BAB 1

PENDAHULUAN

A. Latar Belakang Masalah

Bacaan doa dan dzikir harian adalah bagian penting dari kehidupan umat Muslim. Namun, sering kali materi ini kurang mendapat perhatian dalam pendidikan formal maupun informal. Banyak umat Muslim mengalami kesulitan dalam menghafal dan memahami bacaan doa dan dzikir karena keterbatasan waktu atau akses ke sumber belajar yang memadai.

Di era digital ini, diperlukan solusi yang praktis dan efisien untuk mengatasi tantangan tersebut. Aplikasi *mobile learning* berbasis Android yang dirancang khusus untuk bacaan doa dan dzikir harian bisa menjadi solusi yang tepat. Aplikasi ini tidak hanya memberikan kemudahan akses bagi pengguna, tetapi juga memungkinkan mereka belajar kapan saja dan di mana saja.

Dengan fitur-fitur interaktif dan *user-friendly*, aplikasi *mobile learning* dapat membantu pengguna menghafal dan memahami bacaan doa dan dzikir dengan lebih efektif. Selain itu, aplikasi ini dapat menyediakan materi yang terstruktur dan terstandarisasi, sehingga pengguna bisa belajar secara sistematis.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun aplikasi *mobile learning* untuk bacaan doa dan dzikir harian berbasis Android. Diharapkan, aplikasi ini dapat membantu umat Muslim meningkatkan pengetahuan

dan pemahaman mereka terhadap bacaan doa dan dzikir harian, serta mendukung upaya peningkatan kualitas ibadah mereka.

Berdasarkan pemamparan masalah diatas, maka penulis tertarik untuk mengambil permasalahan yang berjudul "**Rancang Bangun Aplikasi *Mobile Learning* Bacaan Doa dan Dzikir Harian Berbasis Android**".

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang dipaparkan, rumusan masalah dalam penelitian ini adalah bagaimana membangun aplikasi *mobile learning* bacaan doa dan dzikir harian berbasis android?

C. Tujuan Penelitian

Berdasarkan yang telah dipaparkan sebelumnya, maka tujuan dari penelitian ini adalah membuat sebuah aplikasi *mobile learning* bacaan doa dan dzikir harian berbasis android yang dapat berkontribusi secara positif dalam meningkatkan akses dan pemahaman bacaan doa dan dzikir harian, serta meningkatkan keimanan dan ketakwaan umat islam pada umumnya.

D. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah:

1. Aplikasi ini dirancang dan dikembangkan dengan berfokus pada penggunaan bacaan doa dan dzikir harian sebagai inti pembelajaran. Dalam upaya memberikan pembelajaran yang mendalam dan bermanfaat kepada pengguna dalam menjalankan aktivitas ibadah sehari-hari.

2. Peneliti membatasi bacaan doa dan dzikir harian yang terdapat dalam aplikasi android hanya pada bacaan yang paling umum dan dikenal luas dalam agama Islam.

E. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis

   Penulis dapat mengimplementasikan ilmu yang didapat selama proses perkuliahan.

2. Bagi Akademik

   Sebagai bahan referensi bagi penulis lain untuk mengembangkan kemampuan dibidang yang sama.

3. Bagi Masyarakat

   Memberikan kemudahan akses terhadap bacaan doa dan dzikir harian yang memungkinkan masyarakat untuk belajar dan mempraktikan ajaran agama islam secara mendalam.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

1. Sambri (2019). "Aplikasi Dzikir Berbasis Android". Universitas Muhammadiyah Parepare. Tujuan dari penelitian ini adalah membuar aplikasi sederhana yang menyediakan konten dzikir seperti buku digital yang berbasis android dan menampilkan dzikir dalam bahasa arab yang dilengkapi dengan artinya.
2. Alda, M. (2023). "Rancang Bangun Aplikasi Doa Harian untuk Perangkat *Mobile* Berbasis Android". Universitas Islam Negeri Sumatera Utara. Tujuan penelitian ini yaitu membangun aplikasi bacaan doa harian untuk perangkat mobile android yang memudahkan pengguna untuk dapat mengakses rangkaian doa-doa harian.
3. Armandanato, A.C. (2023). "Perancangan Aplikasi *Mobile* Pengingat Dan *Monitoring* Ibadah Sholat Wajib Untuk Siswa SMP Berbasis Android". Universitas Nusantara PGRI Kediri. Tujuan penelitian ini dilakukan untuk merancang aplikasi pengingat dan *Monitoring* para siswa untuk melaksanakan ibadah shalat wajib tepat waktu.

B. Kajian Teori

1. Mobile Learning

(Hernawan, 2017) *Mobile learning (m-learning)* adalah pendekatan pembelajaran yang menggunakan perangkat mobile seperti ponsel, PDA, laptop, dan tablet PC. Dengan *m-learning*, pengguna dapat mengakses materi, petunjuk, dan aplikasi terkait dengan pembelajaran tanpa dibatasi oleh ruang dan waktu, sehingga mereka dapat belajar di mana pun dan kapan pun. M-learning merupakan bagian dari *e-learning* dan secara inheren juga merupakan bagian dari *d-learning*.

Menurut Darmawan (dalam Hernawan, 2017) Pengembangan pembelajaran *mobile* dilatarbelakangi oleh beberapa alasan. Pertama, pembelajaran *mobile* memungkinkan penggunaannya kapan saja dan di mana saja, baik dalam jaringan maupun di luar jaringan. Kedua, pembelajaran *mobile* memiliki cakupan yang luas, karena dapat digunakan dengan jaringan seluler komersial seperti *GSM* dan *CDMA* tanpa memerlukan pembangunan infrastruktur jaringan sendiri, karena jaringan tersebut sudah tersedia di mana-mana. Ketiga, pembelajaran *mobile* terintegrasi dengan baik dengan sistem yang ada, termasuk sistem *e-learning*, sistem informasi akademik, dan sistem lainnya seperti pesan instan.

Menurut ElHussen dan Cronje (dalam Hernawan, 2017) *Mobile learning* adalah model pembelajaran yang menggabungkan teknologi dan pembelajaran di lingkungan yang bergerak. Dengan kemajuan teknologi, *mobile learning*

memiliki potensi untuk menjadi salah satu cara paling efisien dalam memberikan pendidikan tinggi di masa depan.

Mobile Learning atau pembelajaran berbasis seluler menawarkan sej*uml*ah keunggulan yang sangat signifikan dalam konteks pendidikan dan pengembangan pribadi, beberapa diantaranya meliputi:

1. Belajar melalui telepon seluler atau *m-learning* memiliki keuntungan bagi pengguna untuk belajar setiap kesempatan yang mereka miliki seperti di bus, atau dimanapun jika ada waktu kosong setelah bekerja mereka dapat belajar di mana saja, kapan saja sesuai keinginan penggunanya.
2. Dua fitur terpenting dari perangkat *mobile* adalah portabilitas dan konektivitas. Untuk konektivitas, sistem seluler dirancang untuk terhubung dan berinteraksi dengan situs web pembelajaran melalui jaringan nirkabel untuk mengakses materi pelajaran dari mana saja, baik melalui teks atau email. Portabilitas perangkat seluler ini memungkinkan siswa untuk dengan mudah memindahkan dan membawa konten pembelajaran mereka.
3. Dibandingkan dengan perangkat lain seperti laptop, ponsel (ponsel) relatif lebih murah dan dapat digunakan untuk fungsi *browsing* internet seperti yang tersedia pada sebagian besar perangkat. Dapat diakses bahkan di daerah terpencil, perangkat berbiaya rendah ini memiliki kemampuan email atau *SMS*, sehingga memungkinkan untuk mentransfer informasi ke atau dari ponsel antara guru dan siswa tanpa kesulitan.

Dalam era teknologi dan informasi yang berkembang pesat saat ini, perangkat seluler telah menjadi perangkat multifungsi yang membawa banyak

manfaat dan kemudahan kepada penggunanya. Beberapa layanan yang dapat diakses melalui perangkat seluler mecakup beragam aspek kehidupan dan penggunaan, yang sangat memperluas kapabilitas dan kenyamanan pengguna. Berikut adalah beberapa di antaranya:

1. Portabilitas: Perangkat seluler ini bisa dibawa kemana mana ukuran kecil dan ringan;
2. Interaksi sosial: pertukaran data dan kerjasama dengan siswa lain dapat dilakukan melalui perangkat seluler;
3. Sensitivitas konteks: data tersedia di perangkat seluler dikumpulkan dan dijawab oleh pengguna lain di tempat dan waktu berbeda;
4. Konektivitas: Perangkat seluler bisa terhubung ke perangkat lain, seperti alat untuk mengumpulkan data, atau jaringan umum yang dibuat untuk jaringan bersama.
5. Individu: platform aktivitas bisa disesuaikan untuk pengguna secara individu.

2. Doa

(Rahman, 2022) Doa merupakan pintu yang paling besar diantara pintu-pintu ibadah lain dalam menghambakan diri kepada allah. Bahkan doa merupakan saka guru ibadat, sebagaimana di sebutkan dalam hadist :

الدعاء مخ العبادة

Artinya: Doa itu adalah otaknya ibadah (HR. Bukhari)

Dipandang sebagai otaknya ibadah karena doa adalah suatu ibadat yang secara jelas memperlihatkan kehambaan diri seseorang kepada Tuhan-Nya. Oleh karena itu sungguh tidak layak jika kita berdoa selain Allah. Berdoalah kepada Allah saja. Sebagaimana allah berfirman dalam QS. Yunus ayat 106:

وَلَا تَدْعُ مِنْ دُوْنِ اللّٰهِ مَا لَا يَنْفَعُكَ وَلَا يَضُرُّكَ ۚفَاِنْ فَعَلْتَ فَاِنَّكَ اِذًا مِّنَ الظّٰلِمِيْنَ

Artinya: Dan janganlah kamu berdoa kepada selain Allah, yaitu sesuatu yang mendatangkan manfaat kepadamu dan tidak pula mendatangkan mudarat. (QS. Yunus : 106)

Adapun sepuluh adab dalam berdoa:

1. Waktu yang mustajab
2. Mempergunakan kesempatan dalam keadaan yang mulia
3. Menghadap ke kiblat dan mengangkat kedua tangan
4. Melunakkan suara
5. Tidak memaksakan diri dengan bersajak
6. Merendahkan diri dan khusyu’
7. Mantap dan yakin
8. Bersungguh-sungguh dan mengulanginya sampai tiga kali

9. Menyebut nama Allah
10. Bersih batin

3. Dzikir

(Fatihuddin, 2010) Dzikir secara etimologis berasal dari kata "Dzakaro" yang berarti mengingat. Kata Dzikir berasal dari bentuk masdar, yaitu dzikron, yang kemudian dikenal dengan istilah dzikir. Beberapa orang memahami dzikir sebagai "menyimpan dalam ingatan". Ketika berdzikir kepada Allah, berarti menjaga pikiran agar senantiasa mengingat Allah Ta'ala. Oleh karena itu, menurut syariat, dzikir adalah mengingat Allah dengan cara-cara tertentu yang ditetapkan oleh Al-Qur'an dan hadis, dengan tujuan mensucikan hati dan mengagungkan Allah.

Allah telah menunjukkan prinsip dasar bahwa dzikir dapat menenangkan hati manusia. Hasrat manusia tidak akan pernah terpuaskan (nafsu), kecuali jika dihujani dengan dzikir. Hanya dengan berdzikir hati bisa menjadi tenang, sehingga pikiran jahat tidak tumbuh.

Allah berfirman dalam Q.S Ar Ra'd ayat 28:

ٱلَّذِينَ ءَامَنُواْ وَتَطۡمَئِنُّ قُلُوبُهُم بِذِكۡرِ ٱللَّهِۗ أَلَا بِذِكۡرِ ٱللَّهِ تَطۡمَئِنُّ ٱلۡقُلُوبُ

Artinya: (yaitu) orang-orang yang beriman dan hati mereka manjadi tenteram dengan mengingat Allah. Ingatlah, hanya dengan mengingat Allah-lah hati menjadi tenteram.

Mengingat Allah sangatlah mudah. Cukuplah hati dan pikiran kita "Mengingat"-Nya, itulah yang disebut dengan dzikir. Lalu ada tahapan-tahapan dzikir menurut tingkatan dan derajat hamba-Nya. Ada yang dzikirnya sempurna,

ada pula yang catatannya kurang sempurna. Ini pada taraf betapa mudahnya kita mengingat atau merindukan sanak saudara yang jauh, sehingga cukup disebut dengan mengingat (dzikir) juga. Dan jika orang tersebut ingin mencapai dzikir yang sempurna, tentu saja ia berteman. Persahabatan adalah suatu tindakan atau amal. Jika kita memahami dzikir kepada Allah Ta'ala, maka yang disebut dengan amal dan amalan adalah etika dari dzikir itu sendiri yang disertai akibat dari mengikuti perintah-Nya dan menjauhi larangan-Nya.

Fungsi dan tujuan dzikir adalah untuk mensucikan hati dan mengagungkan Allah. Allah sebenarnya tidak memerlukan dzikir yang diucapkan oleh manusia. Perintah Allah untuk mengingat-Nya semata-mata bertujuan untuk kemaslahatan manusia itu sendiri. Sesuai dengan sifat Allah Yang Maha Mengetahui, manusia sebenarnya tidak perlu berdoa karena Allah sudah mengetahui permintaan mereka sebelum mereka memintanya. Namun, tetap ada perintah untuk berdoa, yang tujuannya adalah demi kebaikan manusia itu sendiri, karena setiap permohonan mereka dicatat sebagai amalan orang yang bertakwa. Salah satu ciri orang beriman adalah keinginan mereka untuk berdoa kepada Allah.

4. Android

Gambar ***2. 1*** *Logo Android*

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler, mencakup sistem operasi, *middleware*, dan aplikasi. Sistem ini awalnya dikembangkan oleh sebuah perusahaan kecil di Silicon Valley bernama Android Inc. Pada tahun 2005, *Google* mengadopsi Android dan mengumumkannya sebagai sistem operasi "*Open Source*". Akibatnya, siapa pun dapat menggunakannya secara gratis, termasuk kode sumber yang diperlukan untuk mengkompilasi sistem operasi tersebut. Android kini menjadi sistem operasi dengan jumlah pengguna terbanyak di dunia, karena sifatnya yang terbuka memungkinkan para *developer* untuk membuatnya lebih fungsional.

Android menyediakan lingkungan yang unik untuk pengembangan aplikasi. Semua aplikasi beroperasi pada level yang sama, tanpa pembedaan antara aplikasi inti dan aplikasi pihak ketiga. *Application Programming Interface (API)* yang disertakan memberikan akses ke perangkat keras serta data ponsel cerdas

atau data sistem itu sendiri. Pengguna bahkan memiliki kemampuan untuk menghapus aplikasi inti dan menggantinya dengan aplikasi pihak ketiga. Adapun Kelebihan dari android yaitu:

1. *Platform* yang komprehensif: Menawarkan berbagai alat yang bermanfaat untuk menciptakan aplikasi, yang dapat terus dikembangkan oleh pengembang.
2. *Platform Open Source*: Karena bersifat terbuka, pengembang memiliki kemudahan dalam mengembangkan Android.
3. *Platform* gratis: Pengembang bebas untuk mengembangkan, mendistribusikan, dan menjual sistem operasi Android tanpa perlu membayar royalti atau memperoleh lisensi.

5. XML (Extensible Markup Language)

(Rahman, 2011) *XML,* singkatan dari *Extensible Markup Language*, merupakan bagian dari *Standart Generalized Markup Language (SGML)* yang ditetapkan oleh *World Wide Web Consortium (W3C)*. *XML* termasuk dalam bahasa yang dapat diperluas, memungkinkan pengguna untuk menentukan elemen *markup* mereka sendiri. Ini berbeda dengan *HTML*, yang memerlukan elemen-elemen tertentu untuk tampilan di *browser* web.

