BAB I

PENDAHULUAN

A. Latar Belakang

Aplikasi pembelajaran scale kecapi berbasis android mungkin dikembangkan untuk memudahkan orang dalam mempelajari alat musik tradisional Indonesia yang bernama kecapi. Kecapi adalah alat musik petik yang memiliki banyak peminat di Indonesia dan sering dimainkan dalam berbagai acara tradisional.

Dalam pembelajaran scale kecapi, terdapat beberapa skala yang harus dipelajari, termasuk skala dasar, skala mayor, dan skala minor. Selain itu, pembelajaran scale kecapi juga meliputi teknik-teknik dasar seperti memetik senar dan memperhatikan nada yang dihasilkan.

Dengan adanya aplikasi pembelajaran scale kecapi berbasis android, pengguna dapat dengan mudah mempelajari dan mempraktikkan skala kecapi dimanapun dan kapanpun tanpa harus bergantung pada buku atau guru. Aplikasi ini juga dapat menyediakan fitur-fitur interaktif seperti rekaman suara dan video, serta latihan-latihan yang dirancang khusus untuk membantu pengguna dalam memahami materi pembelajaran.

B. Rumusan Masalah

Rumusan masalah yang diajukan dalam aplikasi pembelajaran scale kecapi berbasis android antara lain:

1. Bagaimana cara memperkenalkan alat musik kecapi pada pengguna yang belum familiar dengan alat musik tersebut?

2. Bagaimana cara mengajar skala dasar, skala mayor, dan skala minor pada pengguna melalui aplikasi pembelajaran scale kecapi berbasis android?

3. Bagaimana cara memberikan feedback dan evaluasi terhadap kemampuan pengguna dalam memainkan skala kecapi melalui aplikasi ini?

4. Bagaimana cara membuat pengguna tertarik dan terus menggunakan aplikasi pembelajaran scale kecapi berbasis android ini untuk belajar?

C. Tujuan Penelitian

Tujuan penelitian aplikasi pembelajaran scale kecapi berbasis android adalah untuk mengembangkan sebuah aplikasi yang efektif dalam membantu pengguna dalam mempelajari dan mempraktikkan skala kecapi dengan mudah dan interaktif. Tujuan penelitian ini meliputi:

1. Meningkatkan pemahaman pengguna tentang alat musik kecapi dan skala-skala yang terdapat di dalamnya melalui aplikasi pembelajaran yang interaktif dan mudah digunakan.

2. Meningkatkan keterampilan pengguna dalam memainkan skala kecapi melalui latihan-latihan dan *feedback* yang diberikan oleh aplikasi.

3. Memberikan solusi alternatif bagi pengguna yang tidak memiliki akses atau waktu untuk belajar langsung dengan guru atau dalam kelas.

4. Menarik minat pengguna dalam mempelajari alat musik tradisional Indonesia dan mempromosikan kecapi sebagai alat musik yang bernilai budaya.

5. Menghasilkan sebuah aplikasi pembelajaran yang dapat dikembangkan lebih lanjut untuk memasukkan materi-materi pembelajaran kecapi yang lebih kompleks dan memungkinkan pengguna untuk berinteraksi dengan pengguna lainnya dalam komunitas musik kecapi secara online.

D. Batasan Masalah

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

1. Pembuatan aplikasi ini hanya ditujukan untuk pengguna yang ingin menggunakan aplikasi pembelajaran scale kecapi.

2. Aplikasi ini hanya membahas mengenai pembelajaran scale kecapi.

3. Aplikasi yang dibuat berbasis Android.

E. Manfaat Penelitian

Manfaat penelitian aplikasi pembelajaran scale kecapi berbasis android adalah sebagai berikut:

1. Memberikan alternatif pembelajaran yang lebih modern dan fleksibel bagi pengguna dalam mempelajari alat musik tradisional Indonesia.

2. Meningkatkan aksesibilitas pengguna dalam mempelajari skala kecapi kapan saja dan di mana saja melalui perangkat seluler mereka.

3. Meningkatkan efisiensi dan efektivitas pembelajaran skala kecapi dengan menyediakan fitur interaktif seperti rekaman suara dan video serta latihan-latihan yang dirancang khusus.

4. Meningkatkan minat masyarakat pada alat musik tradisional Indonesia dan mempromosikan kecapi sebagai alat musik yang bernilai budaya.

5. Memberikan kontribusi pada pengembangan aplikasi pembelajaran di Indonesia dan meningkatkan kualitas industri teknologi di Indonesia.

6. Menjadikan aplikasi pembelajaran scale kecapi berbasis android sebagai referensi bagi pengembang aplikasi pembelajaran alat musik tradisional Indonesia lainnya.

F. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan proposal ini adalah sebagai berikut :

BAB I PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

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

BAB III METODE PENELITIAN

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

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

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

BAB V KESIMPULAN DAN SARAN

Dalam bab ini menjelaskan mengenai kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (Game), dan sebagainya. Menurut Harip Santoso, Aplikasi adalah suatu kelompok file (Form, Class, Report) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi *fixed asset*, dan lain-lain. Rosyadi, (2020).



Gambar 2.1 Contoh Aplikasi

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*Application Suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, Bahasa Pemrograman yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi – aplikasi dalam suatu paket biasanya memiliki

antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

2. Kecapi

Kecapi adalah sebuah alat musik tradisional yang berasal dari Indonesia, terutama daerah Sumatra dan Jawa Barat. Kecapi terbuat dari kayu dengan bentuk yang datar dan berlubang di tengahnya, serta memiliki 10-20 senar yang terbuat dari nilon atau sutra. Kecapi dimainkan dengan cara dipetik dengan jari-jari tangan atau dengan menggunakan alat pemukul kecil. Alat musik ini sering dimainkan sebagai pengiring musik tradisional, seperti lagu-lagu daerah atau dalam acara-acara tradisional seperti pernikahan atau upacara adat. Kecapi juga sering dimainkan secara solo atau dalam kelompok kecapi sebagai bentuk kesenian tradisional yang diwarisi turun-temurun. (Nita, Budi Hermawan.,2021)

a. Bagian-bagian kecapi terdiri dari:

1. Rangka (*body*): Rangka kecapi terbuat dari kayu dan memiliki bentuk yang pipih dan lebar. Bagian ini yang menentukan suara dan karakteristik dari kecapi.

2. Lubang suara (*sound hole*): Lubang suara kecapi biasanya berbentuk bulat atau persegi dan berfungsi untuk memperkuat suara yang dihasilkan dari senar.

3**.** Senar (*strings*): Senar kecapi terbuat dari nilon atau sutra dan dipegang oleh peg senar pada bagian atas rangka kecapi.

4. Papan nada *(fretboard*): Papan nada terletak di atas rangka kecapi dan biasanya terbuat dari kayu yang lebih halus dan licin. Pada bagian ini terdapat

garis-garis fret (*finger position markers*) yang menunjukkan posisi jari pada saat memetik senar.

5. Peg (*tuning pegs*): Peg kecapi berfungsi untuk menegangkan atau mengendurkan senar agar mendapatkan nada yang sesuai.

6. Jembatan (*bridge*): Jembatan kecapi terletak di bagian bawah rangka kecapi dan berfungsi untuk menopang senar dan menghantarkan getaran senar ke rangka kecapi.

7. Pick (*plectrum*): Pick kecapi adalah alat bantu yang digunakan untuk memetik senar kecapi dengan lebih mudah dan cepat. Pick biasanya terbuat dari bahan plastik atau kayu tipis.

Itulah beberapa bagian-bagian kecapi yang umumnya ada pada kebanyakan jenis kecapi. Bagian-bagian tersebut bekerja sama untuk menghasilkan suara yang indah dan khas dari kecapi.

b. Teknik bermain kecapi

Berikut adalah beberapa teknik bermain kecapi yang umumnya digunakan.

1. *Melody*: Teknik ini digunakan untuk memetik senar satu per satu untuk menghasilkan sebuah melodi yang jelas dan terdengar harmonis. Pada teknik ini, setiap jari pada tangan dimanfaatkan untuk memetik senar dengan cepat dan akurat.

2. *Harmoni*: Teknik ini digunakan untuk memetik dua atau lebih senar pada saat yang sama untuk menghasilkan akord dan harmoni yang berbeda-beda.

Teknik ini memerlukan keterampilan dan kreativitas dalam memilih kombinasi senar dan posisi jari.

3. *Trill*: Teknik trill dilakukan dengan memetik senar yang sama secara cepat dan berulang-ulang dengan jari yang sama. Teknik ini umumnya digunakan untuk menciptakan efek suara yang cepat dan energik.

4. *Tremolo*: Teknik tremolo dilakukan dengan memetik satu senar secara cepat secara terus-menerus dengan jari yang sama. Teknik ini umumnya digunakan untuk memberikan efek suara yang konstan dan berirama.

5. *Sliding*: Teknik sliding dilakukan dengan memetik senar lalu menggeser jari ke posisi yang lebih tinggi atau rendah untuk menghasilkan nada yang berbeda. Teknik ini umumnya digunakan untuk menciptakan suara yang halus dan eksperimental.

Itulah beberapa teknik bermain kecapi yang umumnya digunakan. Selain teknik-teknik di atas, masih banyak teknik lain yang bisa dipelajari dan dikuasai untuk menghasilkan suara yang indah dan khas dari kecapi.

c. *Chord* Kecapi

Chord kecapi adalah kombinasi beberapa senar pada kecapi yang dimainkan secara bersamaan untuk menghasilkan akor atau suara yang harmonis. Berikut adalah beberapa chord kecapi yang umumnya digunakan:

C: Dalam *chord* ini, senar pertama pada posisi ketiga dan senar keempat pada posisi kedua dimainkan bersama-sama.

D: Dalam *chord* ini, senar pertama dan kedua pada posisi kedua dimainkan bersama-sama.

E: Dalam *chord* ini, senar pertama pada posisi kelima dan senar keempat pada posisi kedua dimainkan bersama-sama.

F: Dalam *chord* ini, senar pertama pada posisi satu dan senar kedua pada posisi dua dimainkan bersama-sama.

G: Dalam *chord* ini, senar pertama pada posisi ketiga dan senar kedua pada posisi empat dimainkan bersama-sama.

A: Dalam *chord* ini, senar kedua pada posisi empat dan senar ketiga pada posisi lima dimainkan bersama-sama.

B: Dalam *chord* ini, senar pertama pada posisi ketujuh dan senar kedua pada posisi enam dimainkan bersama-sama.

Setiap *chord* dapat dimainkan dalam berbagai variasi, baik itu menggunakan teknik melodi maupun harmoni. Hal ini tergantung pada keahlian dan kreativitas dari pemain kecapi itu sendiri. Selain itu, masih banyak chord kecapi lainnya yang dapat dipelajari dan dikuasai.

d. Birama/Nada Kecapi

Birama adalah istilah yang merujuk pada jumlah ketukan dalam satu melodi. Dengan kata lain, birama merupakan gabungan dari ketukan-ketukan nada yang teratur sehingga jadi satu kesatuan melodi yang utuh pada sebuah musik. Tanda birama umumnya ditulis dalam bentuk pecahan, seperti 2/4, ¾, dan 4/4.

