

**PENGEMBANGAN APLIKASI MASJID AL-IKHLAS
SOROSUTAN BERBASIS *PROGRESSIVE WEB APP* (PWA)
MENGUNAKAN METODE *DESIGN THINKING***



Disusun Oleh:

N a m a : Revansha Zulfikar Alfaz
NIM : 20523177

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING
PENGEMBANGAN APLIKASI MASJID AL-IKHLAS
SOROSUTAN BERBASIS *PROGRESSIVE WEB APP* (PWA)

TUGAS AKHIR



N a m a : Revansha Zulfikar Alfaz
NIM : 20523177

Yogyakarta, 28 Oktober 2024

Pembimbing,


(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI MASJID AL-IKHLAS
SOROSUTAN BERBASIS *PROGRESSIVE WEB APP* (PWA)
MENGUNAKAN METODE *DESIGN THINKING***

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 11 November 2024

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom.



Anggota 1

Ir. Dhomas Hatta Fudholi, S.T., M.Eng.,
Ph.D



Anggota 2

Moh. Idris, S.Kom., M.Kom.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Revansha Zulfikar Alfaz

NIM : 20523177

Tugas akhir dengan judul:

PENGEMBANGAN APLIKASI MASJID AL-IKHLAS SOROSUTAN BERBASIS *PROGRESSIVE WEB APP* (PWA)

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 28 Oktober 2024

A 10,000 Rupiah Indonesian postage stamp is shown, featuring the Garuda Pancasila emblem and the text 'METERAI TEMPEL'. The stamp is partially obscured by a handwritten signature in blue ink. The serial number 'AAAMX021545844' is visible at the bottom of the stamp.

(Revansha Zulfikar Alfaz)

HALAMAN PERSEMBAHAN

Dengan dituliskannya skripsi ini, penulis menyadari bahwa perjalanan masa kuliah telah mencapai garis akhir. Skripsi ini menjadi salah satu wujud nyata dari kerja keras dan dedikasi yang telah penulis lalui selama menempuh pendidikan. Sebagai bentuk apresiasi, penulis mempersembahkan karya ini kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung.

Pertama dan yang paling utama, skripsi ini penulis persembahkan kepada kedua orang tua tercinta, Papah dan Mamah yang selalu menjadi sumber energi dan cahaya dalam setiap langkah hidup penulis. Penulis menyadari bahwa sejauh ini, mungkin belum banyak yang dapat dibanggakan dari diri penulis, namun satu hal yang selalu menjadi tujuan adalah melihat senyum bahagia di wajah mereka berdua. Setiap prestasi yang penulis raih adalah persembahan kecil yang lahir dari kasih sayang dan doa tanpa henti yang mereka panjatkan dalam sujud terakhir setiap salat mereka.

Kedua, tentunya tak lupa kepada saudara penulis, kakak perempuan yang selalu memberikan arahan, dukungan, dan tuntunan bagi penulis. Beliau adalah sosok kakak yang dengan penuh kasih sayang menjalankan peran sebagai pelindung dan pengarah, yang senantiasa hadir saat penulis membutuhkan nasihat, baik dalam hal keputusan-keputusan kecil maupun besar.

Tentunya, skripsi ini penulis persembahkan kepada dosen pembimbing penulis, Ibu Arrie Kurniawardhani, S.Si., M.Kom. yang telah membimbing penulis pada masa penulisan tugas akhir ini dengan tulus. Penulis menyadari bahwasanya diri penulis masih jauh dari kata sempurna sebagai anak bimbingan beliau, namun beliau tetap sabar untuk terus membimbing penulis hingga dapat menyelesaikan skripsi ini.

Tak lupa, skripsi ini juga penulis persembahkan kepada organisasi yang telah menempa dan membentuk karakter penulis, Himpunan Mahasiswa Islam Fakultas Teknologi Industri (HMI FTI). Organisasi ini tidak hanya menjadi wadah bagi pengembangan diri, tetapi juga memaksa dan membimbing setiap kadernya, termasuk penulis, untuk terus tumbuh dan berkembang menjadi pribadi yang lebih baik. HMI FTI telah menjadi tempat di mana penulis menemukan keluarga baru, yang memberikan banyak pelajaran berharga serta mengubah pribadi penulis secara signifikan menuju arah yang lebih positif.

Terakhir, kepada seluruh orang yang pernah hadir dalam kehidupan penulis dan memberikan warna yang berarti, penulis mengucapkan terima kasih. Maaf, penulis tidak dapat menyebutkan satu per satu, namun setiap kehadiran dan pengalaman yang kalian berikan telah

mengenalkan penulis pada hal-hal baru yang berharga. Semua itu akan menjadi bagian dari kisah klasik yang akan selalu penulis kenang di masa yang akan datang. Terima kasih atas kehadiran kalian yang telah turut mewarnai perjalanan hidup ini.

HALAMAN MOTO

“Dan janganlah kamu (merasa) lemah, dan jangan (pula) bersedih hati, sebab kamu paling tinggi (derajatnya), jika kamu orang beriman”

– Q.S Ali Imran: 139

“Dan (ingatlah) ketika Tuhanmu memaklumkan, "Sesungguhnya jika kamu bersyukur, niscaya Aku akan menambah (nikmat) kepadamu, tetapi jika kamu mengingkari (nikmat-Ku), maka pasti azab-Ku sangat berat."

- Q.S Ibrahim: 7

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan nikmat yang diberikan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini. Tak lupa sholawat dan salam penulis panjatkan kepada baginda junjungan alam, Nabi Muhammad SAW sebagaimana beliau adalah sebaik-baiknya panutan bagi umat manusia termasuk penulis karena didalam diri beliau terdapat suri tauladan yang baik, *laqad kana lakum fi rasulillahi uswatun hasanah*.

Proses penyusunan skripsi ini tidak mungkin dilakukan tanpa adanya dukungan, doa, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa syukur dan terimakasih yang tulus kepada:

1. Kedua orang tua dan kakak perempuan penulis yang selalu memberikan doa yang tidak pernah terputus setiap hembusan nafasnya.
2. Kakak perempuan penulis yang selalu memberikan nasihat dan dukungan bagi penulis.
3. Ibu Arrie Kurniawardhani, S.Si., M.Kom. selaku dosen pembimbing penulis yang selalu memberikan saran dan masukan yang membangun hingga penulis dapat menyelesaikan skripsi ini.
4. Bapak Hari Purnomo, Prof., Dr., Ir., M.T., IPU, ASEAN.Eng. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
5. Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Progam Studi Informatika Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia.
6. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
7. Bapak Nuri Subandi dan seluruh anggota takmir Masjid Al-Ikhlas Sorosutan.
8. Seluruh pihak yang telah kebersamai penulis dalam melaksanakan perlombaan Program Kreativitas Mahasiswa (PKM) hingga sampai ditahap Pekan Ilmiah Mahasiswa Nasional (PIMNAS).
9. Penulis juga ingin menyampaikan rasa terima kasih kepada seluruh orang yang pernah penulis temui selama masa perkuliahan. Baik itu teman-teman dalam organisasi, kepanitiaan, maupun berbagai perlombaan yang telah penulis ikuti. Setiap pertemuan dengan mereka memberikan penulis pengalaman dan pelajaran berharga serta mendorong penulis untuk terus belajar dan berkembang setiap harinya.