XML adalah dokumen serbaguna yang strukturnya mencerminkan struktur data yang dikandungnya. Secara umum, struktur dokumen *XML* dapat dianggap sebagai pohon hierarki yang terdiri dari elemen-elemen seperti root, parent, dan child. *XML* menetapkan struktur dan konten dokumen melalui serangkaian

elemen, di mana setiap elemen mencakup bagian tertentu dari dokumen. Elemen *XML* terdiri dari atas:

1. *Start Tag* <
2. Isi atau elemen (Dapat berupa teks atau angka)
3. *End Tag* />

Tag Pembuka menunjukkan awal dari suatu elemen, sedangkan Tag Penutup menandakan akhir dari elemen tersebut. Aturan penulisan nama elemen dalam *XML* peka terhadap huruf besar dan kecil serta tidak boleh mengandung spasi. Setiap elemen *XML* bisa memiliki satu atau lebih elemen anak, dan setiap elemen anak tersebut juga merupakan elemen.

Elemen dalam *XML* dapat memiliki penjelasan atau keterangan yang disebut atribut. Sebuah elemen dapat memiliki beberapa atribut. Penulisan atribut dilakukan dengan menuliskan nama atribut, diikuti oleh tanda sama dengan (=), dan nilai atribut.

```
<?xml version="1.0"?>                                  --> Deklarasi
<belajar pelajaran="Ilmu Pengetahuan Alam"></belajar>  --> Elemen
<murid nama="Imam" kelas="12 SMA" absen="15">          --> Atribut
```

Gambar 2.2 *Struktur Dokumen XML*

Untuk dokumen *XML* bisa dibaca dengan benar, harus mematuhi dua tingkat validasi, yang mencakup:

1. Terbentuk Dengan Baik: dokumen *XML* harus memiliki sintaks yang benar, termasuk kesesuaian dalam penulisan elemen, dengan persyaratan bahwa tag awal (<>) harus diikuti oleh tag penutup (< />).

2. Valid: Konten dari elemen-elemen dalam dokumen *XML* harus sesuai dengan aturan semantik atau tipe data yang ditetapkan. Sebagai contoh, konten elemen yang tidak memiliki nilai yang diharapkan dianggap tidak valid.

Proses pembacaan dokumen *XML* yang juga dikenal sebagai *parsing* dapat dilakukan menggunakan berbagai pustaka atau perangkat lunak. Beberapa contohnya melibatkan MS*XML* yang dikembangkan oleh *Microsoft* untuk pengembangan di sistem operasi *Windows* dan *Xerces* yang digunakan untuk pengembangan aplikasi *Java*.

6. Android Studio

Gambar 2. 2 Logo Android Studio

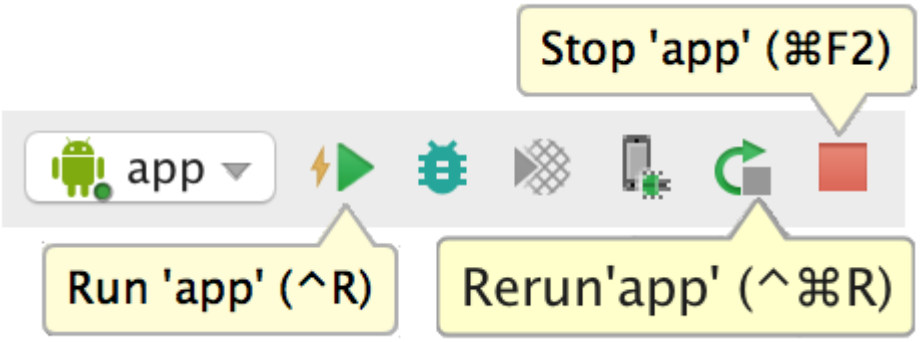
Android Studio adalah sebuah Lingkungan Pengembangan Terpadu *(Integrated Development Environment atau IDE)* yang digunakan untuk membuat aplikasi *Android*. Aplikasi ini dikeluarkan oleh Google pada tanggal 16 Mei 2013 dan dapat diunduh secara gratis dengan *lisensi Apache 2.0. Android Studio* menggantikan perangkat lunak pengembangan *Android* sebelumnya, yakni *Eclipse*. Sebagai sebuah *IDE, Android Studio* menyediakan berbagai fungsi yang terintegrasi yang dibutuhkan oleh pengembang perangkat lunak, termasuk editor kode, *debugger*, *kompiler*, dan sebagainya.

Android Studio merupakan sebuah IDE yang dibangun berdasarkan *IntelliJ IDEA*, memiliki kemiripan dengan *Eclipse*, dan dilengkapi dengan plugin *ADT (Android Development Tools)*. Beberapa fitur yang dimiliki Android Studio antara lain:

1. Proyek berdasarkan *Gradle Build*.
2. *Refactoring* cepat dan perbaikan bug.
3. Alat baru yang disebut "*Lint*" di *Android Studio* mengklaim dapat secara cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.
4. Android Studio mendukung Proguard dan proses penandatanganan aplikasi untuk meningkatkan keamanan. Memiliki *GUI* aplikasi Android lebih mudah
5. Setiap aplikasi yang dikembangkan dengan *Android Studio* didukung oleh *Google Cloud Platform*.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para *programmer*, khususnya *programmer* tingkat dasar yang ingin mempelajari lebih lanjut tentang *Android*. Meski menggunakan *Android Studio* memakan cukup banyak *RAM* pada perangkat *PC*, namun *Android Studio* memiliki sejumlah keunggulan lain, yaitu:

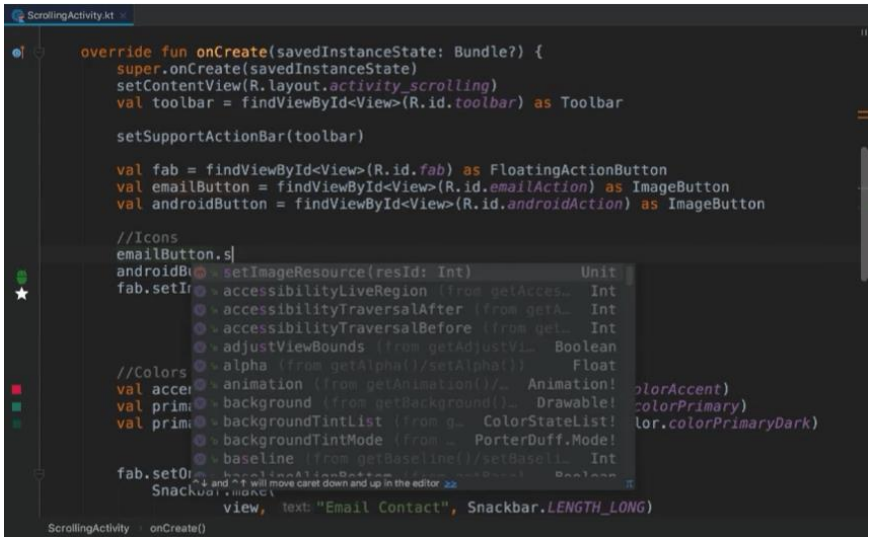
1. *Instant RUN*



Gambar 2. 3 *Instant Run*

Fitur *Instant Run* memungkinkan program berjalan dengan cepat tanpa perlu melakukan kompilasi ulang aplikasi atau menghasilkan ulang APK setiap kali ada perubahan kode. Hal ini mengakibatkan proses pengembangan menjadi lebih cepat.

2. *Intelligent Code Editor*



Gambar 2. 4 *Intelligent Code Editor*

Android Studio dilengkapi dengan *Intelligent Code Editor* yang mempermudah analisis kode dan memberikan saran kode dengan menggunakan fitur *auto-complete*. Ketika kita mengetik kode, *Android*

Studio secara otomatis akan memberikan saran kelas yang tersedia, dan kita dapat menekan tombol TAB untuk memasukkan kode jika sesuai dengan kebutuhan. Fitur-fitur ini secara signifikan meningkatkan produktivitas pembuatan program dengan mempercepat proses pembuatan dan penulisan kode.

3. Sistem Versi yang Fleksibel

   Android Studio menyediakan otomatisasi versi, manajemen dependensi, dan konfigurasi versi yang dapat disesuaikan. Anda dapat mengatur proyek Anda untuk menyertakan pustaka lokal atau di-*hosting*, serta menentukan varian-versi yang berisi kode yang berbeda. Ini memungkinkan Anda untuk mengonfigurasi dan menginstal perpustakaan yang mempermudah dalam pembuatan aplikasi *Android*. Fitur ini mencerminkan fleksibilitas yang dimiliki oleh *Android Studio*.

4. Dioptimalkan untuk semua perangkat Android

   Android Studio menyediakan lingkungan untuk membuat aplikasi yang ditujukan untuk berbagai perangkat Android, termasuk t*ablet Android, Android Wear, Android TV,* dan *Android Auto.* Fitur terstruktur ini memungkinkan Anda membagi proyek menjadi unit-unit fungsional yang dapat dibuat, diuji, dan didebug sesuai kebutuhan Anda.

5. Didesain untuk Tim

   Android Studio terintegrasi dengan berbagai sistem kontrol versi populer seperti Git dan *Subversion*. Bahkan untuk mempermudah kolaborasi, kita dapat menggunakan layanan GitHub langsung dari Android Studio.

Dengan cara ini, pengembang dan tim dapat terus bekerja secara efektif dengan proyek-proyek yang mudah diakses satu sama lain.

7. *Java*

Gambar 2. 5 *Logo Java*

Java merupakan bahasa pemrograman berorientasi objek yang dirancang untuk keperluan komputasi. Dikembangkan dengan tujuan ukurannya kecil, kemudahan, dan portabilitas, *Java* dapat dijalankan di berbagai platform dan sistem operasi. Program yang ditulis dalam bahasa *Java* dapat berupa *applet* (aplikasi kecil yang berjalan di dalam browser web) atau aplikasi mandiri yang dieksekusi menggunakan program *Java Interpreter*.

Bahasa pemrograman *Java* pertama kali muncul dari proyek yang dikenal sebagai Proyek Hijau, yang berlangsung selama 18 bulan dari awal 1991 hingga musim panas 1992. Pada saat itu, proyek tersebut belum menggunakan nama "*Java*". Inisiatif ini diprakarsai oleh Patrik Naughton, Mike Sheridan, James Gosling, dan Bill Joy, bersama sembilan *programmer* lainnya dari *Sun*

Microsystems. Salah satu hasil dari proyek ini adalah penciptaan maskot Duke, yang dibuat oleh Joe Cross.

Mereka menciptakan *browser (Mosaic)* sebagai landasan pertama dalam pembuatan browser *Java* pertama yang diberi nama *Web Runner*, yang terinspirasi dari film *Blade Runner* tahun 1980-an. Pada tahap pengembangan awal, *Web Runner* kemudian diubah namanya menjadi Hot *Java*.

Pada sekitar Maret 1995, kode sumber *Java* versi 1.0a2 pertama kali dibuka. Kesuksesan awal ini pertama kali dilaporkan dalam surat kabar *San Jose Mercury News pada* tanggal 23 Mei 1995. Nama Oak tidak digunakan untuk versi rilis *Java* karena ada merek dagang lain yang sudah menggunakan nama tersebut. Sebagai penggantinya, nama "*Java*" diambil, terinspirasi dari kopi murni yang digiling langsung dari bijinya (*java coffee*), yang merupakan favorit James Gosling. Kopi ini biasanya berasal dari Pulau Jawa. Jadi, nama bahasa pemrograman *Java* dipilih karena kesamaannya dengan kata "*Java*" yang merupakan nama dari pulau tersebut.

8. Firebase

(Maulana, 2020) *Firebase Realtime Database* adalah sistem *database real-time* yang disimpan di *cloud* dan mendukung *multiplatform* seperti Android, iOS, dan Web. Data dalam *Firebase* disimpan dalam format struktur JSON (*JavaScript Object Notation*). *Database Firebase* secara otomatis melakukan sinkronisasi dengan aplikasi klien yang terhubung kepadanya. Aplikasi multiplatform yang menggunakan SDK untuk Android, iOS, dan *JavaScript*

akan secara otomatis menerima pembaruan data terbaru saat terhubung ke server *Firebase*.

Gambar 2. 6 *Logo Firebase*

Firebase Realtime Database adalah sebuah sistem penyimpanan data yang beroperasi dalam mode real-time. Saat ada perubahan dalam data, setiap aplikasi yang terhubung ke Firebase akan secara otomatis mendapatkan pembaruan data di setiap perangkat, termasuk di situs web dan perangkat mobile. *Firebase* menyediakan beragam pustaka untuk berbagai platform, baik web maupun mobile, serta mendukung integrasi dengan berbagai kerangka kerja lainnya seperti *Node.js*, *Java*, *Java*Script, dan lainnya. Beberapa fitur yang ditawarkan oleh *Firebase* termasuk:

1. *Analytics*: Fitur ini berguna untuk memantau perilaku pengguna saat menggunakan aplikasi dan hasilnya dipresentasikan dalam satu dashboard.
2. *Develop*: Fitur ini mencakup berbagai layanan seperti pesan *cloud*, otentikasi, *database real-time*, penyimpanan, *hosting*, laboratorium pengujian, dan pelaporan kegagalan.
3. *Grow*: Fitur ini berguna untuk meluncurkan produk aplikasi ke publik.

Gambar 2. 7 *Fitur Firebase*

9. *UML (Unified Modelling Language)*

UML (Unified Modeling Language) adalah sebuah alat untuk merancang sistem berorientasi objek. Secara filosofis, kemunculan *UML* terinspirasi dari konsep yang telah ada sebelumnya, yaitu konsep pemodelan berorientasi objek (OO). Konsep ini menggambarkan sistem seperti kehidupan nyata yang didominasi oleh objek, dan direpresentasikan atau didokumentasikan dalam simbol-simbol yang khusus. *UML* menyediakan proses standar dan independen.

Tujuan utama dari diagram *UML* adalah untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau penulis program. Komponen dan notasi *UML* berasal dari tiga notasi yang sudah ada sebelumnya, yaitu Grady Booch dari *OOD (Object-Oriented Design)*, Jun Rumbaugh dari *OMT (Object Modeling Technique)*, dan Ivar Jacobson dari *OOSE (Object-Oriented Software Engineering)*.

UML terbagi menjadi tiga kategori utama: diagram struktur, diagram perilaku, dan diagram interaksi. Tiap kategori ini menggambarkan arsitektur sistem dan saling terintegrasi.

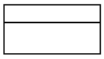
Adapun daftar simbol *UML* yaitu:

Tabel 2. 1 *Symbol Use Case Diagram*

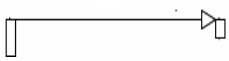
No	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpunan peran yang dimainkan oleh pengguna saat berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan di mana perubahan pada suatu elemen independen akan memengaruhi elemen yang bergantung padanya disebut sebagai hubungan ketergantungan.
3		*Generalization*	Hubungan di mana objek anak mewarisi perilaku dan struktur data dari objek yang berada di atasnya, disebut sebagai hubungan pewarisan atau *inheritance*.
4		*Include*	Menyatakan *Use Case* sumber secara eksplisit.
5		*Extend*	Menyatakan secara eksplisit bahwa *use case* target memperluas perilaku dari *use case* sumber pada titik tertentu.
6		*Association*	Menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menentukan paket yang menampilkan sistem dalam lingkup yang terbatas.
8		*Use Case*	Deskripsi dari serangkaian langkah-langkah yang dilakukan oleh sistem untuk menghasilkan hasil yang terukur bagi seorang aktor.
9		*Collaboration*	Interaksi antara aturan-aturan dan elemen lainnya yang bekerja bersama untuk menciptakan perilaku yang lebih besar daripada sekadar jumlah dan elemen-elemennya, disebut sebagai sinergi.