Sedangkan pengertian ketukan adalah panjang pendeknya durasi nada. Adapun panjang atau pendeknya suatu nada tidak menggunakan dimensi maupun

satuan waktu, melainkan menggunakan ketukan. Nilai ketukan terbesar merupakan satu ketukan dan bisa jadi lebih kecil yang kemudian dipecah lagi menjadi ½, ¼, hingga 1/16.

Contoh Nada 2 Ketukan dalam Musik

Mengutip dari buku *Mandiri Belajar Tematik SD/MI Kelas 2 Semester 1* karya *Nidaul Janah* (2021:56), tanda birama 2/4 artinya setiap birama terdapat dua hitungan dan setiap hitungan bernilai seperempat atau ada dua not seperempat dalam setiap birama. Dengan kata lain, tiap birama terdiri dari dua ketukan.

Berikut adalah pola ketukan birama 2/4.

a. Ọ ● / Ọ ● / Ọ ● b. ● Ọ / ● Ọ / ● Ọ

1 2 1 2 1 2 1 2 1 2 1 2

T K T K T K K T K T K T

Sedangkan untuk contoh nada 2 ketukan pada lagu nusantara yang menggunakan birama 2/4 adalah sebagai berikut.

Hari Merdeka (lagu nasional).

e. Algoritma Memainkan Kecapi

Algoritma memainkan kecapi dapat dilakukan dengan beberapa teknik, di antaranya:

1) Dijambret

Teknik ini dilakukan dengan menggunakan tiga jari tangan kanan, yaitu jempol, jari tengah, dan telunjuk, untuk memetik tiga senar secara bersamaan. Teknik ini menghasilkan harmoni dari tiga nada yang

berbeda. Teknik ini biasanya digunakan untuk mengiringi lagu-lagu dengan tempo cepat.

2) Sinteruk toel

Teknik ini dilakukan dengan menjentikan senar kecapi menggunakan ujung jari telunjuk tangan kanan dan kiri.

3) Dijeungkalan

Teknik ini dilakukan dengan menggunakan tangan kanan untuk memainkan tiga senar nada secara bersamaan, sedangkan tangan kiri untuk memetik senar dan memainkan nada dengan bebas. Teknik ini menghasilkan tiga nada di dalam satu waktu.

Selain teknik-teknik tersebut, kecapi juga dapat dimainkan dengan cara menekan bagian bawahnya untuk menghasilkan suara baru, atau dengan membunyikan keseluruhan kawat secara bersamaan.

3. Scale Kecapi

Skala pada kecapi biasanya disebut dengan istilah "laras" dan terdiri dari lima buah laras. Lima laras tersebut adalah laras pelog, laras slendro, laras sorog, laras manyura, dan laras barang. Setiap laras memiliki karakteristik yang berbeda dan dapat digunakan untuk membentuk melodi yang unik.

Laras pelog terdiri dari tujuh not atau "sanga" dan memiliki karakteristik suara yang agak seram atau menakutkan. Laras slendro terdiri dari lima not atau "lima" dan memiliki karakteristik suara yang lebih ceria dan cerah. Laras sorog terdiri dari sembilan not atau "nava" dan memiliki karakteristik suara yang sangat beragam, dari yang seram hingga yang cerah. Laras manyura terdiri dari delapan

not atau "asta" dan memiliki karakteristik suara yang lembut dan indah. Laras barang terdiri dari enam not atau "enem" dan memiliki karakteristik suara yang kuat dan agak keras.

Pemain kecapi dapat memilih salah satu laras yang ingin digunakan dalam sebuah lagu atau juga dapat mengkombinasikan beberapa laras untuk menciptakan variasi melodi yang lebih kaya dan menarik. Selain itu, pemain kecapi juga dapat menggunakan teknik-teknik seperti glissando, hammer-on, pull-off, dan vibrato untuk memperkaya suara yang dihasilkan dari laras-laras tersebut.

4. Android

Android adalah sistem operasi untuk perangkat mobile yang dikembangkan oleh Google. Sistem operasi ini dirancang khusus untuk perangkat mobile seperti smartphone, *tablet*, dan *smartwatch.* Android sangat populer dan banyak digunakan di seluruh dunia karena kemampuannya untuk disesuaikan dan diubah sesuai dengan kebutuhan penggunanya.

Android memiliki antarmuka pengguna yang sederhana dan mudah digunakan, serta menyediakan akses mudah ke berbagai aplikasi dan layanan Google seperti Google Play Store, Google Maps, Gmail, dan banyak lagi. Selain itu, Android juga mendukung banyak fitur seperti multi-tasking, notifikasi, dan sinkronisasi data.

Saat ini, Android menjadi sistem operasi mobile yang paling banyak digunakan di seluruh dunia, dan terus berkembang dengan fitur-fitur baru dan

inovasi yang dibuat oleh para pengembang aplikasi dan perangkat mobile. Jika Anda memiliki perangkat mobile seperti smartphone atau tablet, kemungkinan besar perangkat tersebut menggunakan sistem operasi Android.

Android merupakan sistem operasi yang didistribusikan secara *open source* oleh Google, atau dengan kata lain operating system ini dapat dikelola oleh berbagai pihak tanpa membutuhkan lisensi khusus (Lee, 2012). Sistem operasi yang diperuntukan bagi smartphone ini berbasis sistem operasi Linux. Pada setiap versi Android memiliki versi API tersendiri, hingga pertanggal 19 Agustus 2015, Android telah merilis 23 tingkatan API. API (*Application Programming Interface*) merupakan sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu (Wahyu Setia Bintara.2023).



Gambar 2.3 Android
(Sumber. https://dianisa.com/pengertian-android/)

a. Sejarah Android

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc. pada tahun 2003. Perusahaan ini didirikan oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada awalnya, Android Inc. bertujuan untuk mengembangkan sistem operasi yang dapat digunakan pada kamera digital. Namun, pada tahun 2005, perusahaan ini diakuisisi oleh Google.

Setelah diakuisisi oleh Google, pengembangan Android terus berlanjut dan pada tahun 2007, Google mengumumkan Android sebagai sistem operasi mobile yang terbuka dan berbasis Linux. Pada tahun 2008, Android resmi diluncurkan dengan dukungan dari berbagai produsen perangkat mobile, termasuk HTC, Samsung, dan Motorola.

Versi pertama dari Android, Android 1.0, diluncurkan pada September 2008 dan dilengkapi dengan fitur-fitur seperti notifikasi, kamera, dan akses ke Google Maps. Android 1.0 juga memperkenalkan aplikasi Google Search dan Google Maps, serta dukungan untuk akses ke Gmail dan YouTube.

Sejak itu, Android terus mengalami perkembangan dengan fitur-fitur baru dan inovasi seperti multi-tasking, pengenalan suara, dan kontrol gerakan. Android juga semakin populer dengan munculnya berbagai aplikasi dan game yang tersedia di Google Play Store. Saat ini, Android menjadi sistem operasi mobile yang paling banyak digunakan di seluruh dunia dan terus berkembang dengan inovasi dan fitur-fitur baru. (Wahyu Setia Bintara.2023)

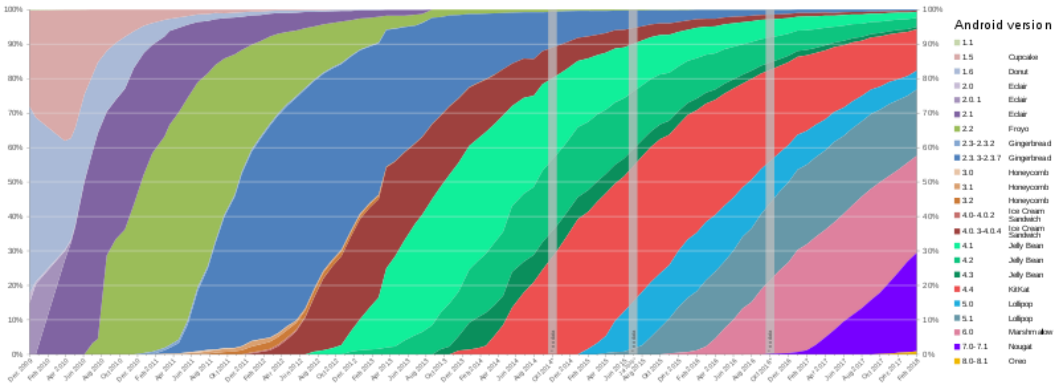
b. Versi Android

Berikut adalah beberapa versi dari sistem operasi Android yang pernah dirilis oleh Google:

Android 1.0 (2008), Android 1.5 Cupcake (2009), Android 1.6 Donut (2009), Android 2.0/2.1 Eclair (2009), Android 2.2 Froyo (2010), Android 2.3 Gingerbread (2010), Android 3.0/3.1/3.2 Honeycomb (2011), Android 4.0 Ice Cream Sandwich (2011), Android 4.1/4.2/4.3 Jelly Bean (2012), Android 4.4 KitKat (2013), Android 5.0/5.1 Lollipop (2014), Android 6.0 Marshmallow

(2015), Android 7.0/7.1 Nougat (2016), Android 8.0/8.1 Oreo (2017) , Android 9.0 Pie (2018), Android 10 (2019), Android 11 (2020), Android 12 (2021). (Wahyu Setia Bintara.2023).

Setiap versi Android memiliki fitur-fitur baru dan perbaikan yang membuat sistem operasi menjadi lebih baik dan lebih efisien. Selain itu, setiap versi Android juga memiliki nama kode yang berbeda-beda, seperti Cupcake, Donut, dan Jelly Bean. Nama kode ini biasanya diambil dari makanan pencuci mulut atau jajanan yang sering dikonsumsi di Amerika Serikat.



Gambar 2.4 Distribusi Versi Android Secara Global

(Sumber: https://id.wikipedia.org/wiki/Berkas:Android_historical_version_distribution_-_vector.svg)

c. Fitur Android

Android tersedia secara terbuka bagi manufaktur perangkat keras untuk memodifikasi sesuai kebutuhan. Konfigurasi perangkat android tidak sama antara satu perangkat dengan perangkat lainnya, namun android sendiri mendukung fitur-fitur tertentu, menurut Nazruddin Safaat (2020) .

beberapa fitur-fitur android yang paling penting adalah :

1. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable.*

2. Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.

3. Integrated browser berdasarkan *engine open source* WebKit.

4. grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).

5. SQLite untuk penyimpanan data.

6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).

7. Bluetooth, EDGE, 3G dan WiFi (tergantung hardware).

8. Kamera, GPS, kompas, dan Accelerometer (tergantung hardware).

9. Lingkungan development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

d. Arsitektur Android

Nazruddin Safaat (2020), secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. *Applications* dan *Widgets,* adalah layer di mana kita berhubungan dengan aplikasi saja, kita download aplikasi kemudian kita lakukan instlasi dan jalankan aplikasi tersebut.

2. *Applications Frameworks,* adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android.