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini masih ada kekurangan yang perlu diperbaiki. Oleh karena itu, penulis menghargai setiap saran dan masukan yang membangun untuk penyempurnaan laporan ini di masa mendatang. Semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang membutuhkan. Amin.

Yogyakarta, 21 Oktober 2024

A handwritten signature in black ink, appearing to read 'Revansha Zulfikar Alfaz', with a stylized flourish underneath.

(Revansha Zulfikar Alfaz)

SARI

Masjid Al-Ikhlas Sorosutan Yogyakarta menghadapi empat permasalahan utama, yaitu keamanan kotak infak, keterlambatan azan salat, pencatatan keuangan yang masih dilakukan secara manual, serta kegiatan pembelajaran di TPA yang belum optimal. Penelitian ini menerapkan *Progressive Web Application* (PWA) karena PWA mampu menghadirkan aplikasi web yang lebih cepat, responsif, dan dapat diakses secara *offline*, sehingga dapat meningkatkan efektivitas operasional masjid. Metodologi yang digunakan adalah *design thinking*, yang dimulai dari tahap *empathize* dengan mengumpulkan data dari calon pengguna aplikasi hingga tahap testing, yaitu pengujian aplikasi setelah dikembangkan untuk memastikan kesesuaian dengan standar dan penerimaan oleh pengguna. Hasil penelitian menunjukkan bahwa aplikasi Masjid Al-Ikhlas Sorosutan telah memenuhi standar *lighthouse* testing dan diterima oleh pengguna melalui pengujian SUS (*System Usability Scale*).

Kata kunci: masjid, *design thinking*, *lighthouse testing*, *Progressive Web App* (PWA).

GLOSARIUM

API	<i>Application Programming Interface</i> adalah cara agar aplikasi atau program bisa berkomunikasi dan berbagi data dengan aplikasi lain.
<i>Backend</i>	<i>Backend</i> adalah bagian dari pengembangan perangkat lunak yang mengelola proses-proses di balik layar, yang berjalan di server dan tidak langsung terlihat oleh pengguna.
Baitul Mal	Lembaga yang memiliki fungsi untuk mengelola harta milik umat.
Compile	Proses untuk mengubah berkas kode program dengan berkas lain yang terkait menjadi berkas yang siap untuk dieksekusi oleh sistem operasi secara langsung.
Debug	langkah untuk menelusuri kesalahan kode program.
<i>Framework</i>	Struktur pemrograman yang menyediakan seperangkat alat dan pustaka untuk memudahkan pengembangan aplikasi dengan cara yang lebih terorganisir dan efisien.
<i>Internet of Things</i>	Jaringan perangkat yang saling terhubung untuk bertukar data melalui internet.
<i>Push notification</i>	Teknologi yang dipakai oleh aplikasi untuk mengirimkan pesan kepada pengguna.
Takmir	Organisasi masjid yang berperan dalam pembangunan, pemeliharaan, serta pengelolaan kegiatan untuk memakmurkan masjid.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vii
KATA PENGANTAR	viii
SARI.....	x
GLOSARIUM.....	xi
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	6
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
1.6 Metode Penelitian	7
1.7 Sistematika Penulisan	8
BAB II LANDASAN TEORI	10
2.1 <i>Progressive Website App (PWA)</i>	10
2.1.1 Struktur Aplikasi <i>Progressive Web App (PWA)</i>	11
2.1.2 <i>Manifest File</i>	12
2.1.3 <i>Service Worker</i>	12
2.1.4 <i>Lighthouse Testing</i>	12
2.2 React JS.....	12
2.3 Firebase	13
2.4 <i>Design thinking</i>	13
2.4.1 <i>Empathize</i>	14
2.4.2 <i>Define</i>	14
2.4.3 <i>Ideate</i>	15
2.4.4 <i>Prototype</i>	18
2.4.5 <i>Testing</i>	19
2.5 Kajian Pustaka.....	26
BAB III METODE PENELITIAN	29
3.1 <i>Empathize</i>	29
3.1.1 Observasi.....	29
3.1.2 Wawancara.....	30
3.1.3 <i>Empathy Map</i>	31
3.2 Define.....	35
3.2.1 Pernyataan Masalah	35
3.2.2 <i>Pengguna Persona</i>	36
3.2.3 <i>The “How Might We” Method</i>	37
3.3 Ideate.....	39
3.3.1 Identifikasi Aktor	39
3.3.2 Identifikasi <i>Use Case</i>	39

3.3.3	Diagram <i>Use case</i>	40
3.3.4	Skenario <i>Use case</i>	42
3.3.5	<i>Activity Diagram</i>	57
3.3.6	<i>Class diagram</i>	73
3.4	Prototype	74
3.4.1	Alur Proses Bisnis Aplikasi	74
3.4.2	Perancangan Antarmuka Pengguna	80
3.4.3	Perancangan <i>Database</i>	86
3.4.4	Implementasi Proyek.....	88
3.4.5	Pengujian Aplikasi	99
	BAB IV HASIL DAN PEMBAHASAN	101
4.1	Hasil Implementasi Aplikasi	101
4.1.1	Tampilan Halaman Beranda Aplikasi	101
4.1.2	Tampilan <i>Monitoring</i> Posisi Kotak Infak	102
4.1.3	Tampilan Notifikasi Kotak Infak	104
4.1.4	Tampilan Halaman Catatan Keuangan	105
4.1.5	Tampilan Halaman Jadwal Waktu Salat	108
4.1.6	Tampilan Notifikasi Azan Salat	109
4.1.7	Tampilan Halaman Panduan Mengajar TPA	110
4.1.8	Tampilan Halaman Lihat Laporan Keuangan	113
4.1.9	Tampilan Halaman <i>Profile</i> Pengguna	116
4.1.10	Tampilan Halaman Beranda Admin	117
4.1.11	Tampilan Halaman Tabel Data Keuangan	118
4.1.12	Tampilan Halaman Kelola Data Pengguna	119
4.2	Hasil Pengujian Aplikasi.....	120
4.2.1	Pengujian <i>Black Box</i>	120
4.2.2	Pengujian <i>Lighthouse</i>	133
4.2.3	Pengujian <i>System Usability Scale (SUS)</i>	134
	BAB V KESIMPULAN DAN SARAN.....	140
5.1	Kesimpulan	140
5.2	Saran.....	140
	DAFTAR PUSTAKA	142
	LAMPIRAN	145