No	GAMBAR	NAMA	KETERANGAN
10		*Note*	Elemen fisik yang ada saat aplikasi berjalan dan merepresentasikan sumber daya komputasi disebut sebagai infrastruktur.

Tabel 2. 2 *Symbol Class Diagram*

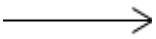
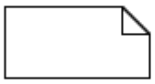
No.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan di mana objek anak memperoleh perilaku dan struktur data dari objek induknya disebut sebagai hubungan pewarisan atau *inheritance.*
2		*Nary Association*	Upaya menghindari yang lebih dari 2 objek.
3		*Class*	Kelompok objek-objek yang memiliki atribut dan operasi yang sama disebut sebagai kelas.
4		*Collaboration*	Deskripsi urutan langkah-langkah yang dilakukan oleh sistem untuk menghasilkan hasil yang dapat diukur bagi seorang aktor.
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek disebut metode.
6		*Dependency*	Hubungan di mana perubahan pada suatu elemen mandiri akan mempengaruhi elemen-elemen yang bergantung padanya disebut sebagai ketergantungan.
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 *Symbol Sequence Diagram*

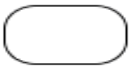
No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Entitas objek yang berinteraksi melalui antarmuka.
2		*Message*	Spesifikasi komunikasi antar objek yang mencakup informasi tentang aktivitas yang terjadi disebut sebagai protokol komunikasi.
3		*Message*	Spesifikasi komunikasi antar objek yang mencakup informasi

			tentang aktivitas yang terjadi disebut sebagai protokol komunikasi.

Tabel 2. 4 *Symbol State Chart Diagram*

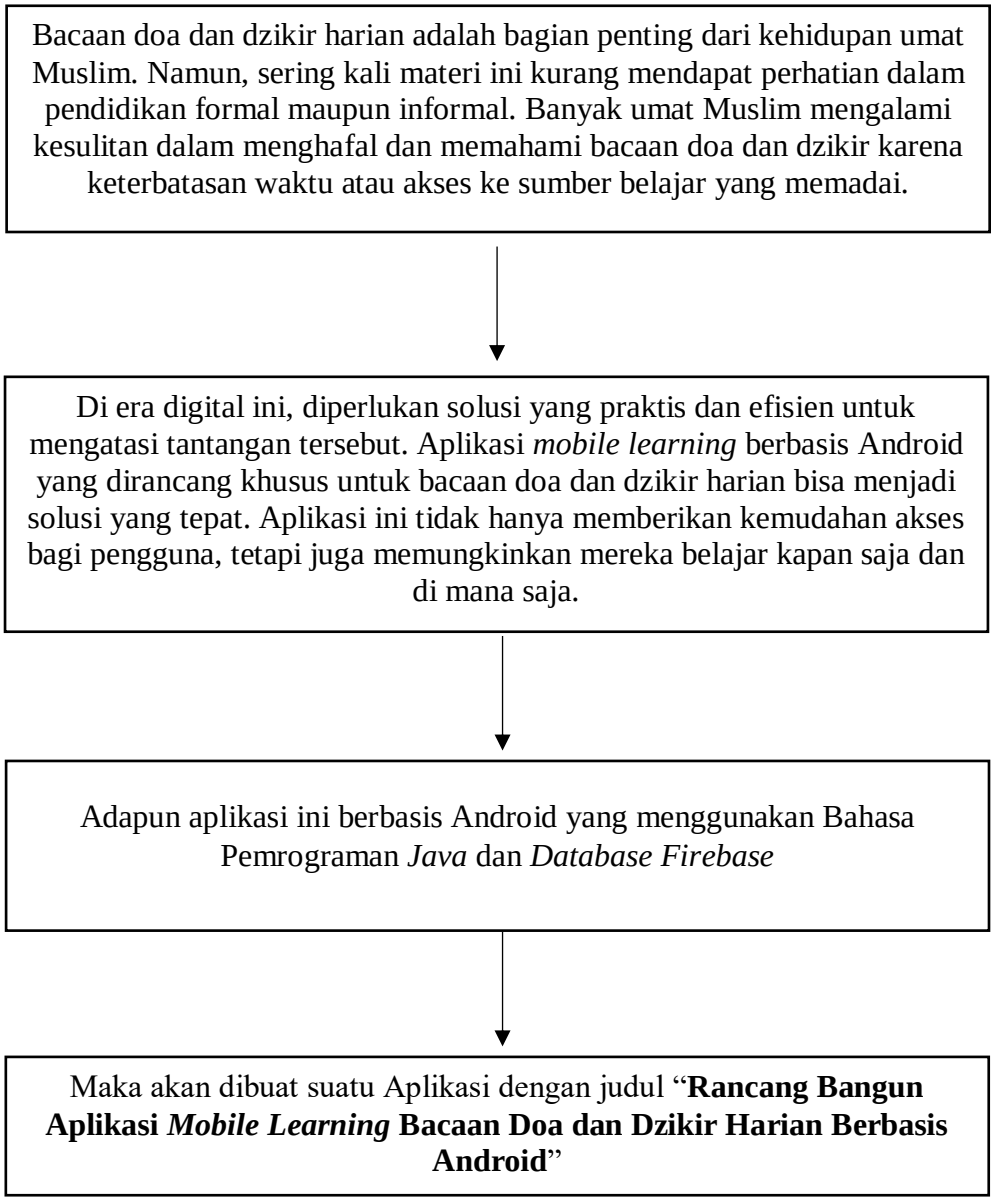
No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai tautan pada suatu objek pada waktu tertentu disebut sebagai status objek.
2		*Initial Pseudo State*	objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan diselesaikan
4		*Transition*	Sebuah kejadian yang menginisiasi perubahan pada suatu objek dengan memperbarui satu atau lebih nilai atributnya disebut sebagai *trigger state* objek.
5		*Association*	menyambungkan objek satu dengan objek lain.
6		*Node*	Elemen fisik yang ada saat aplikasi dijalankan dan mencerminkan sumber daya komputasi disebut sebagai infrastruktur komputasi.

Tabel 2. 5 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Menampilkan interaksi antara setiap kelas antarmuka dalam sebuah diagram aliran.
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi disebut sebagai *state* aksi.
3		*Initial Node*	Bagaimana suatu objek diciptakan atau diinisialisasi.
4		*Actifity Final Node*	Bagaimana objek dibuat dan dihapus.
5		*Fork Node*	Satu aliran yang pada suatu titik terbagi menjadi beberapa aliran.

C. Kerangka Pikir

Untuk Memahami alur peneltian diatas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram ini:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah jenis penelitian kualitatif, dimana metode penelitian ini digunakan untuk memahami persepsi dan kebutuhan pengguna pada aplikasi *mobile learning* bacaan doa dan dzikir harian berbasis android.

B. Waktu dan Tempat Penelitian

Rencana waktu yang digunakan untuk penelitian ini berlangsung selama ± 2 bulan dan bertempat di Masjid Al-Huda Kota Parepare.

C. Alat dan Bahan Penelitian

1. Hardware

Adapun spesifikasi perangkat keras yang digunakan dalam penyusunan proposal ini :

a. Laptop Acer Aspire 3 A315-41-R971
b. *Processor* Amd Ryzen 5 2500U
c. RAM 8,00 GB
d. HDD 500 GB

2. Software

Software yang digunakan dalam pembuatan aplikasi yaitu:

a. Windows 10

b. Android Studio

c. *Java*

d. Firebase

D. Metode Pengumpulan Data

1. Secara tidak langsung

   Metode tidak langsung adalah cara untuk mengumpulkan data atau informasi yang ditemukan dalam sumber-sumber seperti buku, internet, jurnal, dan artikel.

2. Secara Langsung

   Metode secara langsung adalah pendekatan untuk mengumpulkan data atau informasi yang berkaitan langsung dengan ustadz di lokasi penelitian.

E. Tahapan Penelitian

Tahap-tahap penelitian yang dimaksud dalam penelitian ini merujuk pada proses pelaksanaan penelitian, yang melibatkan serangkaian langkah yang sistematis untuk merancang, melaksanakan, dan menganalisis penelitian.

1. Persiapan Penelitian

   Tahap persiapan adalah fase yang dilakukan sebelum memulai penelitian. Pada tahap ini, langkah awalnya adalah memeriksa permasalahan yang ada dan kemudian melakukan studi literatur terkait dengan masalah yang sedang diteliti.

2. Studi Literatur

Pada tahapan ini, terdapat empat langkah yang harus dipenuhi untuk mencapai hasil maksimal dalam penelitian. Langkah-langkah tersebut meliputi pengumpulan data dengan menggunakan berbagai metode pengumpulan data yang telah dijelaskan sebelumnya, pengolahan data, analisis data, dan penafsiran hasil analisis.

Setelah tahap persiapan, langkah berikutnya adalah melakukan tugas lapangan untuk mengumpulkan data yang akan diproses. Proses ini mencakup penyuntingan data, penerapan masalah dalam aplikasi program, serta analisis untuk menarik kesimpulan sebagai hasil akhir.

3. Pengumpulan Data

Pada tahap ini, peneliti melakukan pencarian data dari berbagai sumber untuk dikumpulkan dan kemudian dikaji lebih lanjut.

4. Analisis

Pada tahap analisis, peneliti menganalisis permasalahan yang diteliti dan merumuskan masalah pokok penelitian untuk menyusun alternatif pemecahan masalah.

5. Perancangan

Peneliti kemudian mendesain aplikasi berdasarkan solusi alternatif yang telah diidentifikasi.

6. Pengujian

Setelah menyelesaikan perancangan, peneliti menguji hasilnya. Jika ditemukan kekurangan atau kelemahan, mereka akan kembali ke tahap analisis.

7. Implementasi

   Setelah memastikan tidak ada kekurangan dalam perancangan, aplikasi siap digunakan oleh pengguna.

8. Tahap Penyelesaian

   Tahap penyelesaian adalah tahap akhir dalam penelitian. Pada tahap ini, laporan penelitian disusun.

F. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *blackbox testing* dan *whitebox testing*:

1. *Blackbox testing*

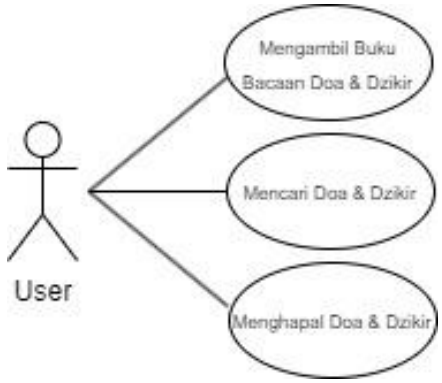
   Blackbox testing berfokus pada fungsionalitas program. Dalam Blackbox testing, program dijalankan dan diamati apakah berfungsi dengan baik atau tidak. Teknik yang digunakan adalah *equivalence partitions*, yaitu pengujian berdasarkan *input* dari setiap menu dalam program. Setiap *input* diuji melalui klasifikasi dan pengelompokan berdasarkan fungsinya

2. *Whitebox testing*

Whitebox testing bertujuan untuk memastikan bahwa struktur aplikasi sesuai dengan ketentuan. Pengujian ini menitikberatkan pada pemeriksaan detail perancangan perangkat lunak. Prosesnya melibatkan mendefinisikan semua alur perangkat lunak, membangun kasus uji, dan kemudian menguji kasus tersebut untuk mendapatkan hasilnya.

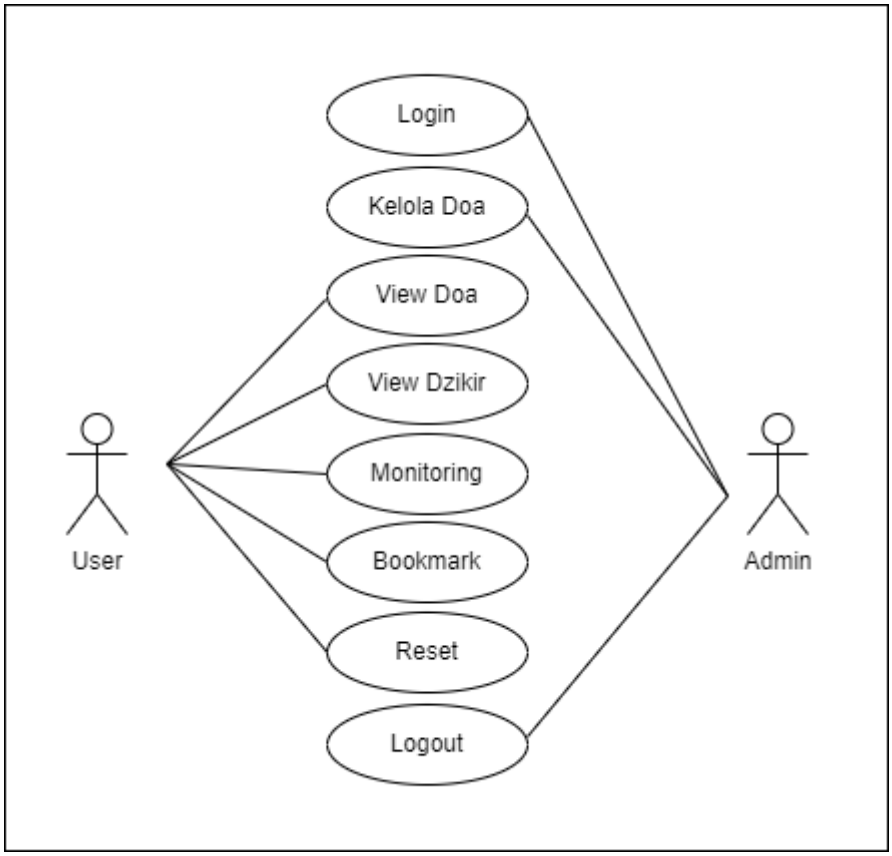
G. Desain Sistem

1. Desain sistem yang berjalan



Gambar 3. 1 *UML* Sistem Yang Berjalan

2. Desain sistem yang diusulkan



Gambar 3. ***2*** *UML* Sistem yang Diusulkan

BAB IV

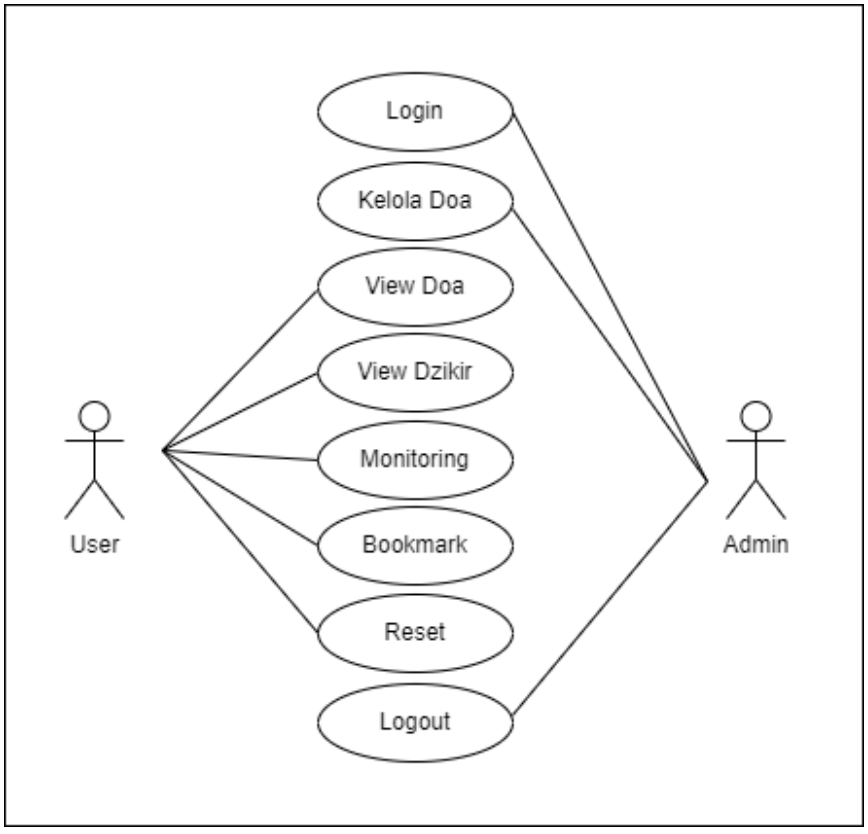
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data *UML*

Rancangan sistem ini peneliti gambarkan menggunakan diagram *UML* yaitu *use case diagram,* dan *Activity diagram.*

1. *Use Case Diagram*

   Use Case Diagram berfungsi untuk melakuka kelebihan sistem jika dilihat menurut sudut pandang individu yang berada di luar sistem (*user* dan admin)



Gambar 4. 1 *Use Case Diagram User* dan *Admin*

Interaksi antara user dan admin dengan aplikasi dijelaskan dalam skenario *use case* sebagai berikut:

Tabel 4. 1 Penjelasan *Use Case Diagram* Admin

Nama *Use Case*	**Deskripsi *Use Case***
Login	*Use case* ini menjelaskan akses masuk untuk administrator. Pengguna diharuskan memasukkan nama pengguna (*username*) dan kata sandi (*password*) yang valid untuk dapat mengakses tambah data.
Kelola Doa	*Use case* ini memungkinkan administrator untuk mengelola daftar doa yang tersedia dalam aplikasi. *Administrator* dapat menambahkan doa baru, mengedit doa yang sudah ada, dan menghapus doa yang tidak diperlukan lagi.
Logout	*Use case* ini menjelaskan fitur untuk keluar dari akun administrator, memastikan bahwa sesi *login* diakhiri dengan aman dan tidak ada akses tanpa izin setelah *logout*.