3. Libraries, adalah layer dimana fitur-fitur Android berada. biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

a) *Libraries* media untuk pemutaran media audio dan video.

b) *Libraries* untuk manajemen tampilan.

c) *Libraries Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D.*

d*) Libraries SQLite* untuk dukungan database.

e) *Libraries SSL* dan WebK.it tenntegrasi dengan web *browser security.*

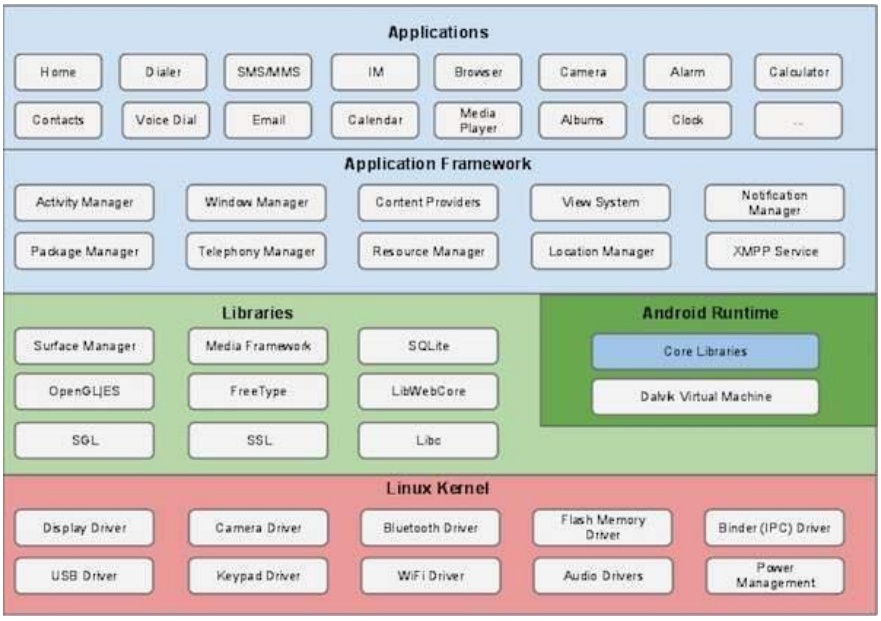
f) *Libraries LiveWebcore mencakup* modem web browser dengan *engine* embeded web view. g) Libraries 3D yang mencakup implementasi OpenGL ES 1.0 APi's

4. *Android RunTime.* Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux. Android Run Time dibagi menjadi dua bagian, yaitu :

a) *Core Libraries* : Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan merupakan Virtual Machine Java, sehingga diperlukan Libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh Core Libraries.

b) *Dalvik Virtual Machine* : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan threading dan manajemen tingkat rendah.

5. *Linux Kernel*. Layer dimana inti dari operating sistem dari Android itu berada. Berisi file-file system yang mengatur sistem *processing, memory. resource, drivers,* dan sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6.



Gambar 2.5 Arsitektur Android

(Sumber: http://www.kapalomen.com/2016/07/android-arsitektur-android-sistem-operasi-android.html)

e. Kelebihan dan Kelemahan Android

Menurut Saiful Jais (2012) beberapa kelebihan-kelebihan android dibandingkan dengan sistem operasi lainnya yaitu sebagai berikut :

1. Android bersifat open source jadi bisa dikembangkan oleh siapa saja

2. Multitasking, dapat menjalankan berbagai aplikasi, itu artinya pengguna bisa browsing, membaca e-book dan mendengarkan lagu.

3. Kemudahan dalam notifikasi, setiap ada SMS, E-mail, atau bahkan artikel terbaru dari RSS Reader akan selalu ada di notifikasi di Home Screen.

4. Google Android Market menyediakan ribuan aplikasi dan games yang dapat didownload dan diinstal sesuai dengan keinginan pengguna

5. Pilihan ponsel yang beranekaragam

6. Bisa menginstal ROM yang dimodifikasikan

7. Widget pada homescreen, pengguna dapat dengan mudah mengakses berbagai fitur dengan cepat.

8. Android telah terintegrasi dengan layanan Google.

Selain beberapa kelebihan android yang telah disebutkan sebelumnya, namun ada beberapa hal yang menjadi kelemahan android, antara lain :

1. Koneksi internet yang terhubung terus menerus.

2. Iklan akan selalu muncul pada bagian atas atau bawah aplikasi.

3. Boros baterai, karena setiap aplikasi yang dijalankan, sinyal, RUM yang digunakan akan menggunakan daya baterai. Apalagi sinyal di Indonesia yang belum stabil maka baterai akan mudah habis karena untuk mempertahankan sinyal agar tetap stabil.

5. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA . Selain sebagai editor kode dan fitur *developer IntelliJ* yang l, Android Studio menawarkan banyak 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 resourceke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi

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

e) Framework dan alat pengujian yang lengkap

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

g) Dukungan C++ dan NDK

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

6. Framework

Salah satu alasan mengapa orang menggunakan *framework* terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah *framework* biasanya sudah tersedia struktur aplikasi yang baik, standard coding, best practice, design pattern, dan common function. Dengan menggunakan framework kita dapat langsung fokus kepada business process yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain.

Menurut Hakim (2010:3) menjelaskan bahwa, *Framework* adalah koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan

sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal.

Sedangkan menurut Raharjo (2015:2), *Framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi web. Jadi, *Framework* adalah kumpalan-kumpalan potongan program yang dipadukan menjadi satu kerja kerja yang digunakan untuk membatu dalam pembuatan sebuah aplikasi.(Ciraces.2023).

7. 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 1 *Prototype* Paradigma

Tahap-tahap pada model prototype diatas adalah sebagai berikut:

1) Mendengarkan Pelanggan Tahap ini dilakukan pengumpulan data untuk memenuhi 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).

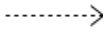
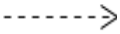
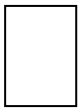
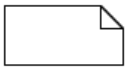
c. UML (*Unified Modelling Language*)

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).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

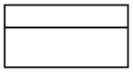
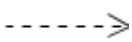
Lanjutan Tabel 2. 1

e. Class Diagram

class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan 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
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

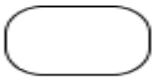
Lanjutan Tabel 2. 2

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		*Fork 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 menunjukkan 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

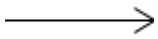
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang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

h. *StateChartDiagram*

*Statechar*t diagram merupakan model perilaku yang dinamis dari class secara individual maupun beberapa bentuk dari obyek. Semua itu menunjukkan 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. Statechart 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

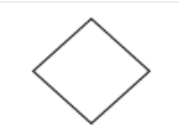
No.	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state 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*

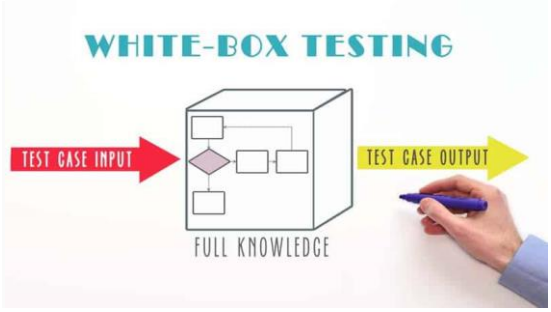
Simbol	Nama dan Funsi
	***Terminator***(Tanda mulai atau selesai), menyatakan titik awal dan titik akhir diagram alir.
	**Proses** (Tanda komputasi), menyatakan sebarang proses.

	**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 didiagramalirkan.
	**Konektor** (Tanda penghubung), digunakan untuk menghubungkan ke berbagai bagian dalam alir diagram.

8. . 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 memenuhi 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. 2 *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. Tester 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 tester 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 conditional 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

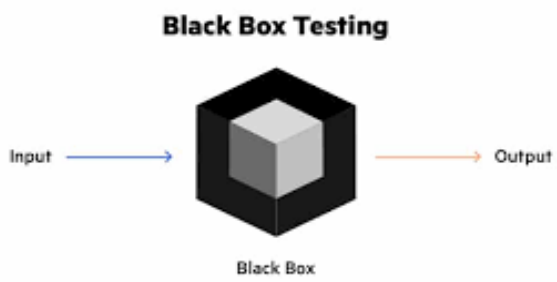
c) 1-2-3-10 yaitu properti yang dimiliki oleh pemain

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://itbox.id/blog/white-box-testing-adalah/)

b. *Black Box*



Gambar 2. 3 *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) Cause effect graph merupakan teknik yang digunakan di mana graphic sebagai pacuannya. Dalam teknik ini di mana graphic 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://itbox.id/blog/black-box-testing-adalah/).

B. Penelitian yang Relevan

Beberapa hasil penelitian yang relevan dengan penelitian ini antara lain :

1. Penelitian tentang Pengenalan Alat Musik Tradisional Berasis Android (Egi Badar Sambana, Yuda & Yopa., 2020) - penelitian ini membahas penggunaan aplikasi mobile dalam pengenalan alat musik, termasuk pendekatan dan metode pembelajaran yang dapat diterapkan pada pengembangan aplikasi pembelajaran scane kecapi.

2. Aplikasi Media Pembelajaran Alat Musik Gitar Berbasis Android Menggunakan Metode SDLC (Rio Saputra, Dani Ahmadani, Ryan Prastiyo, Rizki Hermawan & Andry Maulana., 2022) - penelitian ini membahas tentang pengembangan aplikasi pembelajaran musik tradisional berbasis android.

3. Rancang Bangun Media Pembelajaran Mengenal Alat Musik Tradisional Khas Jawa Barat Menggunakan Teknologi Augmented Reality (Raden Erwin, G.H & Daris, D.A., 2020) - penelitian ini membahas tentang pengenalan alat musik dapat memberikan gambaran dalam pengembangan aplikasi pembelajaran scane kecapi.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penulis menggunakan teknik pengumpulan data kualitatif seperti observasi, wawancara, dan kajian pustaka dalam penelitian ini. Penulis melakukan observasi guna mengumpulkan data yang cukup terkait dengan judul penelitian ini, sehingga tidak menyimpang dari pokok permasalahan

Pengumpulan data dilakukan untuk mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian ini.

1. Observasi, pengamatan dan juga pencatatan sistematik atas unsur-unsur yang muncul dalam suatu gejala atau gejala-gejala yang muncul dalam suatu objek penelitian.
2. Wawancara adalah percakapan dengan maksud tertentu. Percakapan itu dilakukan oleh dua pihak, yaitu pewawancara (interview) yang mengajukan pertanyaan dan terwawancara (interview) yang memberikan jawaban atas pertanyaan itu. Ciri utama wawancara adalah kontak langsung dengan tatap muka antara pencari informasi dan sumber informasi. Dalam wawancara sudah disiapkan berbagai macam pertanyaanpertanyaan tetapi muncul berbagai pertanyaan lain saat meneliti.

B. Waktu dan Tempat Penelitian

Adapun waktu dan tempat penelitian yang dilaksanakan pada bulan Apil tahun 2024, di Sanggar Seni Melati, dimana proses pengambilan data yang akan dimasukkan dalam aplikasi nantinya Waktu yang dipergunakan untuk penelitian ini 2 (dua) bulan dimulai juni sampai dengan juli.

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No.	Uraian Kegiatan	Bulan 2024	
		Juni	Juli
1	Studi Literatur		
2	Pembuatan Aplikasi		
3	Pengujian Aplikasi		
4	Hasil 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, javascript, react native* dan *postgreSql*. 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 proposal yang diterapkan berdasarkan 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.

6. Implementasi

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

E. Jenis Data

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

a. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

b. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, website dan jurnal.

F. Metode Pengujian

Metode pengujian Black Box atau biasa disebut metode pengujian perangkat lunak secara terinci, jalur logika (logica path) perangkat lunak akan di test case yang mengerjakan kumpulan kondisi atau perulangan secara spesifik.

