

**UNIVERSITAS ESA UNGGUL**

**RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK MANAJEMEN INFORMASI DI MASJID**

**JAMI’ AL MU’MININ**

**PROPOSAL TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana**

**NAMA : Junianto Tri Setiawan**

**NIM : 20160801250**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS ESA UNGGUL**

**TAHUN 20****21**

# HALAMAN PERNYATAAN KEASLIAN

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Junianto Tri Setiawan

NIM : 20160801250

Tanda Tangan :

|  |
| --- |
|  |

Tanggal : 6 Agustus 2021

# HALAMAN PENGESAHAN TUGAS AKHIR

Tugas Akhir ini diajukan oleh

Nama : Junianto Tri Setiawan

NIM : 20160801250

Program Studi : Teknik Informatika

Fakultas Ilmu Komputer - Universitas Esa Unggul

Judul Tugas Akhir : Rancang Bangun Aplikasi Berbasis Android untuk

Manajemen Informasi di Masjid Jami’ Al Mu’minin.

**Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Ilmu Komputer, Universitas Esa Unggul.**

**TIM PENGUJI**

Pembimbing : Diah Aryani, S.Kom, M.Kom (...tanda tangan...)

Penguji I : Ir. Kundang Karsono, MMSI (...tanda tangan...)

Penguji II : Popong Setiawati, S.Kom, MMSI (...tanda tangan...)

Ditetapkan di : Jakarta

Ketua Program Studi : M.Bahrul Ulum, S.Kom., M.Kom. (...tanda tangan..)

Tanggal : 6 Agustus 2021

# HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Esa Unggul, saya yang bertanda tangan di bawah ini:

Nama : Junianto Tri Setiawan

NIM : 20160801250

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Jenis Karya Ilmiah : Tugas Akhir

Demi pengembangan ilmu pengetahuan, teknologi, dan seni, menyetujui untuk memberikan kepada Universitas Esa Unggul Hak Bebas Royalti Noneksklusif atas karya ilmiah saya yang berjudul:

**Rancang Bangun Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin**

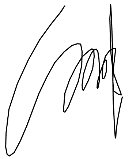
beserta perangkat yang ada (apabila diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Universitas Esa Unggul berhak menyimpan, mengalihmediakan, mengelola dalam bentuk pangkalan data, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada tanggal : 6 Agustus 2021

Yang menyatakan



JuniantoTri Setiawan  
(20160801250)

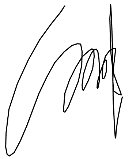
# KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah Swt. yang telah melimpahkan rahmat dan hidayah-Nya karena dengan usaha dan doa, penyusunan Laporan Tugas Akhir yang berjudul **“Rancang Bangun Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin”** dapat diselesaikan dengan baik.

Dalam kesempatan ini penulis juga mengucapkan terima kasih kepada berbagai pihak yang telah memberikan dukungan, bimbingan, dan kerja samanya dalam menyusun Laporan Tugas Akhir ini, karena tanpa dukungan mereka penyusunan Laporan Tugas Akhir ini tidak akan tercapai. Karena itu penulis ingin mengucapkan terima kasih kepada :

1. Kepada Kedua Orang Tua, kakak serta keluarga besar yang senantiasa selalu memberikan motivasi, dukungan, dan doa kepada penulis sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Bapak Habibullah Akbar, S.Si, M.Sc, Ph.D. selaku Dekan Fakultas Ilmu Komputer, Universitas Esa Unggul Jakarta.
3. Bapak M.Bahrul Ulum, S.Kom, M.Kom selaku Ka. Prodi Teknik Informatika Fakultas Ilmu Komputer, Universitas Esa Unggul Jakarta.
4. Ibu Diah Aryani, S.Kom, M.Kom sebagai Dosen Pembimbing Materi yang telah meluangkan banyak waktunya untuk memberikan bimbingan dan arahan kepada penulis dalam menyelesaikan Laporan Tugas Akhir.
5. Bapak Dr. Fransiskus Adikara, M.Kom, M.M sebagai Dosen Pembimbing Akademik yang telah memberikan bimbingan dan arahan selama penulis menyelesaikan Laporan Tugas Akhir.
6. Seluruh Dosen dan Staf Fakultas Ilmu Komputer Universitas Esa Unggul, yang tidak bisa disebutkan satu persatu namanya.
7. Kepada Bapak H. Karjono Sukatma,S.Ag.,M.Pd. selaku Ketua Dewan Kemakmuran Masjid (DKM) Masjid Jami’ Al Mu’minin yang telah memberikan berbagai informasi yang dibutuhkan untuk membangun aplikasi berbasis android untuk manajemen informasi di Masjid Jami’Al Mu’minin.
8. Kepada Bapak Sugono. selaku Bendahara Dewan Kemakmuran Masjid (DKM) Masjid Jami’ Al Mu’minin yang telah memberikan berbagai informasi yang dibutuhkan untuk membangun aplikasi berbasis android untuk manajemen informasi di Masjid Jami’Al Mu’minin.
9. Kepada Bapak Jarwanto selaku Sekretaris Dewan Kemakmuran Masjid (DKM) Masjid Jami’ Al Mu’minin yang telah memberikan berbagai informasi yang dibutuhkan untuk membangun aplikasi berbasis android untuk manajemen informasi di Masjid Jami’Al Mu’minin.
10. Kepada Bapak Karyo selaku Pembimbing lapangan saat melakukan penelitian di Masjid Jami’ Al Mu’minin yang telah memberikan berbagai informasi yang dibutuhkan untuk membangun aplikasi berbasis android untuk manajemen informasi di Masjid Jami’Al Mu’minin.
11. Kepada teman-teman Keluarga Besar Fakultas Ilmu Komputer seluruh angkatan, khususnya Angkatan 2016. Terima kasih atas dukungan dan bantuannya.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih banyak kelemahan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun akan diterima dengan senang hati. Penulis berharap Laporan Tugas Akhir ini dapat memberikan manfaaat, wawasan dan pengetahuan bagi pembaca.

Jakarta, 6 Agustus 2021

Junianto Tri Setiawan

(20160801250)

# ABSTRAK

|  |  |
| --- | --- |
| Judul : | **RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK MANAJEMEN INFORMASI DI MASJID JAMI’ AL MU’MININ** |
| Nama : | Junianto Tri Setiawan |
| Program Studi : | Teknik Informatika |

Masjid Jami’ Al Mu’minin merupakan tempat kegiatan keagamaan bagi warga Kelurahan Rawabuaya, Kecamatan Cengkareng, Jakarta Barat. Pengelolaan informasi di masjid masih menggunakan metode manual, yaitu dengan menempelkan informasi pada papan pengumuman dan melalui pengeras suara. Hal tersebut menjadi kendala bagi masyarakat dalam mencari informasi seputar masjid. Maka dari itu peneliti membangun *Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin* bertujuan untuk meningkatkan pengelolaan informasi bagi pengurus masjid dan membantu masyarakat dalam mencari informasi seputar masjid. Metode pengembangan yang digunakan adalah metode prototype yang memiliki beberapa tahapan yaitu pengumpulan kebutuhan, membangun *prototype*, evaluasi *prototype*, pengkodean sistem, pengujian sistem, dan evaluasi sistem. Metode pengumpulan data menggunakan studi literatur, observasi, dan wawancara. Hasil dari penelitian ini yaitu informasi terkait pengelolaan informasi kegiatan dan keuangan Masjid Jami’ Al Mu’minin. Dengan terciptanya aplikasi ini akan menjadi solusi dari kendala-kendala yang dihadapi masyarakat dalam mencari informasi serta meningkatkan transparasi keuangan masjid.

Kata kunci : Masjid, Manajemen informasi, Android, Metode *prototype*.

**ABSTRACT**

|  |  |
| --- | --- |
| *Title :* | ***DESIGN AND BUILD AN ANDROID-BASED APPLICATION FOR INFORMATION MANAGEMENT AT THE JAMI’ AL MU’MININ MOSQUE*** |
| *Name :* | Junianto Tri Setiawan |
| *Study Program :* | *Computer Science* |

*Jami' Al Mu'minin Mosque is a place for religious activities for residents of Rawabuaya Village, Cengkareng District, West Jakarta. Information management in mosques still uses the manual method, namely by pasting information on notice boards and through loudspeakers. This becomes an obstacle for the community in finding information about mosques. Therefore, researchers built an Android-Based Application for Information Management at the Jami' Al Mu'minin Mosque aimed at improving information management for mosque administrators and helping the public in finding information about mosques. The development method used is the prototype method which has several stages, namely gathering requirements, building prototypes, evaluating prototypes, coding systems, testing systems, and evaluating systems. Methods of data collection using literature study, observation, and interviews. The results of this study are information related to the management of information on activities and finances of the Jami' Al Mu'minin Mosque. With the creation of this application, it will be a solution to the constraints faced by the community in finding information and increasing mosque financial transparency.*

*Keywords : Mosque, Information management, Android, Prototype method.*

# DAFTAR ISI

[HALAMAN PERNYATAAN KEASLIAN ii](#_Toc80454256)

[HALAMAN PENGESAHAN TUGAS AKHIR iii](#_Toc80454257)

[HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS iv](#_Toc80454258)

[KATA PENGANTAR v](#_Toc80454259)

[ABSTRAK vii](#_Toc80454260)

[DAFTAR ISI ix](#_Toc80454261)

[DAFTAR GAMBAR xii](#_Toc80454262)

[DAFTAR TABEL xiv](#_Toc80454263)

[BAB 1 PENDAHULUAN 1](#_Toc80454264)

[1.1 Latar Belakang 1](#_Toc80454265)

[1.2 Identifikasi Masalah 4](#_Toc80454266)

[1.3 Tujuan Penelitian 4](#_Toc80454267)

[1.4 Manfaat Penelitian 4](#_Toc80454268)

[1.5 Batasan Masalah 5](#_Toc80454269)

[1.6 Kerangka Berpikir 5](#_Toc80454270)

[1.7 Sistematika Penulisan 7](#_Toc80454271)

[BAB 2 LANDASAN TEORI 9](#_Toc80454272)

[*2.1* *Smartphone* 9](#_Toc80454273)

[2.2 Android 9](#_Toc80454274)

[2.2.1 Fitur Android 9](#_Toc80454275)

[2.2.2 Arsitektur Android 10](#_Toc80454276)

[2.2.3 Versi Android 11](#_Toc80454277)

[2.3 Masjid 12](#_Toc80454278)

[*2.4* *User Interface* 12](#_Toc80454279)

[2.4.1 Delapan Aturan Emas Perancangan *User Interface* 13](#_Toc80454280)

[*2.5* *User Experience* 14](#_Toc80454281)

[2.6 Java 14](#_Toc80454282)

[2.7 *Application Programming Interface* (API) 15](#_Toc80454283)

[*2.8* *Database* 15](#_Toc80454284)

[2.9 NoSQL 15](#_Toc80454285)

[2.10 Firebase 16](#_Toc80454286)

[2.10.1 Sejarah Firebase 16](#_Toc80454287)

[2.10.2 Fitur Firebase 16](#_Toc80454288)

[2.11 *Java Script Object Notation* (JSON) 18](#_Toc80454289)

[2.12 *Extensible Markup Language* (XML) 18](#_Toc80454290)

[2.13 *Unified Modeling Language* (UML) 18](#_Toc80454291)

[a. *Use Case* Diagram 18](#_Toc80454292)

[b. *Activity* Diagram 20](#_Toc80454293)

[c. *Sequence* Diagram 21](#_Toc80454294)

[d. *Class Diagram* 22](#_Toc80454295)

[2.14 Metode *Prototyping* 23](#_Toc80454296)

[2.14.1 Tahapan *Prototyping* 23](#_Toc80454297)

[BAB 3 METODE PENELITIAN 25](#_Toc80454298)

[3.1 Rencana Penelitian 25](#_Toc80454299)

[3.1.1 Tahapan Penelitian 25](#_Toc80454300)

[3.2 Alur Proses Sistem Yang Akan Diusulkan 27](#_Toc80454301)

[3.3 Waktu dan Populasi Penelitian 27](#_Toc80454302)

[3.4 Struktur Organisisai 28](#_Toc80454303)

[3.5 Metode Analisis *Pieces* 28](#_Toc80454304)

[3.6 Objek Penelitian 29](#_Toc80454305)

[3.7 Metode Pengumpulan Data 29](#_Toc80454306)

[BAB 4 HASIL DAN PEMBAHASAN 32](#_Toc80454307)

[4.1 *Requirement* Analisis 32](#_Toc80454308)

[4.1.1 Kebutuhan Fungsional 32](#_Toc80454309)

[4.1.2 Kebutuhan Non Fungsional 33](#_Toc80454310)

[4.1.3 Analisis Perangkat Keras untuk Pembuatan Aplikasi 34](#_Toc80454311)

[4.1.4 Analisis Perangkat Lunak untuk Pembuatan Aplikasi 34](#_Toc80454312)

[4.1.5 Spesifikasi Minimum Perangkat untuk Menjalankan Aplikasi 35](#_Toc80454313)

[4.2 Hasil Analisis Menggunakan Metode PIECES 35](#_Toc80454314)

[4.3 Data Hasil Penelitian 36](#_Toc80454315)

[4.4 Pemodelan Aplikasi 36](#_Toc80454316)

[4.4.1 Rancangan *Use Case* Diagram 37](#_Toc80454317)

[4.4.2 Rancangan *Activity* Diagram 38](#_Toc80454318)

[4.4.3 Rancangan Sequence Diagram 46](#_Toc80454319)

[*4.4.4* Rancangan *Class Diagram* 56](#_Toc80454320)

[4.5 *Mock Up* Desain Aplikasi 57](#_Toc80454321)

[*4.6* Rancangan *Database* 61](#_Toc80454322)

[4.7 Rancangan *Interface* Aplikasi 63](#_Toc80454323)

[4.7.1 Rancangan *Interface Icon* dan *Splash Screen* Aplikasi 63](#_Toc80454324)

[4.7.2 Rancangan *Interface* Beranda dan Isi Berita Aplikasi 63](#_Toc80454325)

[4.7.3 Rancangan *Interface* Jadwal Salat dan Kiblat 64](#_Toc80454326)

[4.7.4 Rancangan *Interface* Halaman Quran 65](#_Toc80454327)

[4.7.5 Rancangan *Interface* Keuangan Masjid 65](#_Toc80454328)

[4.7.6 Rancangan *Interface Login* Aplikasi 66](#_Toc80454329)

[4.7.7 Rancangan *Interface* Menu Admin Aplikasi 67](#_Toc80454330)

[4.7.8 Rancangan *Interface* Kelola Berita Admin 67](#_Toc80454331)

[4.7.9 Rancangan *Interface* Form Tambah Berita dan Edit Berita 68](#_Toc80454332)

[4.7.10 Rancangan *Interface* Kelola Keuangan Admin 69](#_Toc80454333)

[4.7.11 Rancangan *Interface* Ketika Tekan dan Tahan Daftar Keuangan 70](#_Toc80454334)

[4.7.12 Rancangan *Interface Form* Tambah dan *Edit* Pemasukan Keuangan 71](#_Toc80454335)

[4.7.13 Rancangan *Interface Form* Tambah dan *Edit* Pengeluaran Keuangan 72](#_Toc80454336)

[4.8 Uji Coba 73](#_Toc80454337)

[4.8.1 Pengujian *BlackBox* 73](#_Toc80454338)

[4.8.2 Pengujian *User Acceptance Test* (UAT) 78](#_Toc80454339)

[4.9 Perbandingan Pengelolaan Informasi Masjid 81](#_Toc80454340)

[4.10 Ilustrasi Penggunaan Aplikasi Masjid 82](#_Toc80454341)

[4.11 Studi Literatur 82](#_Toc80454342)

[BAB 5 KESIMPULAN DAN SARAN 84](#_Toc80454343)

[5.1 Kesimpulan 84](#_Toc80454344)

[5.2 Saran 84](#_Toc80454345)

[DAFTAR REFERENSI 85](#_Toc80454346)

[Lampiran 1 87](#_Toc80454347)

[Lampiran 2 88](#_Toc80454348)

[Lampiran 3 90](#_Toc80454349)

# DAFTAR GAMBAR

[Gambar 1.1 Kerangka Berpikir 7](#_Toc80454156)

[Gambar 2.1 Cara Kerja API 15](#_Toc80454157)

[Gambar 2.2 Tahapan *prototyping* 23](#_Toc80454158)

[Gambar 3.1 Flowchat Sistem Usulan 27](#_Toc80454159)

[Gambar 3.2 Struktur Organisasi Masjid Jami’ Al Mu’minin 28](#_Toc80454160)

[Gambar 3.3 Masjid Jami’ Al Mu’minin 29](#_Toc80454161)

[Gambar 4.1 *Use Case* Diagram 37](#_Toc80454162)

[Gambar 4.2 *Activity* Diagram 1 39](#_Toc80454163)

[Gambar 4.3 *Activity* Diagram 2 40](#_Toc80454164)

[Gambar 4.4 *Activity* Diagram 3 41](#_Toc80454165)

[Gambar 4.5 *Activity* Diagram 4 42](#_Toc80454166)

[Gambar 4.6 *Activity* Diagram 5 43](#_Toc80454167)

[Gambar 4.7 *Activity* Diagram 6 44](#_Toc80454168)

[Gambar 4.8 *Activity* Diagram 7 45](#_Toc80454169)

[Gambar 4.9 *Sequence* Diagram 1 46](#_Toc80454170)

[Gambar 4.10 *Sequence* Diagram 2 47](#_Toc80454171)

[Gambar 4.11 *Sequence* Diagram 3 49](#_Toc80454172)

[Gambar 4.12 *Sequence* Diagram 4 51](#_Toc80454173)

[Gambar 4.13 *Sequence* Diagram 5 53](#_Toc80454174)

[Gambar 4.14 *Sequence* Diagram 6 53](#_Toc80454175)

[Gambar 4.15 Sequence Diagram 7 54](#_Toc80454176)

[Gambar 4.16 *Sequence* Diagram 8 55](#_Toc80454177)

[Gambar 4.17 *Sequence* Diagram 9 55](#_Toc80454178)

[Gambar 4.18 *Class Diagram* 56](#_Toc80454179)

[Gambar 4.19 *Mock up* 1 57](#_Toc80454180)

[Gambar 4.20 *Mock up* 2 57](#_Toc80454181)

[Gambar 4.21 *Mock up* 3 58](#_Toc80454182)

[Gambar 4.22 *Mock up* 4 58](#_Toc80454183)

[Gambar 4.23 *Mock up* 5 59](#_Toc80454184)

[Gambar 4.24 *Mock up* 6 59](#_Toc80454185)

[Gambar 4.25 *Mock up* 7 60](#_Toc80454186)

[Gambar 4.26 *Mock up* 8 60](#_Toc80454187)

[Gambar 4.27 *Mock up* 9 61](#_Toc80454188)

[Gambar 4.28 *Database* *Firebase* Akun Admin 61](#_Toc80454189)

[Gambar 4.29 *Database* *Firebase* Informasi Kegiatan Masjid 62](#_Toc80454190)

[Gambar 4.30 *Database* *Firebase* Keuangan Masjid 62](#_Toc80454191)

[Gambar 4.31 *Storage Firebase* 62](#_Toc80454192)

[Gambar 4.32 *Interface* *Icon* dan *Splash Screen* Aplikasi 63](#_Toc80454193)

[Gambar 4.33 *Interface* Beranda dan Isi Berita Aplikasi 63](#_Toc80454194)

[Gambar 4.34 *Interface* Jadwal Salat dan Kiblat 64](#_Toc80454195)

[Gambar 4.35 *Interface* Halaman Quran 65](#_Toc80454196)

[Gambar 4.36 Rancangan *Interface* Keuangan Masjid 65](#_Toc80454197)

[Gambar 4.37 Rancangan *Interface Login* Aplikasi 66](#_Toc80454198)

[Gambar 4.38 Rancangan *Interface* Menu Admin Aplikasi 67](#_Toc80454199)

[Gambar 4.39 *Interface* Kelola Berita Admin 67](#_Toc80454200)

[Gambar 4.40 *Interface Form* Tambah Berita dan *Edit* Berita 68](#_Toc80454201)

[Gambar 4.41 *Interface* Kelola Keuangan Admin 69](#_Toc80454202)

[Gambar 4.42 *Interface* Ketika Tekan dan Tahan Daftar Keuangan 70](#_Toc80454203)

[Gambar 4.43 *Interface Form* Tambah dan *Edit* Pemasukan Keuangan 71](#_Toc80454204)

[Gambar 4.44 *Interface* FormTambah dan *Edit* Pengeluaran Keuangan 72](#_Toc80454205)

[Gambar 4.45 Ilustrasi Penggunaan Aplikasi Masjid 82](#_Toc80454206)

# DAFTAR TABEL

[Tabel 2.1 Versi Android 11](#_Toc80454207)

[Tabel 2.2 Simbol *Use Case* Diagram 19](#_Toc80454208)

[Tabel 2.3 Simbol *Activity* Diagram 20](#_Toc80454209)

[Tabel 2.4 Simbol *Sequence Diagram* 21](#_Toc80454210)

[Tabel 2.5 Simbol *Class* Diagram 22](#_Toc80454211)

[Tabel 4.1 Analisis Kebutuhan Fungsional 32](#_Toc80454212)

[Tabel 4.2 Analisis Kebutuhan Non Fungsional 33](#_Toc80454213)

[Tabel 4.3 Spesifikasi Hardware yang digunakan dalam pembuatan aplikasi 34](#_Toc80454214)

[Tabel 4.4 Spesifikasi Software yang digunakan dalam pembuatan aplikasi 34](#_Toc80454215)

[Tabel 4.5 Spseifikasi Minimum Perangkat Untuk Menjalankan Aplikasi 35](#_Toc80454216)

[Tabel 4.6 Metode Analisis *Pieces* 35](#_Toc80454217)

[Tabel 4.7 Penjelasan *Use Case* 38](#_Toc80454218)

[Tabel 4.8 Tabel Uji Coba *Blackbox* terhadap aplikasi 73](#_Toc80454219)

[Tabel 4.9 Bobot Nilai Jawaban 78](#_Toc80454220)

[Tabel 4.10 Pertanyaan dan Data Jawaban 79](#_Toc80454221)

[Tabel 4.11 Pengolahan Data Jawaban 79](#_Toc80454222)

[Tabel 4.12 Pengelolaan Informasi di Masjid Sebelum Terciptanya Aplikasi 81](#_Toc80454223)

[Tabel 4.13 Pengelolaan Informasi di Masjid Setelah Terciptanya Aplikasi 81](#_Toc80454224)

[Tabel 4.14 Studi Literatur 82](#_Toc80454225)

# PENDAHULUAN

## Latar Belakang

Perkembangan teknologi semakin hari semakin berkembang, salah satunya di bidang teknologi informasi yang saat ini perkembangannya begitu pesat seiring dengan kebutuhan manusia yang semakin meningkat. Dengan berkembangnya ilmu pengetahuan serta tuntutan kebutuhan manusia akan informasi yang tepat, cepat, dan akurat mendorong terbentuknya sebuah jaringan komputer yang mampu melayani berbagai kebutuhan tertentu dalam waktu yang bersamaan. Jaringan komputer itu dikenal dengan nama internet.

Internet adalah sebuah jaringan komputer global, yang terdiri dari jutaan komputer yang saling terhubung dengan protokol yang sama untuk berbagi informasi secara bersama. Jadi internet merupakan kumpulan atau penggabungan jaringan komputer lokal atau *Local Area Network* (LAN) menjadi jaringan komputer global *Wide Area Network* (WAN). Jaringan-jaringan tersebut saling berhubungan atau berkomunikasi satu sama lainnya dengan berbasis *Internet Protokol* (IP) dan *Transmision Control Protokol* (TCP) atau *User Datagram Protoko* (UDP), sehingga setiap pengguna pada setiap jaringan dapat mengakses semua layanan yang disediakan oleh setiap jaringan [1].

*Smartphone* adalah telepon genggam yang mempunyai kemampuan tinggi menyerupai komputer yang dibekali dengan berbagai macam jenis sistem operasi antara lain Windows Phone, iOS, Blackberry OS, Bada OS, FireFox OS, Palm OS, Symbian OS, Tizen OS, Ubuntu OS, dan juga Android OS. Belum ada standar pabrik yang menentukan arti *smartphone*. Bagi beberapa orang *smartphone* merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, *smartphone* hanyalah merupakan sebuah telepon yang

menyajikan fitur canggih seperti surat elektronik, internet dan kemampuan membaca buku elektronik (*e-book*). Dengan kata lain, *smartphone* merupakan komputer kecil yang mempunyai kemampuan sebuah telepon. Pertumbuhan permintaan akan alat canggih yang mudah dibawa ke mana-mana membuat kemajuan besar dalam pemroses, pengingatan, layar dan sistem operasi yang diluar dari jalur telepon genggam sejak beberapa tahun ini [2].