Tabel 4. 2 Penjelasan *Use Case Diagram User*

Nama *Use Case*	**Deskripsi *Use Case***
View Doa	*Use case* ini menjelaskan tentang daftar doa yang tersedia dalam aplikasi, memungkinkan pengguna untuk melihat dan membaca doa-doa yang telah disimpan oleh *administrator*.
View Dzikir	*Use case* ini menjelaskan tentang daftar doa dzikir yang tersedia dalam aplikasi, mengelompokkan doa-doa berdasarkan waktu pelaksanaan: pagi, petang, dan ba'da sholat. Pengguna dapat melihat dan membaca doa-doa dzikir yang relevan dengan kebutuhan mereka.
Monitoring	*Use case* ini menjelaskan tentang fitur untuk memonitor dan merekap data bulanan mengenai aktivitas pengguna dalam membaca doa dan dzikir. Pengguna dapat melihat statistik bacaan mereka setiap bulan.

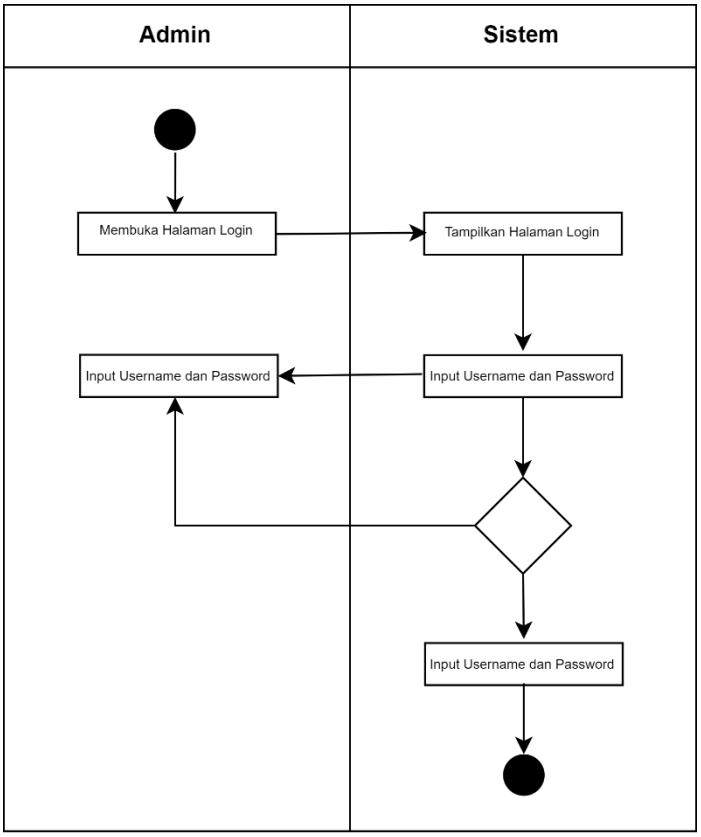
Nama *Use Case*	Deskripsi *Use Case*
Bookmark	*Use case* ini menjelaskan tentang fitur untuk menandai dan mengingat bacaan doa terakhir yang dibaca oleh pengguna, memungkinkan mereka untuk melanjutkan dari titik terakhir tanpa perlu mencari kembali.
Reset	*Use case* ini menjelaskan tentang fitur untuk mereset data dari *Monitoring* dan *Bookmark*, memungkinkan pengguna untuk menghapus semua rekaman dan memulai kembali dari awal.

2. *Activity Diagram*

Activity diagram merupakan bahasa spesifikasi standar yang dipakai untuk mendokumentasikan, mengspesifikasikan, dan mengembangkan perangkat lunak. Diagram ini menggambarkan berbagai aktivitas dalam aliran proses sebuah sistem.

a. *Activity Diagram* Admin

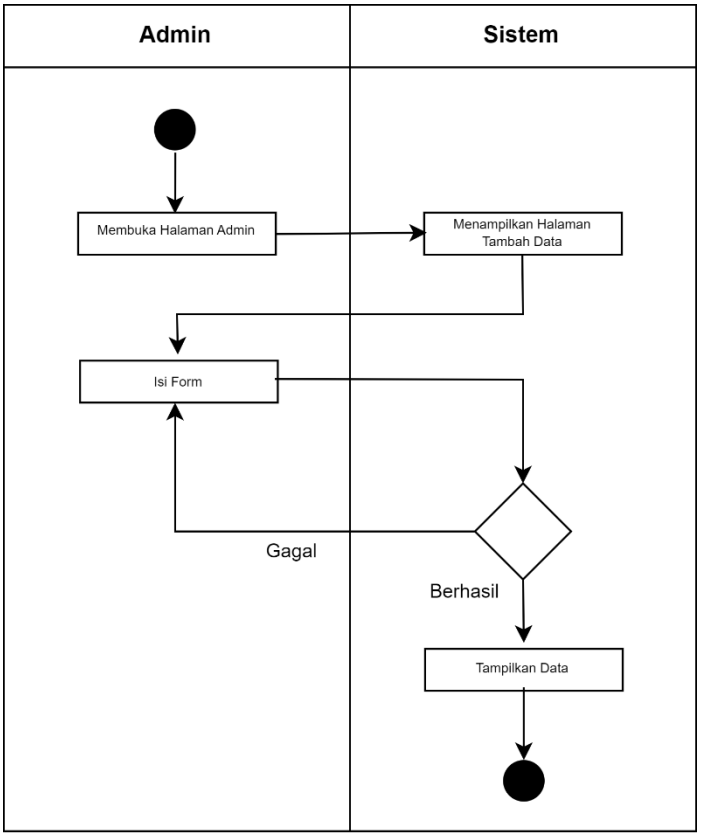
1) *Activity Diagram Login*



Gambar 4. 2 *Activity Diagram Login*

Pada Gambar 4.2 menjelaskan proses *Login* dari admin. Pertama yang dilakukan adalah admin membuka *website* dan memilih menu *Login* kemudian sistem akan menampilkan form *Login*, kemudian admin memasukkan *username* dan *password*. Selanjutnya sistem akan melakukan validasi, jika benar maka akan dilanjutkan ke halaman *dashboard* namun jika salah maka akan kembali ke halaman *login* dengan menampilkan pesan kesalahan.

2) *Activity* Diagram Tambah Data



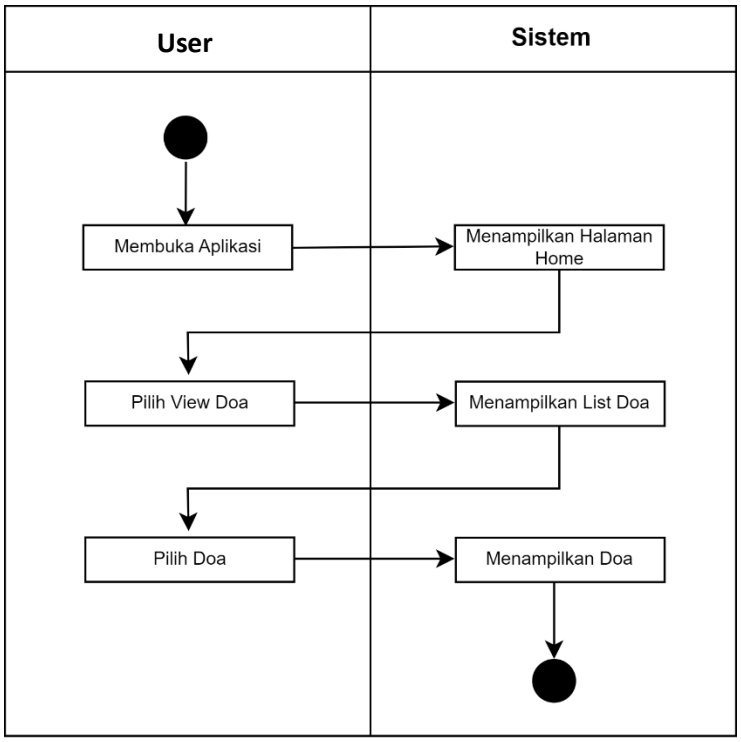
Gambar 4. 3 *Activity Diagram* Tambah Data

Pada gambar 4.3 menjelaskan bagaimana proses yang dilakukan admin jika ingin menambah data. Pertama yang dilakukan adalah membuka halaman admin kemudian sistem akan menampilkan *dashboard*. Selanjutnya *admin* diharapkan memilih menu yang akan ditambahkan data, kemudian sistem akan menampilkan halaman menu yang dipilih admin, selanjutnya admin memilih tambah data. Maka sistem akan menampilkan Form tambah data. Setelah itu *admin* diharapkan mengisi form tambah data, setelah diisi maka sistem akan melakukan validasi data, apabila berhasil maka sistem akan menampilkan halaman menu yang ditampilkan data dan sebaliknya jika

gagal maka sistem akan menampilkan pesan kesalahan dan admin diharapkan mengisi kembali form tambah data tersebut dengan benar

b. *Activity* Diagram User

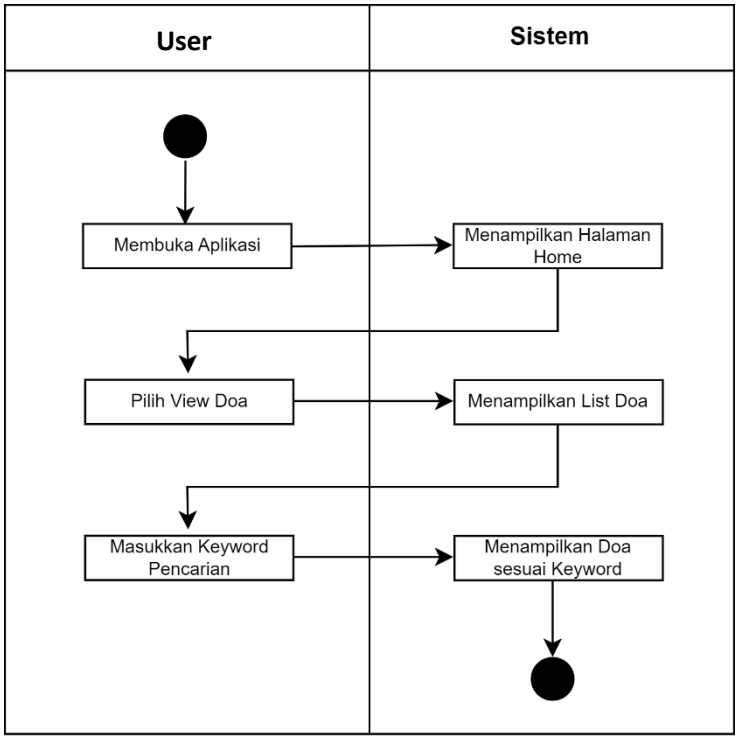
1) *Activity* Diagram Pilih *View* Doa



Gambar 4. 4 *Activity Diagram* Pilih *View* Doa

Pada gambar 4.4 menjelaskan bagaimana proses yang dilakukan *user* jika ingin menampilkan *View* doa. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman *home*. Selanjutnya *user* diharapkan memilih *View* doa kemudian sistem akan menampilkan List doa dan *user* diharapkan memilih doa yang akan dibaca kemudian sistem akan menampilkan doa yang telah dipilih.

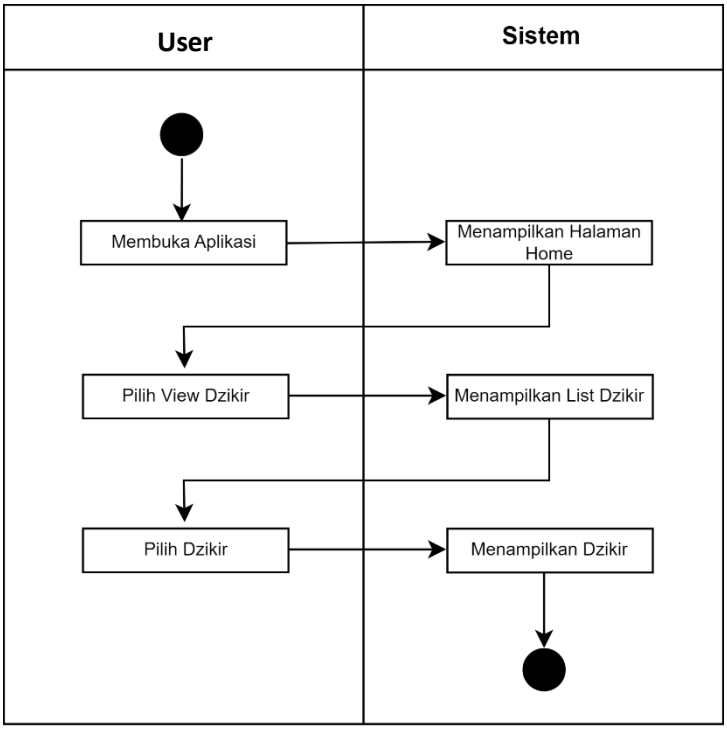
2) *Activity Diagram* Cari Doa



Gambar 4. 5 *Activity Diagram* Cari Doa

Pada gambar 4.5 menjelaskan bagaimana proses yang dilakukan *user* jika ingin mencari doa. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman home. Selanjutnya *user* diharapkan memilih *View* doa kemudian sistem akan menampilkan daftar doa dan *user* diharapkan memasukkan *keyword* doa yang akan dibaca kemudian sistem akan menampilkan doa sesuai *keyword*.