Dengan menggunakan metode Black Box penulis dapat menentukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau salah
2. Kesalahan interfase
3. Kesalahan dalam struktur data atau database eksternal
4. Kesalahan kinerja
5. Instalasi dan kesalahan terminasi.

6. Langkah-langkah pengujian dalam Black Box :
7. Buat test case dari aplikasi.
8. Uji coba data setiap From dari aplikasi.
9. Hasil uji coba sudah sesuai dengan harapan, maka dapat disimpulkan bahwa aplikasi sudah berfungsi dengan baik (bebas dari kesalahan fungsional).

G. Analisis Sistem

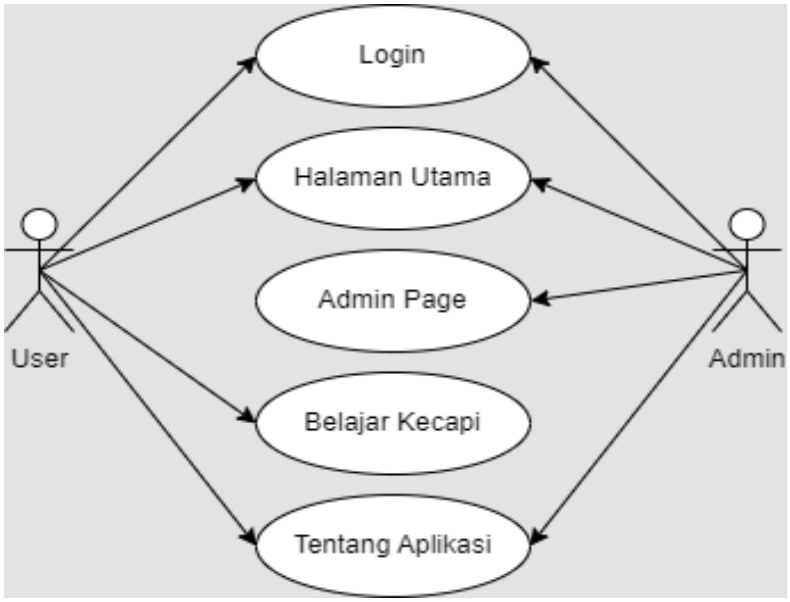
1. Adapun Sistem yang berjalan pada penelitian ini adalah:



Gambar 3.1. Sistem Berjalan

Keterangan dari gambar 3.1. User datang ketempat les atau ke salah satu tempat kesenian kemudian belajar kecapi secara lansung.

2. Adapun Sistem yang diusulkan pada penelitian ini adalah:



Gambar 3.2. Sistem yang diusulkan

Keterangan dari gambar 3.2. Mempunyai dua aktor user dan admin, dimana user dapat mengakses login, halaman utama, belajar ecapi, serta tentang akun, sedangkan admin tentunya dapat mengakses menu login, admin page, dan tentang aplikasi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Selama melakukan penelitian di Sanggar seni melati dalam membangun aplikasi pembelajaran scale kecapi, didapatkanlah hasil penelitian antara lain chord kecapi, yang akan membantu kelancaran dalam menyelesaikan aplikasi berbasis android ini.

Tabel 4. 1 Chord Kecapi

No	Chord	Keterangan	Gambar
1	Do*	Letakkan Jari di Fred 6 Senar 1	
2	Si	Letakkan Jari di Fred 5 Senar 1	
3	Do	Letakkan Jari di Fred 3 Senar 1dan 2	
No	**Chord**	**Keterangan**	**Gambar**

4	Fa	Letakkan Jari di Fred 2 Senar 1	
5	Re	Senar terbuka	
6	La	Letakkan Jari di Fred 4 Senar 1	
7	Sol	Letakkan Jari di Fred 3 Senar 1	
8	Mi	Letakkan Jari di Fred 1 Senar 1	

Dari data yang didapatkan hasil dari wawancara hanya terdapat 1(satu) poin penting diatas chord kecapi Data diatas digunakan untuk diinput di aplikasi pembelajaran scale kecapi berbasis android.

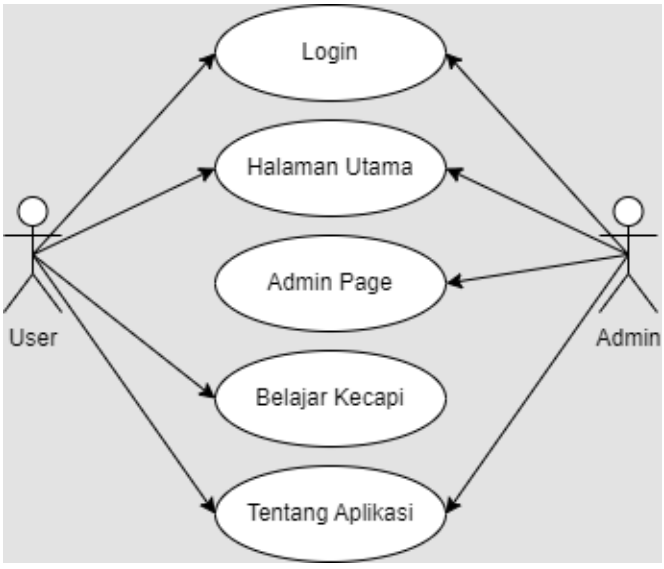
B. Pembahasan

1. Analisis Aliran Data UML

1.1. *Use Case Diagram*

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

Use case juga dapat digunakan untuk merepresentasikan interaksi yang terjadi antara aktor dengan proses sistem yang dibuat. Pada perancangan *use case* juga terdapat skenario, yaitu langkah yang menerangkan urutan kejadian antara pengguna dengan sistem. Diagram *use case* pada pembelajaran scale kecapi berbasis android pada penelitian ini digambarkan seperti berikut ini:



Gambar 4.1. *Use Case Diagram.*

1.2. Tabel *Use Case*

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

Tabel 4.2. 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* karena Admin dapat mengelola Jenis plafon dan halaman keranjang belanja..

Sedangkan untuk penjelasan definisi setiap use case sebagai berikut:

Tabel 4.3. Definisi Use Case Sistem

No	*Use Case*	Keterangan
1	Login	Melakukan Login akun.
2	Home	Menampilkan halaman utama aplikasi
3	Admin Page	Menampilkan halaman Admin
4	Belajar Kecapi	Menampilkan halaman Belajar kecapi
5	Tentang Aplikasi	Menampilkan halaman tentang aplikasi

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 4.4. .Skenario Login.

Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
1. Memasukkan User name dan password.	
2. Menekan tombol Login.	3. Mengecek validitas isian data.
	4. Jika data yang dimasukkan valid, maka 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 4.5. 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 Admin Page

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk menampilkan admin page.

Pre-condition : - .User berada di halaman admin page

Post-condition : - Data dalam sistem akan diperbarui.

Tabel 4.6. Skenario Use Case Admin Page

Aksi Aktor	Reaksi Sistem
1. Memilih menu Admin Page	
	2. Menampilkan form Admin Page

4) Skenario Use Case Belajar Kecapi

Aktor : User

Deskripsi : Sebuah kegiatan untuk melihat Belajar Kecapi

Pre-condition :- Admin berada di halaman Belajar Kecapi

Post-condition : -

Tabel 4.7. Skenario Use Case Belajar Kecapi

Aksi Aktor	Reaksi Sistem
1. Menekan menu Belajar Kecapi	
	2. Menampilkan data yang dipilih (Belajar Kecapi) yang dimaksud dalam sistem.
	3. Menampilkan data yang dipilih sesuai data yang ada.

5) Use Case Kalimat Tentang Aplikasi

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk melihat Tentang Aplikasi

Pre-condition : - User berada di halaman Tentang Aplikasi

Post-condition : -

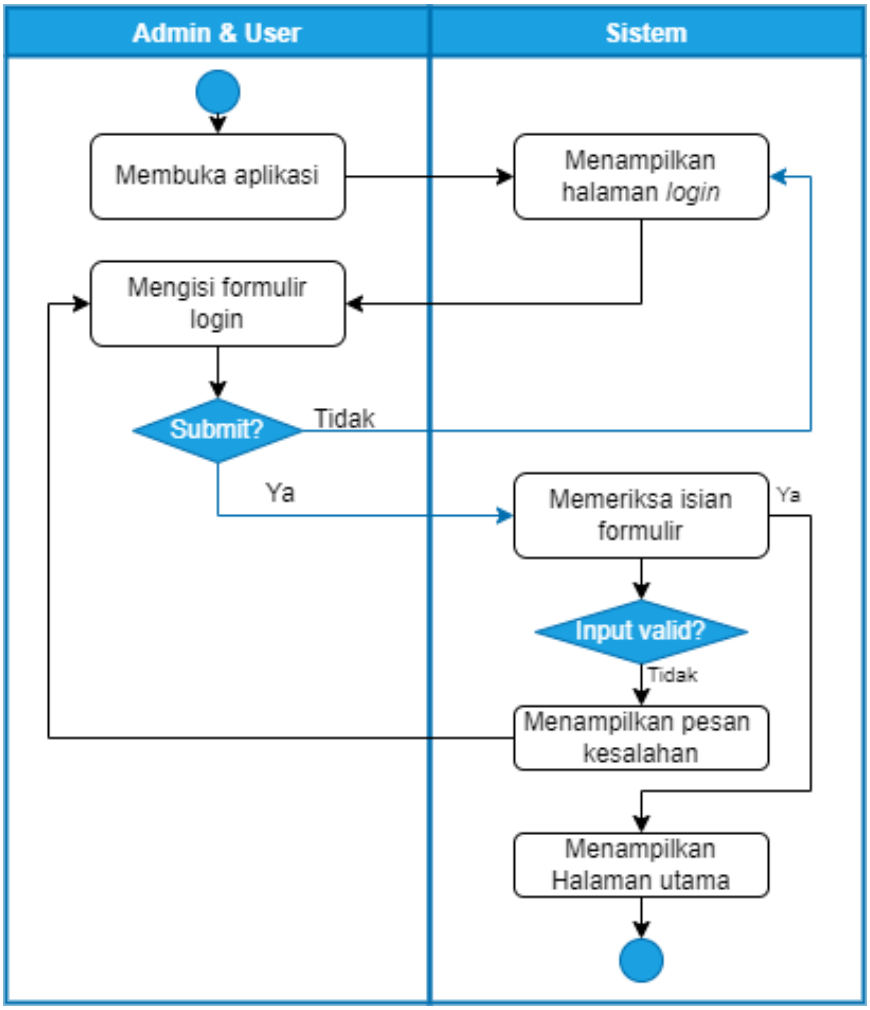
Tabel 4.8. Skenario Use Case Tentang Aplikasi

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

1.3. *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.