DAFTAR TABEL

Tabel 2. 1 Pertanyaan Pengujian <i>System Usability Scale</i> (SUS)	20
Tabel 2. 2 Pilihan Jawaban Pengujian <i>System Usability Scale</i> (SUS).....	20
Tabel 2. 3 Tabel Audit Lighthouse Aksesibilitas	23
Tabel 2. 4 Kajian Pustaka Penelitian Terdahulu	27
Tabel 3. 1 Hasil Observasi	30
Tabel 3. 2 <i>Empathize Map</i> Takmir Masjid Al-Ikhlas Sorosutan.	34
Tabel 3. 3 Pernyataan Masalah Takmir Masjid Al-Ikhlas Sorosutan	36
Tabel 3. 4 Identifikasi <i>Use case</i>	39
Tabel 3. 5 <i>Use case</i> Pantau Posisi Kotak Infak	42
Tabel 3. 6 <i>Use case</i> Notifikasi Kotak Infak	43
Tabel 3. 7 <i>Use case</i> Tambah Catatan Keuangan.....	44
Tabel 3. 8 <i>Use Case</i> Lihat Catatan Keuangan	45
Tabel 3. 9 <i>Use Case</i> Mengelola Data Catatan Keuangan	46
Tabel 3. 10 <i>Use Case</i> Lihat Jadwal Waktu Salat	47
Tabel 3. 11 <i>Use Case</i> Notifikasi Azan Salat.....	48
Tabel 3. 12 <i>Use Case</i> Modul Panduan Mengajar TPA	49
Tabel 3. 13 <i>Use Case</i> Perbarui Profil Pengguna.....	50
Tabel 3. 14 <i>Use Case</i> Registrasi Takmir	51
Tabel 3. 15 <i>Use Case Login</i> Takmir	52
Tabel 3. 16 <i>Use Case Logout</i> Takmir	53
Tabel 3. 17 <i>Use Case</i> Mengelola Data Pengguna.....	54
Tabel 3. 18 <i>Use case</i> Mengelola Data Keuangan	55
Tabel 3. 19 <i>Use Case Login</i> Admin.....	56
Tabel 3. 20 <i>Use Case Logout</i> Admin.....	57
Tabel 4. 1 Hasil Pengujian <i>Black Box</i> Fungsionalitas Aplikasi.....	121
Tabel 4. 2 Hasil Pengujian <i>Black Box</i> Antarmuka Aplikasi	124
Tabel 4. 3 Hasil Pengujian <i>Lighthouse</i>	133
Tabel 4. 4 Pertanyaan Fungsionalitas Sistem Terhadap Permasalahan	135
Tabel 4. 5 Hasil Kuisioner <i>System Usability Scale</i>	135
Tabel 4. 6 Hasil Jawaban SUS yang Telah Dilakukan Perhitungan	136
Tabel 4. 7 Jawaban Pertanyaan Fungsionalitas Sistem Terhadap Permasalahan	136

DAFTAR GAMBAR

Gambar 2. 1 Proses <i>Design thinking</i>	14
Gambar 2. 2 Notasi <i>Use case Diagram</i>	16
Gambar 2. 3 Notasi <i>Activity Diagram</i>	17
Gambar 2. 4 Skala Penilaian <i>System Usability Scale</i>	21
Gambar 3. 1 <i>Empathize Map</i> Takmir Masjid Al-Ikhlas Sorosutan.	32
Gambar 3. 2 <i>Pengguna Persona</i>	37
Gambar 3. 3 Diagram <i>Use case</i>	41
Gambar 3. 4 <i>Activity Diagram</i> Pantau Posisi Kotak Infak	58
Gambar 3. 5 <i>Activity Diagram</i> Notifikasi Kotak Infak.....	59
Gambar 3. 6 <i>Activity Diagram</i> Tambah Catatan Keuangan	60
Gambar 3. 7 <i>Activity Diagram</i> Lihat Catatan Keuangan	62
Gambar 3. 8 <i>Activity Diagram</i> Kelola Catatan Keuangan.....	63
Gambar 3. 9 <i>Activity Diagram</i> Lihat Jadwal Waktu Salat.....	64
Gambar 3. 10 <i>Activity Diagram</i> Notifikasi Azan Salat.....	64
Gambar 3. 11 <i>Activity Diagram</i> Modul Panduan Mengajar TPA.....	65
Gambar 3. 12 <i>Activity Diagram</i> Perbarui Profil Pengguna.....	66
Gambar 3. 13 <i>Activity Diagram</i> Registrasi Takmir	67
Gambar 3. 14 <i>Activity Diagram</i> Login Takmir	68
Gambar 3. 15 <i>Activity Diagram</i> Logout Takmir	69
Gambar 3. 16 <i>Activity Diagram</i> Mengelola Data Pengguna.....	70
Gambar 3. 17 <i>Activity Diagram</i> Kelola Data Keuangan.....	71
Gambar 3. 18 <i>Activity Diagram</i> Login Admin	72
Gambar 3. 19 <i>Activity Diagram</i> Logout Admin.....	73
Gambar 3. 20 <i>Class diagram</i> Aplikasi Masjid Al-Ikhlas Sorosutan	73
Gambar 3. 21 Alur aplikasi Masjid Al-Ikhlas Sorosutan.....	76
Gambar 3. 22 Sub alur fitur aplikasi pengguna (takmir)	77
Gambar 3. 23 Sub alur fitur aplikasi admin	78
Gambar 3. 24 Rancang Halaman Beranda Pengguna	80
Gambar 3. 25 Rancangan halaman fitur <i>monitoring</i> kotak infak.....	81
Gambar 3. 26 Rancangan halaman fitur tambah catatan keuangan	81
Gambar 3. 27 Rancangan halaman fitur jadwal waktu salat.....	82
Gambar 3. 28 Rancangan halaman fitur panduan mengajar TPA	83
Gambar 3. 29 Rancangan halaman fitur lihat laporan keuangan	83

Gambar 3. 30 Rancangan halaman profil pengguna	84
Gambar 3. 31 Rancangan halaman beranda admin.....	85
Gambar 3. 32 Rancangan halaman edit data pengguna	85
Gambar 3. 33 Rancangan halaman registrasi pengguna	86
Gambar 3. 34 Rancangan halaman <i>login</i>	86
Gambar 3. 35 Struktur Basis Data <i>Local Storage</i>	87
Gambar 3. 36 Struktur Basis Data Firebase.....	88
Gambar 3. 37 Perintah Installasi React JS Ke dalam Komputer	88
Gambar 3. 38 Perintah untuk memulai proyek ReactJS	89
Gambar 3. 39 Isi File <i>Serviceworker</i> Aplikasi Masjid Al-Ikhlas Sorosutan.....	90
Gambar 3. 40 Konfigurasi Manifest File PWA pada aplikasi	92
Gambar 3. 41 Perintah memasang plugin React Router pada proyek ReactJS	93
Gambar 3. 42 Contoh isi file <i>App.js</i> sebagai route	97
Gambar 3. 43 Konfigurasi Firebase <i>fire.js</i>	98
Gambar 3. 44 perintah untuk build aplikasi <i>react.js</i>	99
Gambar 4. 1 Tampilan <i>Desktop</i> Halaman Beranda Aplikasi	101
Gambar 4. 2 Tampilan <i>Mobile</i> Halaman Beranda Aplikasi.....	102
Gambar 4. 3 Tampilan <i>Desktop</i> Halaman Pantau Posisi Kotak Infak	103
Gambar 4. 4 Tampilan <i>Mobile</i> Halaman Pantau Posisi Kotak Infak.....	104
Gambar 4. 5 Tampilan <i>Desktop</i> Notifikasi Kotak Infak	105
Gambar 4. 6 Tampilan <i>Mobile</i> Notifikasi Kotak Infak.....	105
Gambar 4. 7 Tampilan <i>Desktop</i> Beranda Catatan Keuangan	106
Gambar 4. 8 Tampilan <i>Desktop</i> Tambah Catatan Keuangan.....	106
Gambar 4. 9 Tampilan <i>Desktop</i> Formulir Pemasukan Keuangan.....	107
Gambar 4. 10 Tampilan <i>Desktop</i> Formulir Pengeluaran Keuangan	107
Gambar 4. 11 Tampilan <i>Mobile</i> Halaman Catatan Keuangan	107
Gambar 4. 12 Tampilan <i>Desktop</i> Halaman Jadwal Waktu Salat	108
Gambar 4. 13 Tampilan <i>Mobile</i> Halaman Jadwal Waktu Salat.....	109
Gambar 4. 14 Tampilan <i>Desktop</i> Notifikasi Azan Salat.....	110
Gambar 4. 15 Tampilan <i>Mobile</i> Notifikasi Azan Salat.....	110
Gambar 4. 16 Tampilan <i>Desktop</i> Beranda Modul Panduan Mengajar TPA.....	112
Gambar 4. 17 Tampilan <i>Desktop</i> Beranda Salah Satu Modul Panduan Mengajar TPA.....	112
Gambar 4. 18 Tampilan <i>Desktop</i> Salah Satu Materi Modul Panduan Mengajar TPA	113