Android adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau *smartphone*. Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan *Operating System* (OS) ini diperangkatnya tanpa lisensi alias gratis. Begitupun dengan para pembuat aplikasi, mereka bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan google. Dengan seperti itu android memiliki jutaan support aplikasi gratis/berbayar yang dapat diunduh melalui google play [3].

Di Indonesia Android merupakan platform yang paling laris dan banyak diminati, karena memiliki *user interface* yang mudah dipahami, fitur-fitur canggih dan menarik yang disediakan, serta harga *smatphone* berbasis Android yang bervariasi mulai dari yang dipasarkan dengan harga sangat mahal , sampai dengan harga yang relatif terjangkau bagi masyarakat Indonesia. Hal tersebut membuat perkembangan *smartphone* berbasis android yang begitu pesat berdampak langsung pada dunia informasi dan komunikasi di Indonesia. Keadaan ini semakin mendorong pengembang untuk terus melakukan berbagai macam percobaan dan penelitian untuk pengembangan suatu sistem maupun penemuan cara-cara baru guna memberikan manfaat lebih untuk mempermudah manusia dalam bertukar informasi. Termaksud dalam kebutuhan masyarakat akan informasi tentang kegiatan. Salah satu contohnya adalah kegiatan pada tempat ibadah, yaitu masjid.

Masjid merupakan sebuah tempat untuk beribadah bagi umat muslim, selain digunakan untuk beribadah masjid juga digunakan sebagai tempat untuk kegiatan masyarakat atau warga sekitar dan komunitas muslim, diantaranya kegiatan pengajian, pembinaan, pendidikan, dan perayaan hari besar. Peran masjid dalam kehidupan masyarakat cukup vital dan memerlukan perhatian khusus.

Perkembangan masjid di era modern ini masih banyak yang belum secara maksimal memanfaatkan kemajuan teknologi. Seperti halnya pada Masjid Jami’ Al Mu’minin. Perlunya inovasi yang sesuai dengan perkembangan teknologi yang ada untuk memajukan potensi masjid, yaitu dengan memberikan pembaharuan dalam manajemen masjid secara internal akan membuat masjid menjadi lebih baik. Kebutuhan masyarakat akan informasi terkait masjid cukup tinggi, dan keterbukaan informasi administrasi yang berhubungan dengan informasi kegiatan dan pengelolaan keuangan merupakan suatu hal yang penting bagi lembaga masyarakat dan keagamaan seperti masjid.

Masjid Jami’ Al Mu’minin merupakan sebuah masjid yang berada di Jalan Cempaka raya, Rt.03/11 Kelurahan Rawabuaya, Kecamatan Cengkareng, Jakarta Barat. Masjid ini merupakan pusat kegiatan keagamaan dan kegiatan sosial warga sekitar. Masjid Jami’ Al Mu’minin saat ini sedang dalam proses peningkatan infrastruktur masjid dengan terus melakukan pembangunan secara berkala pada bangunan masjid. Pendanaan untuk kebutuhan infrastruktur masjid dan kegiatan masjid lainya diperoleh dari kotak amal dan dari jimpitan warga sekitar, dimana seorang pengurus masjid berkeliling ke rumah-rumah warga untuk mengambil uang jimpitan setiap harinya.

Dalam mengelola dan membagikan data informasi kegiatan di masjid, pengurus masjid masih menggunakan cara konvensional dimana dengan cara mencatatnya di kertas lalu ditempelkan pada papan pengumuman atau diinformasikan ketika kegiatan sebelumnya. Begitu juga untuk pengelolaan terkait keuangan masjid, yang dicatat dengan buku dan diinformasikan saat sela-sela pertengahan sholat jum’at. Dengan kondisi seperti ini, sebagian masyarakat kerap tidak mendapatkan informasi dengan jelas terkait informasi seputar masjid, karena informasinya hanya diinformasikan melalui pengeras suara dirasa kurang efektif.

Berdasarkan uraian tersebut, untuk mempermudah dalam memanajemen informasi kegiatan dan keuangan di Masjid Jami’ Al Mu’minin maka diperlukan sebuah aplikasi untuk memanajemen informasi masjid berbasis android. Aplikasi ini akan membantu pengurus masjid dalam menyampaikan informasi seputar masjid kepada masyarakat agar lebih efektif dan mudah diterima oleh masyarakat. Selain itu aplikasi ini juga akan menampilkan detail keuangan masjid yang dapat meningkatan transparasi keuangan masjid. Oleh karena itu, dalam penelitian ini peneliti mengangkat judul *Rancang Bangun* *Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin.*

## Identifikasi Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas, didapatkan identifikasi masalah sebagai berikut.

1. Bagaimana pengelolaan manajemen informasi di Masjid Jami’ Al Mu’minin?
2. Bagaimana perancangan aplikasi berbasis android untuk manajemen informasi di Masjid Jami’ Al Mu’minin?

## Tujuan Penelitian

Tujuan yang ingin dicapai dalam dalam penelitian ini yaitu untuk membangun aplikasi manajemen informasi di Masjid Jami’ Al Mu’minin agar dapat meningkatkan pengelolaan informasi bagi pengurus masjid dan membantu masyarakat dalam mencari informasi seputar masjid.

## Manfaat Penelitian

Hasil penelitian diharapkan dapat memberikan manfaat kepada berbagai pihak diantaranya sebagai berikut.

1. Bagi peneliti

Peneliti dapat menerapkan ilmu pengetahuan dan ketrampilan tentang teknik informatika yang telah diperoleh selama menjadi mahasiswa Universitas Esa Unggul, terutama dalam perancangan aplikasi berbasis android.

1. Bagi Pengguna

Manfaat perancangan aplikasi oleh peneliti untuk mempermudah pengelola masjid dalam membuat dan menyampaikan informasi kegiatan, serta keuangan di Masjid Jami’ Al Mu’minin secara cepat dan akurat, serta membantu masyarakat dalam mendapatkan informasi seputar Masjid Jami’ Al Mu’minin agar lebih cepat dan akurat.

1. Bagi Peneliti Lain

Hasil penelitian ini dapat menjadi bahan rujukan alternatif bagi peneliti yang akan mengangkat topik serupa pada penelitiannya.

## Batasan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah, peneliti perlu membatasi masalah-masalah yang akan dikaji dalam penelitian ini. Pembatasan masalah bertujuan agar peneliti fokus dalam melakukan penelitian. Adapun batasan masalah dalam peneitian ini yaitu sebagai berikut.

1. Aplikasi ini hanya berjalan pada *mobile* yang berbasis Android.
2. Pembuatan aplikasi menggunakan bahasa pemrograman java dan *extensible markup language* (XML).
3. Aplikasi berfokus pada penyampaian informasi kegiatan dan keuangan di Masjid Jami’ Al Mu’minin.
4. Aplikasi tidak memberikan notifikasi pada pengguna.
5. Pengguna umum hanya dapat melihat isi informasi masjid.
6. Pembuatan *database* menggunakan firebase.
7. Pembuatan akun admin dibuat di firebase.
8. Aplikasi harus terhubung dengan jaringan internet.
9. Aplikasi dibuat menggunakan *software* Android Studio.
10. Aplikasi membutuhkan sensor *gyroscope* untuk menjalankan fitur arah kiblat.

## Kerangka Berpikir

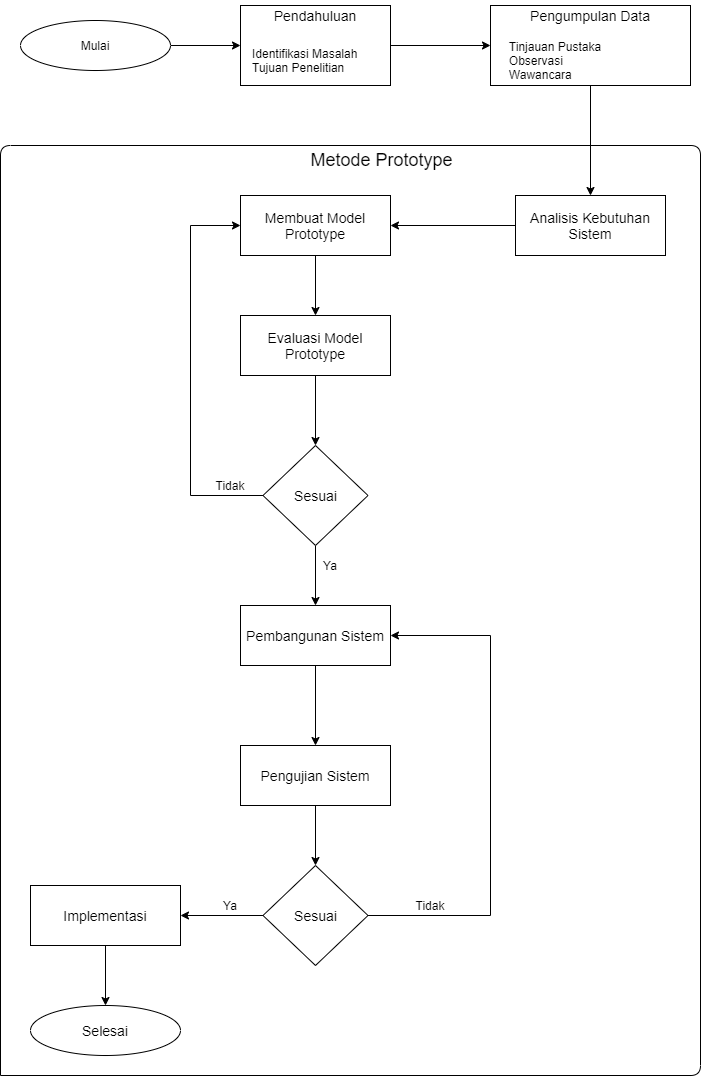
Tahap pertama yang dilakukan oleh peneliti dalam penelitian ini adalah melakukan observasi di Masjid Jami’ Al Mu’minin untuk mengidentifikasi permasalahan terkait pengelolaan manajemen informasi masjid. Setelah itu peneliti menentukan tujuan penelitian, yaitu bertujuan pada Rancang Bangun Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin, yang nantinya dapat membantu pengurus masjid dalam memanajemen informasi kegiatan dan keuangan di masjid, dan juga membantu menginformasikan informasi-informasi yang ada di masjid kepada masyarakat atau jama’ah masjid, agar nantinya informasi dapat diterima masyarakat dengan mudah, cepat, dan akurat.

Tahapan selanjutnya yang dilakukan peneliti yaitu pengumpulan data dengan melakukan peninjauan pustaka dan wawancara. Dalam peninjauan pustaka peneliti mencari literatur sebagai dasar penelitian untuk menyelesaikan masalah secara ilmiah. Tinjauan pustaka menggunakan berbagai jurnal penelitian dan sumber buku yang relevan dan mendukung dalam penelitian ini, sehingga peneliti dapat menggali keterikatan setiap materi pada penelitian yaitu mengenai pengelolaan data informasi dan keuangan di Masjid Jami’ Al Mu’minin, serta cara pembuatan aplikasi berbasis android. Kemudian peneliti mencari data-data yang berkaitan dengan Masjid Jami’ Al Mu’minin. Peneliti menggunakan metode wawancara dalam mengumpulkan data, dengan langsung turun ke lapangan untuk melakukan pengamatan dan tanya jawab secara langsung dengan pengurus masjid dan pihak-pihak lain yang bersangkutan. Data tersebut lalu dianalisis terhadap hasil tinjauan pustaka yang diperoleh sehingga menjadi suatu informasi.

Setelah melakukan pengumpulan data, selanjutnya penelti mendefiniskan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat. Setelah mengumpulkan kebutuhan apa saja yang dibutuhkan, peneliti membangun *prototyping* dengan membuat perancangan sementara dan dilanjutkan dengan evaluasi yang dilakukan oleh pengurus masjid untuk memastian apakah prototyping yang sudah sesuai dengan keinginan pengurus masjid atau belum.

Setelah *prototyping* disepakati oleh pengurus masjid, kemudian peneliti mulai mendesain dan membangun sistem (*coding*), dan setelah pembangunan sistem selesai maka tahap selanjutnya yang dilakukan peneliti yaitu uji coba pada sistem yang telah dibuat untuk mengetahui apakah sistem sudah berjalan sesuai dengan yang diharapkan atau belum sehingga sistem tersebut nantinya bisa digunakan secara optimal dan dapat dimplementasikan.

Agar pembaca lebih mudah memahami kerangka berpikir yang digunakan peneliti dalam penelitian ini, peneliti menyajikannya dalam bentuk *flowchart*. *flowchart* kerangka berpikir dalam penelitian ini digambarkan pada Gambar 1.1 berikut.



Gambar 1.1 Kerangka Berpikir

## Sistematika Penulisan

Sistematika yang digunakan dalam penulisan laporan tugas akhir ini meliputi:

1. BAB I

Pada Bab I akan dijelaskan tentang pendahuluan yang berisi latar belakang, identifikasi masalah, tujuan, manfaat penelitian, batasan masalah, kerangka berpikir, dan sistematika penulisan.

1. BAB II

Pada bab II akan dijelaskan mengenai teori-teori yang menunjang dan mendasari permasalahan dalam pembuatan tugas akhir ini.

1. BAB III

Pada Bab III akan dijelaskan mengenai profil masjid dan metode penelitian yang berisi tahapan-tahapan dalam analisis dan perancangan aplikasi berbasis andorid untuk informasi kegiatan di Masjid Jami’ Al Mu’minin.

1. BAB IV

Pada Bab IV akan dijelaskan tentang hasil dan pembahasan dari perancangan aplikasi berbasis andorid untuk informasi kegiatan di Masjid Jami’ Al Mu’minin.

1. BAB V

Pada Bab V akan dijelaskan kesimpulan yang diperoleh dari penelitian yang telah dilakukan, serta akan dipaparkan pula saran sebagai sumbangan pemikiran bagi penulisan lebih lanjut agar dapat dikembangkan menjadi lebih baik.

# LANDASAN TEORI

Teori yang digunakan sebagai landasan dalam penelitian ini meliputi pengertian *smartphone*, android, masjid, *user interface*, *user experience*, *java*, *application programming interface* (API), *database*, NoSQL, firebase, *JavaScript Object Notation* (JSON), *Extensible Markup Language* (XML), Unified Modeling Language (UML) dan *metode prototyping*.

## *Smartphone*

*Smartphone* adalah telepon selular dengan mikroprosesor, memori, layar dan modem bawaan. *Smartphone* merupakan ponsel multimedia yang menggabungkan fungsionalitas *Personal Computer* (PC) dan *handset* sehingga menghasilkan *gadget* yang mewah, di mana terdapat pesan teks, kamera, pemutar musik, video, *game*, akses email, tv digital, *search engine*, pengelola informasi pribadi, fitur *Global Positioning System* (GPS), jasa telepon internet dan bahkan terdapat telepon yang juga berfungsi sebagai kartu kredit [4].

## Android

Android adalah sistem operasi berbasis java yang beroperasi pada kernel linux 2.6 Sistem ini sangat ringan dan memiliki banyak fitur. Aplikasi android dikembangkan menggunakan java dan dapat dipindahkan ke platform baru dengan mudah. Fitur lain dari android termasuk mesin yang dipercepat dengan grafik 3-D, dukungan SQLite database dan web browser yang terintegrasi[5].

### Fitur Android

Android memiliki beberapa fitur yang mendukung kerja dari android itu sendiri, antara lain [6]:

1. *Application framework*.
2. Dalvik virtual machine (DVM) dioptimalkan untuk perangkat *mobile*.
3. *Browser* yang terintegrasi.
4. Grafis yang dioptimasi, didukung oleh library grafis 2D yang custom grafis 3D berdasarkan spesifikasi OpenGL ES 1.0.
5. SQLite untuk penyimpanan data terstruktur.
6. Media pendukung untuk audio, video dan gambar dalam beberapa format (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
7. *Global System for Mobile Communications* (GSM) Telepon (bergantung pada hardware yang digunakan).
8. Bluetooth, EDGE, 3G dan WiFi (bergantung pada hardware yang digunakan).
9. Kamera, GPS, Kompas dan *accelerometer* (bergantung pada hardware yang digunakan).
10. Lingkungan pengembang yang lengkap, termasuk perangkat emulator, alat untuk debug, memori dan kinerja profil, serta *plugin* untuk Eclipse *Integrated Development Environment* (IDE).

### Arsitektur Android

Sistem operasi Android dibagi menjadi bagian bagian berikut [7]:

1. *Aplications* dan *Widgets* : layer (lapisan) dimana pengguna hanya berhubungan dengan aplikasi saja.
2. *Applications Framework*: lapisan dimana para pengembang melakukan pembuatan aplikasi yang akan dijalankan di sistem operasi Android dengan komponen-komponennya meliputi *views*, *contents* *provider*, *resource* *manager*, *notification manager*, *activity manager*.
3. *Libraries*: lapisan dimana fitur-fitur Android berada yang berada diatas kernel meliputi library C/C++ inti seperti Libc dan *Secure Socket Layer* (SSL).
4. Android *Run Time*: lapisan yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux yang terbagi menjadi dua bagian yaitu *Core Libraries* dan *Dalvik virtual Machine*.
5. Linux Kernel: Layer yang berisi file-file system untuk mengatur *processing*, *memory*, *resource*, *driver*, dan sistem operasi Android lainnya. Sistem operasi yang mendasari Android dilisensikan dibawah *General Public License verse 2* (GNU, GPLv2) yang sering dikenal dengan istilah *copyleft*. Pendistribusian Android dibawah lisensi dari Apache *software* yang memungkinkan untuk distribusi kedua dan seterusnya.

### Versi Android

Android telah melewati cukup banyak pembaruan sejak pertama kali dirilis. Berikut merupakan versi-versi dari android beserta *codename*-nya [8].

Tabel 2.1 Versi Android

|  |  |  |
| --- | --- | --- |
| Codename | Version | API Level |
| Android 11 | 11 | API Level 30 |
| Android 10 | 10 | API Level 29 |
| Pie | 9 | API Level 28 |
| Oreo | 8.1.0 | API Level 27 |
| Oreo | 8.0.0 | API Level 26 |
| Nougat | 7.1 | API Level 25 |
| Nougat | 7.0 | API Level 24 |
| Marshmallow | 6.0 | API Level 23 |
| Lollipop | 5.1 | API Level 22 |
| Lollipop | 5.0 | API Level 21 |
| KitKat | 4.4 – 4.4.4 | API Level 19 |
| Jelly Bean | 4.3.x | API Level 18 |
| Jelly Bean | 4.2.x | API Level 17 |
| Jelly Bean | 4.1.x | API Level 16 |
| Ice Cream Sandwich | 4.0.3 – 4.0.4 | API Level 15, NDK 8 |
| Ice Cream Sandwich | 4.0.1 – 4.0.2 | API Level 14, NDK 7 |
| Honeycomb | 3.2.x | API Level 13 |
| Honeycomb | 3.1 | API Level 12, NDK 6 |
| Honeycomb | 3.0 | API Level 11 |
| Gingerbread | 2.3.3 – 2.3.7 | API Level 10 |
| Gingerbread | 2.3 – 2.3.2 | API Level 9, NDK 5 |
| Froyo | 2.2.x | API Level 8, NDK 4 |
| Eclair | 2.1 | API Level 7, NDK 3 |
| Eclair | 2.0.1 | API Level 6 |
| Eclair | 2.0 | API Level 5 |
| Donut | 1.6 | API Level 4, NDK 2 |
| Cupcake | 1.5 | API Level 3, NDK 1 |
| (No codename) | 1.1 | API Level 2 |
| (No codename) | 1.0 | API Level 1 |

Sumber: (<https://source.android.com/setup/start/build-numbers>)

## Masjid

Masjid pada awal pertama kali dibangun di zaman Rasulullah SAW tidak hanya digunakan sebagai tempat pelaksanaan ibadah melainkan juga menjadi tempat kegiatan utama, tempat berbagi ilmu pengetahuan, berbagi informasi dan pengembangan ekonomi kerakyatan, tempat utama pengaturan strategi perang, serta tempat pembinaan dan pendidikan serta pengembangan sumber daya umat. Fungsi masjid sebagai pusat kegiatan masyarakat bukan hanya konteks sosial yang sederhana, tetapi juga karena proses manajemen sosial masjid yang telah berfungsi sebagai pengikat hubungan sosial [9].

## *User Interface*

*User Interface* merupakan bagian dari sistem informasi yang perlu interaksi dari pengguna untuk membuat *input* dan *output*. Menjelaskan bahwa sebuah sistem informasi baru mempengaruhi banyak dari sistem informasi yang ada lainnya, dan analisis harus memastikan bahwa semua bekerja bersama-sama. Sistem juga harus berinteraksi dengan pengguna baik di dalam maupun di luar organisasi. *User* *Interface* yang lebih dari sekedar layar yaitu, itu adalah serangkai tampilan grafis yang dapat dimengerti oleh pengguna dalam menggunakan sistem, konseptual dan fisik. Dari penjelasan tersebut *User Interface* mempunyai peran yang penting dalam efektivitas suatu sistem informasi. Pembuatan *User Interface* bertujuan untuk menjadikan teknologi informasi tersebut mudah digunakan oleh pengguna [10].

### Delapan Aturan Emas Perancangan *User Interface*

Terdapat delapan aturan emas yang dapat digunakan sebagai petunjuk dasar dalam perancangan tampilan antarmuka pengguna, antara lain [11]:

1. Berusaha untuk konsisten

Konsisten dalam tampilan yang digunakan pada aplikasi, seperti konsisten dalam penggunaan warna, jenis tulisan, ukuran dan tata letak.

1. Menyediakan *usability* yang universal

*Usability* yang universal adalah fungsi-fungsi yang dikenalisecara umum, seperti penggunaan singkatan dan tombol-tombol *shortcu*t yang sudah umum diketahui.

1. Memberikan umpan balik yang informatif

Dalam setiap aksi yang dilakukan oleh pengguna harus disediakan umpan balik yang informatif untuk menunjukkan hasil dari aksi yang dilakukan.

1. Merancang dialog penutupan

Urutan tindakan sebaiknya diatur dalam pengelompokkan menjadi bagian awal, tengah dan akhir. Di mana pada bagian akhir tindakan diberikan umpan balik yang informatif untuk mengindikasikan bahwa proses yang dilalui sudah benar.

1. Memberikan penanganan kesalahan yang sederhana

Rancangan sistem yang baik harus mengurangi terjadinya kesalahan. Apabila pengguna melakukan kesalahan, sistem harus dapat mendeteksi kesalahan tersebut serta memberikan instruksi yang sederhana dan jelas kepada pengguna untuk memperbaiki kesalahan.

1. Memudahkan pembalikan aksi ke kondisi awal

Dalam perancangan sistem harus disediakan fitur yang memungkinkan pengguna untuk kembali ke kondisi semula. Dengan demikian pengguna dapat dengan berani menggunakan pilihan-pilihan yang belum diketahui dengan tidak merasa cemas akan kesalahan yang akan muncul.

1. Mendukung *internal locus of control*

Sistem harus bisa memberikan kendali kontrol kepada pengguna, sehingga pengguna tidak merasa dikendalikan oleh sistem.

1. Mengurangi beban ingatan jangka pendek

Daya ingat manusia yang terbatas untuk memproses informasi sehingga perlu merancangtampilan dengan baik sehingga mudahdimengerti dan diingat oleh pengguna.