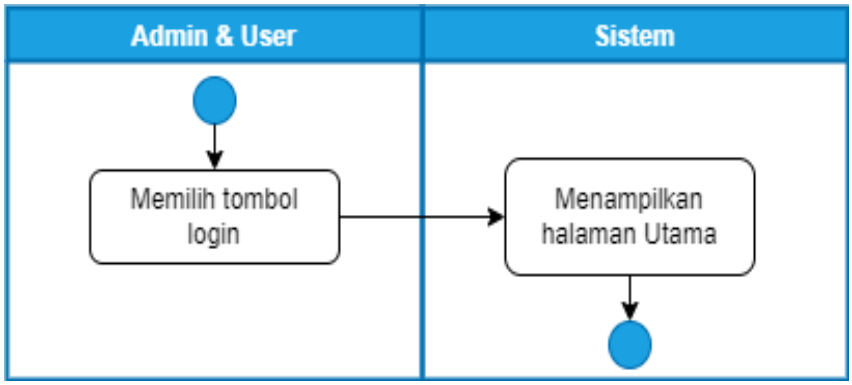
a) *Activity Diagram Login*



Gambar 4.2. *Activity Diagram Login*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas login. Proses dimulai dengan admin atau user memilih tombol login, kemudian sistem menampilkan form isian yang kemudian admin atau user akan mengisi form isian. Ketika sudah disimpan, sistem akan memperbarui data di database yang akan mengecek data yang dikirim, jika terdapat kesalahan, akan menampilkan pesan kesalahan, jika tidak, sistem menampilkan halaman utama.

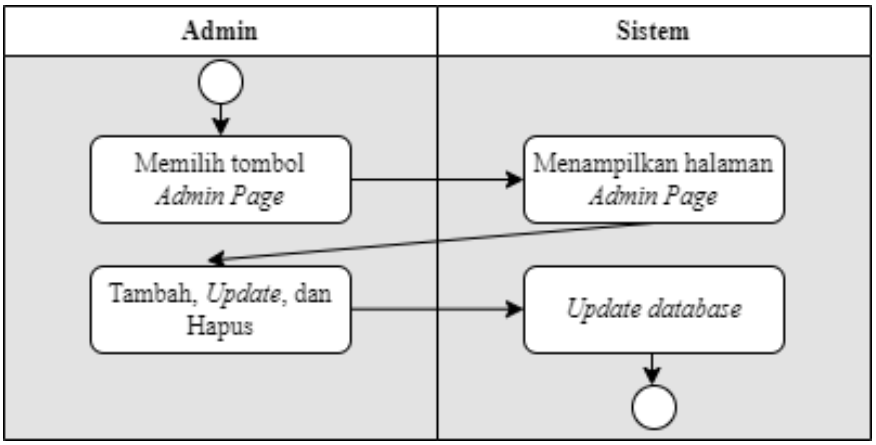
b) *Activity Diagram Home*



Gambar 4.3. *Activity Diagram Home*

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

c) *Activity* Diagram Admin *Page*

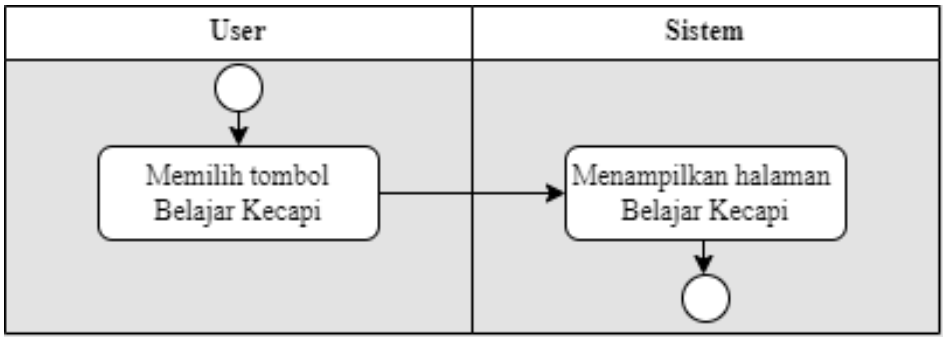


Gambar 4.4. *Activity* Admin *Page*

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas *Admin Page* Proses dimulai dengan admin memilih tombol *admin page*, kemudian sistem menampilkan halaman admin page Jika ingin tambah data, *Update* data serta hapus data, maka dapat memilih salah satu menu maka

sistem akan menambahkan, *update* serta hapus data tersebut dan akan terupdate di *database*.

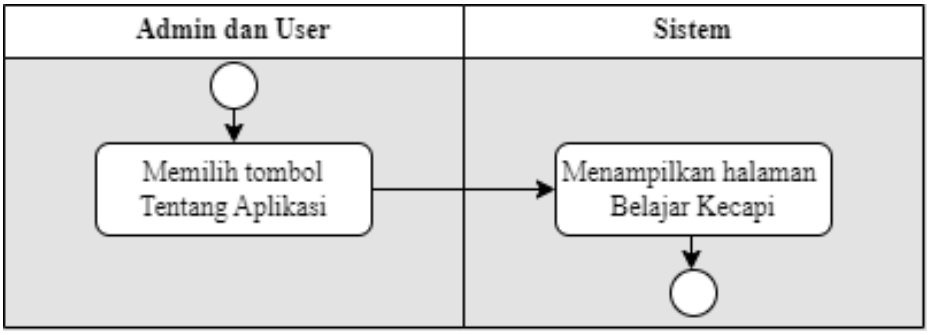
d) *Activity* Diagram Belajar Kecapi



Gambar 4.5. *Activity* Diagram Belajar Kecapi

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Belajar KecapiProses dimulai dengan *use*r memilih tombol Belajar Kecapi, kemudian sistem menampilkan halaman Belajar Kecapi.

e) *Activity* Diagram Tentang Aplikasi



Gambar 4.6. *Activity* Diagram Tentang Aplikasi

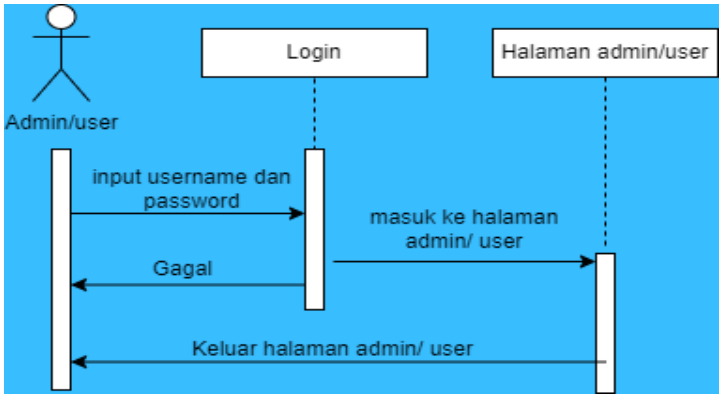
Diagram aktivitas di atas menerangkan alur proses untuk aktivitas Tentang Aplikasi. Proses dimulai dengan *admin* dan *user* memilih Tentang Aplikasi, kemudian sistem menampilkan halaman Tentang Aplikasi.

1.4. *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 pembaha-san proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada sistem informasi yang dikembangkan:

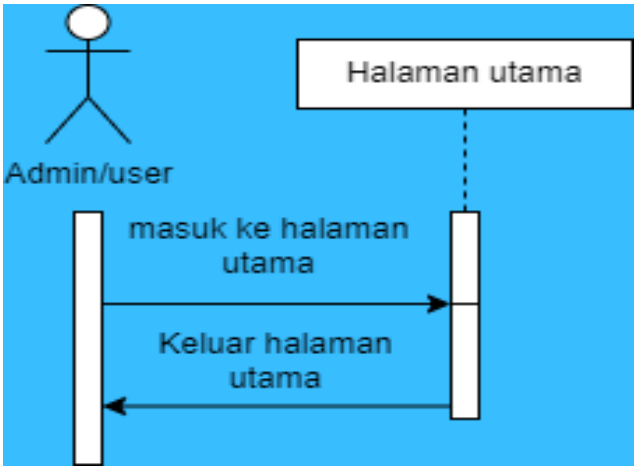
1) *Sequence Diagram*: *Login*



Gambar 4.11 *Sequence Diagram login*

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

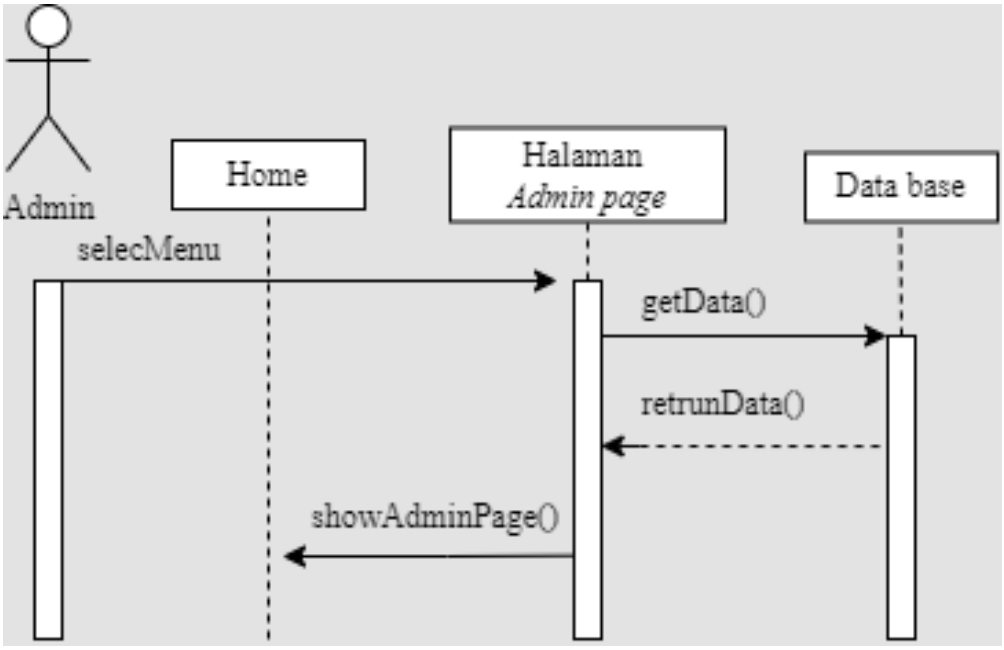
2) *Sequence Diagram*: *Home*



Gambar 4.12. *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.