Gambar 4. 19 Tampilan <i>Mobile</i> Modul Panduan Mengajar TPA.....	113
Gambar 4. 20 Tampilan <i>Desktop</i> Beranda Laporan Keuangan	114
Gambar 4. 21 Tampilan <i>Desktop</i> Tabel Laporan Keuangan.....	115
Gambar 4. 22 Tampilan <i>Desktop</i> Grafik Laporan Keuangan	115
Gambar 4. 23 Tampilan <i>Mobile</i> Laporan Keuangan	116
Gambar 4. 24 Tampilan <i>Desktop</i> Halaman Profil Pengguna.....	117
Gambar 4. 25 Tampilan <i>Desktop</i> Halaman Formulir Ubah Data Pengguna.....	117
Gambar 4. 26 Tampilan <i>Mobile</i> Halaman Profil Pengguna dan Formulir Ubah Data Pengguna	117
Gambar 4. 27 Tampilan <i>Desktop</i> Halaman Beranda Admin	118
Gambar 4. 28 Tampilan Halaman <i>Desktop</i> Admin Tabel Laporan Keuangan Masjid.....	119
Gambar 4. 29 Tampilan Halaman <i>Mobile</i> Admin Tabel Laporan Keuangan Masjid.....	119
Gambar 4. 30 Tampilan <i>Desktop</i> Halaman Admin Tabel Daftar Pengguna.....	120
Gambar 4. 31 Tampilan <i>Mobile</i> Halaman Admin Tabel Daftar Pengguna	120
Gambar 4. 32 Tampilan Hasil Pengujian <i>Lighthouse Testing</i>	133

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan sebuah negara dengan mayoritas penduduknya beragama Islam dan memegang predikat sebagai negara dengan jumlah umat Muslim terbanyak di dunia (Kusnandar, 2021). Selain itu, seperti yang disampaikan oleh Ketua Umum Dewan Masjid Indonesia (DMI), Jusuf Kalla, Indonesia juga terkenal sebagai negara dengan jumlah masjid terbanyak di dunia (Antaraneews, 2020). Menurut Dewan Masjid Indonesia (DMI), saat ini terdapat lebih dari 800 ribu masjid di seluruh Indonesia, dengan tambahan 1 juta musholla yang tersebar di berbagai daerah. Angka ini merupakan pencapaian yang sangat mengesankan, karena tidak ada negara lain di dunia yang bisa menyamai jumlah masjid dan musholla yang ada di Indonesia (Jedi, 2023).

Pada umumnya, masjid bukan hanya tempat ibadah semata, tetapi juga merupakan pusat kegiatan sosial, wadah untuk memperkuat hubungan sosial, serta menjadi sarana pendidikan dan penyebaran ajaran agama bagi masyarakat (Dalmeri, 2014). Selain itu, masjid juga berperan sebagai Baitul Mal, tempat untuk mengumpulkan dan menjaga harta, serta digunakan untuk kegiatan sedekah dan kontribusi yang bermanfaat bagi umat, termasuk proyek-proyek pembangunan masjid, dukungan kepada anak yatim, bantuan kepada mereka yang berada dalam situasi ekonomi sulit, serta berbagai program sosial lainnya.

Masjid Al-Ikhlas Sorosutan berada di Jalan Sorosutan Blok UH 6 Nomor 822, RT. 12/RW. 4, kelurahan Sorosutan, kecamatan Umbulharjo, kota Yogyakarta, Daerah Istimewa Yogyakarta 55162. Berdasarkan data kantor kelurahan Sorosutan, mayoritas masyarakat sekitar Sorosutan menganut agama Islam, yakni lebih dari 93,63%. Begitu juga untuk sekitaran Masjid Al-Ikhlas Sorosutan yang penduduknya mayoritas beragama Islam. Menurut takmir, masyarakat sekitar masjid cukup gemar dalam bersedekah melalui kotak infak. Hal ini berdasarkan jumlah kas bulanan masjid yang mencapai jutaan rupiah. Setiap bulannya, takmir akan merekap uang infak yang didapat dan melakukan pencatatan keuangan tersebut ke dalam buku operasional keuangan masjid.

Sama halnya dengan mayoritas masjid di Indonesia, Masjid Al-Ikhlas Sorosutan mengandalkan kotak infak sebagai sarana utama pengumpulan dana infak. Akan tetapi, tidak jarang kita mendengar tentang kejadian di mana masjid-masjid mengalami kehilangan kotak infak seperti yang terjadi di Masjid Al-Ikhlas Sorosutan. Meskipun sudah dilengkapi dengan teknologi CCTV, Masjid Al-Ikhlas Sorosutan yang terletak di Kota Yogyakarta masih

mengalami kejadian pencurian uang infak pada tahun 2021 dengan perkiraan total kerugian mencapai 50 juta rupiah (Baktora, 2021).

Masjid Al-Ikhlas Sorosutan masih menggunakan sistem pencatatan keuangan secara manual dalam buku keuangan masjid, termasuk pencatatan pemasukan dan pengeluaran. Pak Nuri Subandi selaku ketua takmir menjelaskan bahwa mereka sering mengalami kesulitan dalam merekap keuangan masjid atau menemukan catatan keuangan yang sudah lama karena buku atau kertas seringkali hilang atau terhapus.

Masjid Al-Ikhlas Sorosutan juga seringkali mengalami keterlambatan dalam mengumandangkan azan salat. Hal ini dikarenakan tidak adanya takmir yang menetap di dalam masjid. Selain itu, aktivitas kegiatan yang dilakukan di masjid ini terbatas hanya pada saat Peringatan Hari Besar Islam (PHBI) saja. Taman Pendidikan Al-Qur'an (TPA) merupakan salah satu kegiatan rutin yang terhambat karena kurangnya tenaga pengajar dan tidak adanya modul atau kurikulum sebagai pedoman yang digunakan.

Untuk mengatasi berbagai permasalahan tersebut, solusi yang direncanakan adalah pengembangan *Progressive Web Application* (PWA) yang dapat mengintegrasikan beberapa fitur utama guna meningkatkan keamanan, efisiensi manajemen keuangan, dan optimalisasi kegiatan di masjid. PWA merupakan teknologi modern yang memungkinkan aplikasi web memberikan pengalaman pengguna layaknya aplikasi *native*. Aplikasi ini dirancang untuk diakses melalui berbagai perangkat, bekerja dengan baik meskipun dalam kondisi internet terbatas, serta dapat diinstal di perangkat pengguna.