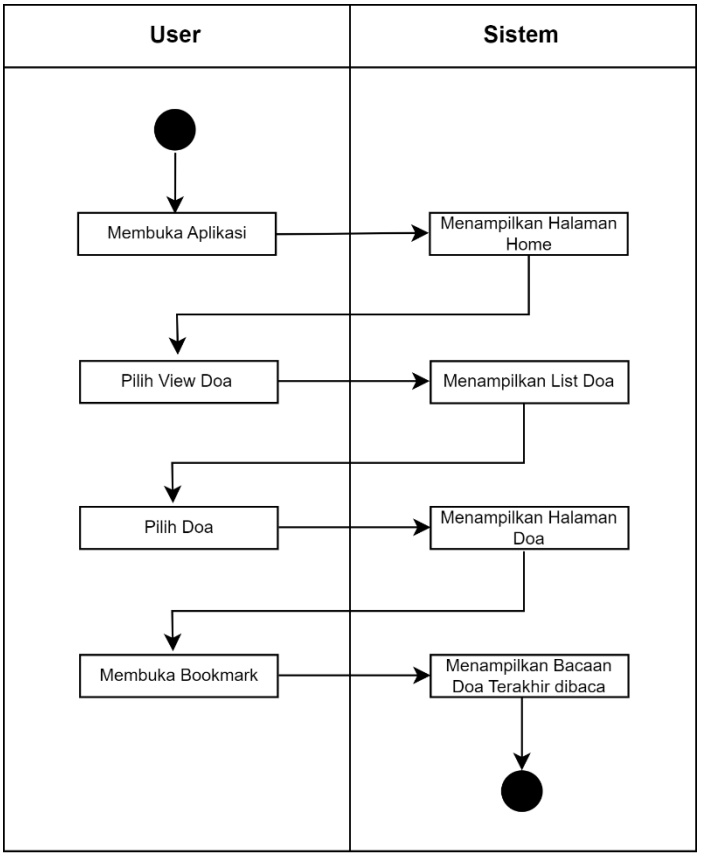
3) *Activity* Diagram Pilih *View* Dzikir



Gambar 4. 6 *Activity Diagram* Pilih *View* Dzikir

Pada gambar 4.6 menjelaskan bagaimana proses yang dilakukan *user* jika menampilkan daftar dzikir. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman *home*. Selanjutnya *user* diharapkan memilih *View* Dzikir kemudian sistem akan menampilkan daftar dzikir. Selanjutnya *user* diharapkan memilih dzikir kemudian sistem akan menampilkan dzikir.

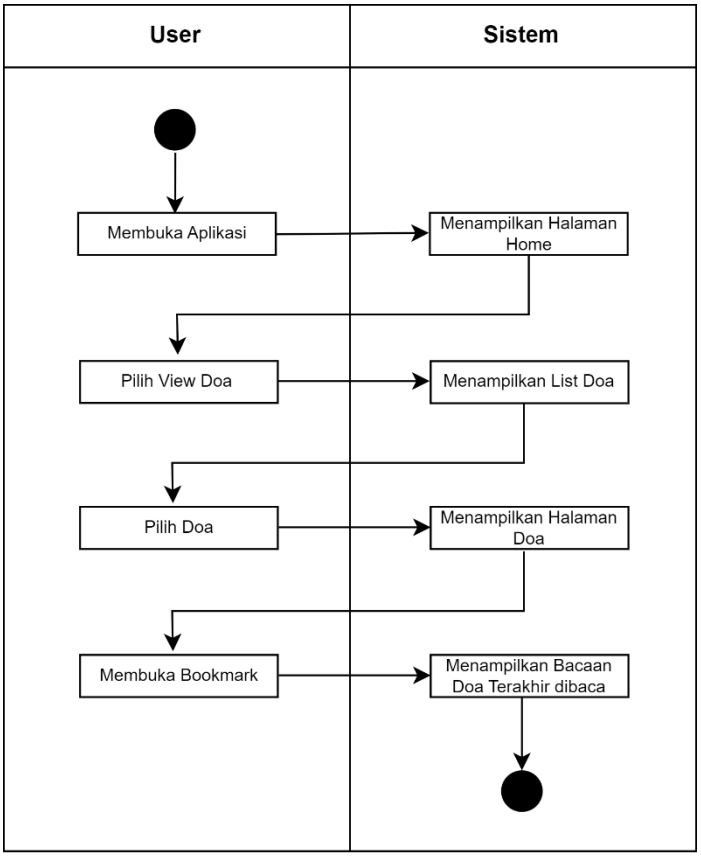
4) *Activity Diagram Monitoring*



Gambar 4. 7 *Activity Diagram Monitoring*

Pada gambar 4.7 menjelaskan bagaimana proses yang dilakukan *user* jika ingin me-*Monitoring* hapalan doa. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman *home*. Selanjutnya *user* diharapkan memilih *View* doa kemudian sistem akan menampilkan daftar doa. Selanjutnya *user* diharapkan memilih doa kemudian sistem akan menampilkan doa, kemudian *user* menekan tombol selesai maka sistem akan me-*Monitoring* doa yang telah dihapal.

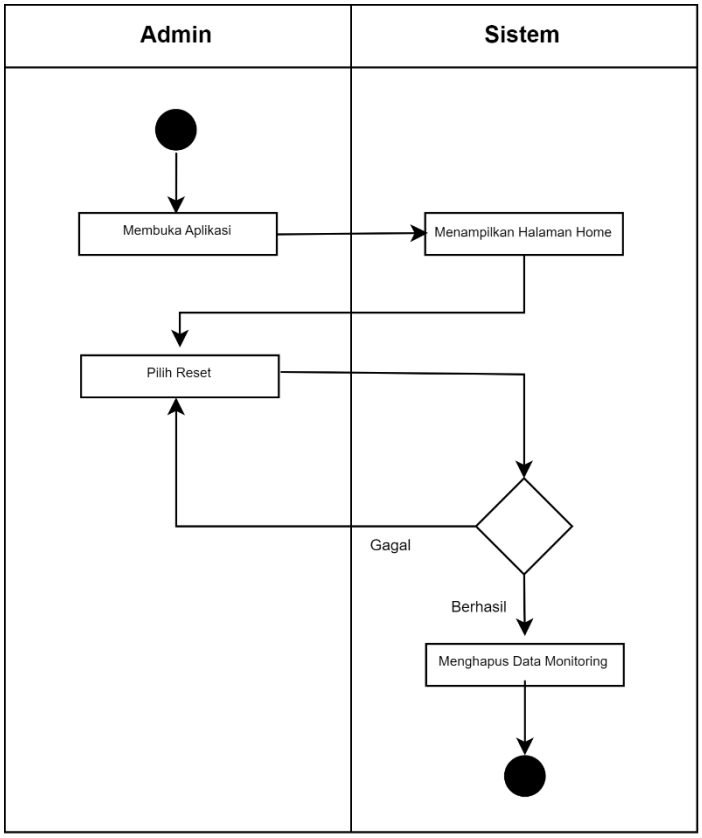
5) *Activity* Diagram Fitur *Bookmark*



Gambar 4. 8 *Activity Diagram Fitur Bookmark*

Pada gambar 4.8 menjelaskan bagaimana proses yang dilakukan *user* jika ingin melihat doa yang terakhir dibaca. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman *home*. Selanjutnya *user* diharapkan memilih *View* doa kemudian sistem akan menampilkan daftar doa. Selanjutnya *user* diharapkan memilih doa kemudian sistem akan menampilkan doa, kemudian sistem akan menampilkan bacaan doa yang terakhir dibaca.

6) *Activity* Diagram Reset



Gambar 4. 9 *Activity Diagram Reset*

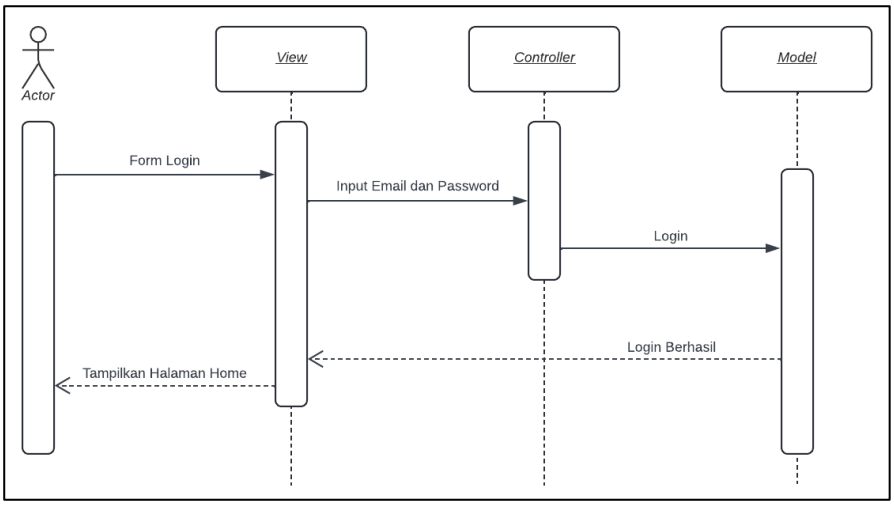
Pada gambar 4.9 menjelaskan bagaimana proses yang dilakukan *user* jika ingin mereset data di *Monitoring* doa. Pertama yang dilakukan *user* adalah membuka aplikasi kemudian sistem akan menampilkan halaman *home*. Selanjutnya *user* diharapkan memilih reset kemudian sistem akan menampilkan pesan apakah anda yakin mereset data. Selanjutnya *user* diharapkan memilih antara ya atau tidak. Jika ya maka sistem akan mereset data yang ada di *Monitoring* doa, tetapi jika tidak maka sistem akan menampilkan tampilan home.

3. *Sequence* Diagram

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

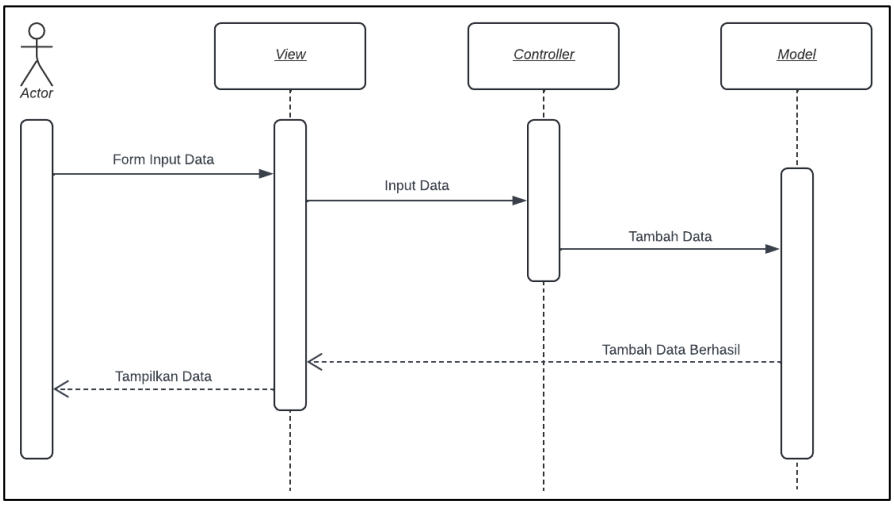
a. *Sequence Diagram* Admin

1) *Sequence Diagram Login*



Gambar 4. 10 *Sequence Diagram Login*

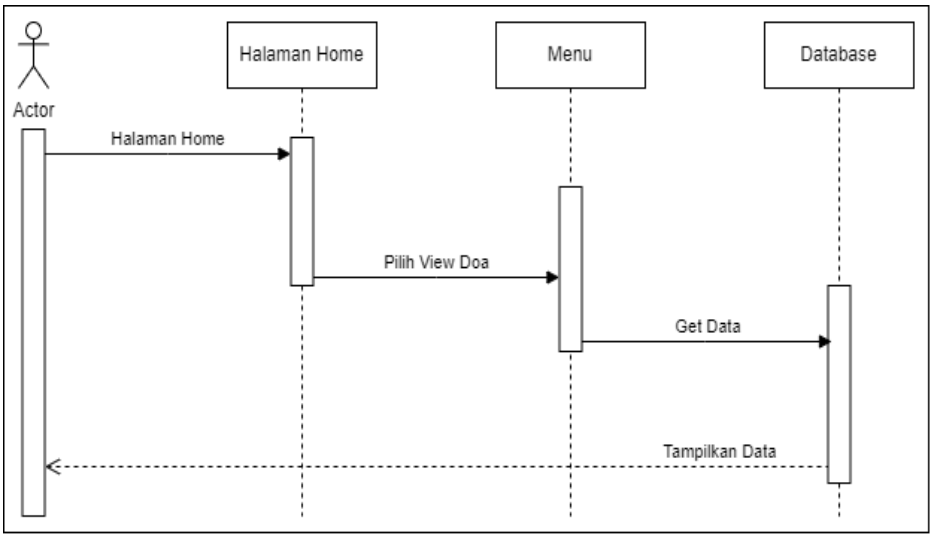
2) *Sequence Diagram* Tambah Data



Gambar 4. 11 *Sequence Diagram Tambah Data*

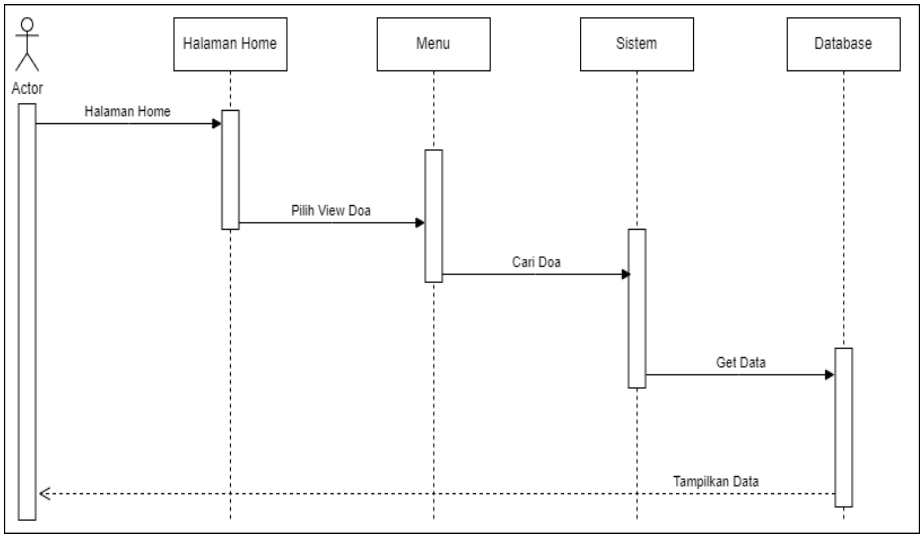
b. *Sequence Diagram User*

1) *Sequence Diagram* Pilih *View* Doa



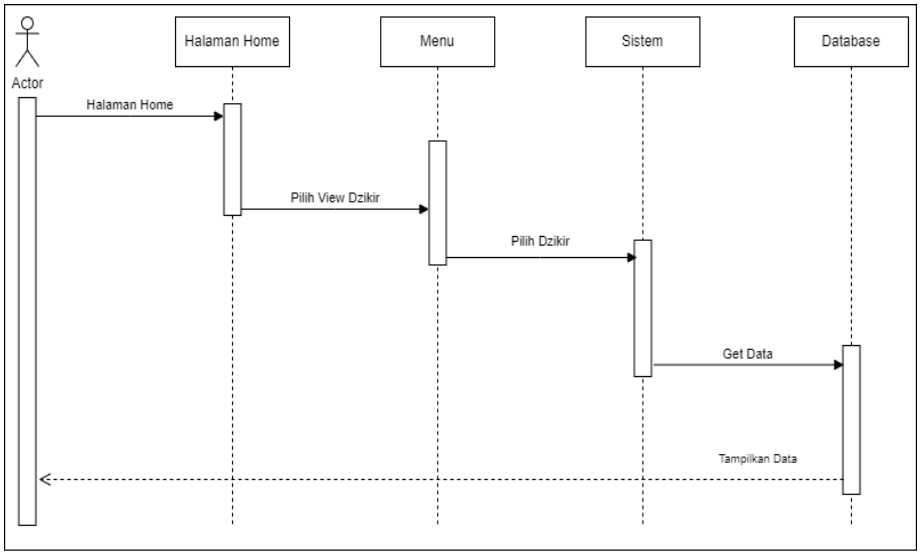
Gambar 4. 12 *Sequence Diagram Pilih View Doa*