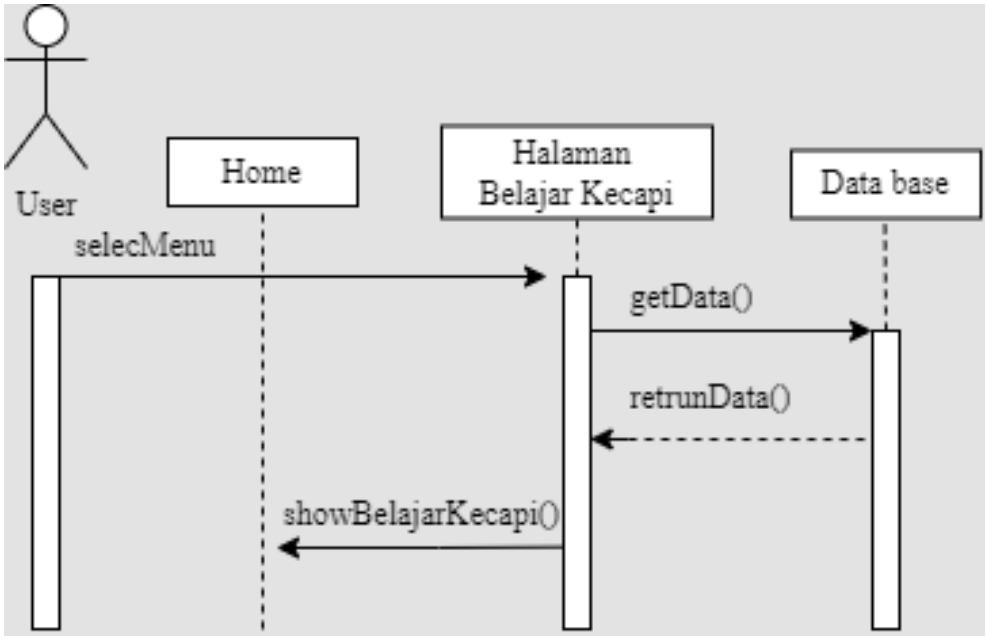
3) *Sequence Diagram*: *Admin Page*



Gambar 4.13. *Sequence Admin Page*

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

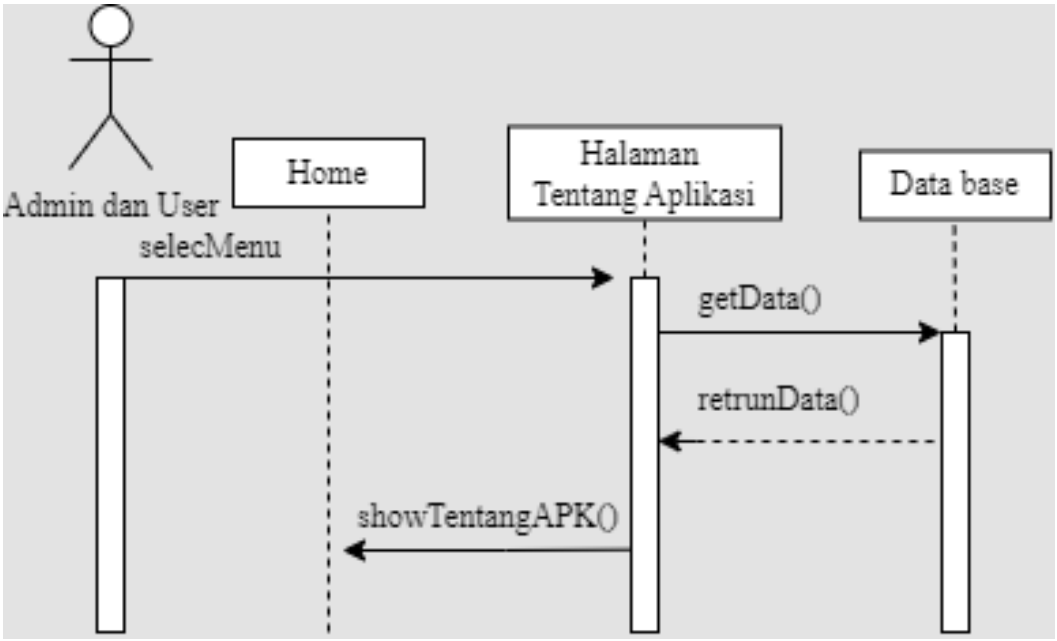
4) *Sequence Diagram*: Belajar Kecapi



Gambar 4.14. *Sequence Diagram* Belajar Kecapi

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

5) *Sequence Diagram*: Tentang Aplikasi



Gambar 4.15. *Sequence Diagram* Tentang Aplikasi

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

1.5. Tabel *Database*

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

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

Tabel 4.10. Tabel User

No	Nama Field	Tipe Data	Ukuran
1	id_user	Varchar	12
2	User_id	Varchar	12
3	Pass_id	Varchar	8
4	nama	Varchar	20
5	alamat	Varchar	50

b. Tabel materi. Nama *table*: Chord, Adapun struktur *table* chord sebagai berikut.

Tabel 4.11. Tabel Chord

No	Nama Field	Tipe Data	Ukuran
1	id_chord	Varchar	10
2	Nama_chord	Varchar	10
4	jumlah	Varchar	20

2. Perancangan Sistem

2.1. Halaman User

1. Halaman *Login* Akun

Berikut adalah tampilan login akun pada aplikasi pembelajaran scale kecapi. Halaman menu login merupakan halaman *login user. Login* merupakan tampilan yang ada pada sistem, yang digunakan user untuk masuk ke halaman selanjutnya.

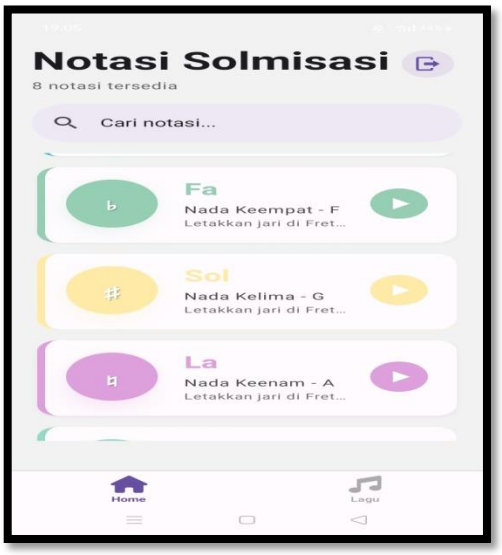


Gambar 4.20. Login akun

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.login
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import
com.abcd.vian_marketplaceweddingorganizer.data.model.UsersModel
importcom.abcd.vian_marketplaceweddingorganizer.databinding.Activi
tyLoginBindingimportcom.abcd.vian_marketplaceweddingorganizer.ui.a
ctivity.admin.AdminMainActivityimportcom.abcd.vian_marketplacewedd
ingorganizer.ui.activity.register.RegisterActivityimportcom.abcd.vi
an_marketplaceweddingorganizer.ui.activity.user.main.MainActivityi
mportcom.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.main
.WeddingOrganizerMainActivityimportcom.abcd.vian_marketplaceweddin
gorganizer.utils.LoadingAlertDialog.
```

2. Halaman Home

Berikut adalah tampilan halaman home pada aplikasi pembelajaran scale kecapi. Halaman ini merupakan halaman yang dapat melihat semua notasi simulasi dari menu Do, Re, Mi, Fa, Sol, La, Si dan Do yang dapat di akses oleh user.

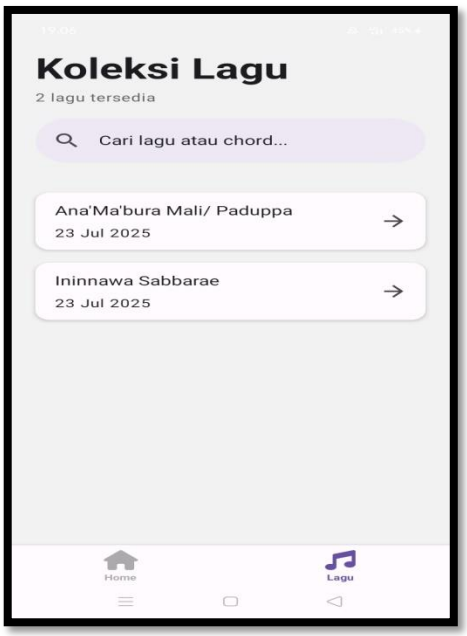


Gambar 4.21. Halaman Home

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import androidx.navigation.compose.*
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            DashboardApp()
        }
    }
}
@Composable
fun DashboardApp() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination =
"dashboard") {
        composable("dashboard") { DashboardScreen(navController) }
        composable("profile") { ProfileScreen(navController) }
    }
```

3 Halaman Koleksi Lagu

Berikut adalah tampilan halaman Koleksi Lagu pada aplikasi pembelajaran scale kecapi. Halaman ini merupakan halaman koleksi lagu yang dapat menampilkan semua lagu pada aplikasi.



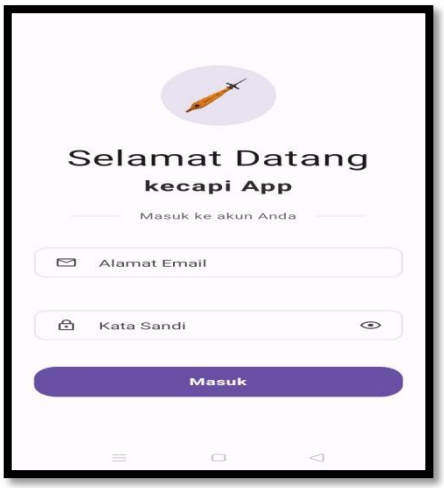
Gambar 4.22. Halaman Koleksi Lagu

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            WeddingOrganizerApp()
        }
    }
}
```

2.2. Halaman Admin

1. Halaman *Login* Akun

Berikut adalah tampilan login akun pada aplikasi pembelajaran scale kecapi. Halaman menu login merupakan halaman *login user. Login* merupakan tampilan yang ada pada sistem, yang digunakan user untuk masuk ke halaman selanjutnya.

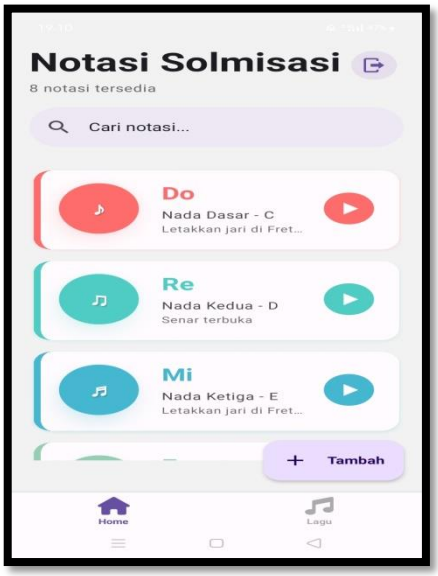


Gambar 4.23. Login akun

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.login
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import
com.abcd.vian_marketplaceweddingorganizer.data.model.UsersModel
importcom.abcd.vian_marketplaceweddingorganizer.databinding.Activi
tyLoginBindingimportcom.abcd.vian_marketplaceweddingorganizer.ui.a
ctivity.admin.AdminMainActivityimportcom.abcd.vian_marketplacewedd
ingorganizer.ui.activity.register.RegisterActivityimportcom.abcd.vi
an_marketplaceweddingorganizer.ui.activity.user.main.MainActivityi
mportcom.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.main
.WeddingOrganizerMainActivityimportcom.abcd.vian_marketplaceweddin
gorganizer.utils.LoadingAlertDialog.
```

2. Halaman Home

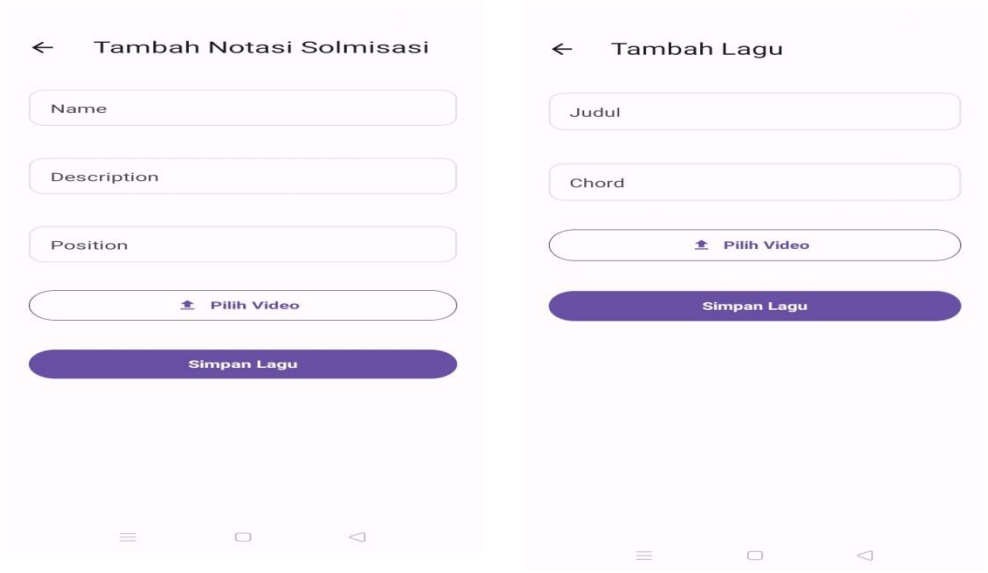
Berikut adalah tampilan halaman Home pada aplikasi pembelajaran scale kecapi. Halaman ini merupakan halaman yang dapat melihat semua menu admin, dan admin dapat menambah, update, dan hapus Lagu serta Notasi solmisasi pada menu home.