Solusi berbasis PWA ini mencakup empat fitur utama. Pertama, melalui implementasi *monitoring* pelacakan kotak infak berbasis IoT untuk meningkatkan keamanan serta pengawasan kotak infak secara efektif. Kedua, penggunaan *push notification* untuk jadwal waktu salat bertujuan memastikan pengumandangan azan salat tepat waktu. Ketiga, dengan pencatatan keuangan digital, diharapkan proses pencatatan pemasukan dan pengeluaran masjid menjadi lebih efisien dan transparan. Terakhir, pembuatan Modul Belajar TPA bertujuan memberikan pedoman yang terstruktur bagi takmir dalam mengajar TPA serta memastikan muatan materi yang diajarkan terorganisir dengan baik.

Solusi ini dirancang menggunakan pendekatan *design thinking*, yang melibatkan tahapan *empathize*, *define*, *ideate*, *prototype*, dan *test*. Metode ini digunakan untuk memastikan bahwa solusi yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna dan mampu memberikan dampak positif bagi pengelolaan Masjid Al-Ikhlas Sorosutan. Dengan penerapan

teknologi PWA dan metode *design thinking*, diharapkan pengelolaan masjid dapat menjadi lebih efektif, aman, dan mendukung keberlanjutan berbagai kegiatan keagamaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan, adapun rumusan masalah yang dapat diambil adalah sebagai berikut:

- a. Bagaimana pengembangan aplikasi sistem *monitoring* keamanan kotak infak berbasis *Internet of Things (IoT)* dapat membantu meningkatkan keamanan kotak infak masjid?
- b. Bagaimana cara untuk memperbaiki manajemen dan pencatatan keuangan Masjid Al-Ikhlas Sorosutan secara lebih efektif dan efisien?
- c. Bagaimana pengembangan aplikasi ini diharapkan dapat mengatasi permasalahan keterlambatan azan salat di Masjid Al-Ikhlas Sorosutan?
- d. Bagaimana pengembangan aplikasi ini diharapkan dapat membantu mengoptimalkan kegiatan Taman Pendidikan Al-Qur'an (TPA) di Masjid Al-Ikhlas Sorosutan?

1.3 Batasan Masalah

Untuk menjaga fokus pada penelitian ini maka ditentukan batasan masalah pada penelitian pengembangan aplikasi ini adalah:

- a. Penelitian ini difokuskan pada pengembangan aplikasi berbasis *Progressive Web Apps (PWA)*.
- b. *Internet of Things (IoT)* terbatas pada sistem *monitoring* kotak infak.
- c. Pengguna aplikasi ini terbatas pada takmir Masjid Al-Ikhlas Sorosutan.
- d. Pengembangan aplikasi untuk Taman Pendidikan Al-Qur'an (TPA) terbatas pada penyediaan modul panduan mengajar TPA.

1.4 Tujuan Penelitian

Adapun tujuan dilaksanakannya penelitian dalam pengembangan aplikasi ini adalah sebagai berikut:

- a. Meningkatkan keamanan kotak infak Masjid Al-Ikhlas Sorosutan melalui pengembangan aplikasi sistem *monitoring* keamanan kotak infak berbasis *Internet of Things (IoT)*.
- b. Memperbaiki manajemen keuangan dan pencatatan keuangan Masjid Al-Ikhlas Sorosutan menjadi lebih efektif dan efisien.
- c. Mengatasi permasalahan keterlambatan azan salat di Masjid Al-Ikhlas Sorosutan.

- d. Membantu mengoptimalkan kegiatan Taman Pendidikan Al-Qur'an (TPA) di Masjid Al-Ikhlas Sorosutan melalui pengembangan modul panduan mengajar TPA.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat kepada masjid dan struktur yang ada di dalamnya, yang diantaranya adalah:

- a. Aplikasi yang diusulkan dapat meningkatkan tingkat keamanan masjid, khususnya dalam melindungi kotak infak dari tindakan pencurian.
- b. Pengembangan aplikasi ini dapat memperbaiki manajemen keuangan dan pencatatan keuangan.
- c. Pengembangan aplikasi ini dapat mengatasi permasalahan keterlambatan azan salat di Masjid Al-Ikhlas Sorosutan. Dengan adanya fitur atau notifikasi yang mengingatkan waktu salat, jamaah akan lebih mudah untuk menjalankan ibadah secara tepat waktu.
- d. Pengembangan aplikasi ini juga dapat membantu mengoptimalkan kegiatan Taman Pendidikan Al-Qur'an (TPA) di Masjid Al-Ikhlas Sorosutan. Dengan adanya modul panduan mengajar dan fitur manajemen kegiatan, takmir diharapkan memiliki alat bantu yang efektif untuk mengelola dan meningkatkan kualitas kegiatan TPA secara terstruktur.

1.6 Metode Penelitian

Adapun metode penelitian ini adalah dengan menggunakan pendekatan metode *Design Thinking* yang meliputi lima tahapan, yakni *empathize* (berempati), *define* (mendefinisikan), *ideate* (memunculkan ide), *prototype* (membuat produk), dan *testing* (uji-coba).

a. *Empathize*

Tahap pertama dalam proses ini adalah tahap empati yang bertujuan untuk memahami sepenuhnya kebutuhan pengguna. Dalam konteks ini, pengguna yang dimaksud adalah pihak takmir Masjid Al-Ikhlas Sorosutan. Metode yang digunakan untuk memahami kebutuhan pengguna adalah dengan melakukan observasi dan wawancara langsung dengan pihak takmir masjid.

b. *Define*

Setelah proses pengumpulan informasi pada tahap sebelumnya, tahap selanjutnya adalah melakukan analisis mendalam terhadap informasi yang telah terkumpul. Analisis ini bertujuan untuk mengidentifikasi masalah inti yang akan menjadi fokus utama pembuatan

aplikasi ini. Dengan memahami masalah inti dengan baik, langkah-langkah selanjutnya dalam pengembangan aplikasi dapat diarahkan dengan lebih tepat dan efektif.

c. Ideate

Pada fase ini, ide atau solusi yang cocok untuk mengatasi masalah dan memenuhi kebutuhan pengguna ditawarkan. Setelah ide tersebut ditawarkan, langkah berikutnya adalah mengembangkan sistem.

d. Prototype

Pada tahap ini sistem yang sudah dikembangkan akan menjadi hasil implementasi sistem berbasis *Progressive Web Application* (PWA). Sistem ini dibuat dengan rancangan yang mudah digunakan oleh takmir, baik fitur yang akan digunakan ataupun proses bisnis yang berlangsung di dalam sistem.

e. Test

Pada tahap pengujian, peneliti melakukan uji coba menggunakan *prototype* yang telah dibuat. Kemudian didapatkan hasil uji yang akan menjadi bahan evaluasi perbaikan pada sistem ini. Langkah pengujian ini bertujuan untuk menguji kesiapan sistem sebelum diimplementasikan secara langsung di Masjid Al-Ikhlas Sorosutan Yogyakarta.