## *User Experience*

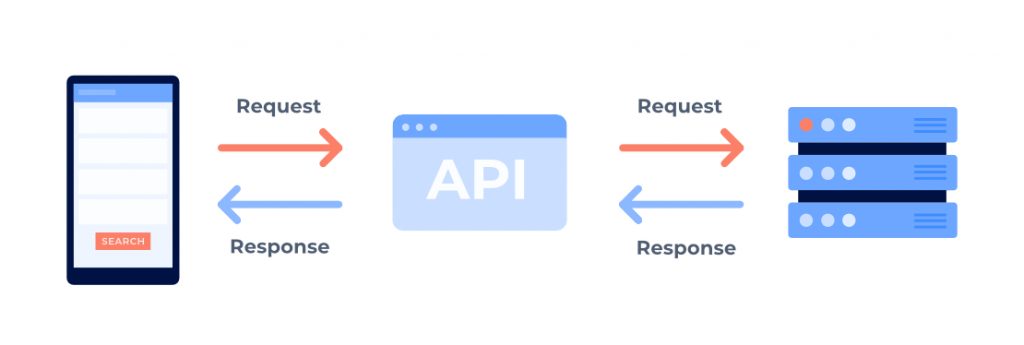
*User Experience* bukanlah tentang cara kerja dari suatu produk atau layanan yang ada. Tetapi bagaimana interaksi antara *user* dengan produk, seperti pengamalan pengguna dalam menggunakan produk, apakah mudah digunakan, sesederhana apa dalam mengoperasikan produk atau layanan hingga pengalaman untuk menemukan, menyerap dan memahami informasi yang tersedia. Hal ini penting karena ketika sebuah produk sedang dikembangkan, biasnya lebih memperhatikan apa yang dilakukannya atau dikembangkan. Sedangkan pengalaman pengguna *user experience* adalah sisi lain yang sering diabaikan, baik dalam segi persamaan hingga bagaimana cara kerjanya. Padahal *user experience* dapat membuat perbedaan antara produk yang sukses dan gagal [10].

## Java

Java adalah suatu teknologi di dunia *software* komputer. Selain merupakan suatu bahasa pemrograman, Java juga merupakan suatu platform. Java merupakan teknologi dimana teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman sendiri. Juga mencakup java sebagai platform dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java [5].

## *Application Programming Interface* (API)

API atau *Application Programming Interface* adalah sebuah *interface* yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform. API sendiri bisa digunakan untuk komunikasi dengan berbagai [bahasa pemrograman](https://www.niagahoster.co.id/blog/bahasa-pemrograman/) yang berbeda. Hal ini memudahkan *developer*, dimana *developer* tidak perlu menyediakan semua data sendiri karena cukup mengambil data yang dibutuhkan dari platform lain melalui API [12].

Gambar 2.1 Cara Kerja API

Sumber: (https://www.niagahoster.co.id/blog/api-adalah/)

## *Database*

*Database* merupakan kumpulan data yang mepunyai relasi logika yang dapat dibagikan, mempunyai deskripsi pada data dan didesain untuk mendapatkan informasi yang dibutuhkan oleh organisasi [13].

## NoSQL

NoSQL merupakan makna dari “*Not Only SQL*” merupakan *database* jenis non-relasional yang menjadi alternatif dari *database* SQL yang merupakan *database* relasional. Perbedaan dari kedua jenis database tersebut terdapat di bentuk skema, SQL memiliki skema yang kaku sedangkan *database* NoSQL memiliki bentuk skema yang lebih fleksibel dan mudah diubah tanpa menggangu sistem yang sedang berjalan [14].

## Firebase

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan bagi para pengembang aplikasi dalam mengembangkan aplikasinya. Firebase alias *Backend as a Service* (BaaS) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan pengembang. Dengan menggunakan Firebase, pengembang aplikasi bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar dalam urusan *backend* [15]*.*

### Sejarah Firebase

Sejarah dari Firebase didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk Firebase yang pertama kali adalah Realtime Database. Realtime Database digunakan pengembang untuk menyimpan data dan *synchronize* ke banyak *user* yang kemudian berkembang sebagai layanan pengembang aplikasi. Pada bulan Oktober 2014, perusahaan tersebut diakuisisi oleh Google [15].

### Fitur Firebase

Berikut adalah fitur-fitur yang diberikan oleh Firebase [15].

1. Fitur *Analytics*

Fitur *Analytics* adalah salah satu fitur pada Firebase yang digunakan sebagai koleksi data dan *reporting* untuk aplikasi Android maupun iOS.

1. Firebase *Cloud Messaging* (FCM)

*Firebase Cloud Messaging* (FCM) merupakan fitur dari Firebase menyediakan koneksi yang handal dan hemat baterai antar server maupun antar *device*. Sehingga dapat mengirim dan menerima pesan serta notifikasi di Android, iOS, dan web tanpa perlu biaya.

1. Firebase *Authentication*

Firebase *Authentication* adalah salah satu layanan *back-end* untuk Android dan iOS. Memiliki SDK yang mudah digunakan, dan memiliki tampilan *interface* yang siap pakai untuk mengautentikasi pengguna ke aplikasi. Firebase *Authentication* mendukung autentikasi menggunakan nomor telepon, sandi, penyedia identitas gabungan populer seperti seperti Google, Facebook, dan sebagainya. Firebase *Authentication* terintegrasi dengan fitur layanan Firebase lainnya. Sistem ini memanfaatkan berbagai jenis standar industri, seperti OAuth 2.0 dan OpenID *Connect*, yang memudahkan integrasi dengan *backend* aplikasi.

1. *Cloud Firestore*

*Cloud Firestore* adalah *database* yang bersifat fleksibel dan terukur untuk pengembangan perangkat seperti seluler, web, dan server di Firebase dan Google *Cloud* Platform. Seperti halnya Firebase Realtime Database, *Cloud Firestore* membuat data tetap terkoneksi di aplikasi *user* melalui *listener realtime* dan menawarkan layanan secara *offline* untuk aplikasi seluler dan web. Dengan begitu, aplikasi dapat dibuat dengan *powerfull*, responsif, dan mampu bekerja tanpa bergantung pada latensi koneksi internet.

1. Firebase *Realtime Database*

Firebase *Realtime Database* adalah *database* yang di-host melalui *cloud*. Data disimpan dan dieksekusi dalam bentuk *JavaScript Object Notation* (JSON) dan disinkronkan secara *realtime* ke setiap *user* yang terkoneksi. Hal ini berfungsi memudahkan pengembang dalam mengelola suatu *database* dengan skala yang cukup besar. Ketika pengembang membuat aplikasi lintas-platform / multi platform menggunakan SDK Android, iOS, dan juga JavaScript (JS), semua pengguna akan berbagi sebuah *instance Realtime Database* dan menerima *update*-an data secara serentak dan otomatis. Kemampuan lain dari Firebase *Realtime Database* adalah tetap responsif bahkan saat *offline* karena *Software Development Kit* (SDK) Firebase *Realtime Database* menyimpan data langsung ke *disk* *device* atau memori lokal. Setelah perangkat terhubung kembali dengan internet, perangkat pengguna akan menerima setiap perubahan yang terjadi.

1. Firebase *Hosting*

Firebase *Hosting* merupakan suatu layanan *hosting* konten web. Hanya dengan satu instruksi dapat mengimplementasikan aplikasi web serta menyajikan konten statis maupun dinamis ke *Content Delivery Network* (CDN) global dengan cepat. Kegunaan dari Firebase *Hosting* itu sendiri yaitu mampu menayangkan konten melalui koneksi yang begitu aman, mengirimkan konten secara cepat, dan mendukung semua jenis konten untuk di hosting, mulai dari file *HyperText Markup Language* (HTML) dan *Cascading Style Sheet* (CSS) hingga API atau layanan mikro Express.js [15].

## *Java Script Object Notation* (JSON)

JSON adalah format pertukaran data yang ringan,mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll [7].

## *Extensible Markup Language* (XML)

*Extensible Markup Language* (XML) adalah bahasa markup serba guna yang direkomendasikan oleh *World Wide Web Consortium* (W3C) untuk mendeskripsikan berbagai macam data. XML menggunakan *markup tags* seperti halnya *Hypertext Markup Language* (HTML) namun penggunaannya tidak terbatas pada tampilan halaman web saja [16].

## *Unified Modeling Language* (UML)

*Unified Modelling Language* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali di populerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan dua metodologi terkenal yaitu Booch dan OMT, kemudian Ivar Jacobson yang menciptakan *Object Oriented Software Engineering* (OOSE) ikut bergabung. Standar UML dikelolah oleh *Object Management Group* (OMG) [7].

1. *Use Case* Diagram

*Use Case* Diagrammerupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interkasi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut adalah simbol-simbol yang ada pada diagram *Use Case* [7].

Tabel . Simbol *Use Case* Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Pengertian** | **Keterangan** |
|  | *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case* |
|  | Aktor | Orang, proses, atau lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang lain, tapi aktor belum tentu merupaka orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
|  | Asosiasi | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor |
| <<extend>> | Ekstensi | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemprograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. |

Sumber: (Rosa A.S dan M. Shalahuddin, 2016)

1. *Activity* Diagram

*Activity* Diagram merupakan diagram yang menggambarkan *workflow* (alirankerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada padaperangkat lunak. Berikut adalah simbol-simbol yang ada pada *activity* diagram [7].

Tabel . Simbol *Activity* Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Pengertian** | **Keterangan** |
|  | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas  memiliki sebuah status awal |
| Aktivitas | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya di awali dengan kata kerja |
|  | Percabangan | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
|  | Penggabungan | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
|  | *Swimlane* | Memisahkan organsasi bisnis yang bertanggung |
| State |  | jawab terhadap aktivitas yang terjadi sistem pada waktu tertentu. State dapat berubah jika ada event tertentu yang memicu perubahan tersebut |

Sumber: (Rosa A.S dan M. Shalahuddin, 2016)

1. *Sequence* Diagram

*Sequence* Diagram merupakan diagram yang menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada *sequence* diagram [7].

Tabel . Simbol *Sequence Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Aktor  Nama actor   |  | | --- | | Nama aktor |   Atau  Tanpa waktu aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
| Garis hidup / lifeline | Menyatakan kehidupan suatu objek |
| Objek  Nama objek: nama kelas | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  1: *login*() 2: cek status *login*()    3: *open*()    Maka cek status *login*() dan *open*() dilakukan di dalam metode *login*() Aktor tidak memiliki waktu aktif |
| Pesan tipe create  <<create>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| Pesan tipe call  1: nama\_metode() | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  1: nama\_metode() |

Sumber: (Rosa A.S dan M. Shalahuddin, 2016)

### *Class Diagram*

*Class* Diagram merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [7].

Tabel . Simbol *Class* Diagram

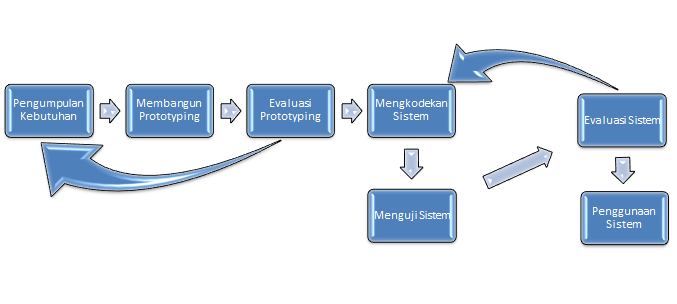
|  |  |  |
| --- | --- | --- |
| **Simbol** | **Pengertian** | **Keterangan** |
| |  | | --- | | Nama\_kelas | | +atribut | | +operasi() | | Kelas | Kelas pada struktur sistem |
|  | Antarmuka | Sama dengan konsep *interface* dalam pemprograman  berorientasi objek |
|  | Asosiasi | Relasi antarmuka dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
|  | Asosiasi berarah | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
|  | Generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
|  | Kebergantungan | Kebergantungan antar kelas |
|  | Agregasi | Relasi antarmuka dengan makna semua bagian *(whole-part)* |

Sumber: (Rosa A.S dan M. Shalahuddin, 2016)

## Metode *Prototyping*

Metode *Prototyping* adalah salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (working model). Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih rendah [17].

### Tahapan *Prototyping*



Gambar . Tahapan *prototyping*

(Sumber : https://medium.com/dot-intern/sdlc-metode-prototype-8f50322b14bf)

Dalam metode pengembangan *prototyping* terdapat tahapan-tahapan yang dilakukan yaitu sebagai berikut.[17]

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

1. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan contoh *output*-nya).

1. Evaluasi *protoptyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka *prototyping* diperbaiki dengan mengulang langkah 1, 2 , dan 3.

1. Pengkodean sistem

Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

1. Menguji system

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan *dengan White Box, Black Box, Basis Path*, pengujian arsitektur dan lain-lain.

1. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.

1. Menggunakan system

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

# METODE PENELITIAN

Pada bab ini dijelaskan tentang bagaimana penelitian ini dilakukan. Oleh karena itu, pada bab ini diuraikan tentang rencana penelitian dan tempat penelitian, serta metode pengumpulan data terkait dalam Rancang Bangun Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin.

## Rencana Penelitian

Penelitian adalah tindakan penyelidikan yang sistematis, terkendali, empiris, teliti dan kritis terhadap fenomena-fenomena untuk dianalisis dengan menggunakan langkah-langkah tertentu agar menghasilkan jawaban ilmiah.

### Tahapan Penelitian

Tahapan penelitian disesuaikan dengan metode penelitian yaitu metode *prototyping*. Metode ini membantu dalam pembuatan aplikasi berbasis android untuk manajemen informasi di Masjid Jami’ Al Mu’minin, karena metode ini menyederhanakan dan mempercepat desain sistem dengan cara berulang-ulang hingga mendapatan sistem yang tepat. Berikut tahapan-tahapan pembuatan metode *prototyping* yang dilakukan oleh peneliti dalam mengembangkan sistem.

1. Mengidentifikasi Masalah

Peneliti melakukan observasi di Masjid Jami’ Al Mu’minin untuk mengidentifikasi permasalahan terkait pengelolaan manajemen informasi masjid.

1. Pengumpulan Data

Peneliti melakaukan pengumpulan data dengan melakukan peninjauan pustaka dan wawancara terhadap pengelolaan informasi di Masjid Jami’ Al Mu’minin.

1. Pengimplementasian Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan oleh peneliti dalam penelitian ini adalah pengembangan sistem model *prototype,* metode *prototype* merupakan metode pengembangan sistem dimana hasil analisa perbagian sistem langsung diterapkan kedalam sebuah model tanpa menunggu seluruh sistem selesai. Metode pengembangan ini digunakan peneliti agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan *prototype* yang sudah ada sehingga dapat menghasilkan aplikasi yang dapat diterima dan digunakan dengan baik oleh pengguna. Berikut adalah tahapan-tahapan yang dilakukan peneliti dalam pembuatan metode *prototype*.

* 1. Pengumpulan Kebutuhan

Peneliti dan ketua Dewan Kemakmuran Masjid (DKM) Masjid Jami’ Al Mu’minin bertemu dan menentukan tujuan umum, kebutuhan yang diketahui, dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

* 1. Membangun *Prototype*

Peneliti Membuat *prototype* aplikasi sebagai perancangan sementara yang berbentuk gambar rancangan *interface* dan fitur *aplikasi*.

* 1. Evaluasi *Prototype*

Ketua Dewan Kemakmuran Masjid (DKM) mengevaluasi *prototype* yang dibuat peneliti. Lalu peneliti mengulangi tahap 1, 2, dan 3 hingga sampai dimana prototype yang diajukan peneliti disetujui oleh ketua Dewan Kemakmuran Masjid (DKM) Masjid Jami’ Al Mu’minin.

* 1. Pengkodean sistem

Dalam tahap ini peneliti melakukan penerjemahan *prototype* ke dalam bahasa pemrograman android sebagai dasar basis aplikasi.

* 1. Pengujian sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, peneliti melakukan pengujian kelayakan aplikasi menggunakan metode *blackbox*.

* 1. Evaluasi sistem

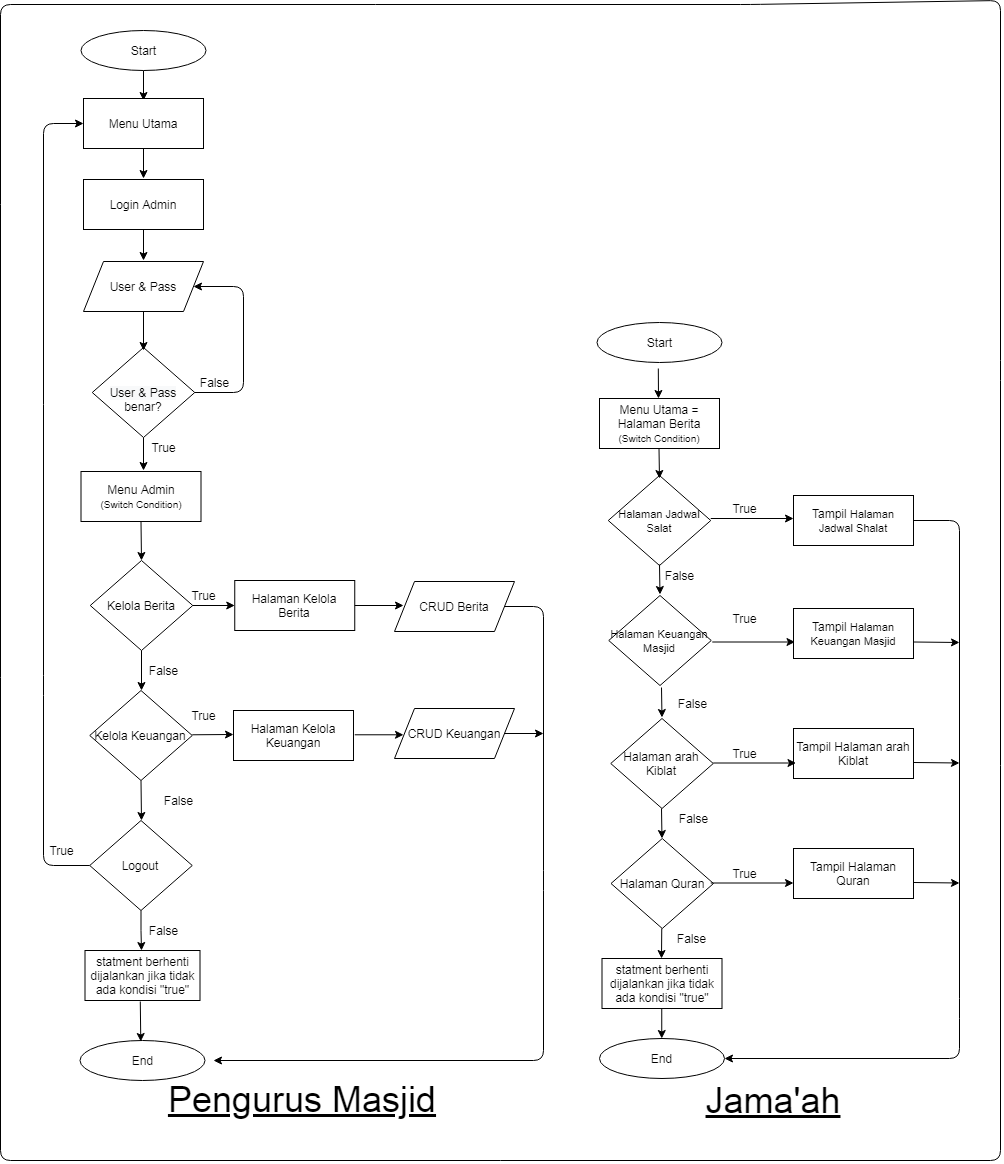
Ketua DKM kembali melakukan evaluasi terhadap aplikasi yang telah dibuat, untuk melihat apakah aplikasi sudah sesuai dengan yang diharapkan.

* 1. Menggunakan sistem

Sistem telah diuji dan siap untuk digunakan.

## Alur Proses Sistem Yang Diusulkan

Berikut adalah gambaran alur kerja atau proses dari sistem yang diusulkan oleh peneliti, digambarkan dalam *flowchart* berikut.



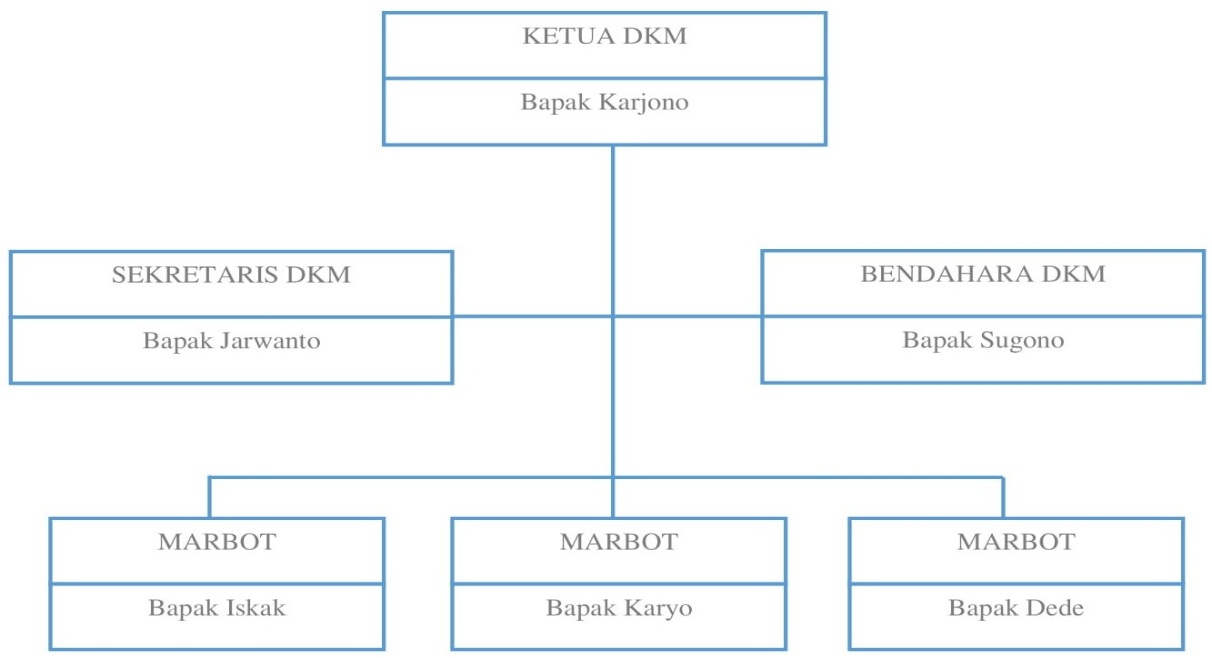
Gambar . *Flowchat* Sistem Usulan

## Waktu dan Populasi Penelitian

Waktu yang digunakan peneliti dalam melakukan penelitian ini yaitu dimulai dari tanggal 7 Maret 2021 sampai dengan selesai. Populasi dalam penelitian ini yaitu jama’ah dan jajaran pengurus Masjid Jami’ Al Mu’minin.

## Struktur Organisisai

Berikut ini merupakan struktur organisasi Dewan Kemakmuran Masjid yang ada pada Masjid Jami’ Al Mu’minin.



Gambar 3.2 Struktur Organisasi Masjid Jami’ Al Mu’minin

## Metode Analisis *Pieces*

Metode analisis *pieces* merupakan kerangka analisa kebutuhan sistem informasi yang dipakai untuk mengelompokkan suatu *problem*, *opportunities*, dan *directives* yang terdapat pada bagian *scope* *definition* analisa dan perancangan sistem. Metode analisis *pieces* menggunakan enam variable untuk menentukan kelayakan sistem yang diusulkan yaitu sebagai berikut [9].

a. *Performance* (kinerja) adalah ukuran terhadap kinerja sistem yang akan dibangun sehingga menjadi lebih efektif diukur dari jumlah pekerjaan yang dapat dikerjakan pada saat tertentu dan lama waktu respon.

b. *Information* (informasi) adalah ukuran terhadap kualitas informasi yang dapat disajikan atau ditampilkan.

c. *Economics* (ekonomi) adalah ukuran terhadap manfaat atau keuntungan dan penurunan biaya yang dikeluarkan.

d. *Control* (pengendalian) adalah ukuran terhadap pengendalian untuk mendeteksi serta memperbaiki kesalahan yang akan muncul.

e. *Efficiency* (efisiensi) adalah ukuran peningkatan terhadap efisiensi waktu operasi.

f. *Service* (pelayanan) adalah ukuran terhadap pelayanan yang diberikan oleh sistem terhadap pengguna.

## Objek Penelitian

****Penelitian ini dilakukan di Masjid Jami’ Al Mu’minin yang terletak di Jalan Cempaka raya, RT.033/11 Kelurahan Rawabuaya, Kecamatan Cengkareng, Jakarta Barat.

Gambar 3.3 Masjid Jami’ Al Mu’minin

## Metode Pengumpulan Data

Metode pengumpulan data bertujuan untuk mengungkapkan atau menjaring informasi sesuai dengan lingkup penelitian. Metode yang digunakan oleh peneliti untuk mengumpulkan data dalam penelitian ini, yaitu studi literatur, observasi dan wawancara.