Gambar 4.24. Halaman Home

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import androidx.navigation.compose.*
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            DashboardApp()
        }
    }
}
```

3. Halaman Tambah Notasi & Lagu



Gambar 4.25. Halaman Home

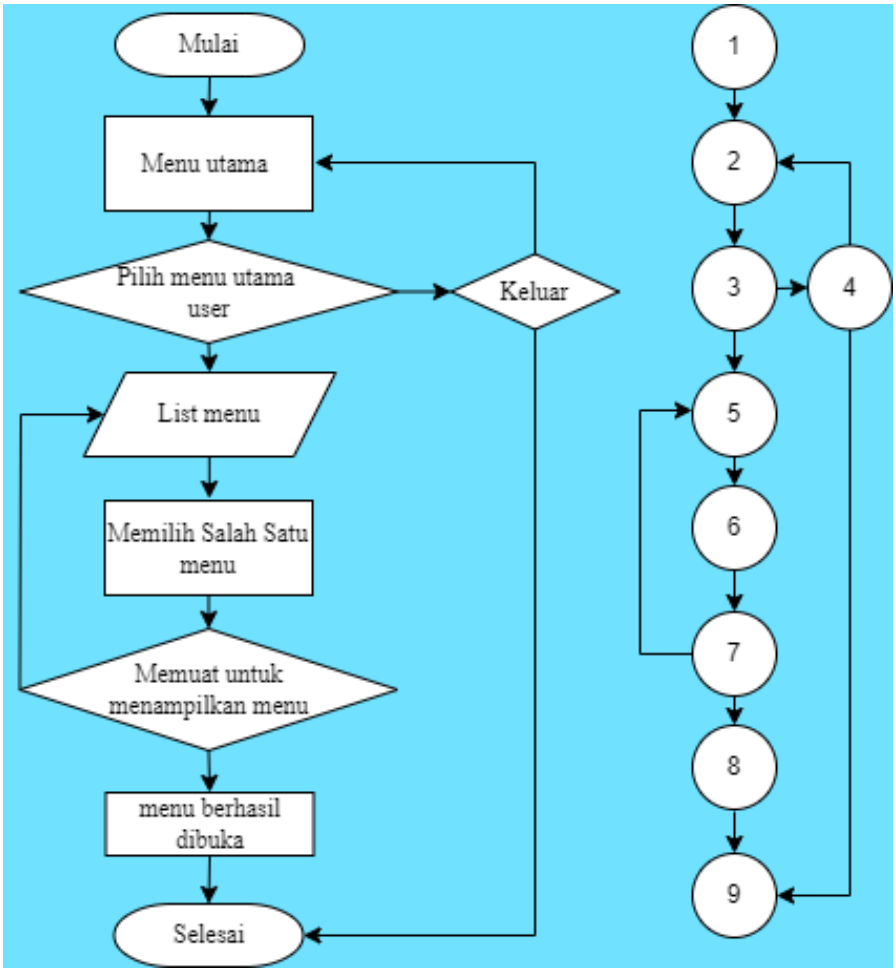
```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            WeddingOrganizerApp()
        }
    }
}
```

3. Pengujian Sistem

3.1. White Box.User

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.

a) *White box* menu aplikasi pembelajaran scale kecapi.



Gambar 4.26 *White box* menu aplikasi pembelajaran scale kecapi.

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 = 9E = 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

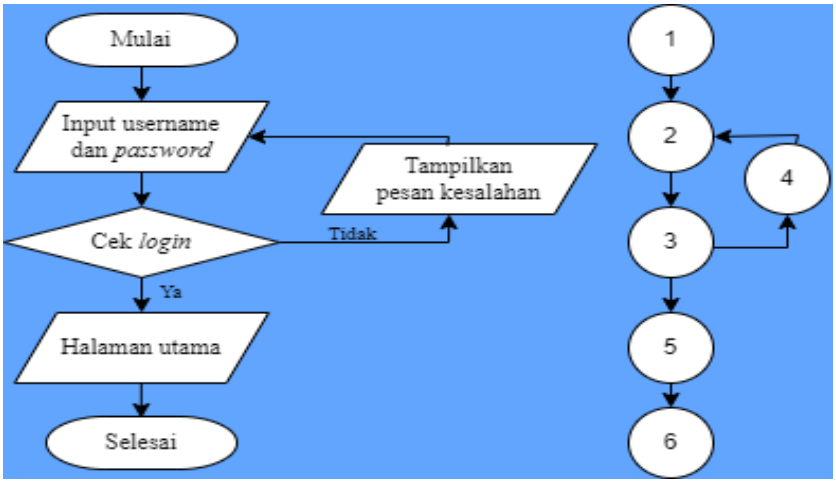
Path 3 = 1 – 2 – 3 – 4 – 9

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

Tabel 4.15. *White Box* Menu aplikasi pembelajaran scale kecapi.

	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

b) *White box* pada menu login



Gambar 4.27.*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 :

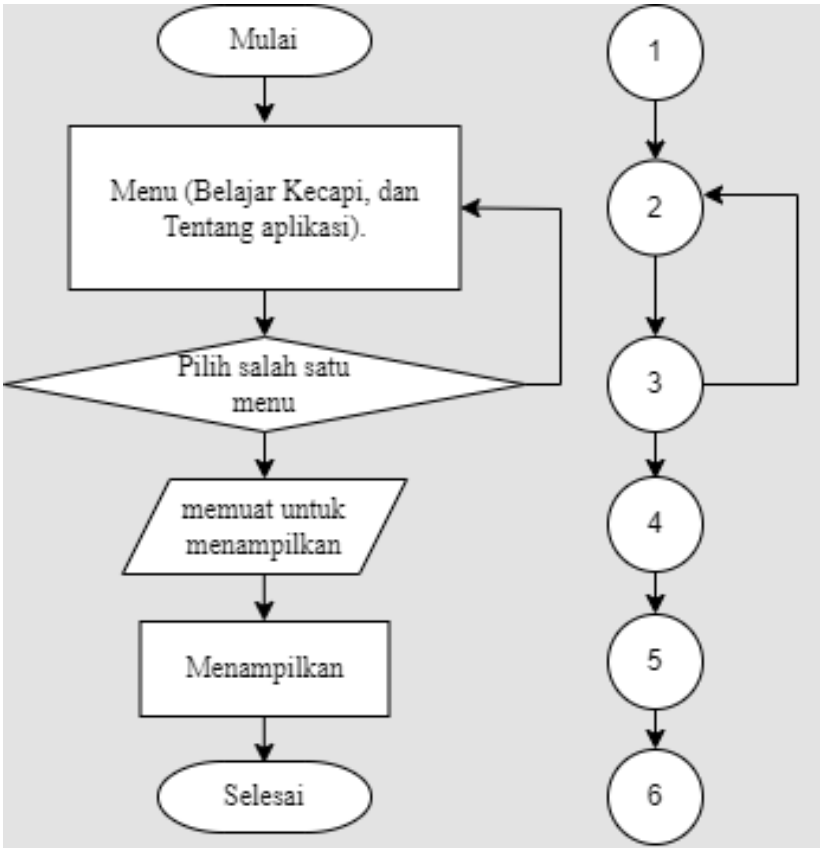
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

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

c) *White box* pada menu aplikasi pembelajaran scale kecapi.



Gambar 4.28.*White box* pada menu aplikasi pembelajaran scale kecapi.

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

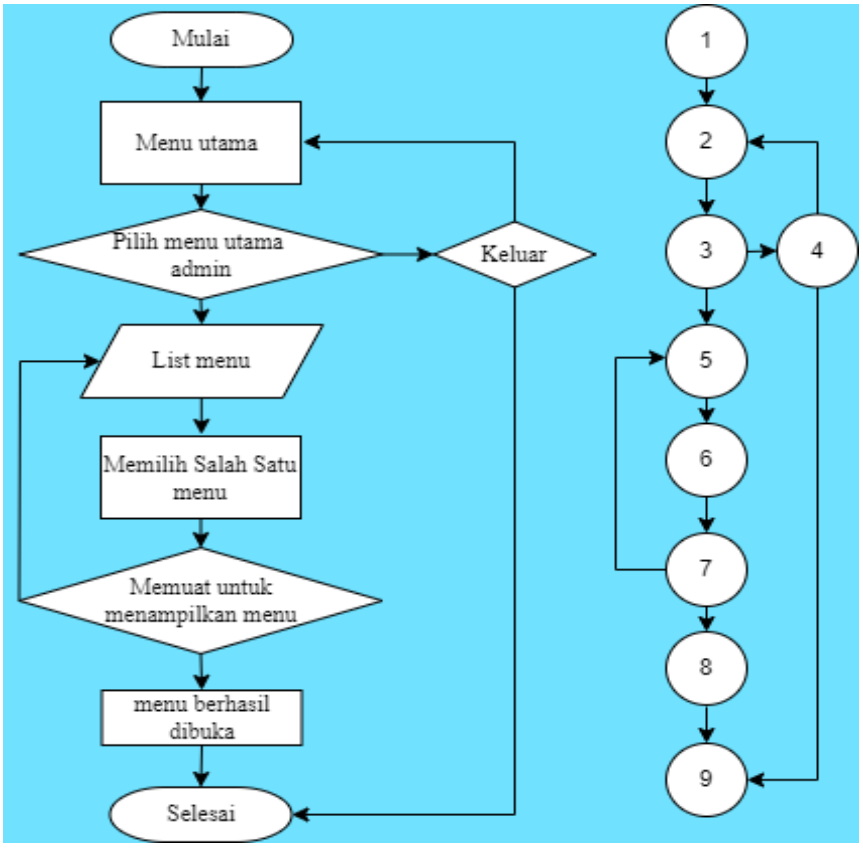
Tabel 4.17. *White Box*Perhituangan menu aplikasi pembelajaran scale kecapi.

	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

3.2. White Box. Admin

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.

a) *White box* menu admin aplikasi pembelajaran scale kecapi.



Gambar 4.29 *White box* menu admin aplikasi pembelajaran scale kecapi.

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 = 9E = 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

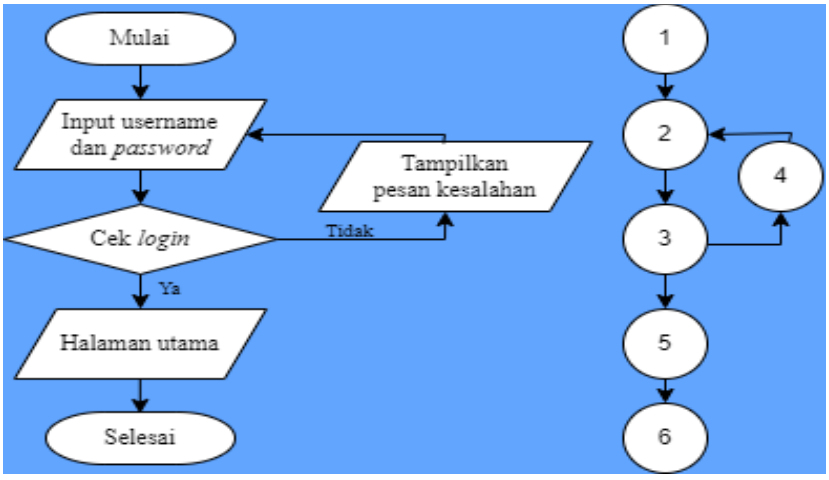
Path 3 = 1 – 2 – 3 – 4 – 9

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

Tabel 4.18. *White Box* menu admin aplikasi pembelajaran scale kecapi.

	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

b) *White box* pada menu login



Gambar 4.30.*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 :

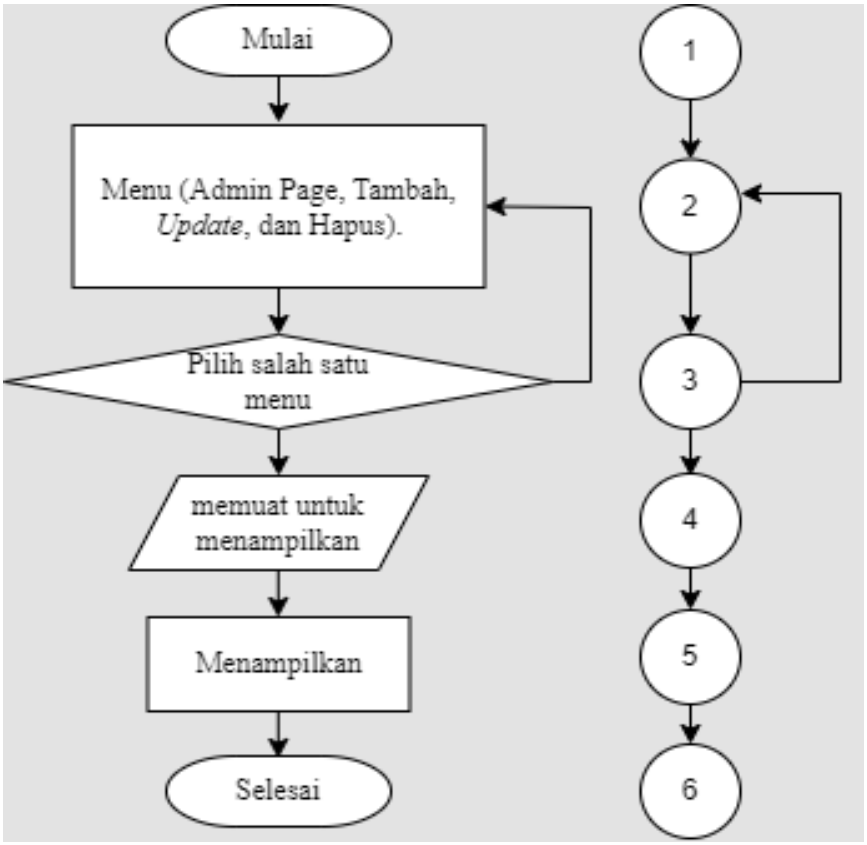
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 –5 – 6

Tabel 4.19. *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

c) *White box* pada menu admin aplikasi pembelajaran scale kecapi.



Gambar 4.31. *White box* pada menu admin aplikasi pembelajaran scale kecapi.

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

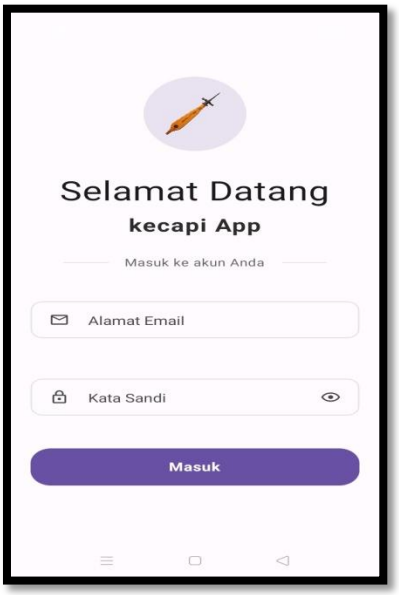
Tabel 4.20. *White Box*Perhituangan menu admin aplikasi pembelajaran scale kecapi.

	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

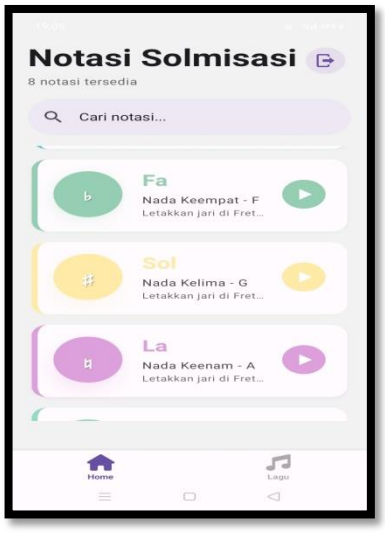
3.3. *Black Box* User

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.

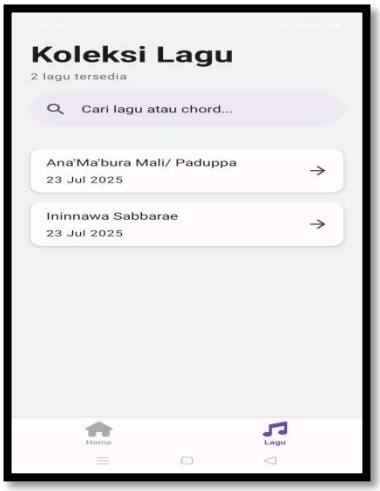
Tabel 4.21. Tabel Pengujian Login

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 Home

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