1.7 Sistematika Penulisan

Penyusunan laporan tugas akhir ini mengikuti langkah-langkah tertentu dalam menyusun berbagai bab, yang melibatkan pemaparan secara menyeluruh mengenai permasalahan dan cara penyelesaiannya. Rincian mengenai susunan laporan ini dapat dijelaskan dalam lima bab:

BAB I PENDAHULUAN

Bab ini berisi pembahasan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi pembahasan mengenai tinjauan terhadap penelitian yang pernah ada berhubungan dengan apa yang akan dirancang dan diimplementasikan serta teori dasar yang digunakan berhubungan dengan sistem dalam mengimplementasikan aplikasi.

BAB III METODOLOGI

Bab ini berisi uraian tentang analisis, desain dan pengembangan aplikasi untuk meningkatkan tingkat keamanan masjid, khususnya dalam melindungi kotak infak masjid dari pencurian.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini, akan dijelaskan mengenai pelaksanaan implementasi aplikasi, termasuk proses implementasi dan evaluasi hasil uji coba aplikasi.

BAB V KESIMPULAN DAN SARAN

Sebagai penutup, bab terakhir ini akan membahas simpulan dan saran terkait dengan penelitian yang telah dilakukan dalam tugas akhir ini.

BAB II LANDASAN TEORI

2.1 *Progressive Website App (PWA)*

Progressive Web App (PWA) adalah metode pengembangan aplikasi web yang menggunakan teknologi modern untuk menciptakan pengalaman pengguna yang mirip dengan aplikasi *native* pada perangkat *mobile* (Sheppard, 2017). PWA merupakan aplikasi *hybrid* yang menggabungkan kemudahan akses peramban di berbagai perangkat dengan kenyamanan aplikasi *mobile* pada ponsel atau tablet, sehingga optimasi yang dilakukan tidak hanya meningkatkan kecepatan situs web tetapi juga menyajikan pengalaman pengguna yang setara dengan aplikasi *mobile* (Kurniawan, 2022). Menurut *Google Developer*, PWA memiliki beberapa karakteristik tersendiri yang membedakannya dengan aplikasi yang lain (Panatagama, 2023):

a. *Progressive*

PWA dirancang untuk bisa diakses oleh semua pengguna, tanpa memandang jenis peramban yang mereka gunakan. Karena pengembangannya bersifat progresif, PWA dapat berfungsi di berbagai kondisi peramban, baik yang terbaru maupun yang lama.

b. *Responsive*

PWA sangat responsif dan dapat berjalan mulus di berbagai jenis perangkat, termasuk *desktop*, tablet, dan *smartphone*. Hal ini memastikan pengalaman pengguna yang konsisten di berbagai ukuran layar dan jenis perangkat.

c. *Connectivity independent*

PWA menggunakan *service workers*, yang memungkinkan aplikasi diakses secara *offline* atau saat kualitas koneksi jaringan rendah.

d. *App-like Display*

PWA memberikan pengalaman pengguna yang mirip dengan aplikasi *native*. Ini terjadi karena PWA menggunakan model aplikasi *shell* yang memisahkan fungsi dari konten aplikasi.

e. *Up-to-date*

PWA akan selalu diperbarui dan terasa *fresh* karena selalu dilakukan proses update oleh *service worker*.

f. *Security*

PWA disajikan melalui HTTPS, yang menjamin keamanan penggunaan dengan mencegah pengintaian dan menjaga integritas konten. Ini memberikan rasa aman bagi pengguna saat mengakses aplikasi.

g. *Discoverable*

PWA dapat dengan mudah diidentifikasi sebagai aplikasi berkat *W3C manifest* dan *service worker registration scope*. Hal ini memungkinkan PWA untuk lebih mudah ditemukan melalui *search engine*, meningkatkan visibilitasnya.

h. *Re-engageable*

Dengan berbagai fitur yang tersedia, PWA dapat dengan mudah mengaktifkan keterlibatan pengguna. Salah satu contohnya adalah penggunaan *push notification* untuk menciptakan interaksi yang menarik.

i. *Installable*

Pengguna dapat menambahkan PWA ke layar utama perangkat mereka tanpa harus melalui *app store* atau *play store*. Ini memberikan kemudahan akses langsung ke aplikasi.

j. *Linkable*

PWA dapat dibagikan dengan mudah melalui URL. Pengguna tidak perlu menginstall aplikasi terlebih dahulu untuk mengaksesnya, cukup dengan membuka tautan yang diberikan.

2.1.1 Struktur Aplikasi *Progressive Web App* (PWA)

Dalam pengembangan web, istilah "*server-side*" dan "*client-side*" merujuk pada dua aspek utama dalam arsitektur aplikasi web. Perbedaan antara kedua sisi ini menentukan cara aplikasi beroperasi, menyajikan konten kepada pengguna, dan memproses data (Sojitra, 2024).

- a. *Server-side rendering (SSR)* adalah proses di mana komponen *website* dibuat di server sebelum dikirimkan ke perangkat pengguna. Meskipun SSR mampu beroperasi secara optimal di berbagai jenis peramban, namun mengharuskan pengguna untuk mengunduh halaman HTML baru setiap kali melakukan navigasi. Setiap permintaan halaman membutuhkan interaksi bolak-balik dengan server.
- b. *Client-side rendering (CSR)* memungkinkan *website* untuk segera memuat komponen-komponen saat pengguna melakukan navigasi halaman. Namun, metode ini memerlukan unduhan yang lebih besar dan proses rendering yang lebih berat saat pengguna mengakses *website* untuk pertama kalinya.

2.1.2 *Manifest File*

Web App Manifest adalah file berformat .JSON yang mengontrol tampilan pengguna dan memberi tahu peramban tentang perilaku aplikasi web saat diinstal pada perangkat seluler atau *desktop* (El Hakim, 2021). *Manifest File* memungkinkan penginstalan aplikasi ke perangkat pengguna dengan menyediakan ikon di layar utama. Ketika ikon ini ditekan, aplikasi dapat dijalankan langsung tanpa perlu membuka peramban atau mengetik URL, mirip dengan aplikasi *native*. Fitur ini memudahkan pengguna dalam mengakses dan berinteraksi dengan aplikasi, meningkatkan pengalaman pengguna secara keseluruhan (Putra, 2023).

2.1.3 *Service Worker*

Service worker adalah skrip yang dijalankan oleh peramban di latar belakang, terpisah dari skrip lain di halaman web. Dengan menggunakan *service worker*, kita dapat memanfaatkan sumber daya yang sudah disimpan dalam *cache* untuk menampilkan konten, bahkan saat pengguna sedang *offline* (Rahman, 2022).

2.1.4 *Lighthouse Testing*

Lighthouse adalah alat audit *open-source* dari peramban *Google Chrome* yang digunakan untuk menguji dan menilai performa, aksesibilitas, SEO, dan kelayakan instalasi PWA dari sebuah aplikasi web (Putra, 2023).

2.2 React JS

React adalah salah satu pustaka JavaScript paling populer, mengungguli pustaka lain seperti Angular dan Vue (Sasks, 2019). React juga dikenal sebagai salah satu pustaka JavaScript yang paling mudah dipelajari. Dari segi kinerja, React memperoleh hasil yang baik dalam pengujian yang dilakukan terhadap tiga pustaka utama: React, Angular, dan Vue. Dalam pengujian tersebut, Vue menempati peringkat pertama, diikuti oleh React, dan terakhir Angular.