1. Studi literatur

Studi literatur adalah tahap dalam mengumpulkan semua informasi yang diperlukan peneliti untuk membangun sistem. Informasi tersebut dapat diperoleh peneliti dari berbagai sumber dengan cara membaca literatur yang terdapat pada jurnal, artikel dan skripsi. Dari literatur tersebut dapat diketahui persamaan ataupun perbedaan terhadap penelitian yang dilakukan oleh peneliti maupun dengan peneliti lainya. Sumber tersebut dijadikan sebagai landasan teori untuk proses pembuatan aplikasi berbasis android untuk informasi kegiatan di Masjid Jami’ Al Mu’minin.

Dari hasil studi literatur, peneliti menemukan referensi sebagai acuan yang relevan dan mendukung dalam penelitian ini yaitu, jurnal internasional yang berjudul *Fitri Tomyam Seafood Menu Order Android Mobile Application* [18]*,* kemudian jurnal nasionalyang berjudul *Sistem Informasi Pengelolaan Masjid Online Menggunakan Metode Watrefall* [19]*,* serta jurnal nasional yang berjudul *Perancangan dan Implementasi Aplikasi Ikhtisar Kas Masjid Istiqomah Berbasis Dekstop* [20]*.*

Jurnal internasional berjudul *Fitri Tomyam Seafood Menu Order Android Mobile Application* yang menghasilkan sebuah aplikasi berbasis *mobile* yang dikembangkan dengan tujuan untuk membantu proses bisnis dalam aspek *delivery* *order* untuk restoran Fitri Tomyam Seafood dengan menggunakan metode pengembangan perangkat lunak berorientasi objek dalam pembangunan aplikasinya dapat menjadi tinjauan pustaka yang relevan karena memiliki kesamaan dalam penelitian yang dilakukan oleh peneliti, yaitu terletak pada dasar pembuatan aplikasi menggunakan android dan menggunakan *database* firebase.

Jurnal nasional yang berjudul *Sistem Informasi Pengelolaan Masjid Online Menggunakan Metode Waterfall* yang menghasilkan sebuah sistem informasi berbasis web yang dikembangkan dengan tujuan untuk membantu pengurus masjid dalam pengelolaan manajemen informasi masjid serta membantu masyarakat dalam memperoleh informasi seputar Masjid An-Nur Bekasi dengan menggunakan metode pengembangan *waterfall* dalam pembangunan sistem informasinya dapat menjadi tinjauan pustaka yang relevan karena memiliki kesamaan dalam penelitian yang dilakukan oleh peneliti, yaitu terletakpada topik penelitian yang sama yaitu untuk informasi kegiatan dan keuangan masjid.

Jurnal yang berjudul *Perancangan dan Implementasi Aplikasi Ikhtisar Kas Masjid Istiqomah Berbasis Dekstop* yang menghasilkan sebuah sistem informasi berbasis dekstop yang dikembangkan dengan tujuan untuk mengatasi masalah dalam pencatatan dan transparasi keuangan kas Masjid Istiqamah dengan menggunakan metode pengembangan *prototype* dalam pembangunan sistem informasinya dapat menjadi tinjauan pustaka yang relevan karena memiliki kesamaan dalam penelitian yang dilakukan oleh peneliti, yaitu terletak pada penyajian terhadap transparasi keuangan masjid dan metode pengembangan sistem yang sama.

1. Observasi

Obervasi dilakukan dengan cara melakukan pengamatan atau peninjauan langsung terhadap objek penelitian yang dalam hal ini yaitu pada Masjid Jami’ Al Mu’minin. Observasi dilakukan pada tanggal 10-15 Maret 2021.

1. Wawancara

Pada tahap ini peneliti melakukan tanya jawab dengan pengurus masjid dan masyarakat yang merupakan jama’ah Masjid Jami’ Al Mu’minin untuk memperoleh keterangan mengenai objek penelitian dan berbagai kebutuhan pengguna.

# HASIL DAN PEMBAHASAN

Hasil dan pembahasan merupakan bab keempat yang berisi mengenai data hasil penelitian, pembahasan hasil penelitian, serta perbandingan dengan penelitian lain.

## *Requirement* Analisis

*Requirement* analisis merupakan suatu proses untuk mendapatkan informasi mengenai sistem dan kebutuhan apa saja yang diperlukan untuk membangun sebuah sistem.

### Kebutuhan Fungsional

Kebutuhan sistem secara fungsional (*Fungsional Requirements*) adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional berisi informasi-informasi apa saja yang harus ada dan dilakukan oleh aplikasi berbasis android untuk manajemen informasi di Masjid Jami’ Al Mu’minin yang akan dibangun. Adapun rencana kebutuhan sistem yang akan dibuat terdapat pada tabel berikut.

Tabel 4.1 Analisis Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Peran** |
| 1. | Admin | *Login* |
| Melihat Berita/Agenda Kegiatan |
| Membuat Berita/Agenda Kegiatan |
| Mengedit Berita/Agenda Kegiatan |
| Menghapus Berita/Agenda Kegiatan |
| Melihat Data Keuangan (Pemasukan & Pengeluaran) |
| Membuat Data Keuangan (Pemasukan & Pengeluaran) |
| Mengedit Data Keuangan (Pemasukan & Pengeluaran) |
| Menghapus Data Keuangan (Pemasukan & Pengeluaran) |
| 2 | Pengguna Umum | Melihat Berita/Agenda Kegiatan |
| Melihat Jadwal Salat |
| Melihat Data Keungan Masjid |
| Melihat Arah Kiblat |
| Melihat Halaman Qur’an |

### Kebutuhan Non Fungsional

Kebutuhan Non-Fungsional adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh aplikasi yang sedang dibangun. Adapun analisis kebutuhan non-fungsional pada sistem akan dijabarkan dalam tabel berikut.

Tabel 4.2 Analisis Kebutuhan Non Fungsional

|  |  |  |
| --- | --- | --- |
| **No.** | **Klasifikasi** | **Requirement** |
| 1 | Operasinal | Aplikasi yang dibangun hanya dapat dijalankan pada sistem operasi android. |
| Website admin dijalankan dengan browser yang mendukung HTML5. |
| 2 | Keamanan | Sistem Aplikasi bagi admin dilengkapi dengan *password*. |
| Akses *database* dilengkapi dengan *password* dan token. |
| 3 | Informasi | Aplikasi menyediakan informasi terkait agenda kegiatan dan keuangan masjid. |
| 4 | Kinerja | Aplikasi bersifat *realtime* terkait semua informasi yang disajikan oleh aplikasi. Agar informasi dapat diterima oleh pengguna dengan cepat. |
| Aplikasi hanya tersedia dalam satu bahasa, yaitu Bahasa Indonesia. |

### Analisis Perangkat Keras untuk Pembuatan Aplikasi

Spesifikasi *Hardware* atau perangkat keras yang digunakan oleh peneliti dalam aplikasi dijabarkan dalam tabel berikut.

Tabel 4.3 Spesifikasi Hardware yang digunakan dalam pembuatan aplikasi

|  |  |
| --- | --- |
| **Perangkat** | **Spesifikasi** |
| Merk | Dell Inspiron N4030 |
| Processor | Intel(R) Core(TM) i5 CPU M 450 @ 2,40GHz (4CPus), ~2,4GHz |
| Memory | VGEN 6144MB |
| VGA | Intel(R) HD Graphics |
| Storage | ADATA 128GB |

### Analisis Perangkat Lunak untuk Pembuatan Aplikasi

Spesifikasi *software* atau perangkat lunak yang digunakan oleh peneliti dalam pembuatan aplikasi dijabarkan dalam tabel berikut.

Tabel 4.4 Spesifikasi Software yang digunakan dalam pembuatan aplikasi

|  |  |
| --- | --- |
| **Perangkat** | **Nama** |
| Sistem Operasi | Windows 7 Ultimate 64-bit (6.1, Build 7601) |
| Aplikasi Database | Google Firebase |
| Aplikasi Pembangun | Android Studio |
| Apliaksi Pemodelan Sistem | Draw I.O |
| Aplikasi Testing (Emulator) | Android Virtual Device |

### Spesifikasi Minimum Perangkat untuk Menjalankan Aplikasi

Spesifikasi minimum perangkat yang dibutuhkan untuk menjalankan aplikasi dijabarkan dalam tabel berikut.

Tabel 4.5 Spseifikasi Minimum Perangkat Untuk Menjalankan Aplikasi

|  |  |
| --- | --- |
| **Perangkat** | **Spesifikasi** |
| Versi Android | Android Marshmallow 6.0 |
| Processor | Dual-Core 1,2GHz |
| Memory | 2GB |
| Display | TFT Capacitive touchscreen, 16M colors |
| GPS | GPS |
| SIM card | GSM & CDMA |

## Hasil Analisis Menggunakan Metode PIECES

Untuk menentukan kelayakan sistem yang akan diusulkan, peneliti melakukan analisis menggunakan metode analisis *PIECES* terhadap masalah-masalah yang ada pada Masjid Jami’ Al Mu’minin. Berikut adalah hasil analisis menggunakan metode PIECES yang digambarkan dalam tabel berikut:

Tabel 4.6 Metode Analisis *Pieces*

|  |  |
| --- | --- |
| **ANALISIS** | **HASIL** |
| *Perfomance* | * Dibutuhkan aplikasi yang simple, mudah digunakan, dan dapat dioperasikan kapan pun, dan dari mana pun. * Proses cepat dalam mengolah data dan pencarian data. |
| *Information* | * Dibutuhkan aplikasi yang dapat menghasilkan informasi yang akurat, tepat guna dan bermanfaat. |
| *Economics* | * Dibutuhkan aplikasi yang tidak menghabiskan banyak biaya operasional. |
| *Control* | * Dibutuhkan aplikasi yang aman, data-data tersimpan dengan baik. * Memiliki hak akses masing-masing pengguna. |
| *Efficiency* | * Dibutuhkan aplikasi yang *simple* * Tidak menghabiskan waktu dalam menggunakan * Tidak perlu bidang ahli tertentu untuk menggunakan aplikasi |
| *Service* | * Dibutuhkan aplikasi yang bisa digunakan kapan saja dan dimana saja * Dapat menghasilkan informasi-informasi yang dibutuhkan. |

## Data Hasil Penelitian

Peneliti melakukan observasi dan wawancara terhadap ketua DKM dan pengurus Masjid Jami’ Al Mu’minin untuk mengetahui masalah yang terjadi. Wawancara dilakukan untuk mengetahui hal-hal berikut.

1. Bagaimana cara pengurus masjid mengelola di Masjid Jami’ Al Mu’minin.
2. Bagaimana cara merancang aplikasi untuk membantu pengurus masjid dalam mengelola dan menyampaikan informasi kepada masyarakat.
3. Bagaimana cara agar informasi dapat dikelola dan disampaikan dengan baik, dengan memanfaatkan sumber daya manusia yang ada berdasarkan pengetahuan dan informasi yang dimiliki.

## Pemodelan Aplikasi

Perancangan dan Implementasi Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin ini diharapkan dapat memberikan solusi terhadap masalah yang terjadi. Dan untuk menggambarkan pemodelan aplikasi dalam pembangunan sistem, peneliti menggunakan *Unified Modeling Language* (UML) sebagai berikut.

1. *Use Case* Diagram
2. *Activity* Diagram
3. *Sequence* Diagram
4. *Class* Diagram

Pada perancangan sistem aplikasi ini ada 2 (dua) aktor yang berhubungan langsung dengan sistem ini:

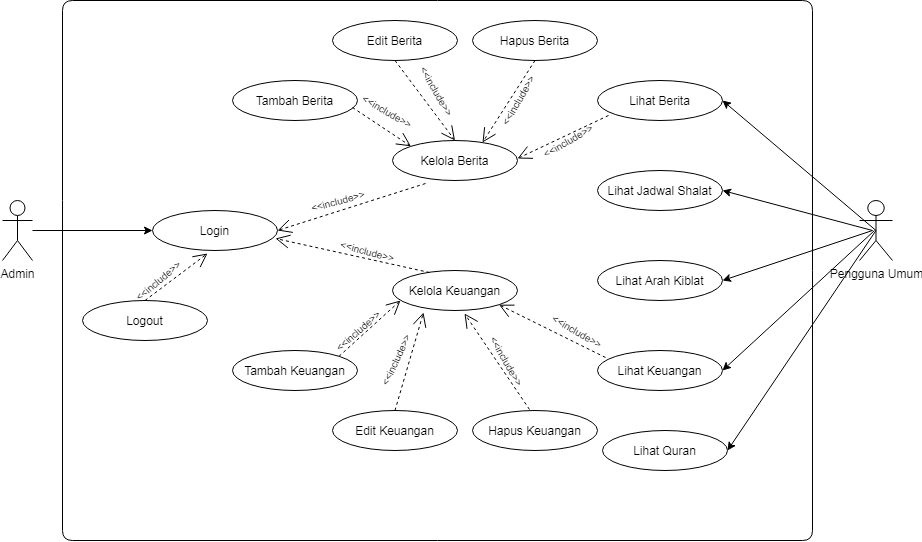
1. Admin/Pengurus masjid

Bertugas untuk mengelola informasi program kegiatan masjid dan pemasukan dan pengeluaran keuangan masjid.

1. Pengguna Umum

Jama’ah atau masyarakat yang merupakan pengguna layanan aplikasi ini untuk melihat informasi terkait program kegiatan dan keuangan masjid.

### Rancangan *Use Case* Diagram



Gambar 4.1 *Use Case* Diagram

*Use Case* Diagram digunakan untuk menggambarkan urutan interaksi yang saling berkaitan antara aktor dan sistem. Penjelasan nya dapat dilihat pada tabel berikut ini.

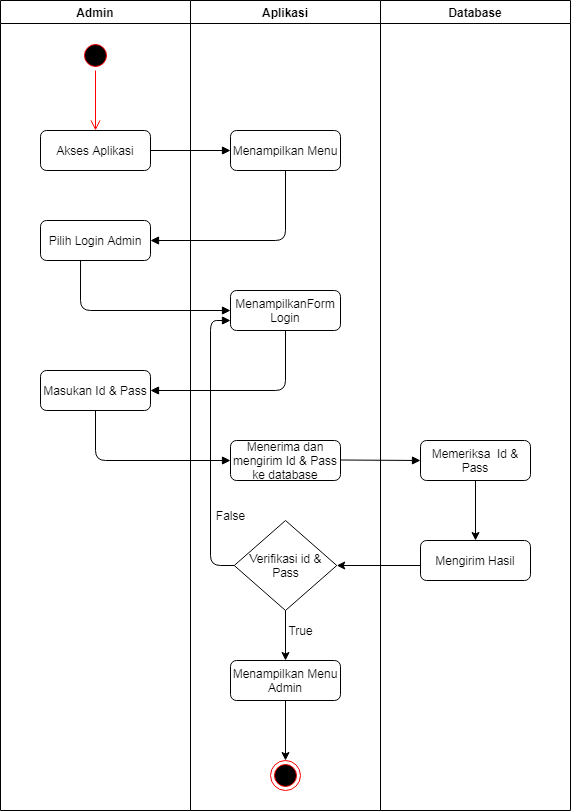
Tabel 4.7 Penjelasan *Use Case*

|  |  |
| --- | --- |
| **Actor** | **Event** |
| Admin (Pengurus masjid) | Melakukan *login*  Melihat isi berita/agenda kegiatan.  Menambah isi berita/agenda kegiatan.  Mengedit isi berita/agenda kegiatan.  Menghapus isi berita/agenda kegiatan  Menampilkan menu keola keuangan  Melihat data keuangan  Menambah data keuangan  Mengedit data keuangan  Menghapus data keuangan  *Logout*. |
| Pengguna Umum | Melihat halaman berita/agenda kegiatan masjid.  Melihat halaman jadwal salat  Melihat halaman keuangan masjid  Melihat halaman arah kiblat  Melihat halaman Quran |

### Rancangan *Activity* Diagram

*Activity* Diagram atau diagram aktivitas adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan atau pengulangan. Dalam *Unified Modeling Language* (UML). Tujuan utama dari *activity* diagram adalah menggambarkan proses bisnis agar lebih dipahami. *Activity* juga digunakan untuk menunjukan aliran pesan dari satu aktivitas ke aktivitas lainya.

Gambar berikut ini berisi tentang perancangan sistem yang diusulkan dengan menggunakan *activity* diagram. Diagram ini menggambarkan *activity* antara admin (Pengurus Masjid) dengan sistem ketika *login*.

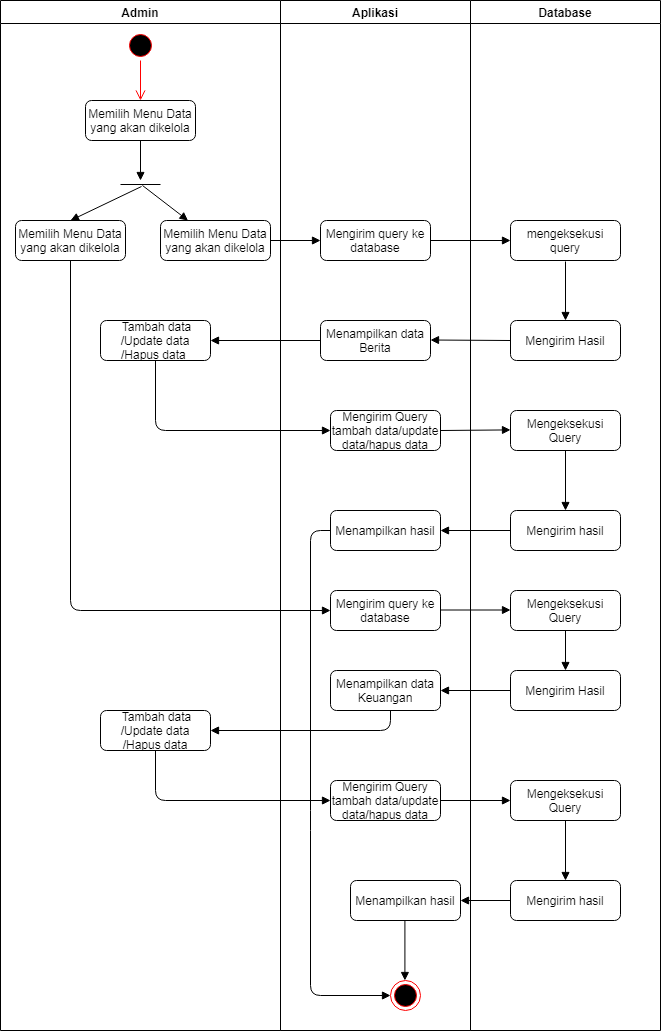


Gambar 4.2 *Activity* Diagram 1

Penjelasan dari gambar 4.2 *activity* diagram 1 adalah sebagai berikut.

1. Admin (Pengurus Masjid) masuk ke dalam aplikasi.
2. Aplikasi menampilkan menu
3. Admin (Pengurus Masjid) pilih menu *login*
4. Aplikasi menampilkan *form login*
5. Admin (Pengurus Masjid) memasukan *username* & *password*
6. Aplikasi mengecek dan memvalidasi *username* & *password* dari *database*
7. Jika *username* & *password salah,* aplikasi akan kembali menampilkan form *username* & *password*
8. Jika *username* & *password* benar aplikasi masuk ke menu admin
9. Selesai

Kemudian gambar berikut ini merupakan perancangan *activity* diagram antara admin dengan sistem ketika mengelola informasi berita dan data keuangan.

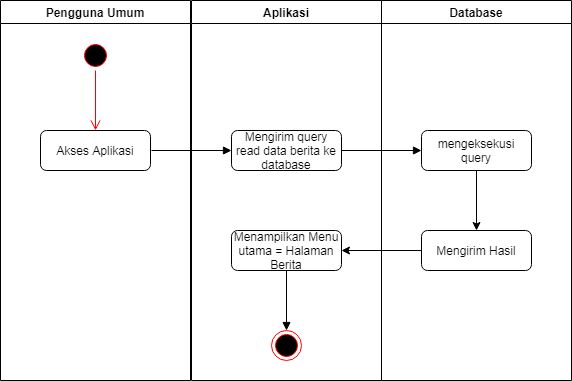


Gambar 4.3 *Activity* Diagram 2

Penjelasan dari gambar 4.3 *activity* diagram 2 adalah sebagai berikut.

1. Admin (Pengurus Masjid) berada di menu admin dan memilih “kelola berita” atau “kelola keuangan”
2. Bila admin (Pengurus Masjid) memilih kelola berita maka aplikasi akan menampilkan daftar berita yang aktif.
3. Admin (Pengurus Masjid) menambah/mengedit/menghapus berita.
4. Aplikasi mengirim data ke *database*
5. *Database* mengeksekusi dan mengirim hasil ke aplikasi
6. Aplikasi menampilan hasil
7. Bila admin (Pengurus Masjid) memilih kelola keuangan maka aplikasi akan menampilkan daftar total pemasukan dan pengeluaran keuangan.
8. Admin (Pengurus Masjid) menambah/mengedit/menghapus keuangan, baik pemasukan keuangan maupun pengeluaran keuangan.
9. Aplikasi mengirim data ke *database*
10. *Database* mengeksekusi dan mengirim hasil ke aplikasi
11. Aplikasi menampilan hasil
12. Selesai

Selanjutnya gambar berikut ini merupakan perancangan *activity* diagram antara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika melihat halaman informasi berita.

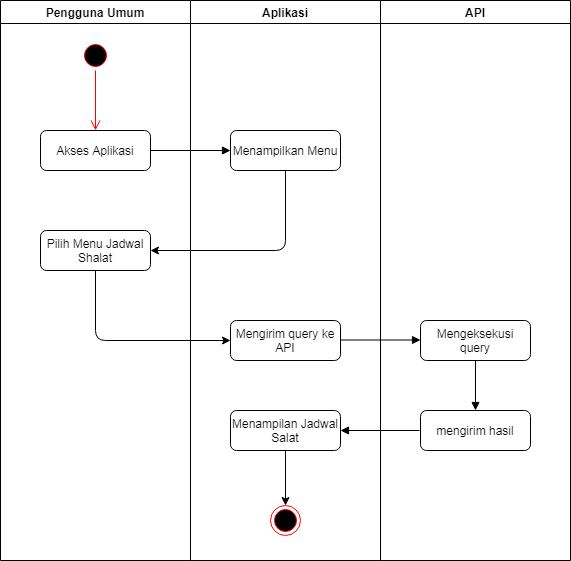


Gambar 4.4 *Activity* Diagram 3

Penjelasan dari gambar 4.4 *activity* diagram 3 adalah sebagai berikut.

1. Pengguna Umum (Masyarakat/jama’ah) mengakses aplikasi
2. Aplikasi mengirim *query* ke *database*
3. *Database* mengeksekusi *query* dan mengirim hasil ke aplikasi
4. Aplikasi menampilkan informasi berita
5. Selesai.

Gambar berikut ini berisi tentang perancangan *activity* diagram antara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika melihat halaman jadwal salat.

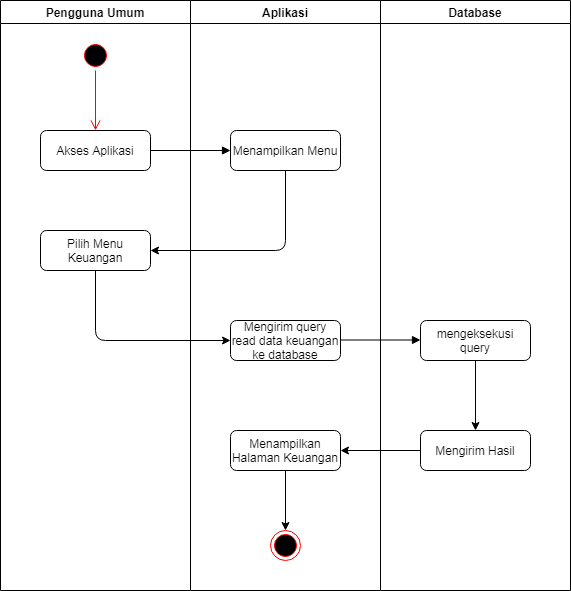


Gambar 4.5 *Activity* Diagram 4

Penjelasan dari gambar 4.5 *activity* diagram 4 adalah sebagai berikut.

1. Pengguna Umum (Masyarakat/jama’ah) mengakses aplikasi
2. Pengguna Umum (Masyarakat/jama’ah) memilih menu jadwal salat
3. Aplikasi mengirim *query* ke API
4. API mengeksekusi *query* dan mengirim hasil
5. Aplikasi menampilkan jadwal salat
6. Selesai.

Kemudian gambar berikut ini berisi tentang penggambaran *activity* diagram antara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika melihat halaman data keuangan masjid.