Tabel 4.23. Tabel Pengujian Halaman Koleksi Lagu

Test Faktor	Hasil	Kesimpulan
Halaman Koleksi Lagu	✔	Sukses, karena ketika *users* sudah *login* sebagai user maka lansung masuk ke halaman Koleksi Lagu.
Screen Shot		
		

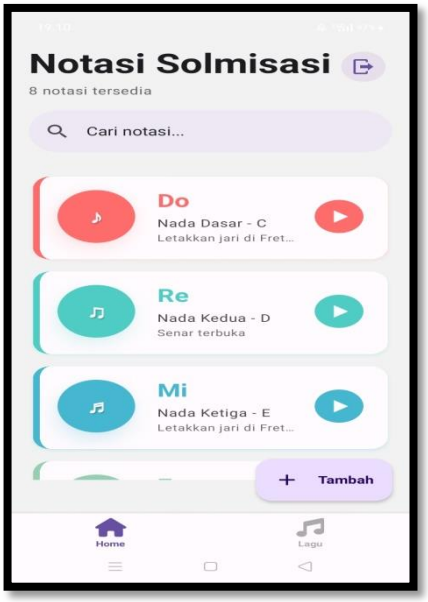
3.4. *Black Box* Admin

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 4.24. 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 4.25. Tabel Pengujian Halaman Home

Test Faktor	Hasil	Kesimpulan
Halaman Home	✔	Sukses, karena ketika *users* sudah *login* sebagai admin maka lansung masuk ke halaman Home di halaman ini admin ketika menekan menu tambah, *update*, dan hapus lagu dan notasi simulasi data tersebut berhasil**.**
Screen Shot		
		

4. Maintenance

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

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dan pada pembahasan bab sebelumnya, maka dapat diperoleh kesimpulan yaitu:

1. Dibangunnya suatu sistem berbasis Android, memberikan kemudahan kepada penggunanya.

2. Sistem ini dapat memberikan kemudahan bagi pengguna dalam proses belajar kecapi.

3. Melalui sistem yang dibangun pengguna dapat dengan mudah mengakses aplikasi ini dimanapun dan kapanpun.

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 fitur *touch* senar kecapi pada aplikasi.

DAFTAR PUSTAKA

Adi Nugroho.2020., *Rekayasa Peragkat Lunak Menggunakan UML dan Java, Pustaka Belajar.* Yogyakarta.

Ciraces. 2023. *Pengertian Mysql & Framework menurut para ahli*.,Admin media.Jakarta Timur.

Egi Badar Sambana, Yuda & Yopa., 2020 - *Aplikasi mobile dalam pengenalan alat music*.Jakarta.

Gilang Ramadhan.2023., *Contoh USE CASE Diagram Legkap Penjelasan, Dicoding Intren*. (Online), https://www.dicoding.com/blog/contoh-use-case-diagram/.

Nita, Budi Hermawan.,2021. *Analisis Studi Perbandingan Sejarah Alat Musik Kecapi di Indonesia dan Guzheng di Tiongkok.* Fakultas Bahasa dan Sastra, Pendidikan Bahasa Mandarin, Universitas Widya Kartika

Raden Erwin, G.H & Daris, D.A., 2020.,*Rancang Bangun Media Pembelajaran Mengenal Alat Musik Tradisional Khas Jawa Barat Menggunakan Teknologi Augmented Reality*.

Rahardja, U., Aini, Q., & Santoso, N. P. L. 2020. Pengintegrasian YII *Framework* Berbasis API pada Sistem Penilaian Absensi. *Sisfotenika*, 8(2), 140. https://doi.org/10.30700/jst.v8i2.403

Rio Saputra, Dani,H, Ryan ,P, Rizki ,H , Andry ,M., 2022., *Media Pembelajaran Alat Musik Gitar Berbasis Android Menggunakan Metode SDLC.*

Rosyadi, Muhammad Hafidz Amri. Aplikasi Media Pembelajaran Alat Musik Daerah di Indonesia dengan Menggunakan Teknologi Augmented Reality. Diss. Universitas Bhayangkara, 2020.

Ruswandi, Tardi. Asep Nugraha & Dody Satya Ekagustdiman,2020. "*Ragam Alat Musik Petik Tradisional Sunda Java Platform Android,XML*." Hasil Penelitian Fundamental. Bandung: Sekolah Tinggi Seni Indonesia (STSI).

Saputra, R., Prastiyo, R., & Hermawan, R. 2020. *APLIKASI MEDIA PEMBELAJARAN ALAT MUSIK GITAR BERBASIS ANDROID*.

Wahyu Setia Bintara.2023.*Pengertian Android Definisi,Fungsi,Sejarah,Kelebihan* . (Online), https://dianisa.com/pengertian-android/.

Nazruddin Safaat.2020 "*Arsitektur Android ,Fitur android Programming*"& sqlite, Penerbitbuku Andi, Semarang.