Pada tahun 2013, tim pengembang dari Facebook memperkenalkan React JS sebagai sebuah perpustakaan tampilan (*view library*), yang merupakan bagian dari pola desain MVC (*Model View Controller*) (Shelli, 2014). React JS adalah sebuah framework JavaScript yang digunakan untuk membangun aplikasi web modern dengan menggunakan teknik yang berbeda dari *vanilla JavaScript* dan *Jquery* (Wieruch, 2017). React JS dapat mendesain tampilan sederhana untuk setiap level dalam aplikasi, sehingga dapat digunakan untuk membuat dan mengembangkan pembuatan aplikasi berbasis web (Ham, 2019).

React JS mengusung pendekatan yang unik dalam pengembangan aplikasi web dengan menggunakan konsep komponen. Pendekatan ini memungkinkan pengembang untuk memecah

antarmuka pengguna (UI) menjadi bagian-bagian kecil yang dikelola secara independen. Dengan cara ini, pengembang dapat fokus pada pengembangan komponen tertentu tanpa harus memperhatikan bagian lain dari aplikasi secara keseluruhan (Shelli, 2014).

2.3 Firebase

Firebase merupakan layanan *Backend as a Service (BaaS)* yang menawarkan berbagai alat dan layanan untuk mempercepat pengembangan aplikasi web dan *mobile*. Sebagai BaaS, Firebase bertanggung jawab atas manajemen *backend* aplikasi, termasuk *database*, otentikasi, hosting, API, dan layanan lainnya (Regita, 2022).

Firebase menawarkan berbagai fitur yang dapat dieksplorasi, beberapa di antaranya adalah (Regita, 2022):

1. *Firebase Realtime Database*

Merupakan layanan yang memungkinkan sinkronisasi data secara real-time antar pengguna, baik dalam kondisi *online* maupun *offline*. Namun, disarankan untuk aplikasi web yang sederhana.

2. *Cloud Firestore*

Merupakan *database* Firebase terbaru yang cocok untuk aplikasi dengan interaksi kompleks dan besar. Fitur sinkronisasi *offline* tersedia untuk iOS, Android, dan web.

3. *Firebase Authentication*

Layanan autentikasi yang mudah digunakan dengan dukungan untuk berbagai metode autentikasi, serta integrasi dengan layanan Firebase lainnya, serta fitur tampilan interfaces siap pakai.

4. *Firebase Cloud Messaging and Notifications (FCM)*

FCM menyediakan koneksi yang handal untuk mengirim dan menerima pesan dan notifikasi di berbagai platform tanpa biaya tambahan, dengan integrasi penuh dengan *Google Analytics for Firebase*.

2.4 Design thinking

Design thinking adalah sebuah metode yang dikenal sebagai proses berpikir komprehensif yang berfokus pada penciptaan solusi, dimulai dengan empati terhadap kebutuhan tertentu yang berpusat pada manusia (*human-centered*) dan berlanjut hingga menghasilkan inovasi berkelanjutan berdasarkan kebutuhan penggunanya (Nadhif et al., 2021). Keunikan metode ini terletak pada sifatnya yang iteratif (berulang), non-linear, dan fleksibel. Metode ini juga menekankan pentingnya kolaborasi antara desainer dan pengguna, dengan fokus pada

pengembangan ide-ide berdasarkan cara berpikir, merasakan, dan berperilaku pengguna sesungguhnya (Plattner et al., 2011).



Gambar 2. 1 Proses *Design thinking*

Sumber: (Shirvanadi & Idris, 2021)

Design Thinking memperhatikan kebutuhan pengguna dan mengintegrasikannya dengan kapabilitas teknologi yang tepat, sehingga menghasilkan produk bisnis yang berkualitas dan sesuai. Pendekatan ini sangat efektif untuk merumuskan kembali tantangan bersama konsumen, menghasilkan berbagai gagasan melalui sesi perbincangan kreatif, serta mengadopsi metode prototipe dan uji coba langsung. *Design thinking* sangat berperan dalam menghadapi situasi masalah yang tidak jelas atau belum teridentifikasi (Saputra & Kania, 2022).

2.4.1 *Empathize*

Tahap empati adalah langkah awal yang berorientasi pada pengguna dalam pembuatan aplikasi. Pada tahap ini, kita berupaya memahami kebutuhan, keinginan, dan masalah yang dihadapi oleh pengguna yang akan menggunakan aplikasi tersebut (Almayda, 2022).

1. *Empathy Map*

Empathy Map digunakan untuk menggambarkan secara visual kebutuhan, sikap, dan perilaku pengguna. Alat ini berguna untuk memperoleh pemahaman yang mendalam tentang target pengguna dan membantu dalam pengambilan keputusan desain. Biasanya, *empathy map* dibagi menjadi empat kuadran: "*Says*," "*Thinks*," "*Does*," dan "*Feels*," dengan pengguna atau persona ditempatkan di pusatnya (Naim et al., 2021).

2.4.2 *Define*

Langkah selanjutnya setelah tahapan *Empathize* dalam metode *Design thinking* adalah tahap *Define*. Pada tahap ini, data yang diperoleh dari tahap *Empathize* dikumpulkan dan dianalisis untuk mengidentifikasi permasalahan yang harus diselesaikan. Tujuan dari tahap ini adalah untuk merumuskan pernyataan masalah yang jelas dan terfokus berdasarkan wawasan yang didapatkan dari pengguna (Almayda, 2022).

1. *Pengguna Persona*

Persona berisi informasi tentang pengguna yang disurvei dan ringkasan survei yang dilakukan sebelumnya. Temuan yang didefinisikan sebagai masalah dirangkum dalam persona, yang berisi informasi detail mengenai pengguna tersebut (Maricar et al., 2022).

2. *How Might We (HMW)*

Pada tahap ini, data yang diperoleh dari hasil wawancara pada tahap *Empathize* dikelola dengan bantuan alat *How Might We* (HMW) untuk menentukan permasalahan yang akan diangkat. Pembuatan HMW dilakukan dengan cara mengubah masalah menjadi pertanyaan. Dengan mengubah masalah menjadi pertanyaan, kita dapat mengubah *mindset* sehingga masalah tersebut tampak lebih dapat diselesaikan. Pertanyaan HMW biasanya diawali dengan kesadaran (wawasan) bahwa ada *glitch* atau hambatan yang membuat pengalaman kurang optimal, serta upaya untuk mengatasinya (Rusniko, 2022).

2.4.3 *Ideate*




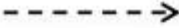
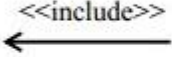

Tahap *Ideate* adalah proses menghasilkan gagasan atau ide yang dapat dijadikan solusi untuk permasalahan yang dihadapi. Pada tahap ini, berbagai ide kreatif dan inovatif dikembangkan untuk mengatasi masalah yang telah diidentifikasi pada tahap sebelumnya. Sesi *brainstorming* dan teknik berpikir kreatif lainnya sering digunakan untuk mendorong munculnya berbagai kemungkinan solusi (Maricar et al., 2022). Dalam konteks ini, UML (*Unified Modeling Language*) dapat digunakan sebagai alat bantu visual untuk memodelkan dan mengkomunikasikan solusi yang dihasilkan pada tahap *Ideate*.