2) *Sequence Diagram* Cari Doa



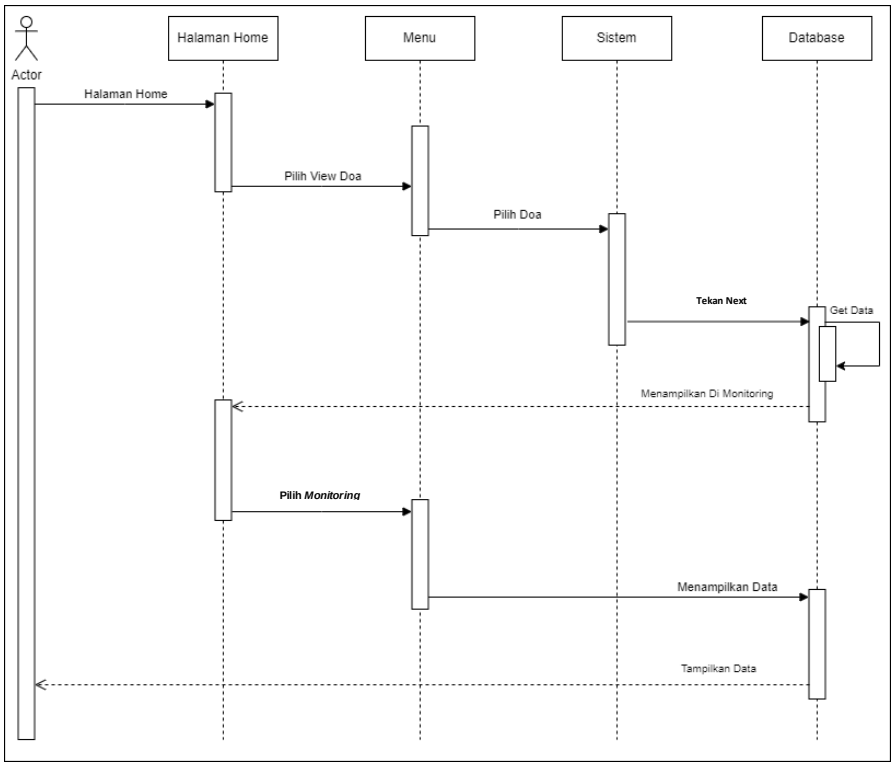
Gambar 4. 13 *Sequence Diagram Cari Doa*

3) *Sequence Diagram* Pilih *View* Dzikir



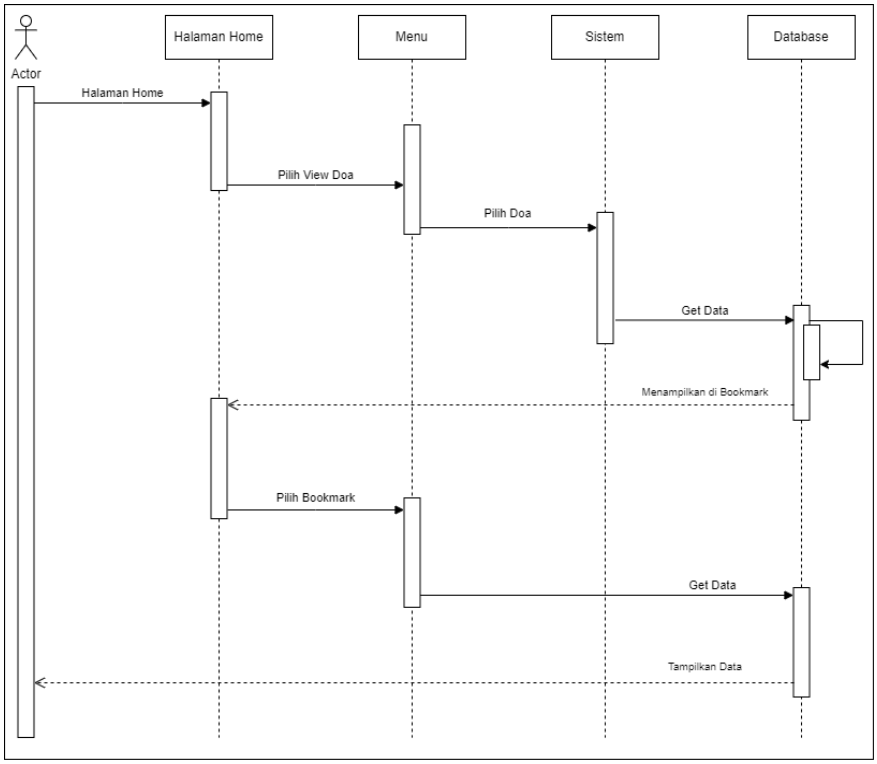
Gambar 4. 14 *Sequence Diagram Pilih View Dzikir*

4) *Sequence Diagram Monitoring*



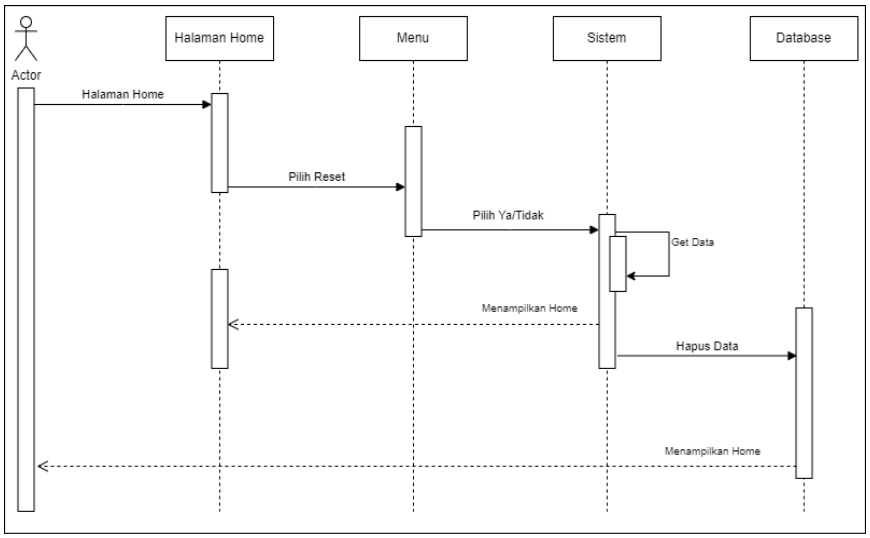
Gambar 4. 15 *Sequence Diagram Monitoring*

5) *Sequence Diagram* Fitur *Bookmark*



Gambar 4. 16 *Sequence Diagram Fitur Bookmark*

6) *Sequence Diagram* Reset



Gambar 4. 17 *Sequence Diagram Reset*

B. Detail Rancangan Database

Berikut ini adalah tabel-tabel yang digunakan pada rancang bangun aplikasi mobile learing bacaan doa dan dzikir harian berbasis android beserta strukturnya:

1. Tabel Doa

   Tabel doa adalah table yang dipakai untuk menyimpan data-data dari bacaan doa

   Tabel 4. 3 *Database* Doa

Field	***Type***	***Description***
arab	*String*	Teks dalam Bahasa arab
artinya	*String*	Terjemahan teks dalam bahasa Indonesia
audio	*String*	*URL* file audio
judul	*String*	Judul dari doa
key	*String*	Kunci unik untuk setiap entri
video	*String*	*URL* file video

2. Tabel badah

   Tabel badah adalah table yang dipakai untuk menyimpan data-data dari bacaan dzikir ba'da sholat.

   Tabel 4. 4 *Database* badah

Field	***Type***	***Description***
arab	*String*	Teks dalam Bahasa arab
artinya	*String*	Terjemahan teks dalam bahasa Indonesia

audio	*String*	*URL* file audio
judul	*String*	Judul dari doa

3. Tabel pagi

Tabel pagi adalah table yang dipakai untuk menyimpan data-data dari bacaan dzikir pagi.

Tabel 4. 5 *Database* Pagi

Field	***Type***	***Description***
arab	*String*	Teks dalam Bahasa arab
artinya	*String*	Terjemahan teks dalam bahasa Indonesia
audio	*String*	*URL* file audio
judul	*String*	Judul dari doa

4. Tabel petang

Tabel pagi adalah table yang dipakai untuk menyimpan data-data dari bacaan dzikir petang.

Tabel 4. 6 *Database* Petang

Field	***Type***	***Description***
arab	*String*	Teks dalam Bahasa arab
artinya	*String*	Terjemahan teks dalam bahasa Indonesia
audio	*String*	URL file audio
judul	*String*	Judul dari doa

5. Tabel *Login*

Tabel *Login* adalah tabel yang dipakai untuk menyimpan informasi terkait proses masuk ke dalam sistem, seperti nama pengguna dan kata sandi.

Tabel 4. 7 Database *Login*

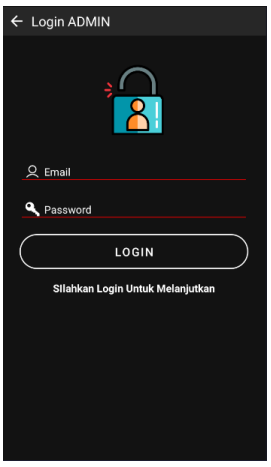
Field	***Type***	***Description***
username	*String*	Nama pengguna yang unik untuk *login* ke sistem
password	*String*	Kata sandi yang aman untuk akses ke akun

C. Detail Sistem

1. Admin

a. Halaman *Login*

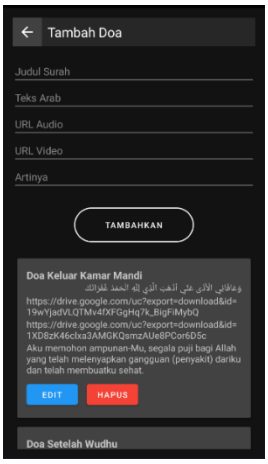
Merupakan halaman awal yang digunakan admin untuk mendapat akses masuk ke halaman tambah data.



Gambar 4. 18 Halaman *Login*

b. Halaman Kelola Doa

Merupakan halaman yang berisi form untuk melakukan tambah, edit dan hapus doa.

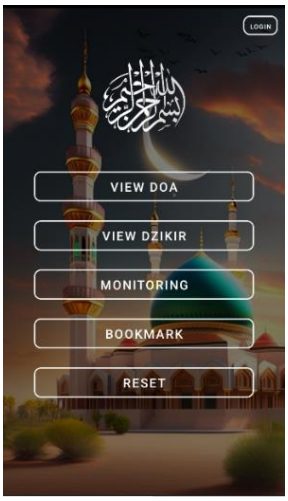


Gambar 4. 19 Halaman Kelola Doa

2. User

a. Tampilan Halaman Utama

Merupakan halaman awal bagi user yang mengakses aplikasi bacaan doa dan dzikir harian yang berisi fitur *View* doa, *View* dzikir, *Monitoring*, *Bookmark*, reset dan tombol *Login* untuk admin.



Gambar 4. 20 Tampilan Halaman Utama

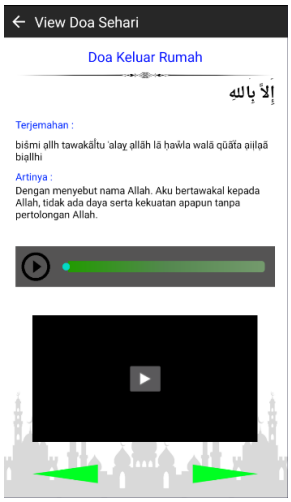
b. Tampilan Halaman *View* Doa

Merupakan halaman yang berfungsi untuk menampilkan daftar doa harian dan dilengkapi dengan fitur pencarian untuk memudahkan mencari doa harian.

Gambar 4. 21 Tampilan Halaman Daftar Doa

c. Tampilan Halaman Detail Doa

Merupakan halaman yang berfungsi untuk menampilkan bacaan doa yang *user* pilih, dihalaman tersebut berisi fitur suara dan video.



Gambar 4. 22 Tampilan Halaman Detail Doa

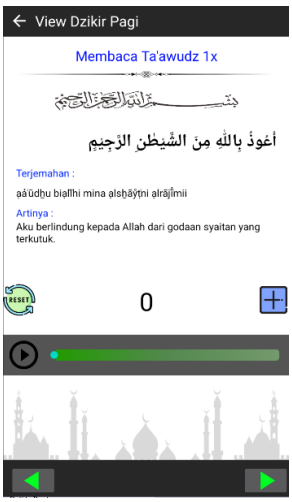
d. Tampilan Halaman *View* Dzikir

Merupakan halaman yang berfungsi untuk menampilkan daftar Dzikir seperti Dzikir pagi, Dzikir Petang dan Dzikir Ba'da Shalat.

Gambar 4. 23 Tampilan *View* Dzikir

e. Tampilan Halaman Dzikir

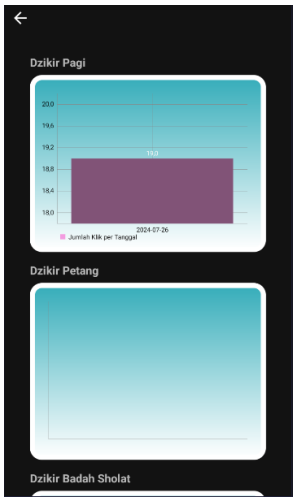
Merupakan halaman yang berfungsi untuk menampilkan dzikir-dzikir. Dihalaman dzikir ini berisi fitur suara dan penghitung dzikir digital.



Gambar 4. 24 Tampilan Halaman Dzikir

f. Tampilan Halaman *Monitoring*

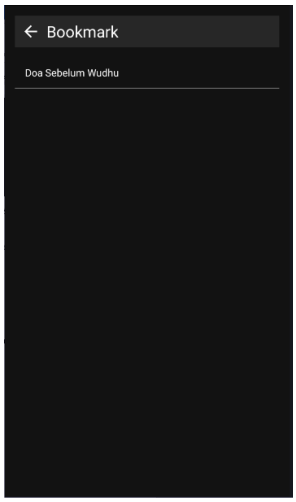
Merupakan halaman yang berfungsi untuk menampilkan grafik pengguna dalam membaca bacaan doa dan dzikir berdasarkan j*uml*ah klik *next* setiap harinya.



Gambar 4. 25 Tampilan Halaman *Monitoring*

g. Tampilan Halaman *Bookmark*

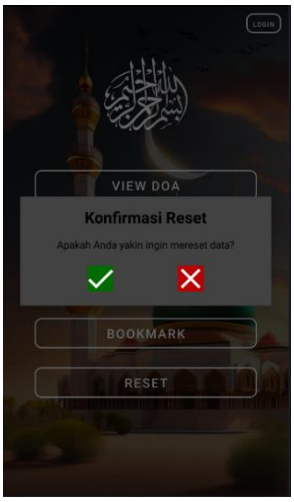
Merupakan halaman yang berfungsi untuk menandai bacaan doa terakhir yang *user* baca.



Gambar 4. 26 Tampilan Halaman *Bookmark*

h. Tampilan Fitur Reset

Merupakan fitur yang berfungsi untuk mereset data yang ada di *Monitoring*.



Gambar 4. 27 *Tampilan Fitur Reset*

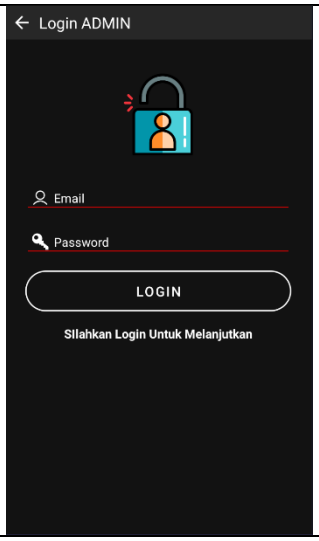
D. Pengujian Sistem

Pengujian Sistem pada penelitian ini menggunakan metode *Black Box Testing* dan juga *white box testing*. Berikut dibawah ini hasil pengujian menggunakan metode tersebut.