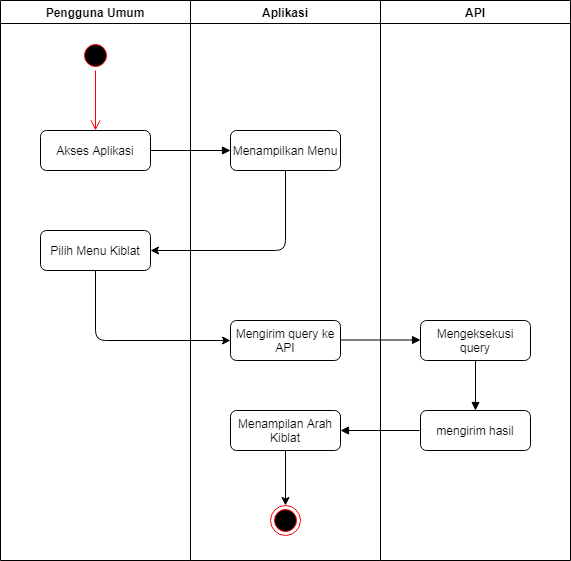


Gambar 4.6 *Activity* Diagram 5

Penjelasan dari gambar 4.6 *activity* diagram 5 adalah sebagai berikut:

1. Pengguna Umum (Masyarakat/jama’ah) mengakses aplikasi
2. Pengguna Umum (Masyarakat/jama’ah) memilih menu keuangan
3. Aplikasi mengirim *query* ke *database*
4. *Database* mengeksekusi *query* dan mengirim hasil ke aplikasi
5. Aplikasi menampilkan data keuangan
6. selesai

Selanjutnya gambar berikut ini berisi tentang penggambaran *activity* diagram antara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika melihat halaman arah kiblat.

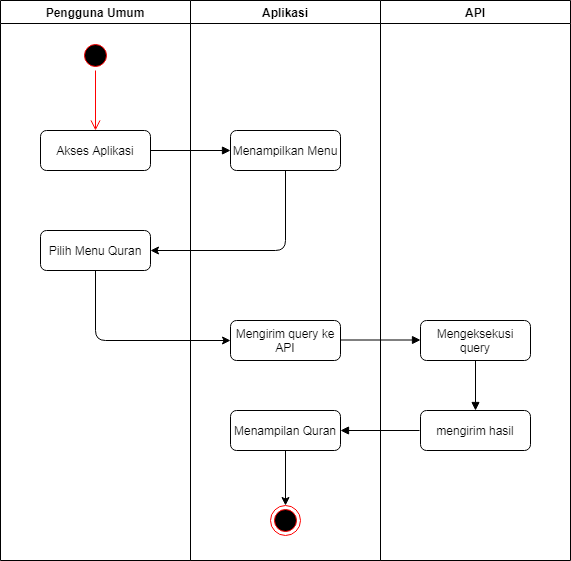


Gambar 4.7 *Activity* Diagram 6

Penjelasan dari gambar 4.7 *activity* diagram 6 adalah sebagai berikut:

1. Pengguna Umum (Masyarakat/jama’ah) mengakses aplikasi
2. Pengguna Umum (Masyarakat/jama’ah) memilih menu kiblat
3. Aplikasi mengirim *query* ke API
4. API mengeksekusi *query* dan mengirim hasil
5. Aplikasi menampilkan arah kiblat
6. Selesai.

Kemudian gambar berikut ini berisi tentang penggambaran *activity* diagram antara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika melihat halaman Quran.



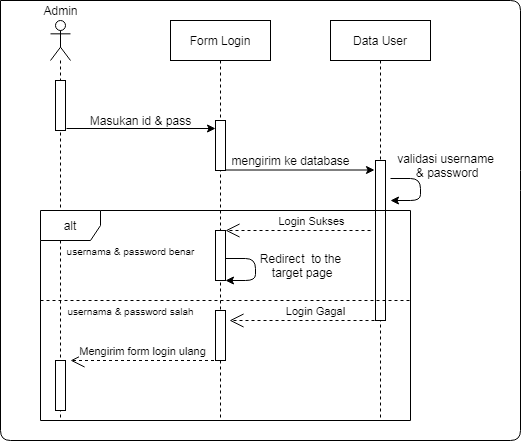
Gambar . *Activity* Diagram 7

Penjelasan dari gambar 4.8 *activity* diagram 7 adalah sebagai berikut:

1. Pengguna Umum (Masyarakat/jama’ah) mengakses aplikasi
2. Pengguna Umum (Masyarakat/jama’ah) memilih menu kiblat
3. Aplikasi mengirim *query* ke API
4. API mengeksekusi *query* dan mengirim hasil
5. Aplikasi menampilkan halaman Quran
6. Selesai.

### Rancangan Sequence Diagram

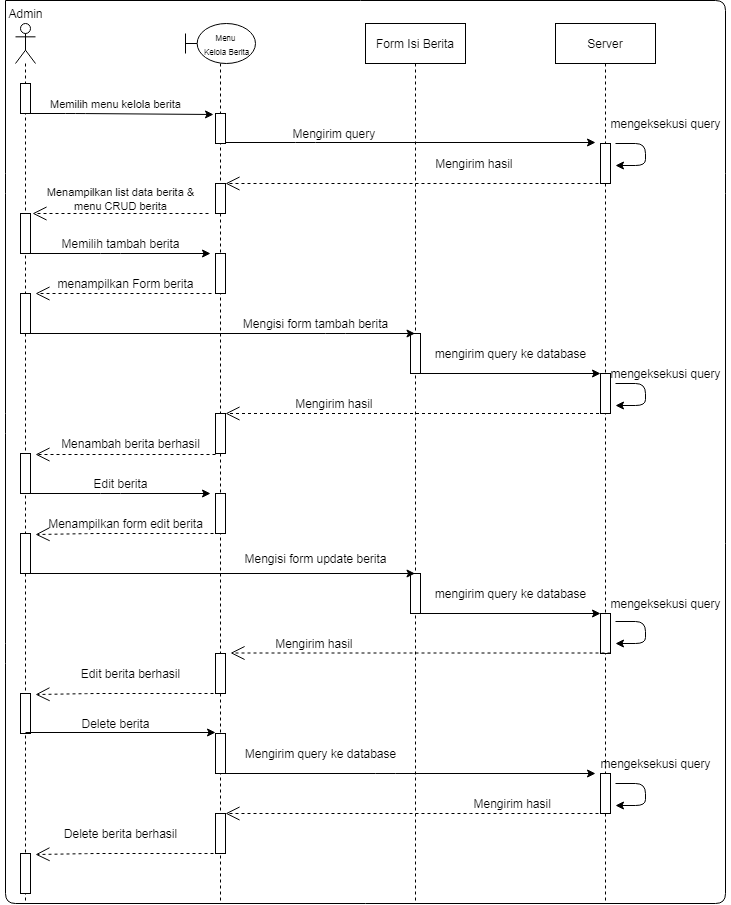
*Sequence* Diagram merupakaan suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence* Diagram digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat berinteraksi. Semua pesan digambarkan dalam urutan pada eksekusi.

Gambar berikut berisi tentang perancangan sistem yang diusulkan dengan menggunakan *sequence* diagram. Diagram ini menggambarkan urutan interaksi admin (Pengurus Masjid) dengan sistem ketika *login*.

Gambar . *Sequence* Diagram 1

Penjelasan dari gambar 4.9 *sequence* diagram 1 adalah sebagai berikut:

1. Admin (pengurus masjid) mengisi *username* & *password* pada form *login*
2. Form *login* akan mengirim hasil inputan ke *database*
3. *Database* mengecek validasi *username* & *password*
4. Bila *username* & *password* valid maka *login* sukses dan menampilkan halaman baru yaitu menu admin
5. Bila *username* & *password* invalid maka *login* gagal akan ada notifikasi dan tetap berada pada *form login.*

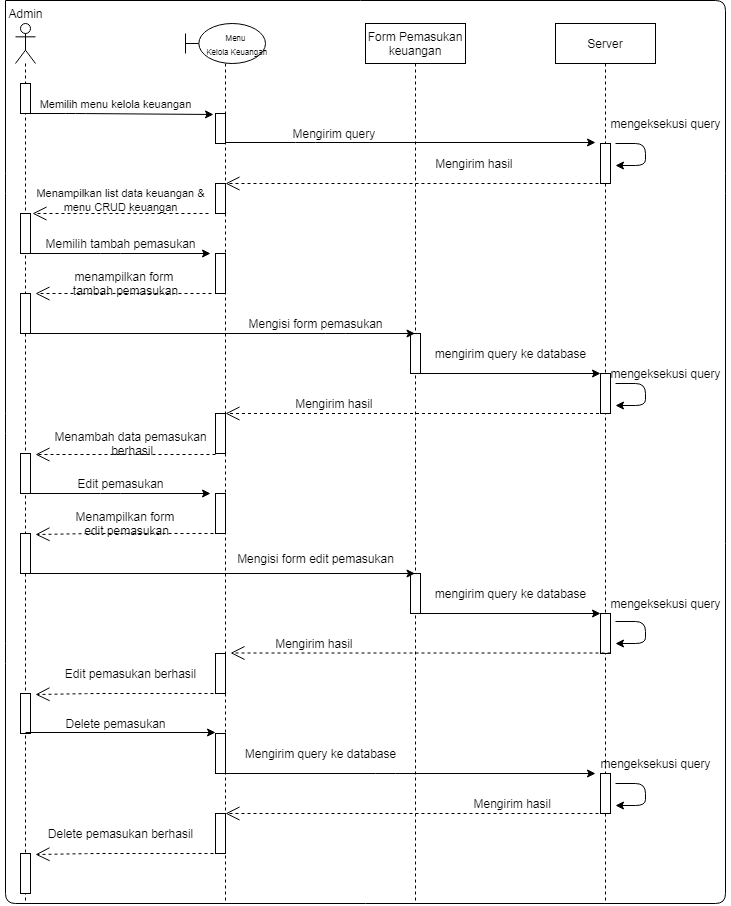
Selanjutnya gambar berikut ini merupakan perancangan *sequence* diagram diantara admin (Pengurus Masjid) dengan sistem ketika mengelola berita kegiatan masjid.

Gambar 4.10 *Sequence* Diagram 2

Penjelasan dari gambar 4.10 *sequence* diagram 2 adalah sebagai berikut:

1. Admin (pengurus masjid) memilih menu kelola berita saat berada di halaman admin
2. Aplikasi mengirim *query* ke server
3. Lalu server mengeksekusi *query* dan dilanjutkan dengan mengirim balik ke aplikasi untuk ditampilkan di menu kelola berita.
4. Aplikasi menampilkan daftar berita yang aktif, dan juga terdapat beberapa opsi yaitu menambah berita baru, mengedit berita, dan menghapus berita.
5. Bila Admin (pengurus masjid) memilih tambah berita,maka menu kelola berita akan menampilkan *form* tambah berita.
6. Admin (pengurus masjid) mengisi *form* tambah berita.
7. *Form* tambah berita dikirim ke server.
8. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu kelola berita.
9. Lalu aplikasi menampilkan list berita terbaru.
10. Bila Admin (pengurus masjid) memilih *edit* berita, maka menu kelola berita akan menampilkan *form* *edit* berita.
11. Admin (pengurus masjid) mengisi *form edit* berita.
12. Form *edit* berita ke server.
13. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu kelola berita.
14. Lalu aplikasi menampilkan *list* berita terbaru.
15. Bila Admin (pengurus masjid) memilih *delete* berita, maka aplikasi akan mengirim *query* ke server.
16. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu kelola berita.
17. Lalu aplikasi menampilkan kembali berita terbaru..
18. Selesai

Kemudian gambar berikut ini merupakan perancangan *sequence* diagram diantara admin (Pengurus Masjid) dengan sistem ketika mengelola pemasukan keuangan masjid.

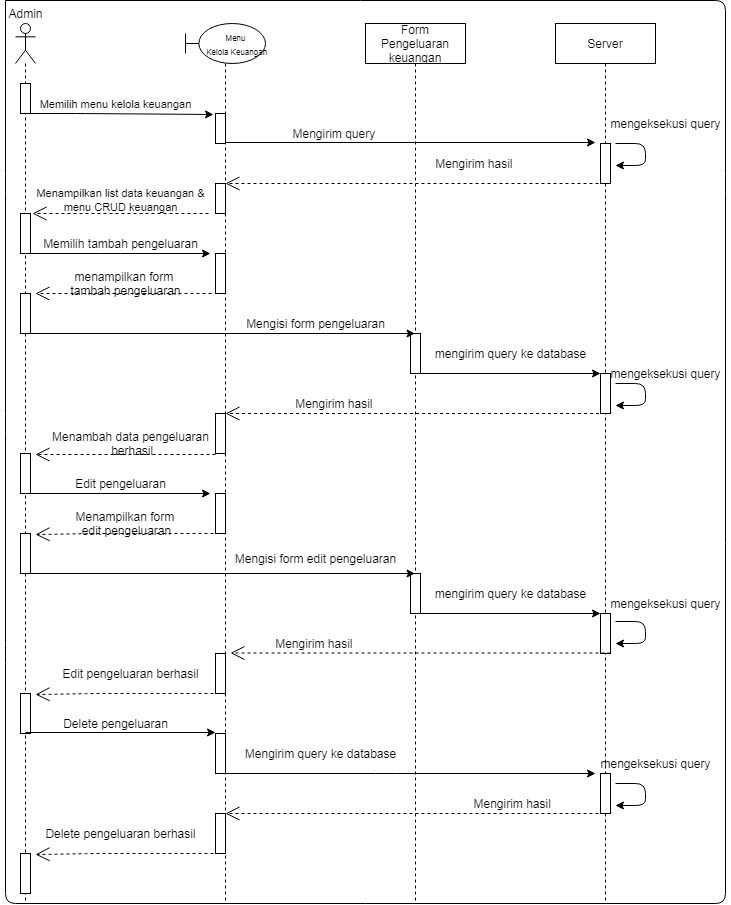


Gambar 4.11 *Sequence* Diagram 3

Penjelasan dari gambar 4.11 *sequence* diagram 3 adalah sebagai berikut:

1. Admin (pengurus masjid) memilih menu kelola keuangan saat berada di halaman admin
2. Aplikasi mengirim *query* ke server
3. Lalu server mengeksekusi *query* dan dilanjutkan dengan mengirim balik ke aplikasi untuk ditampilkan di menu pemasukan keuangan.
4. Aplikasi menampilkan daftar pemasukan keuangan yang aktif, dan juga terdapat beberapa opsi yaitu menambah pemasukan keuangan baru, mengedit pemasukan keuangan, dan menghapus pemasukan keuangan.
5. Bila Admin (pengurus masjid) memilih tambah pemasukan keuangan,maka menu kelola keuangan akan berpindah ke halaman form tambah pemasukan keuangan.
6. Admin (pengurus masjid) mengisi form pemasukan keuangan.
7. Form tambah pemasukan keuangan dikirim ke server.
8. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu pemasukan keuangan.
9. Lalu aplikasi menampilkan list pemasukan keuangan terbaru.
10. Bila Admin (pengurus masjid) memilih *edit* pemasukan keuangan, maka aplikasi akan menampilkan form *edit* pemasukan keuangan.
11. Admin (pengurus masjid) mengisi form *edit* pemasukan keuangan.
12. *Form* *edit* pemasukan keuangan dikirim ke server.
13. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu pemasukan keuangan.
14. Lalu aplikasi menampilkan *list* pemasukan keuangan terbaru.
15. Bila Admin (pengurus masjid) memilih *delete* pemasukan keuangan, maka aplikasi akan mengirim *query* ke server.
16. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu pemasukan keuangan.
17. Lalu aplikasi menampilkan kembali *list* berita terbaru..
18. Selesai

Selanjutnya gambar berikut ini merupakan perancangan *sequence* diagram diantara admin (Pengurus Masjid) dengan sistem ketika mengelola pengeluaran keuangan masjid.

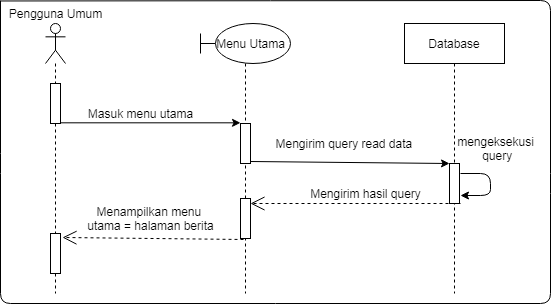


Gambar 4.12 *Sequence* Diagram 4

Penjelasan dari gambar 4.12 *sequence* diagram 4 adalah sebagai berikut:

1. Admin (pengurus masjid) memilih menu kelola keuangan saat berada di halaman admin
2. Aplikasi mengirim *query* ke server
3. Lalu server mengeksekusi *query* dan dilanjutkan dengan mengirim balik ke aplikasi untuk ditampilkan di menu pengeluaran keuangan.
4. Aplikasi menampilkan daftar pengeluaran keuangan yang aktif, dan juga terdapat beberapa opsi yaitu menambah pengeluaran keuangan baru, mengedit pengeluaran keuangan, dan menghapus pengeluaran keuangan.
5. Bila Admin (pengurus masjid) memilih tambah pengeluaran keuangan, maka menu kelola keuangan akan berpindah ke halaman form tambah pengeluaran keuangan.
6. Admin (pengurus masjid) mengisi form pengeluaran keuangan.
7. *Form* tambah pengeluaran keuangan dikirim ke server.
8. Server mengeksekusi *query*, dan mengirim balik ke aplikasi untuk ditampilkan di menu pengeluaran keuangan..
9. Lalu aplikasi menampilkan *list* pengeluaran keuangan terbaru.
10. Bila Admin (pengurus masjid) memilih *edit* pengeluaran keuangan, maka aplikasi akan menampilkan form edit pengeluaran keuangan.
11. Admin (pengurus masjid) mengisi form *edit* pengeluaran keuangan.
12. Form *edit* pengeluaran keuangan dikirim ke server.
13. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu pengeluaran keuangan.
14. Lalu aplikasi menampilkan *list* pengeluaran keuangan terbaru.
15. Bila Admin (pengurus masjid) memilih *delete* pengeluaran keuangan, maka aplikasi akan mengirim *query* ke server.
16. Server mengeksekusi *query* dan mengirim balik ke aplikasi untuk ditampilkan di menu pengeluaran keuangan.
17. Lalu aplikasi menampilkan kembali *list* berita terbaru ..
18. Selesai

Kemudian gambar berikut ini merupakan perancangan *sequence* diagram diantara pengguna umum dengan sistem ketika mengakses halaman berita.

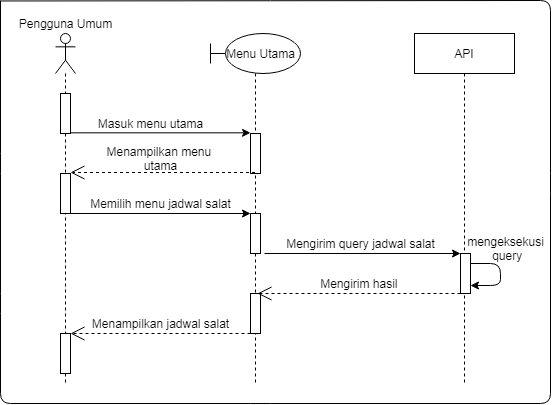


Gambar 4.13 *Sequence* Diagram 5

Penjelasan dari gambar 4.13 *sequence* diagram 5 adalah sebagai berikut:

1. Pengguna umum masuk ke dalam aplikasi
2. Aplikasi mengirim *query* ke *database*
3. *Database* mengeksekusi *query* dan mengirim hasil ke aplikasi utama
4. Menu utama menampilkan halaman berita
5. Selesai

Lalu gambar berikut ini merupakan perancangan *sequence* diagram diantara pengguna umum dengan sistem ketika mengakses menu jadwal salat.

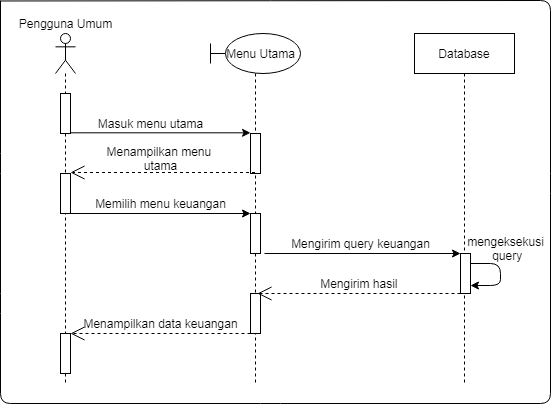


Gambar 4.14 *Sequence* Diagram 6

Penjelasan dari gambar 4.14 *sequence* diagram 6 adalah sebagai berikut:

1. Pengguna umum memilih menu jadwal salat saat berada di menu utama.
2. Aplikasi mengirim *query* jadwal salat ke API
3. API mengeksekusi dan mengirim hasil ke aplikasi
4. Aplikasi menampilkan halaman jadwal salat
5. Selesai

Kemudian gambar berikut ini merupakan perancangan *sequence* diagram diantara pengguna umum (Masyarakat/jama’ah) dengan sistem ketika mengakses menu keuangan masjid.

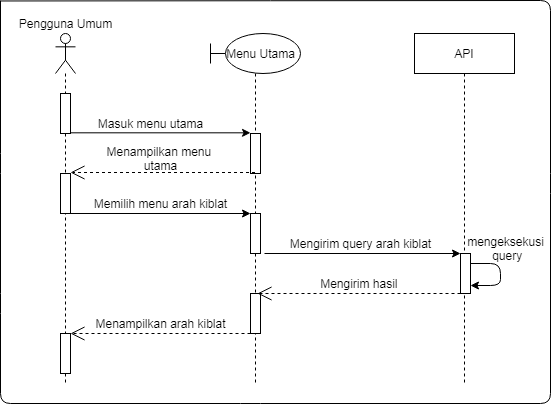


Gambar 4.15 Sequence Diagram 7

Penjelasan dari gambar 4.15 *sequence* diagram 7 diatas adalah sebagai berikut:

1. Pengguna umum memilih menu keuangan saat di menu utama
2. Aplikasi mengirim *query* ke *database*
3. *Database* mengeksekusi *query* dan mengirim hasil ke aplikasi untuk ditampilkan di menu utama
4. Menu utama menampilkan halaman keuangan masjid
5. Selesai

Selanjutnya gambar berikut ini merupakan perancangan *sequence* diagram diantara pengguna umum dengan sistem ketika mengakses halaman arah kiblat.

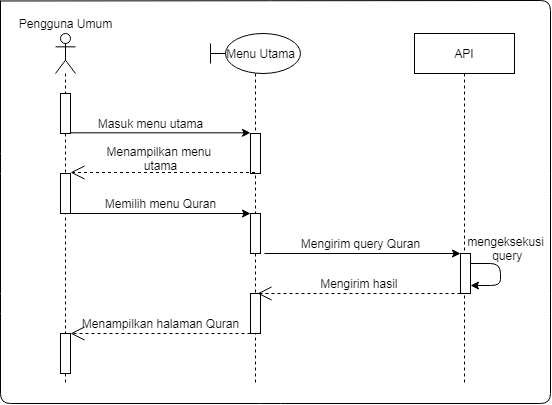


Gambar 4.16 *Sequence* Diagram 8

Penjelasan dari gambar 4.16 *sequence* diagram 8 diatas adalah sebagai berikut:

1. Pengguna umum memilih menu kiblat saat di menu utama
2. Aplikasi mengirim *query* ke API
3. API mengeksekusi *query* dan mengirim hasil ke aplikasi untuk ditampilkan
4. Menu utama menampilkan halaman arah kiblat
5. Selesai

Selanjutnya gambar berikut ini merupakan perancangan *sequence* diagram diantara pengguna umum dengan sistem ketika membuka halaman Quran.



