kesimpulan
Akun	✔	Sukses, karena ketika *user*s menekan *icon* Akun maka menampilkan halaman tersebut
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 implementasi dan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi ini berjalan baik sesuai dengan fungsinya. Aplikasi ini berjalan pada *smartphone* berbasis *Android* yang merupakan penerapan *qr code* sebagai media pelayanan untuk absensi pada siswa SMA Negeri 5 Parepare memberikan kemudahan kepada Guru dan Siswa dalam melakukan absensi.

B. Saran

Saran yang diberikan sebagai bentuk referensi dari pengembangan sistem yang telah dibangun, yaitu:

1. Untuk penelitian selanjutnya diharapkan dapat mengembangkan lagi sistem yang berbasis *android* dengan menambahkan fitur-fitur terbaru.
2. Diharapkan kepada peneliti selanjutnya menciptakan sebuah sistem pengiriman laporan langsung melalui email kepada para pimpinan atau staf tata usaha.

DAFTAR PUSTAKA

Agus Irawa, Tajudin Noor .2020.,Jurnal INTEKNA, Tahun XIV, No. 2, Nopember 2020 : 102 – 209 yang berjudul "*SISTEM IN*FOR*MASI ABSENSI MAHASISWA (STUDI KASUS JURUSAN ADMINISTRASI BISNIS POLIBAN*"

Aini, Q., Rahardja, U., Fatillah, A. 2020., Sistem Informasi STMIK Raharja, D., & STMIK Raharja Jurusan Sistem Komputer, M. (2018). *Penerapan QRCode Sebagai Media Pelayanan Untuk Absensi Pada Website* Berbasis *PHP Native Application of QRCode as Media Services* for *Attendance on the Website Based Php Natively*. Jurnal Ilmiah SISFOTENIKA, 8(1), 47.

Antonius Irianto Sukowati, 2017., Jurnal Ilmiah KOMPUTASI, Volume 16 No :2, September 2017, p-ISSN 1412-9434/e-ISSN 2549-7227 yang *berjudul "Rancang Bangun Sistem Absensi Mahasiswa Sekolah Tinggi Teknik*.

Coreit, J., Setiawan, E. B., Kurniawan, B. 2015., Studi, P., Informatika, T., Indonesia, U. K., Studi, P., Elektro, T., & Indonesia, U. K. *Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan Radio Frequency Identification* (RFId). 1(2), 44–49.

Fitri Andini, Med Irzal, R. A. 2017. *Perancangan Dan ImplementasiI Sistem Absensi Online* Berbasis *Android* Di Lingkungan Universitas Negeri Jakarta Anantassa Fitri Andini, Med Irzal, Ria Arafiyah Program Studi Ilmu Komputer, FMIPA UNJ. Sistem Informasi, 1(1), 1–10.

Irawan, A., & Noor, T. 2014. Sistem Informasi Absensi Mahasiswa (Studi Kasus Jurusan *Admini*strasi Bisnis Poliban). Jurnal INTEKNA, 2(2), 102–209.

Joseph, A., & Komputer, D. T. 2019. Sistem Kehadiran Mahasiswa yang Cerdas Menggunakan *QR Code*.

Masalha, F. 2018. Sistem Kehadiran Siswa Menggunakan *QR Code*. September.

Muhammad Priyono Tri S, I. N. H. R. P. 2014. Aplikasi Presensi Siswa Menggunakan *Kode Qr (Qr Code)* Berbasis *Android* di Smk Shifa Kalipare Malang. Bimasakti.

Pradana, A. A. 2018. Pemanfaatan Teknologi *Android* Untuk Absensi Mahasiswa Politeknik Negeri Sriwijaya. E-Prints Polsri, 5–20.

Pulungan, A. 2019. *pemanfaatan QR Code dalam Memudahkan proses absensi siswa* berbasis *aplikasi mobile. Masyarakat Telematika Dan In*for*masi : Jurnal Penelitian Teknologi In*for*masi Dan Komunikasi*, 10(1), 1. https://doi.org/10.17933/mti.v10i1.148

Rahardja, U., Aini, Q., & Santoso, N. P. L. (2018). Pengintegrasian YII *Framework* Berbasis API pada Sistem Penilaian Absensi. *Sisfotenika*, 8(2), 140. https://doi.org/10.30700/jst.v8i2.403

Raharjo (2015). Pengertian *Framework dan MySQL* Sisfotenika. https://doi.org/10.30700/jst.

Rendy Rian Chrisna Putra , Fransiskus Panca Juniawan .2017., Jurnal Telematika Vol. 10 No. 1 Februari 2017 146 ISSN : 1979 – 925X e-ISSN : 2442 – 4528 yang berjudul "*PENERAPAN ALGORITMA FISHERFACES UNTUK PENGENALAN WAJAH PADA SISTEM KEHADIRAN MAHASISWA* BERBASIS *ANDROID*

Sukowati, A. I., Fauziah, H., & Purwanto, I. (2017). Rancang Bangun Sistem Absensi dan jenis-jenis absensi Mahasiswa Sekolah Tinggi Teknik Cendekia (STTC) Berbasis *Radio Frequency Identi cation* (RFID) menggunakan Abstrak Pendahuluan Tinjauan Pustaka Radio *Frequency Identi cation*. Jurnal Ilmiah KOMPUTASI, 16(September), 93–100.

Santoso dan Yulianto.2020. Sistem Aplikasi Absensi Menggunakan Teknologi Bar*code* Scanner Berbasis *Android. It Journal Research and Development*, 4(2), 94–103. https://doi.org/10.25299/itjrd.2020.vol4(2).3642