1. *Black Box Testing*

   a. *Black Box Testing* Kesalahan *Email* dan *Password*

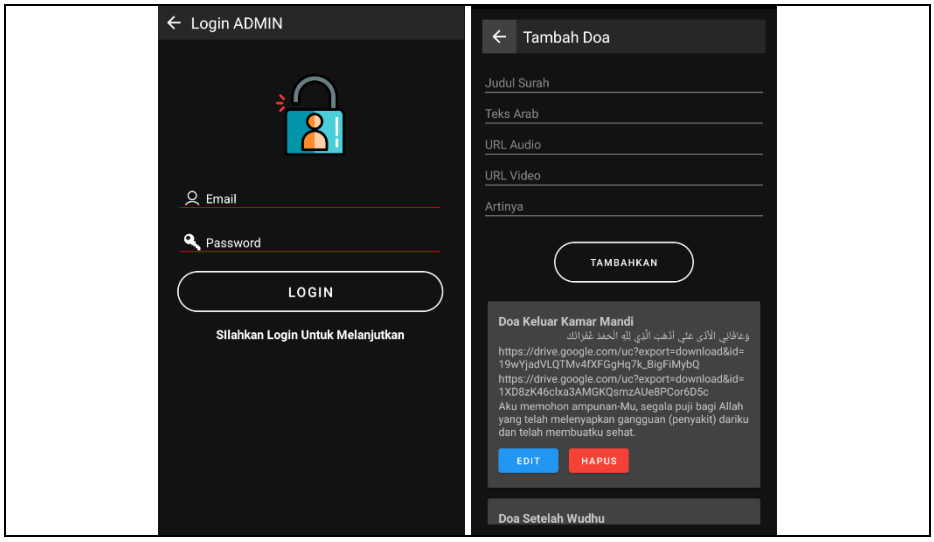
Tabel 4. 8 *Black Box Testing* Kesalahan *Email* dan *Password*

Tes Faktor	Hasil	Keterangan
Memasukkan *email* atau *password* yang tidak sesuai	✓	Berhasil, ketika *email* atau *password* tidak sesuai maka tampil pesan *Login Failed!!!*
Screenshot		
		

   b. *Black Box Testing Login* Berhasil

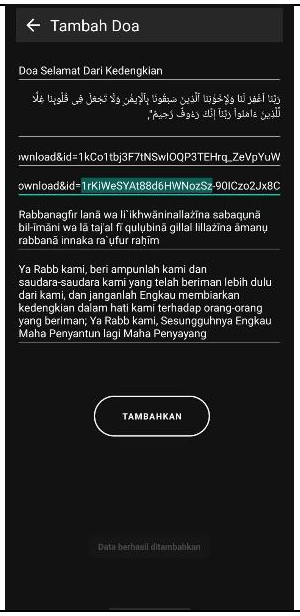
Tabel 4. 9 *Black Box Testing Login* Berhasil

Tes Faktor	Hasil	Keterangan
Memasukkan *email* atau *password* yang sesuai	✓	Berhasil, sistem dapat menampilkan halaman tambah data
Screenshot		



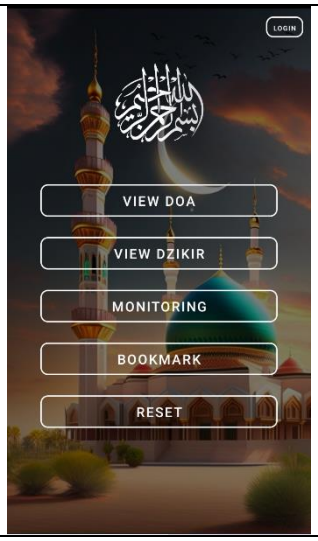
c. *Black Box Testing* Tambah Data

Tabel 4. 10 *Black Box Testing* Tambah Data

Tes Faktor	Hasil	Keterangan
Admin mengisi *form* tambah dan menekan tombol simpan	✓	Berhasil, tampil informasi bahwa data berhasil ditambahkan
Screenshot		
		

e. *Black Box Testing* Tampilan Halaman Utama

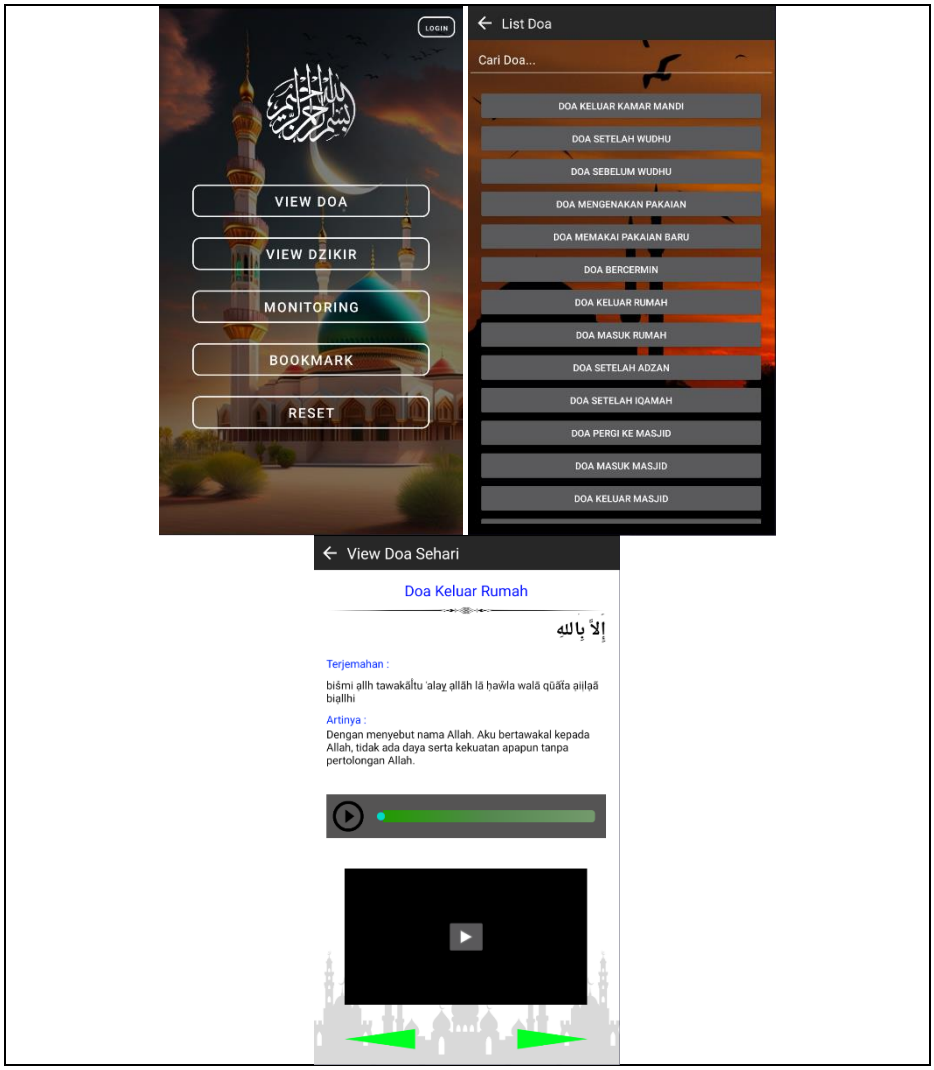
Tabel 4. 11 *Black Box Testing* Tampilan Halaman Utama

Tes Faktor	Hasil	Keterangan
User Pertama kali mengakses Aplikasi bacaan doa dan dzikir harian	✓	Berhasil, Tampil halaman utama
Screenshot		
		

f. *Black Box Testing* Tampilan Halaman *View* Doa

Tabel 4. 12 *Black Box Testing* Tampilan Halaman *View* Doa

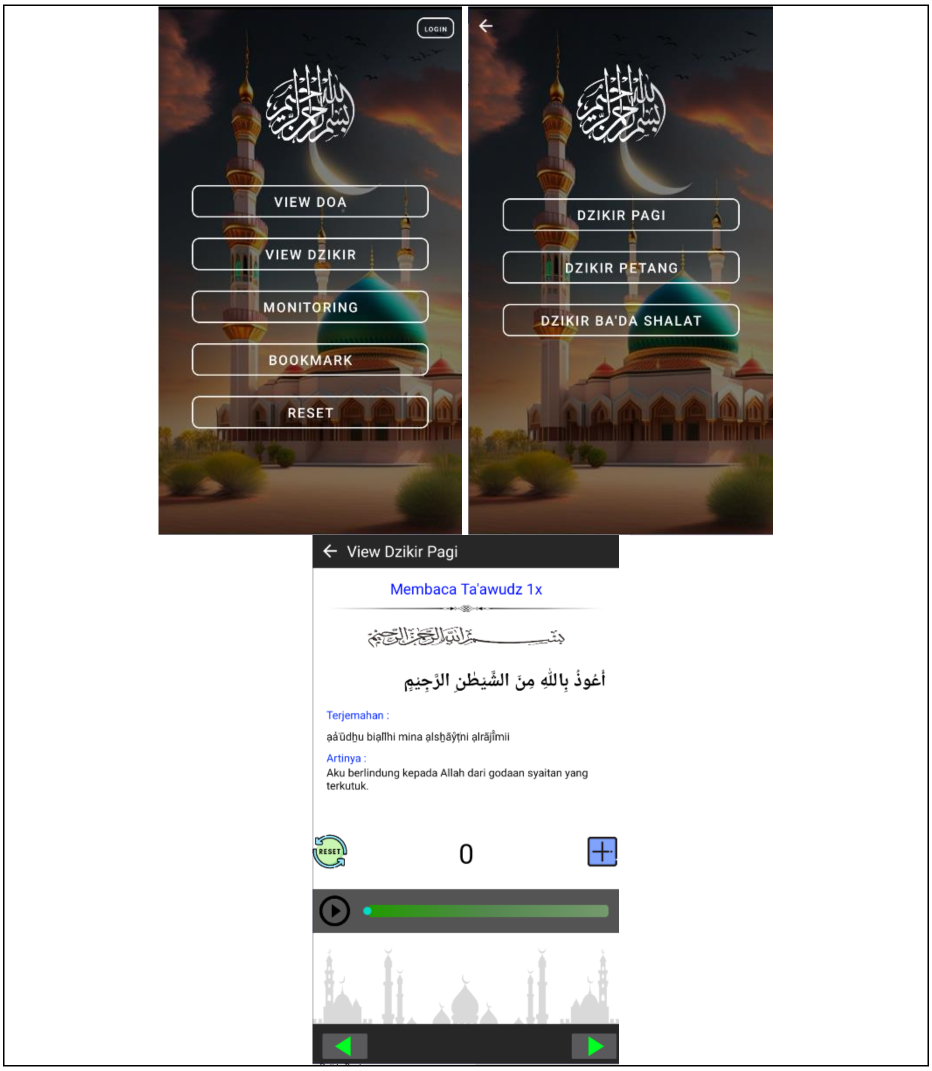
Tes Faktor	Hasil	Keterangan
User Menekan tombol menu *View* doa dan memilih doa yang *user* inginkan	✓	Berhasil, Tampil halaman daftar doa-doa dan tampil doa
Screenshot		



g. *Black Box Testing* Halaman *View* Dzikir

Tabel 4. 13 *Black Box Testing* Halaman *View* Dzikir

Tes Faktor	Hasil	Keterangan
User Menekan tombol menu *View* dzikir dan memilih dzikir yang *user* inginkan	✓	Berhasil, Tampil halaman daftar dzikir dan tampil dzikir
Screenshot		


LOGIN
VIEW DOA
VIEW DZIKIR
MONITORING
BOOKMARK
RESET
DZIKIR PAGI
DZIKIR PETANG
DZIKIR BA'DA SHALAT
View Dzikir Pagi
Membaca Ta'awudz 1x
Terjemahan :
Artinya :
Aku berlindung kepada Allah dari godaan syaitan yang terkutuk.
0

h. *Black Box Testing* Tampilan *Monitoring*

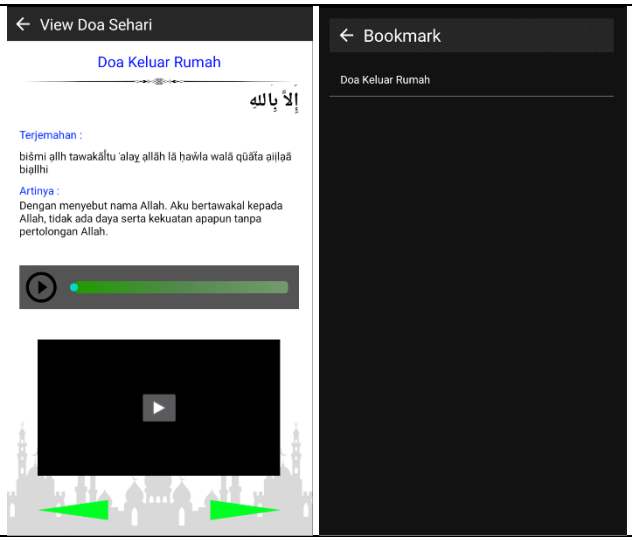
Tabel 4. 14 *Black Box Testing* Tampilan *Monitoring*

Tes Faktor	Hasil	Keterangan
User Menekan tombol *button next*, lalu sistem akan menampilkan di *Monitoring*	✓	Berhasil, tampil data di *Monitoring*
Screenshot		

i. *Black Box Testing* Tampilan *Bookmark*

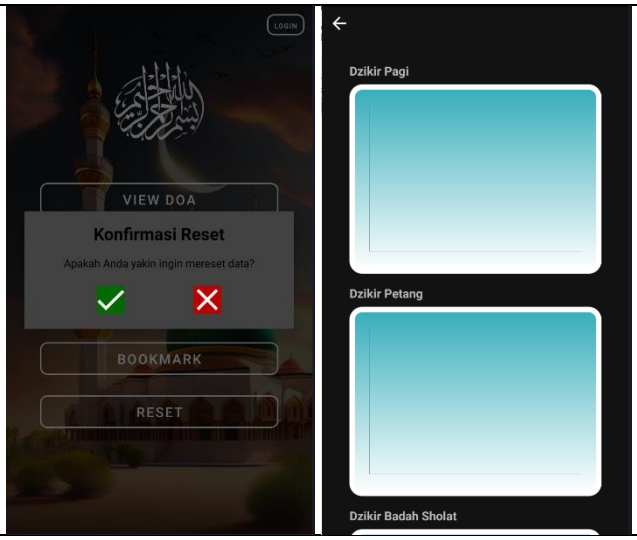
Tabel 4. 15 *Black Box Testing* Tampilan *Bookmark*

Tes Faktor	Hasil	Keterangan

User membaca doa dan sistem menampilkan bacaan terakhir di *Bookmark*	✓	Berhasil, tampil doa di *Bookmark*
Screenshot		
		

j. *Black Box Testing* Fitur Reset

Tabel 4. 16 *Black Box Testing* Fitur Reset

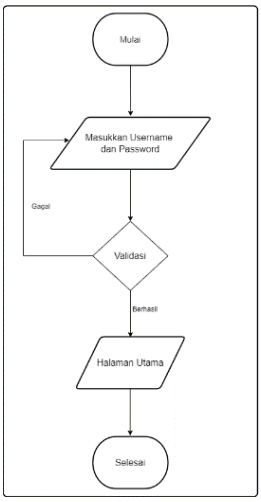
Tes Faktor	Hasil	Keterangan
User menekan tombol reset dan menekan tombol ceklis (ya)	✓	Berhasil, mereset data *Monitoring* doa
Screenshot		
		

2. White Box Testing

Pada metode pengujian ini akan ditampilkan *flowchart* dan *flowgraph* dari sistem yang telah dibuat. Berikut dibawah ini merupakan hasil pengujian menggunakan metode white box testing.