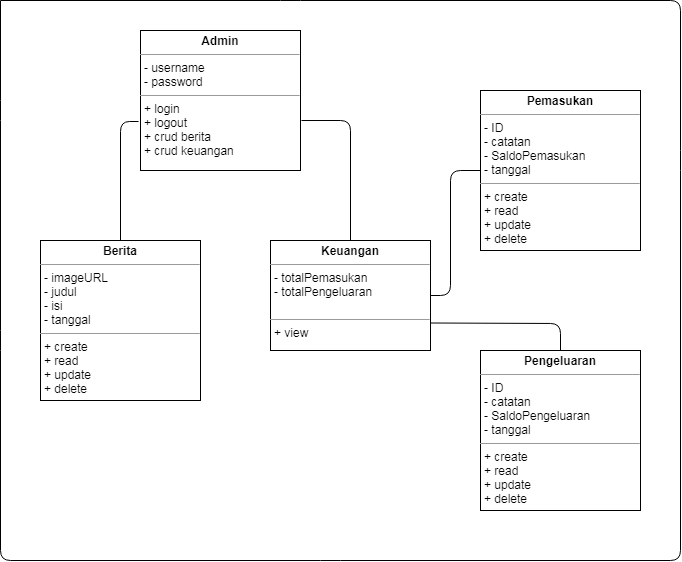
Gambar . *Sequence* Diagram 9

Penjelasan dari gambar 4.17 *sequence* diagram 9 diatas adalah sebagai berikut:

1. Pengguna umum memilih menu kiblat saat di menu utama
2. Aplikasi mengirim *query* ke API
3. API mengeksekusi *query* dan mengirim hasil ke aplikasi untuk ditampilkan di menu utama
4. Menu utama menampilkan halaman Quran
5. Selesai

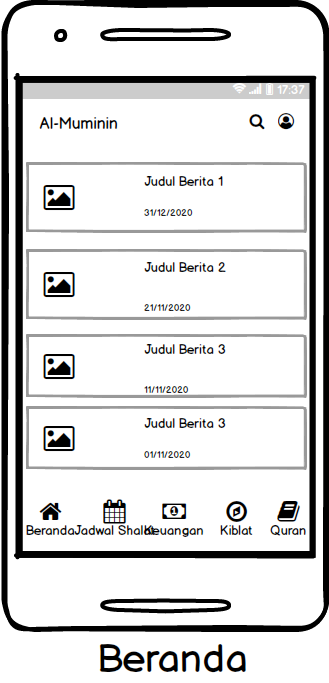
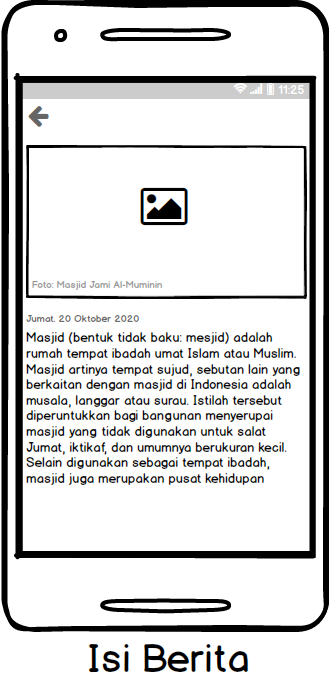
### Rancangan *Class Diagram*

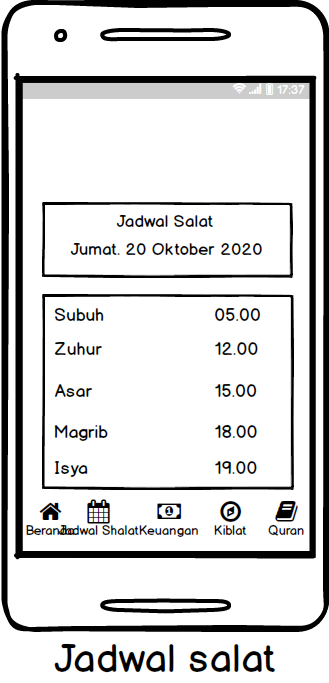
*Class* *diagram* merupakan jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukan sistem *class*, attributnya, metode, dan hubungan antar objek. Berikut ini merupakan *class diagram* dari aplikasi yang diajukan oleh peneliti.

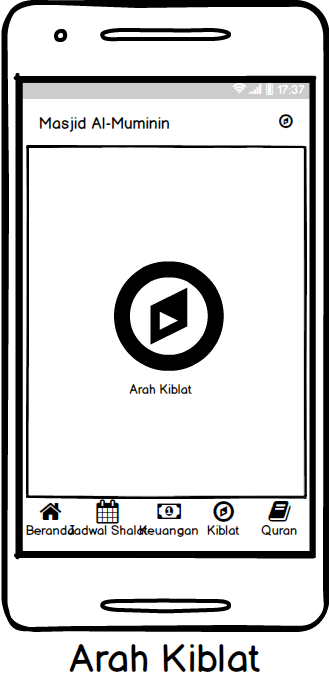


Gambar 4.18 *Class Diagram*

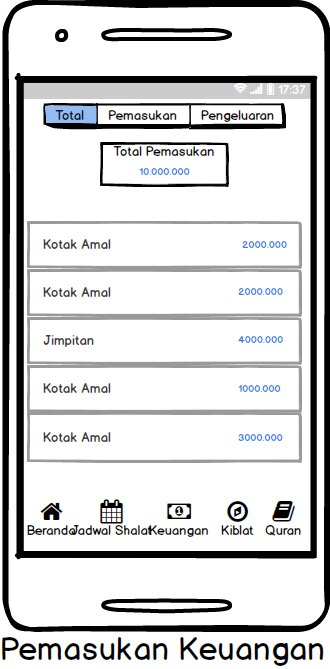
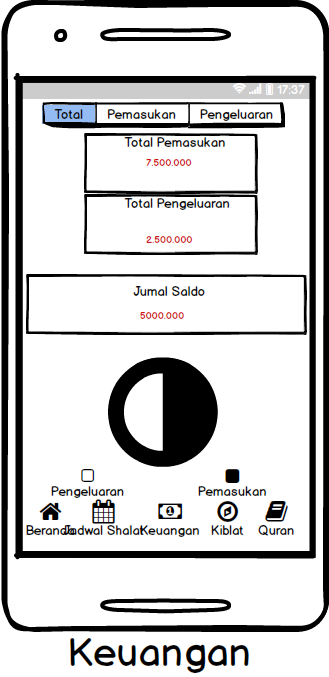
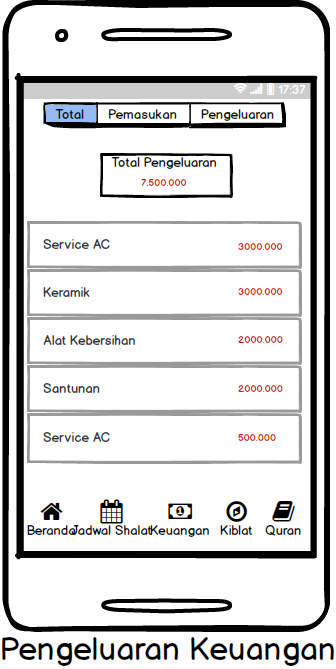
## *Mock Up* Desain Aplikasi

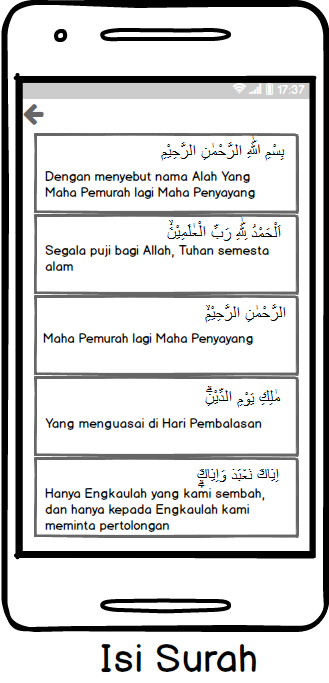
Berikut ini merupakan *final* *mockup* desain aplikasi yang diajukan oleh peneliti sebagai visualisasi konsep desain aplikasi atau sebagai gambaran nyata rancangan aplikasi.

Gambar 4.19 *Mock up* 1

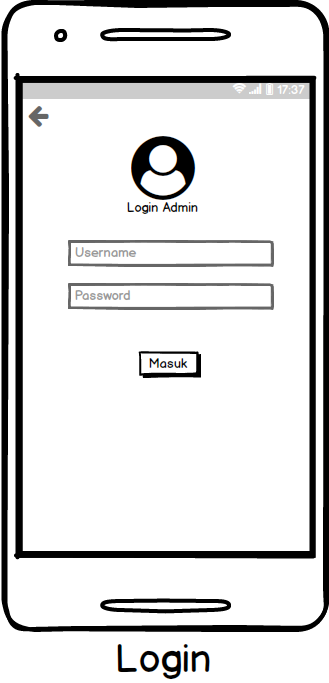
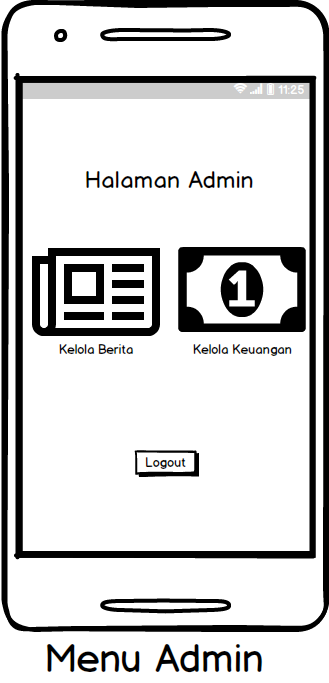


Gambar 4.20 *Mock up* 2

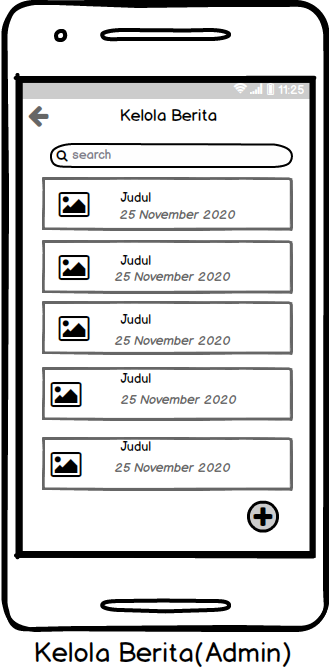


Gambar 4.21 *Mock up* 3

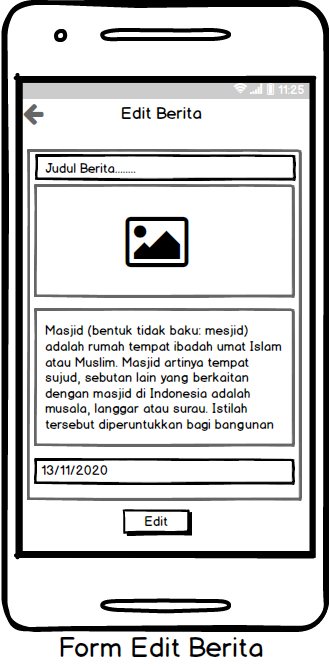
Gambar .* Mock up* 4



Gambar . *Mock up* 5

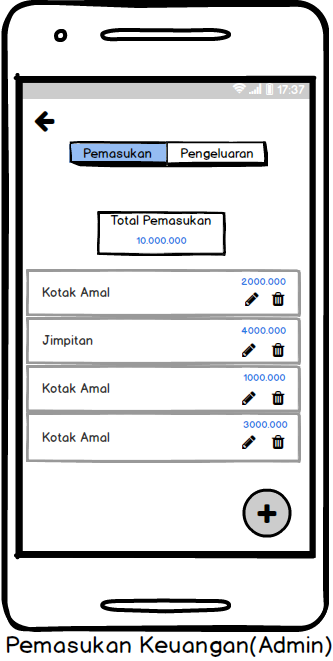


Gambar 4.24 *Mock up* 6

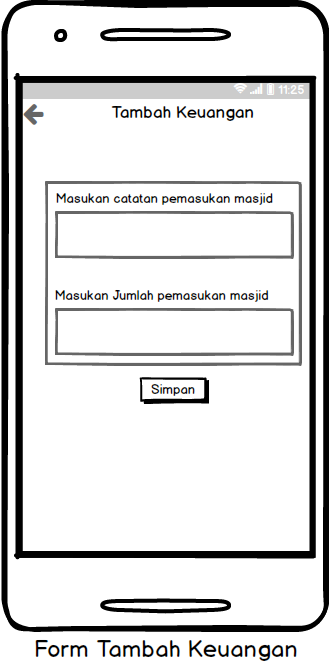
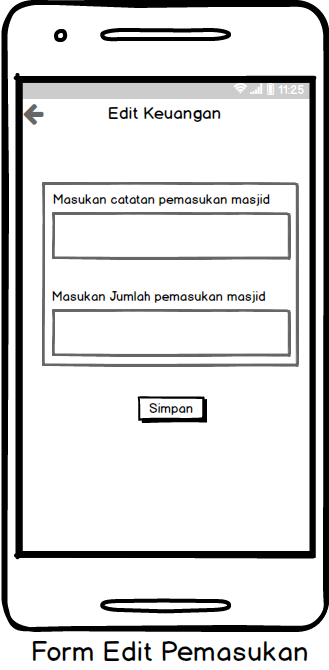


Gambar 4.25 *Mock up* 7





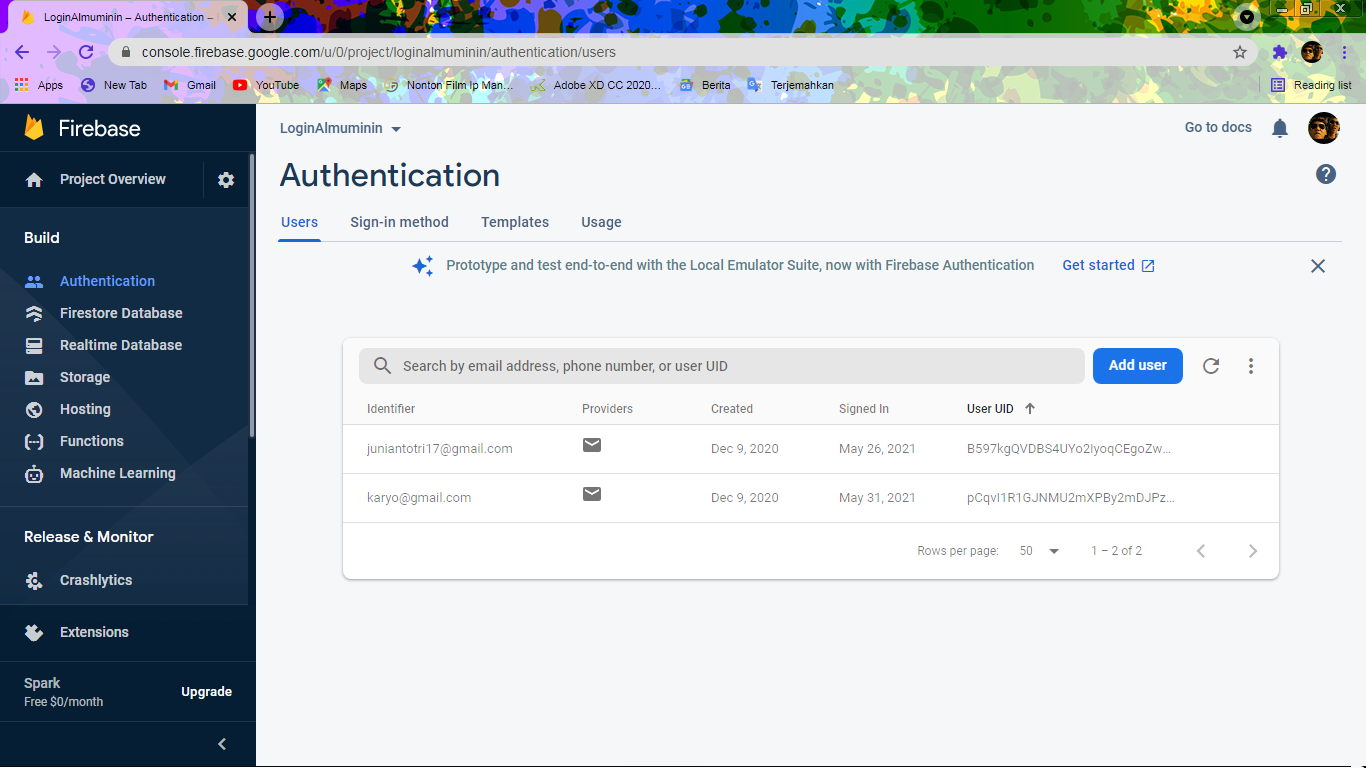
Gambar 4.26 *Mock up* 8



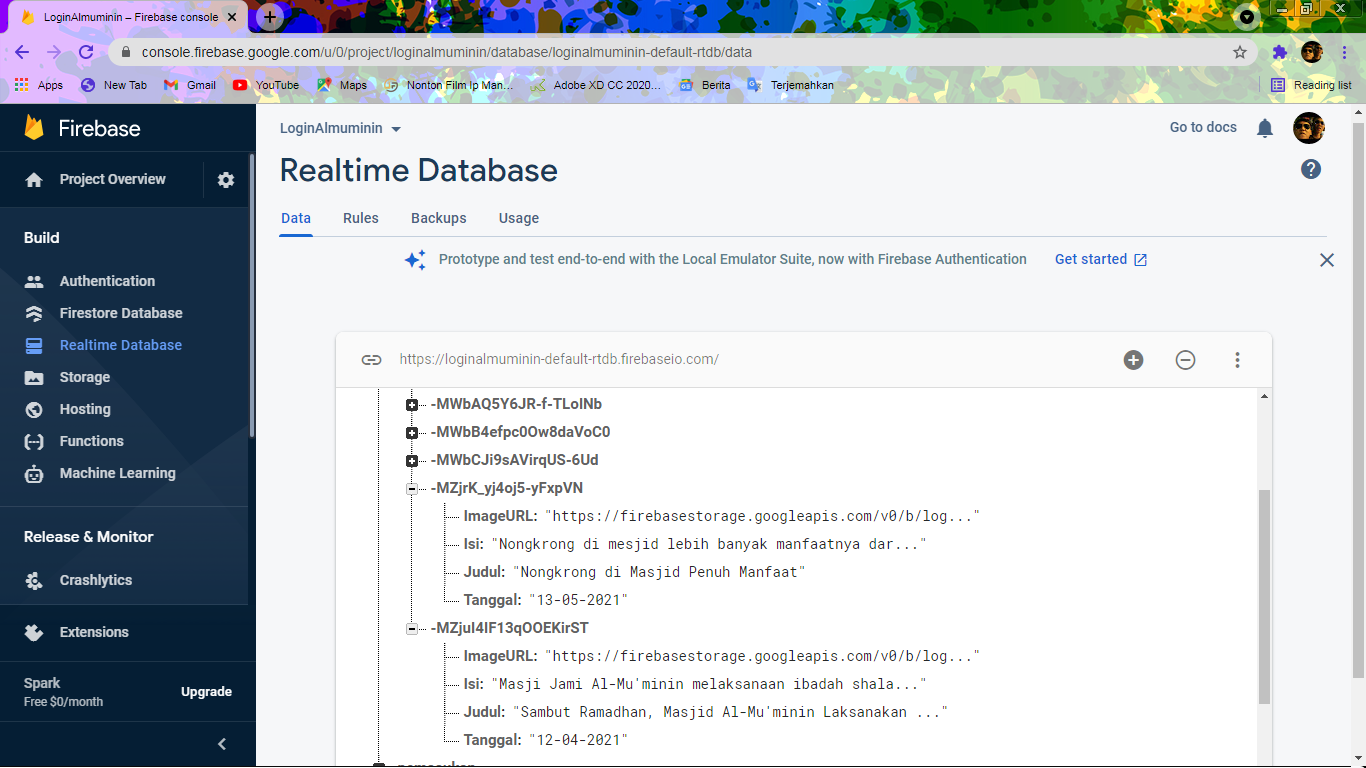
Gambar 4.27 *Mock up* 9

## Rancangan *Database*

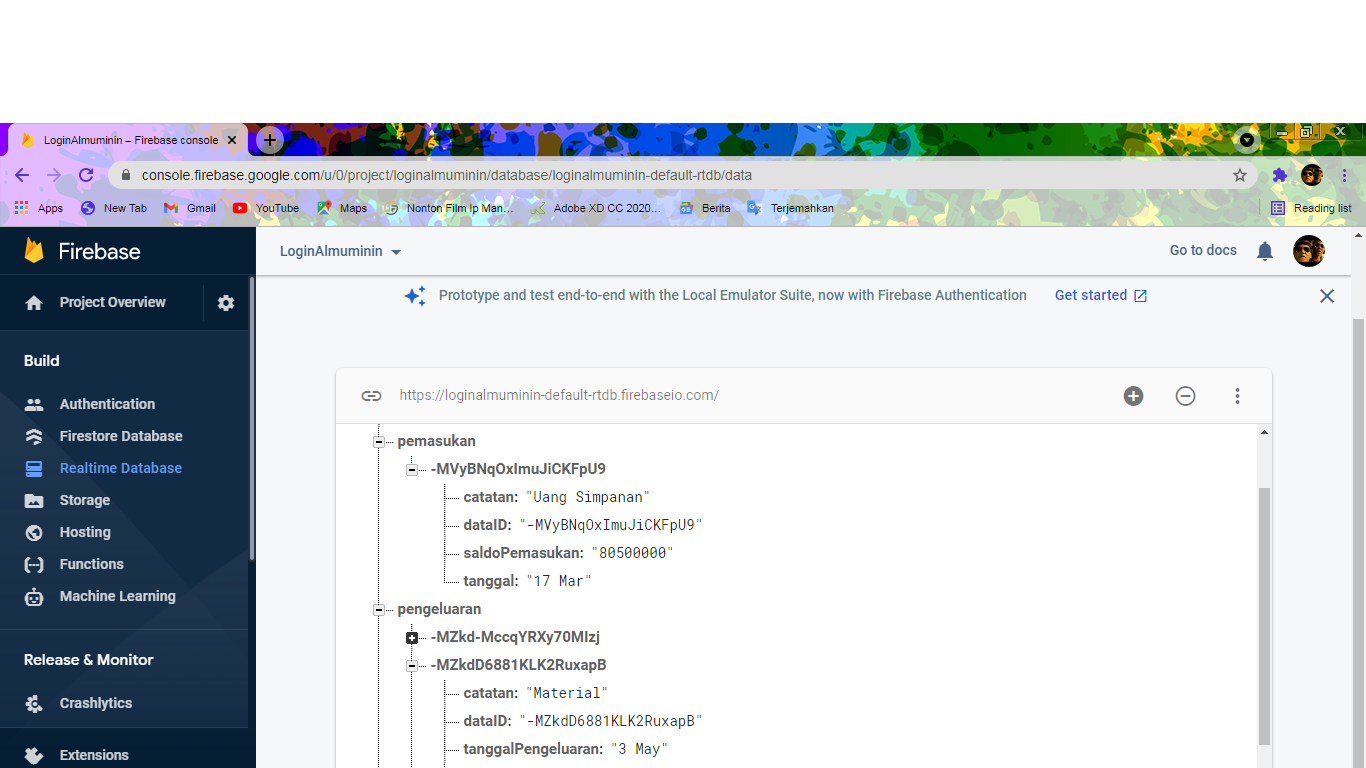
Berikut ini merupakan rancangan *database* dari aplikasi yang diajukan oleh peneliti. Disini peneliti menggunakan firebase sebagai tempat untuk penyimpanan data aplikasi.



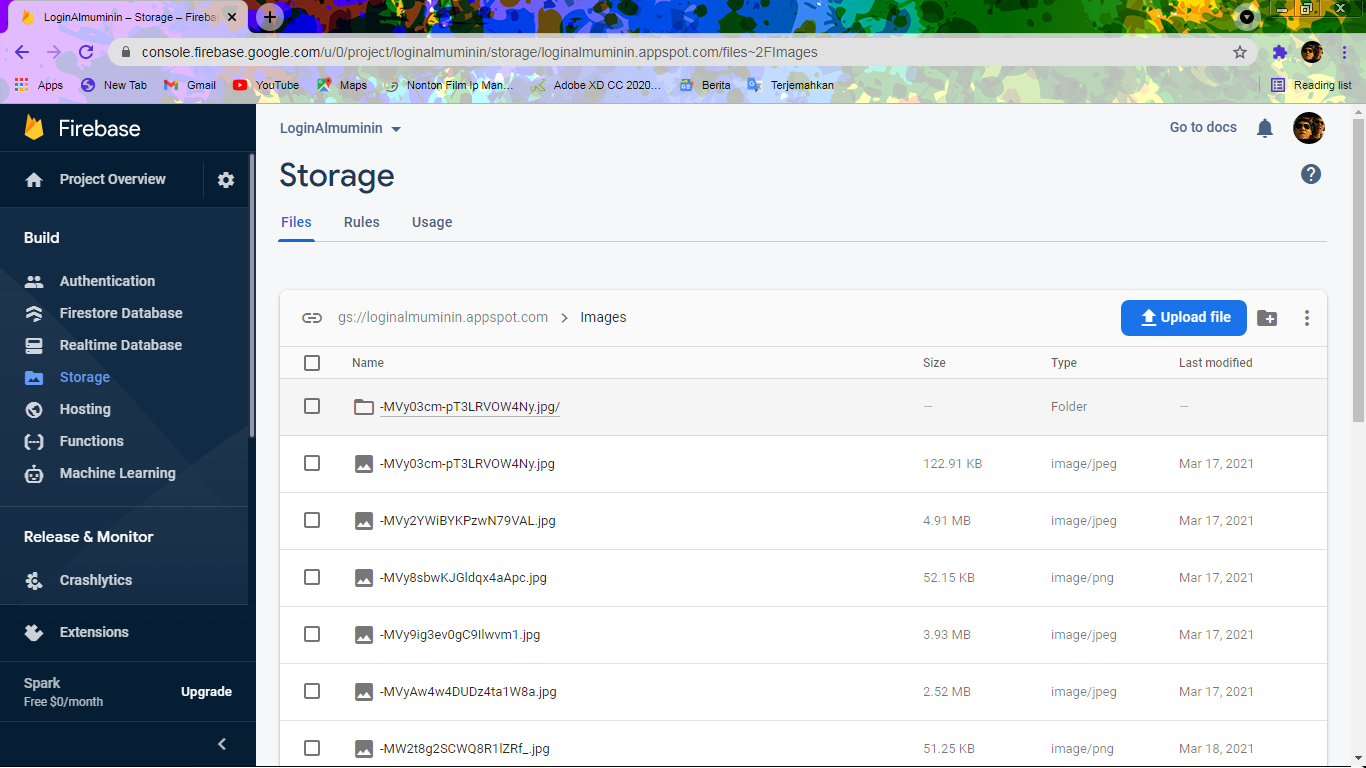
Gambar 4.28 *Database* *Firebase* Akun Admin

**

Gambar 4.29 *Database* *Firebase* Informasi Kegiatan Masjid



Gambar 4.30 *Database* *Firebase* Keuangan Masjid



Gambar 4.31 *Storage Firebase*

## Rancangan *Interface* Aplikasi

### Rancangan *Interface Icon* dan *Splash Screen* Aplikasi

Gambar 4.32 *Interface* *Icon* dan *Splash Screen* Aplikasi

Gambar diatas merupakan icon dari *aplikasi* dan *splashcreen* yang tampil ketika membuka aplikasi sebelum masuk ke dalam beranda.

### Rancangan *Interface* Beranda dan Isi Berita Aplikasi

Gambar 4.33 *Interface* Beranda dan Isi Berita Aplikasi

Gambar 4.33 merupakan tampilan dari beranda dan isi berita aplikasi. Halaman beranda aplikasi adalah tampilan utama yang muncul ketika aplikasi dibuka. Dalam halaman ini terdapat beberapa *icon* dan menu. Dimulai dari bagian atas terdapat kolom pencarian dan *icon* bergambar manusia, yang digunakan untuk masuk akun sebagai admin. Lalu ada tepat dibawahnya berada pada tengah halaman terdapat daftar berita kegiatan masjid yang dapat di *scroll* ke bawah untuk melihat berita kegiatan masjid yang lain, sedangkan bila bagian atas halaman ditarik dan tahan kebawah maka halaman ini akan di *refresh*. Lalu jika daftar kegiatan masjid tersebut di klik maka aplikasi akan membuka halaman isi berita aplikasi seperti pada Gambar 4.33 diatas. Dibagian paling bawah halaman beranda aplikasi terdapat menu navigasi untuk berpindah ke halaman jadwal salat, keuangan masjid, dan arah kiblat dan Quran. Lalu gambar yang kedua yaitu halaman isi berita. Halaman ini merupakan isi dari berita yang dipilih sebelumnya pada daftar isi berita yang terdapat pada beranda.

### Rancangan *Interface* Jadwal Salat dan Kiblat

Gambar 4.34 *Interface* Jadwal Salat dan Kiblat

Gambar diatas adalah *interface* dari jadwal salat dan kiblat pada aplikasi. Gambar pertama yaitu jadwal salat, halaman ini menampilkan jadwal salat wajib lima waktu yang otomatis diperbarui setiap harinya. Lalu yang ke-2 adalah halaman arah kiblat, fitur ini berfungsi sebagai kompas untuk menentukan letak arah kiblat. Dibagian kanan atas halaman ini terdapat icon yang dapat digunakan untuk me-*refresh* halamanarah kiblat.

### Rancangan *Interface* Halaman Quran

Gambar . *Interface* Halaman Quran

Gambar diatas adalah *interface* dari halaman Quran pada aplikasi. Gambar pertama merupakan tampilan daftar surat dan gambar kedua tampilan ketika surat di klik akan membuka isi surat.

### Rancangan *Interface* Keuangan Masjid

Gambar . Rancangan *Interface* Keuangan Masjid

Gambar 4.36 merupakan *interface* dari halaman keuangan. Di halaman keuangan ini terdapat tiga *fragement.* Yang pertama adalah *fragment* total keuangan, *fragment* ini merupakan yang pertama muncul ketika membuka halaman keuangan. Dalam *fragment* ini terdapat informasi total pemasukan, total pengeluaran, dan sisa saldo keuangan masjid serta dilengkapi dengan fitur diagram statistik keuangan masjid. Yang ke-2 adalah *fragment* pemasukan keuangan, dalam *fragment* ini berisi daftar pemasukan keuangan dengan rincian nominal serta catatan pemasukan keuangan. Lalu yang ke-3 adalah *fragment* pengeluaran keuangan, dalam *fragment* ini berisi daftar pengeluaran keuangan dengan rincian nominal serta catatan pengeluaran keuangan. Untuk berpindah antar *fragment* dapat dilakukan dengan mengusap *fragment* ke arah samping.

### Rancangan *Interface Login* Aplikasi

Gambar . Rancangan *Interface Login* Aplikasi

Gambar diatas adalah tampilan dari *login* aplikasi. Admin harus mengisi kolom *username* dan *password* dengan benar. Jika salah satu kolom tidak terisi maka akan muncul notifikasi seperti gambar yang ke-2 diatas. Jika *username* atau *password* tidak benar maka akan muncul notifikasi seperti gambar yang ke-3 diatas.

### Rancangan *Interface* Menu Admin Aplikasi

Gambar 4.38 Rancangan *Interface* Menu Admin Aplikasi

Gambar 4.38 adalah tampilan dari halaman menu admin pada aplikasi. Halaman ini hanya dapat diakses oleh pengguna yang mempunyai *username* dan *password* yaitu admin. Dalam halaman ini terdapat dua pilihan menu yaitu yang pertama kelola berita dan yang ke-2 kelola keuangan. Lalu di bawah kedua menu tersebut terdapat tombol untuk *logout* dari akun admin.

### Rancangan *Interface* Kelola Berita Admin

Gambar 4.39 *Interface* Kelola Berita Admin

Gambar diatas adalah tampilan dari halaman kelola berita admin. Pada bagian atas halaman ini terdapat kolom pencarian, lalu tepat dibawahnya terdapat daftar berita yang aktif, serta dibagian pojok kanan bawah terdapat *icon* tambah yang digunakan untuk menambahkan berita baru. Jika daftar berita diklik maka akan muncul halaman detail berita seperti yang ada pada gambar ke-2 diatas, dimana dibawahnya juga terdapat dua tombol yaitu tombol *edit* dan hapus.

### Rancangan *Interface* Form Tambah Berita dan Edit Berita

Gambar 4.40 *Interface Form* Tambah Berita dan *Edit* Berita

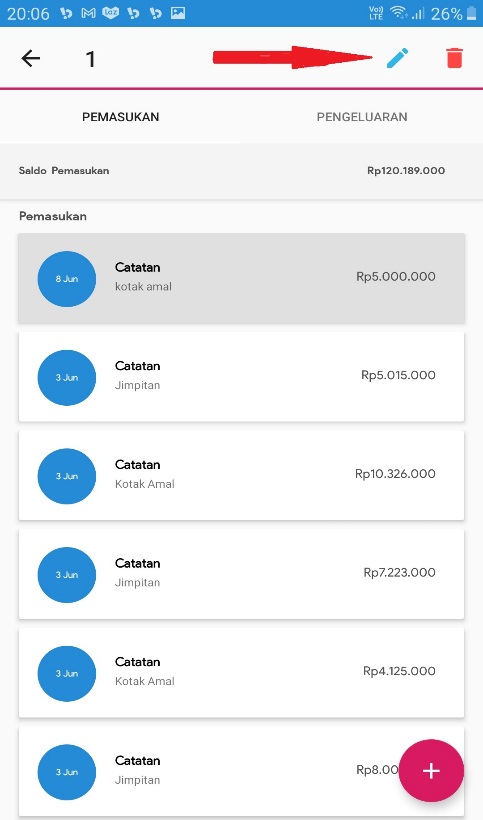
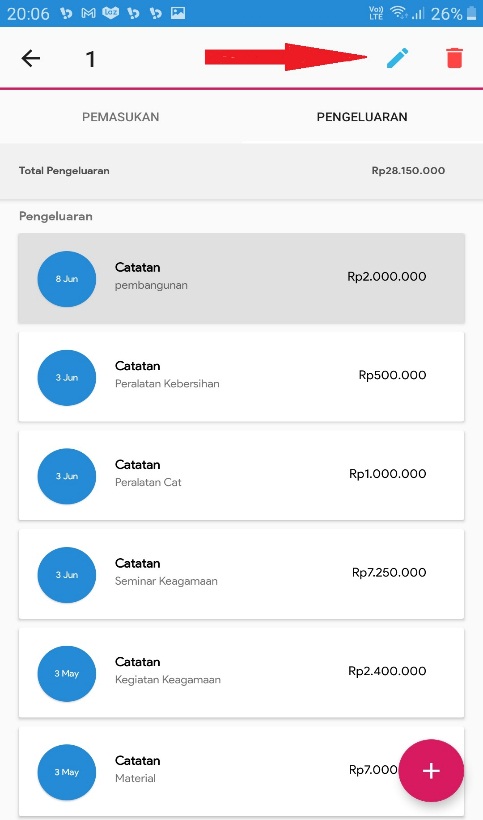
Gambar diatas adalah tampilan dari *form* tambah berita dan *edit* berita. Gambar pertama merupakan tampilan *form* tambah berita yang di dalam nya terdapat kolom-kolom yang harus diisi untuk menambahkan sebuah berita baru. Kolom-kolom yang harus diisi yaitu kolom judul berita, foto berita, isi berita, dan tanggal berita. Jika ada salah satu kolom dari form tersebut tidak terisi maka akan muncul *toast* atau notifikasi perintah untuk mengisi semua kolom yang ada, serta form tersebut juga tidak akan ter-*upload.* Sebalinya jika semua kolom pada formtersebut diisi maka data akan ter-*upload* dan berita akan muncul pada daftar berita. Lalu Gambar yang ke-2 merupakan tampilan dari form *edit* berita, sama dengan form tambah berita semua kolom yang ada pada form *edit* berita juga harus terisi, yang membedakan hanya form *edit* berita sudah terisi dengan data yang sudah ada sebelumnya, dan admin hanya perlu mengedit bagian yang dirasa perlu diperbaiki.

### Rancangan *Interface* Kelola Keuangan Admin

Gambar 4.41 *Interface* Kelola Keuangan Admin

Gambar diatas merupakan *interface* dari halaman kelola keuangan. Halaman ini adalah tempat bagi admin mengelola pemasukan dan pengeluaran keuangan. Di halaman kelola keuangan ini terdapat dua *fragement.* Yang pertama adalah pemasukan keuangan, dalam *fragment* ini berisi daftar pemasukan keuangan dengan rincian nominal serta catatan pemasukan keuangan. Jika daftar pemasukan ditekan dan tahan maka akan muncul menu opsi pada bagian atas yang berguna untuk menampilkan pilihan *edit* dan hapus daftar keuangan, serta di bagian bawah halaman pemasukan juga terdapat *icon* tambah yang berfungsi untuk menambah data pemasukan keuangan baru. Lalu yang ke-2 adalah *fragment* pengeluaran keuangan, dalam *fragment* ini berisi daftar pengeluaran keuangan dengan rincian nominal serta catatan pengeluaran keuangan. Jika daftar pengeluaran ditekan dan tahan maka akan muncul menu opsi pada bagian atas yang berguna untuk menampilkan pilihan *edit* dan hapus daftar keuangan, serta di bagian bawah halaman pengeluaran juga terdapat *icon* tambah yang berfungsi untuk menambah data pengeluaran keuangan baru.Untuk berpindah antar *fragment* dapat dilakukan dengan mengusap *fragment* ke arah samping.

### Rancangan *Interface* Ketika Tekan dan Tahan Daftar Keuangan



Gambar 4.42 *Interface* Ketika Tekan dan Tahan Daftar Keuangan

Gambar diatas adalah menu opsi lanjutan pada *fagment* pemasukan dan pengeluaran keuangan. Menu opsi lanjutan ini muncul dengan cara menekan dan tahan daftar pemasukan atau pengeluaran keuangan yang dipilih. Menu opsi lanjutan ini akan muncul pada bagian atas halaman yang di dalamnya terdapat *icon* untuk perintah *edit* dan hapus data. Admin juga dapat menghapus beberapa data keuangan sekaligus dengan cara menekan dan tahan daftar keuangan lebih dari satu lalu mengeklik *icon* hapus pada menu opsi lanjutan. .

### Rancangan *Interface Form* Tambah dan *Edit* Pemasukan Keuangan

Gambar 4.43 *Interface Form* Tambah dan *Edit* Pemasukan Keuangan

Gambar diatas adalah tampilan dari *form* tambah pemasukan keuangan dan *edit* pemasukan keuangan. Gambar pertama merupakan tampilan form tambah pemasukan keuangan yang di dalam nya terdapat kolom-kolom yang harus diisi untuk menambahkan data pemasukan keuangan baru. Kolom-kolom yang harus diisi yaitu kolom catatan pemasukan keuangan serta nominal saldo pemasukan keuangan. Jika ada salah satu kolom dari *form* tersebut tidak terisi maka akan muncul *toast* atau notifikasi perintah untuk mengisi semua kolom yang ada, serta form tersebut juga tidak akan ter-*upload.* Sebaliknya jika semua kolom pada formtersebut diisi maka data akan ter-*upload* dan data pemasukan keungan baru akan muncul pada daftar pemasukan keuangan. Lalu Gambar yang ke-2 merupakan tampilan dari *form edit* pemasukan keuangan, sama dengan *form* tambah pemasukan keuangan semua kolom yang ada pada form *edit* pemasukan keuangan juga harus terisi, yang membedakan hanya *form* *edit* pengeluaran keuangan sudah terisi dengan data yang sudah ada sebelumnya, dan admin hanya perlu mengedit bagian yang dirasa perlu diperbaiki.

### Rancangan *Interface Form* Tambah dan *Edit* Pengeluaran Keuangan

Gambar 4.44 *Interface* FormTambah dan *Edit* Pengeluaran Keuangan

Gambar diatas adalah tampilan dari *form* tambah pengeluaran keuangan dan *edit* pengeluaran keuangan. Gambar pertama merupakan tampilan *form* tambah pengeluaran keuangan yang di dalam nya terdapat kolom-kolom yang harus diisi untuk menambahkan data pengeluaran keuangan baru. Kolom-kolom yang harus diisi yaitu kolom catatan pengeluaran keuangan serta nominal saldo pengeluaran keuangan. Jika ada salah satu kolom dari form tersebut tidak terisi maka akan muncul *toast* atau notifikasi perintah untuk mengisi semua kolom yang ada, serta *form* tersebut juga tidak akan ter-*upload.* Sebaliknya jika semua kolom pada *form* tersebut diisi maka data akan ter-*upload* dan data pengeluaran keungan baru akan muncul pada daftar pengeluaran keuangan. Lalu Gambar yang ke-2 merupakan tampilan dari *form edit* pengeluaran keuangan, sama dengan *form* tambah pengeluaran keuangan semua kolom yang ada pada *form* edit pengeluaran keuangan juga harus terisi, yang membedakan hanya form *edit* pengeluaran keuangan sudah terisi dengan data yang sudah ada sebelumnya, dan admin hanya perlu mengedit bagian yang dirasa perlu diperbaiki.

## Uji Coba

Setelah aplikasi selesai dibuat, maka yang tahapan selanjutnya yang diperlukan yaitu tahap uji coba untuk memastikan agar aplikasi bebas dari kesalahan fungsi. Uji coba fungsi dilakukan dengan menggunakan metode *blackbox* dan *User Acceptance Test* (UAT) dapat mendefinisikan skenario-skenario yang terdapat di dalam aplikasi berserta hasil yang diharapkan dari masing-masing skenario tersebut.

### Pengujian *BlackBox*

Berikut ini merupakan tabel uji coba *blackbox* terhadap aplikasi.

Tabel 4.8 Tabel Uji Coba *Blackbox* terhadap aplikasi

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Skenario yang diuji** | **Hasil yang diharapkan** | **Hasil** |
| 1 | Membuka aplikasi | Menampilkan *Splah Screen* | Sesuai |
| 2 | Menu utama | Menampilkan menu aplikasi berjumlah 6 menu. | Sesuai |
| 3 | *Icon* gambar jadwal | Menampilkan halaman jadwal salat | Sesuai |
| 4 | *Icon* gambar uang | Menampilkan halaman keuangan masjid | Sesuai |
| 5 | *Icon* gambar kiblat | Menampilkan halaman arah kiblat | Sesuai |
| 6 | *Icon* gambar Quran | Menampilkan halaman Quran | Sesuai |
| 7 | *Icon* gambar orang | Menampilkan halaman *login* admin | Sesuai |
| 8 | Mengetik di kolom pencarian | Menampilkan hasil pencarian sesuai *keyword* | Sesuai |
| 9 | Klik daftar berita | Menampilkan halaman isi berita | Sesuai |
| 10 | Klik tab pemasukan di halaman keuangan | Menampilkan rincian pemasukan keuangan | Sesuai |
| 11 | Klik tab pengeluaran di halaman keuangan | Menampilkan rincian pengeluaran keuangan | Sesuai |
| 12 | Klik tab total di halaman keuangan | Menampilkan rincian total saldo keuangan | Sesuai |
| 13 | Klik *icon* GPS di halaman kiblat | Me-*refresh* arah kiblat | Sesuai |
| 14 | *Scroll* ke atas daftar berita | Menggulung daftar berita kebawah | Sesuai |
| 15 | Menarik bagian atas menu utama | Me-*refresh* halaman berita | Sesuai |
| 16 | Tekan tombol *back* satu kali di menu utama | Muncul notifikasi “Tekan sekali lagi untuk keluar” | Sesuai |
| 17 | Tekan tombol *back* dua kali di menu utama | Keluar dari aplikasi | Sesuai |
| 18 | Tidak mengisi kolom username atau password saat *login* | Muncul notifikasi ”*username* atau *password* belum terisi” sesuai kolom. | Sesuai |
| 19 | Salah saat mengisi *username* atau *password* saat *login* | Muncul notifikasi ”*username* atau *password* salah”. | Sesuai |
| 20 | Mengisi *username* dan *password* dengan benar saat *login* | Masuk ke menu admin | Sesuai |
| 21 | Masuk menu admin | Menampilkan halaman Menu admin di dalamnya terdapat 2 fitur | Sesuai |
| 22 | Klik tombol kelola berita di menu admin | Menampilkan halaman kelola berita yang berisi daftar berita | Sesuai |
| 23 | Klik tombol tambah di halaman kelola berita. | Menampilkan form tambah berita | Sesuai |
| 24 | Tidak mengisi semua kolom yang ada pada form tambah berita | Muncul notifikasi “semua kolom harus” diisi. | Sesuai |
| 25 | Mengisi semua kolom yang ada pada form tambah berita | Meng-*upload* berita baru | Sesuai |
| 26 | Klik tombol *back* pada form tambah berita | Kembali ke halaman kelola berita | Sesuai |
| 27 | Klik daftar berita di halaman kelola berita | Menampilkan halaman detail berita serta menu *edit* dan *hapus* | Sesuai |
| 28 | Klik *icon* *edit* pada halaman detail berita | Menampilkan form *edit* berita | Sesuai |
| 29 | Klik *icon* hapus pada detail berita | Menghapus daftar berita | Sesuai |
| 30 | Tekan tombol *back* pada form *edit* berita | Kembali ke halaman detail berita | Sesuai |
| 31 | Tekan tombol *back* pada halaman detail berita | Kembali ke halaman kelola berita | Sesuai |
| 32 | Tekan tombol *back* pada halaman kelola berita | Kembali ke halaman menu admin | Sesuai |
| 33 | Klik tombol kelola keuangan di menu admin | Menampilkan menu halaman keuangan | Sesuai |
| 34 | Klik tab pemasukan di halaman keuangan | Masuk ke halaman pemasukan keuangan yang berisi daftar pemasukan | Sesuai |
| 35 | Klik *icon* tambah pada halaman pemasukan keuangan | Menampilkan menu form tambah keuangan | Sesuai |
| 36 | Mengisi semua kolom form tambah pemasukan | Berhasil menambah data pemasukan keuangan | Sesuai |
| 37 | Tidak mengisi semua kolom form tambah keuangan | Akan ada notifikasi “semua kolom harus terisi” | Sesuai |
| 38 | Tekan tombol *back* pada form tambah pemaukan keuangan | Kembali ke halaman pemasukan keuangan | Sesuai |
| 39 | Tekan dan tahan daftar pemasukan keuangan | Menampilkan *toolbar* *edit* dan hapus data pemasukan keuangan | Sesuai |
| 40 | Tekan *icon* *edit* di *toolbar* pemasukan | Menampilkan form *edit* pemasukan keuangan | Sesuai |
| 41 | Mengisi semua kolom form *edit* pemasukan keuangan | Berhasil meng-*edit* data pemasukan keuangan | Sesuai |
| 42 | Tekan tombol *back* pada form *edit* pemasukan keuangan | Kembali ke halaman daftar pemasukan keuangan | Sesuai |
| 43 | Klik tombol hapus di *toolbar* pemasukan | Menghapus data pemasukan keuangan | Sesuai |
| 44 | Klik tab pengeluaran di halaman kelola keuangan | Masuk ke halaman pengeluaran keuangan yang berisi daftar pengeluaran keuangan | Sesuai |
| 45 | Klik *icon* tambah pada halaman pengeluaran keuangan | Menampilkan menu form tambah pengeluaran keuangan | Sesuai |
| 46 | Mengisi semua kolom form tambah pengeluaran keuangan | Berhasil menambah data pengeluaran keuangan | Sesuai |
| 47 | Tidak mengisi semua kolom form tambah pengeluaran keuangan | Akan ada notifikasi “semua kolom harus terisi” | Sesuai |
| 48 | Tekan tombol *back* pada form tambah pengeluaran keuangan | Kembali ke halaman pengeluaran keuangan | Sesuai |
| 49 | Tekan dan tahan daftar pengeluaran keuangan | Menampilkan *toolbar* *edit* dan hapus data pengeluaran keuangan | Sesuai |
| 50 | Tekan *icon* *edit* di *toolbar* pengeluaran keuangan | Menampilkan form *edit* pengeluaran keuangan | Sesuai |
| 51 | Mengisi semua kolom form *edit* pengeluaran keuangan | Berhasil meng-*edit* data pengeluaran keuangan | Sesuai |
| 52 | Tekan tombol *back* pada form *edit* pengeluaran keuangan | Kembali ke halaman daftar pengeluaran keuangan | Sesuai |
| 53 | Klik tombol hapus di *toolbar* pengeluaran | Menghapus data pengeluaran keuangan | Sesuai |
| 54 | Klik tab total di halaman keuangan | Menampilkan total saldo keuangan masjid | Sesuai |
| 55 | Tekan tombol *back* pada halaman keuangan | Menampilkan menu admin | Sesuai |
| 56 | Klik tombol *logout* di menu admin | Keluar dari akun dan masuk ke beranda | Sesuai |

Uji coba tersebut dilakukan terhadap fungsi-fungsi yang terdapat di dalam aplikasi. Uji coba dilaksanakan sampai dengan level pengaksesan fitur, dimana ini berguna untuk memastikan agar fungsi aplikasi sesuai dengan yang diharapkan. Sub-fitur yang terdapat di dalam fitur utama juga dilakukan uji coba agar proses dan hasil dari sebuah fitur dapat terlihat. Uji coba ini dilakukan untuk mencoba fungsi-fungsi aplikasi agar kesalahan yang mungkin terjadi dapat terlihat oleh pengembang saat mode *debug* (uji coba) sedang dieksekusi. Akan tetapi berdasarkan hasil uji coba tersebut tidak ditemukan kesalahan-kesalahan baik dalam semantik maupun sintaksis karena fungsi-fungsi aplikasi sudah sesuai dengan harapan sehingga memungkinkan untuk diimplementasikan.