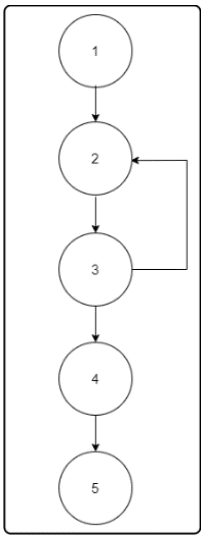
a. *White Box Testing* Kesalahan Email dan Password

1) *Flowchart*



Gambar 4. 28 *Flowchart Kesalahan Username dan Password*

2) *Flowgraph*



Gambar 4. 29 *Flowgraph Kesalahan Username dan Password*

Berdasarkan gambar 4. 29 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks kesalahan email dan password

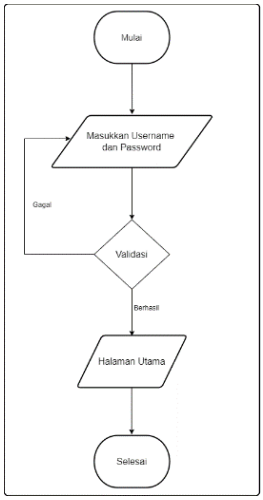
Tabel 4. 17 Grafik Matriks Kesalahan Username dan Password

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1

4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

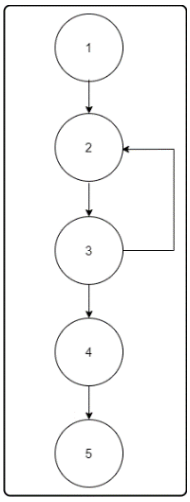
b. *White Box Testing Login* Berhasil

1) *Flowchart*



Gambar 4. 30 *Flowchart Login* Berhasil

2) *Flowgraph*



Gambar 4. 31 *Flowgraph Login* Berhasil

Berdasarkan gambar 4. 32 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks *Login* berhasil

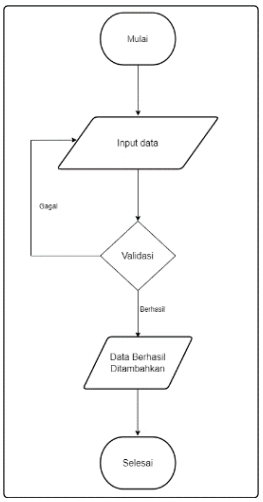
Tabel 4. 18 Grafik Matriks *Login* Berhasil

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0

5						0
	SUM (E + 1)					1 + 1 = 2

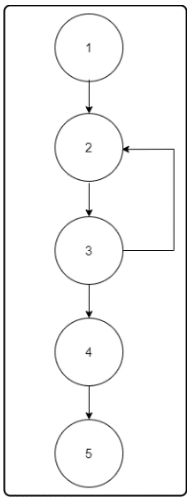
c. *White Box Testing* Tambah Data

1) *Flowchart*



Gambar 4. 32 *Flowchart* Tambah Data

2) *Flowgraph*



Gambar 4. 33 *Flowgraph* Tambah Data

Berdasarkan gambar 4. 34 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks Tambah Data

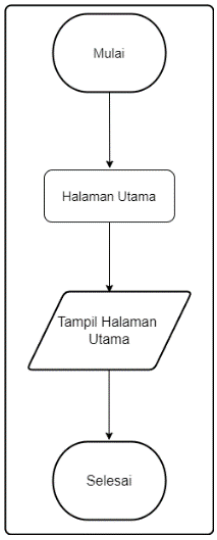
Tabel 4. 19 Grafik Matriks Tambah Data

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0

	SUM (E + 1)	1 + 1 = 2

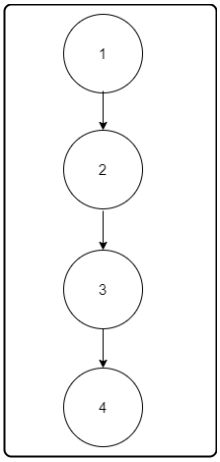
d. *White Box Testing* Halaman Utama

1) *Flowchart*



Gambar 4. 34 *Flowchart* Halaman Utama

2) *Flowgraph*



Gambar 4. 35 *Flowgraph* Halaman Utama

Berdasarkan gambar 4. 36 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 3

N (*node*) = 4

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 3 – 4 + 2

= 2

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

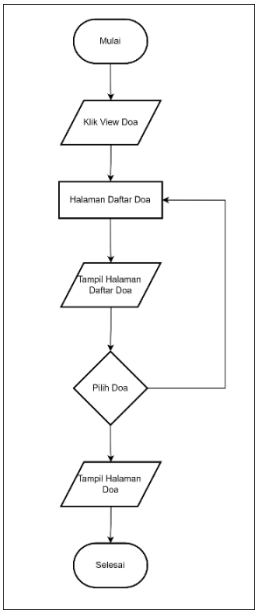
(4) Grafik matriks Halaman Utama

Tabel 4. 20 Grafik Matriks Halaman Utama

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

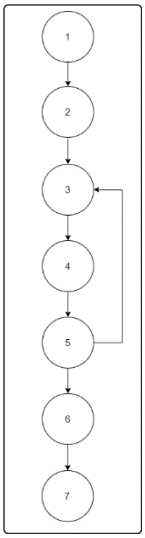
e. White Boc Testing *View* Doa

1) *Flowchart*



Gambar 4. 36 *Flowchart* Halaman *View* Doa

2) *Flowgraph*



Gambar 4. 37 *Flowgraph* Halaman *View* Doa

Berdasarkan gambar 4. 38 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5 – 3

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

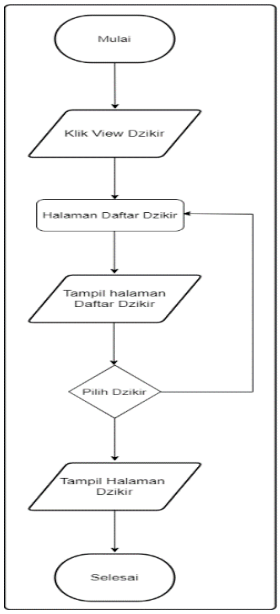
(4) Grafik matriks *View* Doa

Tabel 4. 21 Grafik Matriks *View* Doa

	1	2	3	4	5	6	7	E – 1
1		1						1 – 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0
4			1		1			2 – 1 = 1
5						1		1 – 1 = 0
6							1	1 – 1 = 0
7								0
		SUM (E + 1)						1 + 1 = 2

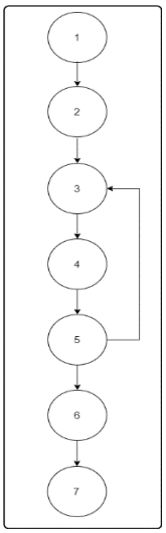
f. *White Box Testing View* Dzikir

1) *Flowchart*



Gambar 4. 38 *Flowchart View* Dzikir

2) *Flowgraph*



Gambar 4. 39 *Flowgraph View* Dzikir

Berdasarkan gambar 4. 40 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5 – 3

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

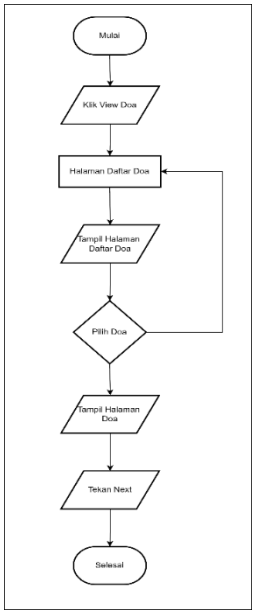
(4) Grafik matriks *View* Dzikir

Tabel 4. 22 Grafik Matriks *View* Dzikir

	1	2	3	4	5	6	7	E – 1
1		1						1 – 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0
4			1		1			2 – 1 = 1
5						1		1 – 1 = 0
6							1	1 – 1 = 0
7								0
		SUM (E + 1)						1 + 1 = 2

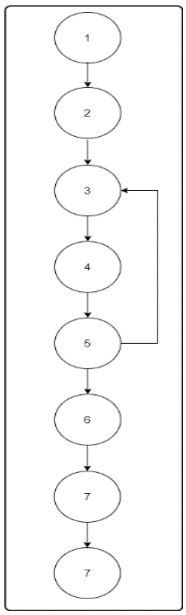
g. *White Box Testing Monitoring*

1) *Flowchart*



Gambar 4. 40 *Flowchart Monitoring*

2) *Flowgraph*



Gambar 4. 41 *Flowgraph Monitoring*

Berdasarkan gambar 4. 42 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 8

N (*node*) = 8

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5 – 3

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

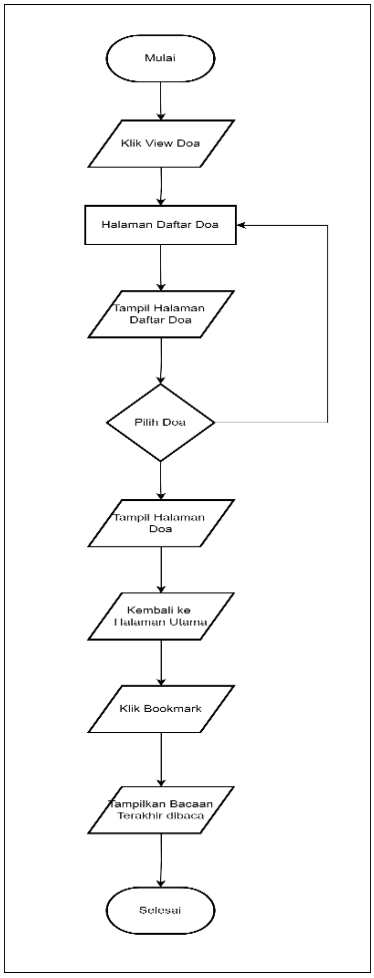
(4) Grafik matriks *Monitoring*

Tabel 4. 23 Grafik Matriks *Monitoring*

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

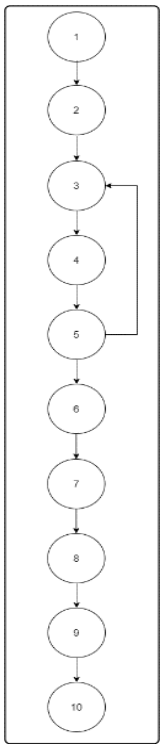
h. *White Box Testing* Fitur *Bookmark*

1) *Flowchart*



Gambar 4. 42 *Flowchart* Fitur *Bookmark*

2) *Flowgraph*



Gambar 4. 43 *Flowgraph* Fitur *Bookmark*

Berdasarkan gambar 4. 42 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 10

N (*node*) = 10

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 10 – 10 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5 – 3

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

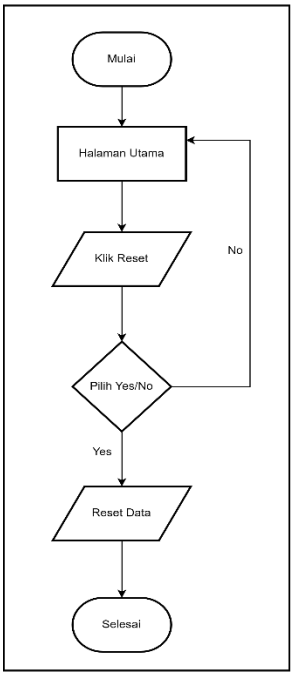
(4) Grafik matriks Fitur *Bookmark*

Tabel 4. 24 Grafik Matriks Fitur *Bookmark*

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

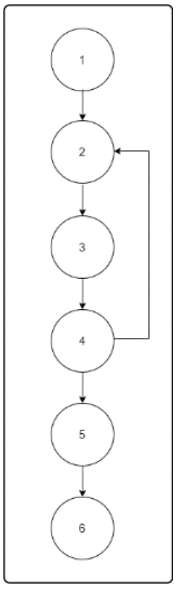
i. *White Box Testing* Fitur Reset

1) *Flowchart*



Gambar 4. 44 *Flowchart* Fitur *Reset*

2) *Flowgraph*



Gambar 4. 45 *Flowgraph* Fitur *Reset*

Berdasarkan gambar 4. 46 diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 6

N (*node*) = 6

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 2

Path 2 = 1 – 2 – 3 – 4 – 5 – 6

(4) Grafik matriks Fitur Reset

Tabel 4. 25 Grafik matriks Fitur Reset

	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							1 - 1 = 0
7							1 - 1 = 0
8							0
		SUM (E + 1)					1 + 1 = 2

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut. Aplikasi *mobile learning* bacaan doa dan dzikir harian berbasis android dibangun menggunakan bahasa pemrograman *JAVA* dengan *Android Studio* sebagai *text editor* serta *Firebase* sebagai *database*nya yang dapat memudahkan pengguna dalam mencari bacaan doa dan dzikir harian kapanpun dan dimanapun.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat di perlu perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Mengembangkan aplikasi ini terutama pada *database android* sehingga data dapat diakses secara *offline*.
2. Aplikasi ini dapat dikembangkan kembali dengan desain *interface* yang lebih menarik.

DAFTAR PUSTAKA

Aulia, S. C. I. (2022). *Pemanfaatan Uml (Unified Modeling Language) Dalam Perancangan Sistem Informasi Rekam Medis Sederhana Pada Kegiatan Posbindu PTM. Jurnal Ilmiah Sains dan Teknologi*, *6*(1), 38-44.

Edy, E., Adhinugraha, D., & Priko, L. (2022). *Implementasi Web Service pada Aplikasi Pemesanan Makanan Berbasis Android. ALGOR*, *3*(2), 60-66.

Fajri, A. A. (2020). *Pengembangan Aplikasi Mobile Guru Tajwid Berbasis Android.* Jurnal MediaTIK, 3(3), 37.

Fatihuddin. (2010). *Tentramkan Hati Dengan Dzikir.* Ciamis: Delta Prima Press.

Hernawan, H. (2017). *Penggunaan Aplikasi Mobile Learning Berbasis Html 5 Untuk Meningkatkan Pemahaman Mahasiswa pada Mata Kuliah Mikrobiologi*. PEDAGOGIA, 15(2), 124-131.

Matlubah, H., Anekawati, A., & Ngadi, N. (2016). *Aplikasi Mobile Learning Berbasis Smartphone Android Sebagai Sumber Belajar Mahasiswa Program Studi Pendidikan Ipa Universitas Wiraraja Sumene*p. LENSA (Lentera Sains): Jurnal Pendidikan IPA, 6(2).

Maulana, IF (2020). Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* , *4* (5), 854-863.

Rahman, A. (2011). Sistem pemroses lembar jawab komputer berbasis *XML*. *Jurnal Sistem Informasi Indonesia*, *1*(1).

Rahman, F. (2022). *Himpunan Doa & Dzikir Pilihan Rasulullah.* Surabaya: CV. Pustaka anyMedia.

Saputra, A. W., Kusuma, W. A., & Suharso, W. (2020). *Rancang Bangun Aplikasi Pemesanan Molly Molen Malang Berbasis Android Menggunakan Metode Waterfall.* Jurnal Repositor, 2(7), 855-862.

Setiawan, Y. P. (2020). *Rancang Bangun Aplikasi Media Pembelajaran Bahasa Korea Sederhana Menggunakan Android.* (Doctoral dissertation, Institut Teknologi Nasional).

Taufiqurrahman, M., Yahya, F., & Ratu, T. (2018). *Pengembangan Mobile Learning (M-Learning) Berbasis Android pada Materi Gerak Melingkar Kelas X SMAN 3 Sumbawa Besar.*

Wahyuni, E. S. (2021). *Analisis Cara Kerja CRUD Dengan Menggunakan Android Studio.* Lampung: Universitas Bandar Lampung.

Yunus, R. M., & Sujadi, H. (2015*). Sistem Keamanan Pesan Dengan Algoritma Rivest Code 6 (RC-6) Menggunakan Java Pada Smartphone Berbasis Android.* J-ENSITEC, 2(01)