### Pengujian *User Acceptance Test* (UAT)

Pengujian UAT dilakukan agar mendapatkan respon dan *feedback* dari pengguna mengenai sistem yang telah dibuat. Pengujian *user* ini dengan cara menberikan 5 peryataan yang kemudian akan diisi oleh pengguna sistem informasi manajemen masjid untuk memperoleh tanggapan tentang sistem ini. Pertanyaan pengujian sistem informasi manajemen Masjid Jami’ Al Mu’minin dapat dilihat pada berikut.

Tabel . Bobot Nilai Jawaban

|  |  |  |
| --- | --- | --- |
| Nilai | Jawaban | Bobot |
| A | Sangat Bagus | 5 |
| B | Bagus | 4 |
| C | Cukup | 3 |
| D | Kurang | 2 |
| E | Sangat Kurang | 1 |

Tabel . Pertanyaan dan Data Jawaban

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NO | Pertanyaan |  | Respon | | |  |
| A | B | C | D | E |
| 1 | Semua fungsi dan fitur aplikasi berjalan dengan baik | 3 | 4 | 3 | 0 | 0 |
| 2 | Aplikasi mudah dimengerti dan mudah untuk digunakan. | 2 | 6 | 2 | 0 | 0 |
| 3 | Aplikasi mempunyai tampilan yang menarik dan tidak membosankan bagi pengguna | 1 | 4 | 5 | 0 | 0 |
| 4 | Aplikasi bermanfaat dan membantu masyarakat untuk memperoleh informasi tentang masjid | 1 | 5 | 4 | 0 | 0 |
| 5 | Aplikasi layak untuk digunakan masyarakat. | 1 | 6 | 3 | 0 | 0 |

Data yang didapat di atas diolah dengan cara mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan sesuai dengan tabel bobot nilai jawaban. Dari hasil perhitungan dengan mengalikan setiap jawaban bobot yang sudah ditentukan maka didapat hasil sebagai berikut.

Tabel . Pengolahan Data Jawaban

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO | Pernyataan |  | Nilai | | |  | Jumlah |
| Ax5 | Bx4 | Cx3 | Dx2 | Ex1 |
| 1 | Semua fungsi fitur menu aplikasi berjalan dengan baik | 15 | 16 | 9 | 0 | 0 | 40 |
| 2 | Aplikasi mudah dimengerti dan mudah untuk digunakan. | 10 | 24 | 6 | 0 | 0 | 40 |
| 3 | Aplikasi mempunyai tampilan yang menarik dan tidak membosankan bagi pengguna. | 5 | 16 | 15 | 0 | 0 | 36 |
| 4 | Aplikasi bermanfaat dan membantu masyarakat untuk memperoleh informasi tentang masjid. | 5 | 20 | 12 | 0 | 0 | 37 |
| 5 | Aplikasi layak untuk digunakan masyarakat. | 5 | 24 | 9 | 0 | 0 | 38 |

* 1. Analisa pertanyaan pertama

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 40. Nilai rata-ratanya adalah 40/10 = 4.0 Prosentase nilainya adalah 4.0/5 x 100 = 80%.

* 1. Analisa pertanyaan kedua

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 40. Nilai rata-ratanya adalah 40/10 = 4.0 Prosentase nilainya adalah 4.0/5 x 100 = 80%.

* 1. Analisa pertanyaan ketiga

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 40. Nilai rata-ratanya adalah 36/10 = 3.6 Prosentase nilainya adalah 3.6/5 x 100 = 72%.

* 1. Analisa pertanyaan keempat

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 37. Nilai rata-ratanya adalah 37/10 = 3.7 Prosentase nilainya adalah 3.7/5 x 100 = 74%.

* 1. Analisa pertanyaan kelima

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 40. Nilai rata-ratanya adalah 38/10 = 3.8 Prosentase nilainya adalah 4. 3.8/5 x 100 = 76%.

Dari data diatas dapat disimpulkan bahwa *Aplikasi berbasis Android Untuk Manajemen Informasi di Masjid Jami’ Al Mu’mimin* berjalan dengan baik, mudah dimengerti, mempunyai tampilan yang cukup baik, bermanfaat bagi masyarakat, dan layak untu digunakan.

## Perbandingan Pengelolaan Informasi Masjid

Berikut adalah tabel analisis perbandingan pengelolaan informasi di Masjid Jami’ Al Mu’minin sebelum dan sesudah terciptanya aplikasi.

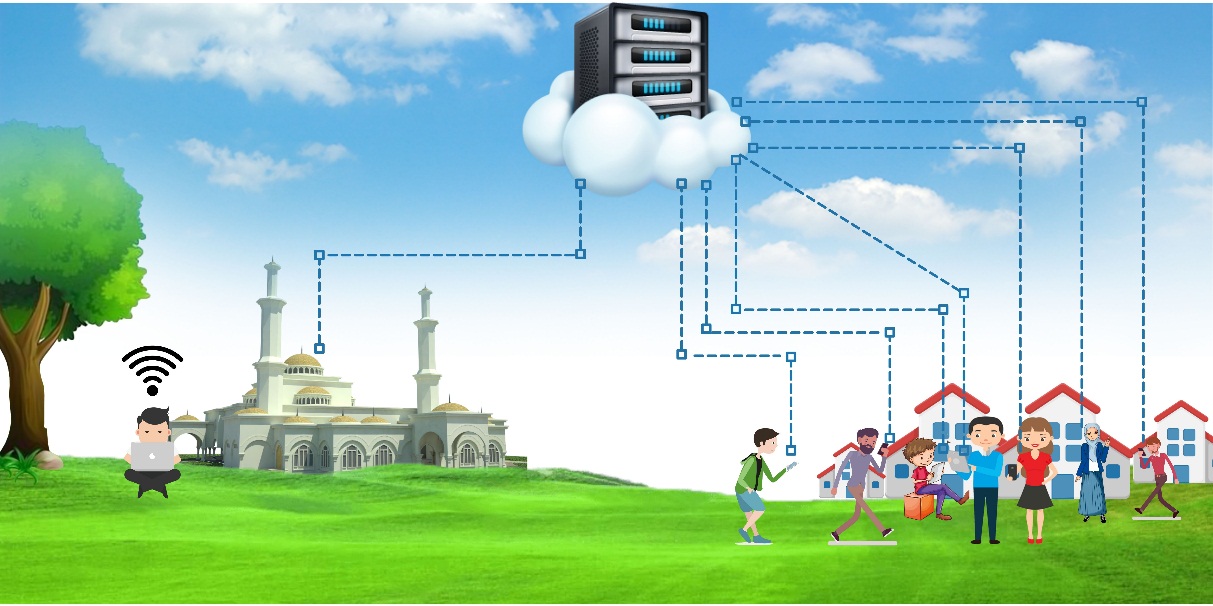
Tabel . Pengelolaan Informasi di Masjid Sebelum Terciptanya Aplikasi

|  |  |  |
| --- | --- | --- |
| **No** | **Aktifitas** | **Metode Yang Digunakan** |
| 1 | Pencatatan pengelolaan informasi masjid | Dicatat menggunakan kertas |
| 2 | Penyampaian pengelolaan informasi masjid | Disampaikan menggunakan pengeras suara dan papan pengumuman |
| 3 | Pencatatan pengelolaan keuangan masjid | Dicatat menggunakan kertas |
| 4 | Penyampaian pengelolaan keuangan masjid | Disampaikan menggunakan pengeras suara saat sela-sela salat jumat |
| 5 | Penyampaian jadwal pengajian masjid | Disampaikan saat kegiatan sebelumnya |
| 6 | Penyampaian jadwal salat harian | Ditempelkan pada papan pengumuman |

Tabel . Pengelolaan Informasi di Masjid Setelah Terciptanya Aplikasi

|  |  |  |
| --- | --- | --- |
| **No** | **Aktifitas** | **Metode Yang Digunakan** |
| 1 | Pencatatan pengelolaan informasi masjid | Menggunakan aplikasi |
| 2 | Penyampaian pengelolaan informasi masjid | Menggunakan aplikasi |
| 3 | Pencatatan pengelolaan keuangan masjid | Menggunakan aplikasi |
| 4 | Penyampaian pengelolaan keuangan masjid | Menggunakan aplikasi |
| 5 | Penyampaian jadwal pengajian masjid | Menggunakan aplikasi |
| 6 | Penyampaian jadwal salat harian | Menggunakan aplikasi |

## Ilustrasi Penggunaan Aplikasi Masjid

Berikut ini adalah ilustrasi penggunaan aplikasi berbasis android untuk manajemen informasi di Masjid Jami’ Al Mu’minin yang menggunakan firebase *database*.

Gambar . Ilustrasi Penggunaan Aplikasi Masjid

## Studi Literatur

Berikut ini adalah jurnal yang digunakan sebagai literatur oleh peneliti untuk mendukung topik tugas akhir yang diangkat oleh peneliti.

Tabel 4.14 Studi Literatur

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Peneliti** | **Judul Penelitian** | **Hasil Penelitian** |
| 1 | 1 Anjasmara  2 Linda Marlinda  3 Ahmad Fauzi  Teknik Informatika  STMIK  Nusa Mandiri | Sistem Informasi Pengelolaan Masjid Online Menggunakan Metode Waterfall | Hasil dari penelitian ini adalah menciptakan sebuah aplikasi berbasis web yang bertujuan untuk mengelola informasi terkait kegiatan keagamaan dan keuangan masjid |
| 2 | 1 Amarudin,  2 Agung Sofiandri  1 Teknik Elektro  Universitas Teknorat Indonesia  2 Sistem Informasi Universitas Teknorat Indonesia | Perancangan dan Implementasi Aplikasi Ikhtisar Kas Masjid Istiqomah Berbasis Dekstop | Hasil dari penelitian ini adalah menciptakan sebuah aplikasi berbasis dekstop yang bertujuan untuk mengelola catatan pemasukan dan pengeluaran keuangan Masjid Istiqomah. |
| 3 | 1 Mazuda Mohamad Zaki  2 Nurezayana Zainal  Faculty of Computer Science and Information Technology  Universiti Tun Hussein Onn Malaysia | Fitri Tomyam Seafood Menu Order Android Mobile Application | Hasil dari penelitian ini adalah membangun aplikasi berbasis *mobile* yang dikembangkan dengan tujuan untuk membantu proses bisnis dalam aspek *delivery* order untuk restoran Fitri Tomyam Seafood |

# KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan pembahasan yang telah diuraikan pada bab sebelumnya, maka dapat disimpulkan sebagai berikut.

1. Proses pengelolaan informasi kegiatan dan keuangan di Masjid Jami’ Al Mu’minin pada mulanya masih menggunakan metode manual yaitu dengan menempelkan informasi di papan pengumuman dan melalui pengeras suara. Dengan adanya *Aplikasi Berbasis Android untuk Manajemen Informasi di Masjid Jami’ Al Mu’minin* dapat meningkatkan pengelolaan informasi bagi pengurus masjid dan membantu masyarakat dalam mencari informasi seputar masjid.
2. Aplikasi untuk manajemen informasi masjid ini berbasis android menggunakan bahasa pemrograman Java dan XML, serta menggunakan database firebase. Metode pengembangan yang digunakan adalah metode prototype. Metode pengumpulan data menggunakan studi literatur, observasi, dan wawancara. Untuk memastikan setiap fungsi pada aplikasi berjalan dengan baik dilakukan uji coba *blackbox*. Aplikasi ini menampilkan informasi kegiatan dan informasi keuangan yang membantu masyarakat menerima informasi serta meningkatkan transparasi keuangan masjid. Aplikasi ini juga dilengkapi dengan jadwal salat, kiblat, serta Quran untuk meningkatkan kepuasan pengguna.

## Saran

Diharapkan aplikasi yang telah dibuat ini menjadi langkah awal dalam pengembangan teknologi informasi yang berdaya guna bagi Masjid Jami’ Al Mu’minin. Diantara langkah kedepan yang dapat dilakukan adalah membuat interface yang lebih menarik, layanan interaktif, notifikasi aplikasi, serta beberapa fitur lain yang bermanfaat serta fleksibel bagi jama’ah Masjid Jami’ Al Mu’minin.

# DAFTAR REFERENSI

[1] C. Rachmijati, “Penggunaan Internet Sebagai Optimalisasi Media Pembelajaran Bahasa Inggris (Program Pengabdian Pada Masyarakat Di Desa Margaluyu Kecamatan Cipendeuy),” *Abdimas Siliwangi*, vol. 1, no. 2, p. 61, 2018, doi: 10.22460/as.v1i2p61-74.106.

[2] Intan Trivena Maria Daeng, N. . Mewengkang, and E. R. Kalesaran, “91161-ID-penggunaan-smartphone-dalam-menunjang-ak,” *e-journal “Acta Diurna,”* vol. 1, no. 1, pp. 1–15, 2017.

[3] Putra, “PENGERTIAN ANDROID: Sejarah, Kelebihan & Versi Sistem Operasi Android OS,” *Salamadian.Com*, 2019. https://salamadian.com/pengertian-android/.

[4] D. Tanaya, “Hubungan Smartphone Addiction Dengan Prokrastinasi Akademik Pada Siswa / Siswi Pengguna Smartphone Di Sma N 105 Jakarta,” *J. Educ. Soc. Stud.*, pp. 1–25, 2017.

[5] D. Koswara and A. Agustinus, “DENGAN TEKNOLOGI NFC PADA SMARTPHONE ANDROID,” vol. 16, no. 3, pp. 275–281, 2017.

[6] D. Android, “Membuat Apa Saja di Android,” *Developer Android*, 2020. ttp://developer.android.com/about/versions/index.html.

[7] F. B. Zulmi, “Sistem Pendukung Keputusan Pemilihan Pegawai Berprestasi Pada Badan Kepegawaian Daerah (BKD) Binjai Menggunakan Metode Profile Matching Berbasis Android.” 2019.

[8] D. Android, “Codenames, Tags, and Build Numbers,” p. 1, 2021, [Online]. Available: https://source.android.com/setup/start/build-numbers.

[9] B. T. Mahardika, “Analisis Dan Perancangan Sistem Informasi Managemen Masjid Berbasis Web ( Studi Kasus : Yayasan Masjid Darul Maarif Daaima Jakarta ),” pp. 1–12, 2018, [Online]. Available: http://ejournal.urindo.ac.id/index.php/TI/article/view/307.

[10] M. Iqbal, G. I. Marthasari, and I. Nuryasin, “Penerapan Metode UCD (User Centered Design) pada Perancangan aplikasi Darurat Berbasis Android,” *J. Repos.*, vol. 2, no. 2, p. 201, 2020, doi: 10.22219/repositor.v2i2.221.

[11] Y. Arifin, B. Handoko, and V. K. Nurtanio, “Aplikasi Game Quiz Animals Berbasis Windows 8,” *ComTech Comput. Math. Eng. Appl.*, vol. 4, no. 2, p. 757, 2013, doi: 10.21512/comtech.v4i2.2509.

[12] Alexandrome Lawrance, “API: Pengertian, Fungsi, dan Cara Kerjanya,” *NIAGAHOSTER*, 2020. https://www.niagahoster.co.id/blog/api-adalah/.

[13] M. Seminar and K. Praktek, “Perancangan Database Untuk Aplikasi Sistem Informasi Absensi,” pp. 15–23, 2013.

[14] E. S. Laksono and I. H. Al Amin, “Penerapan NoSQL Pada Portal Berita Berbasis Android Dengan Menggunakan Metode First In First Out,” *Pros. SENDI\_U 2019*, pp. 978–979, 2019.

[15] R. Juliarto, “Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya,” *Dicoding*, 2021. https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/.

[16] Y. Husain, “Rancang Bangun Aplikasi Al-Qur ’ an Dan Jadwal Solat Berbasis Android,” pp. 1–10.

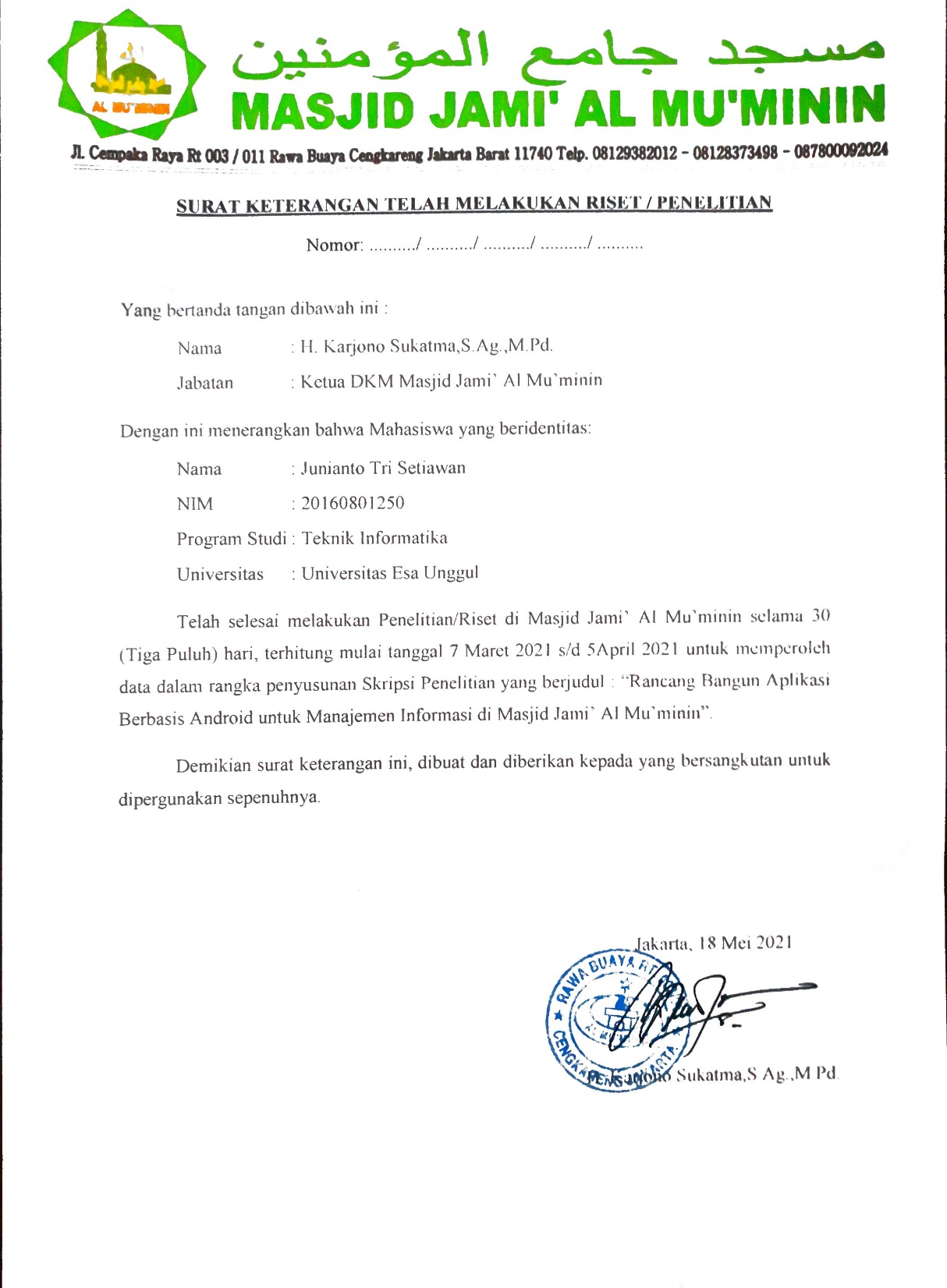
[17] D. Rizky, “Mengenal Prototyping,” *DOT Intern*, 2019. https://medium.com/dot-intern/sdlc-metode-prototype-8f50322b14bf.

[18] M. M. Zaki and N. Zainal, “Fitri Tomyam Seafood Menu Order Android Mobile Application,” vol. 2, no. 1, pp. 161–173, 2021.

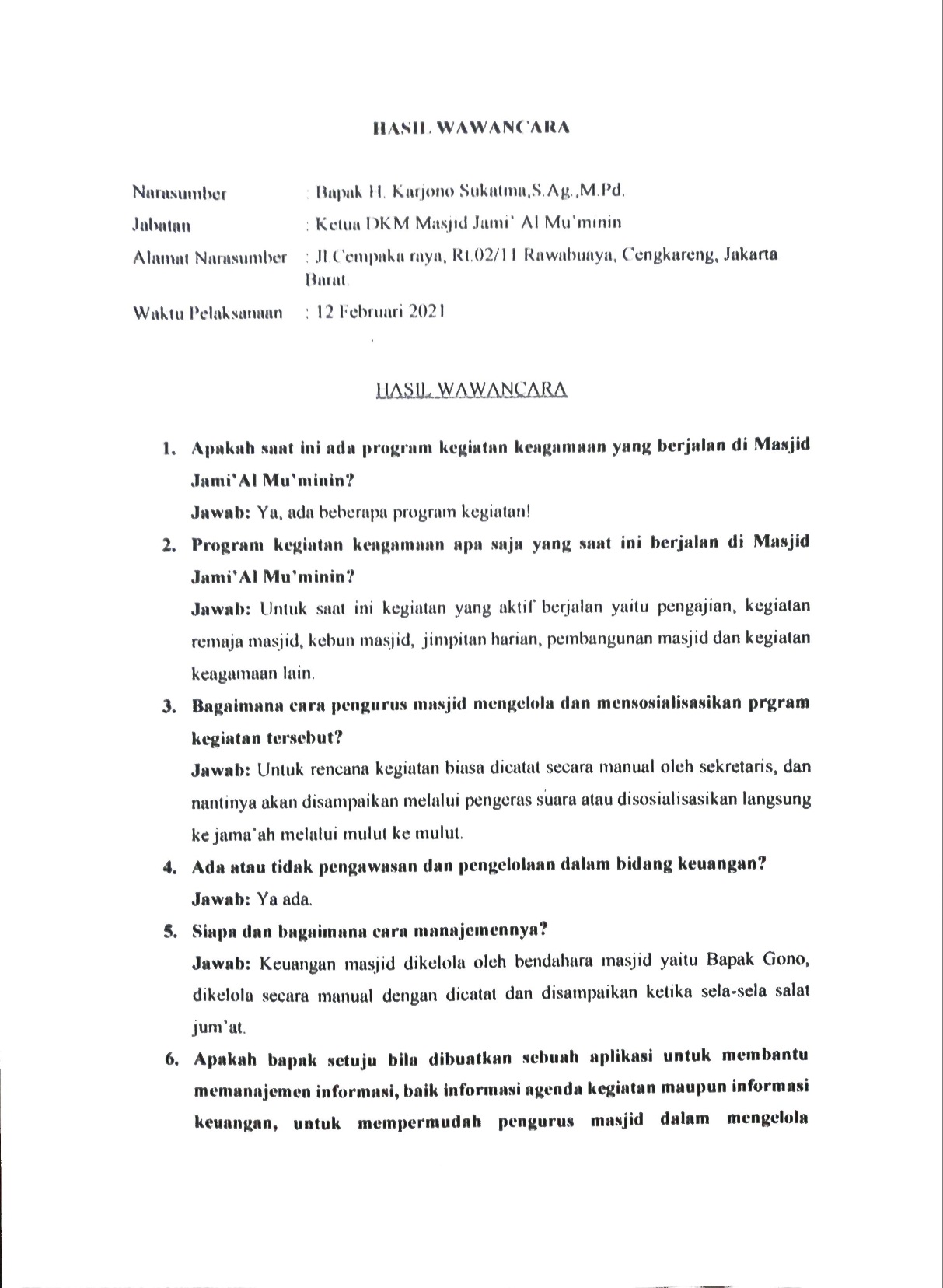
[19] L. Marlinda, A. Hermawan, and A. Fauzi, “Sistem Informasi Pengelolaan Masjid Online Menggunakan Metode Waterfall,” *Edik Inform.*, vol. 6, no. 1, pp. 20–27, 2019, doi: 10.22202/ei.2019.v6i1.3634.

[20] A. Amarudin and A. Sofiandri, “Perancangan dan Implementasi Aplikasi Ikhtisar Kas Masjid Istiqomah Berbasis Desktop,” *J. Tekno Kompak*, vol. 12, no. 2, p. 51, 2018, doi: 10.33365/jtk.v12i2.148.

# Lampiran 1



# Lampiran 2

****

****

# Lampiran 3

****