UML adalah singkatan dari "*Unified Modeling Language*" yang merupakan metode pemodelan visual untuk merancang sistem berorientasi objek. UML dapat dianggap sebagai bahasa standar untuk visualisasi, perancangan, dan dokumentasi sistem perangkat lunak. Saat ini, UML telah menjadi bahasa standar dalam pembuatan *blueprint* perangkat lunak (Raharjo, 2018). Terdapat beberapa diagram UML yang dapat digunakan untuk menggambarkan aplikasi yang akan dikembangkan seperti *use case diagram*, *Activity Diagram*, dan *class diagram*.

A. *Use Case Diagram*

Use Case Diagram adalah model yang menggambarkan perilaku atau behavior dari suatu sistem yang akan dibangun. *Use Case Diagram* digunakan untuk mengidentifikasi fungsi-fungsi yang ada dalam sistem tersebut serta siapa saja yang memiliki hak untuk

menggunakan fungsi-fungsi tersebut (Siagian, 2020). Notasi yang digunakan dalam *Use case* diagram dijelaskan pada gambar 2.2.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi


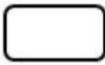
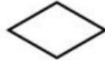


Gambar 2. 2 Notasi *Use case Diagram*

Sumber (Dicoding Team Intern, 2021)

Gambar 2.2 merupakan notasi *use case diagram* dimana terdapat 5 komponen yaitu aktor untuk mendefinisikan atau mewakili peran pengguna aplikasi, *use case* merupakan interaksi antara aktor dengan sistem, *association* merupakan penghubung antara actor dengan *use case*, *generalisasi* merupakan spesialisasi actor untuk berpartisipasi dengan *use case*, *<<include>>* menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya, dan *<<extend>>* menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari suatu *use case* lainnya.

B. Activity Diagram

Activity Diagram adalah diagram yang menjelaskan alur kerja atau kegiatan dalam program yang sedang dirancang. Alur atau aktivitas ini bisa berupa menu-menu atau proses bisnis yang ada dalam sistem tersebut (Pujiyanti, 2020). Notasi yang digunakan dalam diagram aktivitas dijelaskan pada gambar 2.3.

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2. 3 Notasi *Activity Diagram*

Sumber (Pujiyanti, 2020)

Gambar 2.3 menunjukkan notasi dalam *Activity Diagram*, yang digunakan untuk menggambarkan alur aktivitas dalam sistem. Diagram ini mencakup elemen-elemen seperti status awal yang menandakan titik mulai aktivitas, aktivitas yang mewakili tugas yang dilakukan sistem, serta percabangan dan penggabungan yang menunjukkan pilihan atau penggabungan alur berdasarkan kondisi. Selain itu, terdapat status akhir untuk menandakan

titik selesai dalam alur proses, serta swimlane yang memisahkan aktivitas berdasarkan tanggung jawab atau peran dalam organisasi, memberikan gambaran yang jelas tentang siapa yang bertanggung jawab terhadap setiap aktivitas.

C. *Class diagram*

Class diagram adalah salah satu jenis diagram struktur dalam UML yang menggambarkan detail struktur dengan jelas. *Class diagram* dapat mendeskripsikan kelas, metode, atribut, dan hubungan antar objek. Diagram ini bersifat statis, yang berarti hanya menjelaskan hubungan yang terjadi tanpa menjelaskan apa yang terjadi saat kelas-kelas tersebut berhubungan. Fungsi utama *class diagram* adalah menggambarkan struktur sebuah sistem (Saniyyah, 2023).

2.4.4 *Prototype*

Tahap *Prototype* adalah langkah berikutnya dalam proses *Design thinking*, yang bertujuan untuk membuat rancangan awal dari suatu produk yang akan dikembangkan. Pada tahap ini, ide-ide yang dihasilkan pada tahap Ideate diwujudkan dalam bentuk prototipe yang dapat diuji dan dievaluasi (Nursyifa et al., 2023).

1. Alur Proses Bisnis Aplikasi

Alur proses bisnis merupakan rangkaian aktivitas yang disusun oleh perusahaan untuk mencapai tujuannya, baik dalam bentuk produk maupun jasa. Aktivitas-aktivitas ini diatur secara sistematis agar dapat mewujudkan tujuan perusahaan secara efektif. Dalam praktiknya, proses ini dibagi menjadi beberapa sub-proses, di mana setiap bagian memiliki tugas dan tanggung jawab yang berbeda. Secara keseluruhan, tujuan dari proses bisnis adalah untuk memenuhi kebutuhan pelanggan dan mencapai target organisasi (Mahatmavidya, 2021).

2. Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna disini menggunakan *low fidelity wireframe*. *Low fidelity wireframe* merupakan desain kasar yang dibuat tanpa memperhatikan ukuran dan akurasi piksel. Tujuan pembuatan *low fidelity wireframe* adalah untuk menentukan tata letak dasar, sehingga biasanya dibuat saat sebuah ide pertama kali disusun (Rahmalia, 2023).

3. Perancangan *Database*

Perancangan *database* adalah langkah untuk membuat desain sistem basis data yang akan menunjang operasional perusahaan (Connolly & Bagg, 2005). Proses ini membutuhkan sebuah metodologi yang dapat membantu pada tahap perancangan.

Pendekatan struktural digunakan dalam metodologi ini, yang melibatkan prosedur, alat, teknik, serta dokumen pendukung agar proses perancangan dapat berjalan lebih efisien. Melalui penggunaan metode ini, proyek pengembangan basis data dapat direncanakan, diatur, dikelola, diawasi, dan dievaluasi dengan lebih efektif (Connolly & Bagg, 2005).

4. Implementasi Program

Implementasi program dilakukan sebagai tahap penting dalam penerapan PWA pada aplikasi Masjid Al-Ikhlas Sorosutan. Proses ini dimulai dengan persiapan instalasi framework yang diperlukan, seperti React.js untuk memastikan aplikasi berfungsi optimal di perangkat *desktop* dan *mobile*. Tahap terakhir adalah pengujian dan hosting aplikasi agar dapat diakses oleh pengguna secara aman dan lancar.

5. Perancangan pengujian aplikasi

Pengujian perangkat lunak merupakan metode yang digunakan untuk memastikan apakah aplikasi telah memenuhi persyaratan yang diinginkan serta memastikan produk bebas dari kerusakan atau cacat. Proses ini mencakup pemeriksaan komponen dan sistem perangkat lunak dengan menggunakan alat manual maupun otomatis.

2.4.5 *Testing*

Tahap terakhir dalam *design thinking* adalah tahap pengujian (*Testing*). Pada tahap ini, *prototype* yang telah dibuat diuji secara nyata oleh pengguna. Tujuan dari tahap ini adalah untuk memperoleh pemahaman tentang pengalaman pengguna dan mendapatkan umpan balik terhadap evaluasi produk atau aplikasi yang telah diimplementasikan (Almayda, 2022).

1. *System Usability Scale (SUS)*

System Usability Scale (SUS) adalah metode pengujian yang digunakan untuk mengukur kinerja sebuah sistem. Pengujian SUS dilakukan dengan memberikan kuesioner kepada pengguna, yang terdiri dari 10 pertanyaan dengan 5 pilihan jawaban. Skala jawaban berkisar dari 1 hingga 5, yang masing-masing berarti sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju (Susilo, 2019).

Pertanyaan dalam pengujian *System Usability Scale (SUS)* memiliki dua tipe, yaitu pertanyaan bernada positif dan negatif. Pertanyaan bernomor ganjil (1, 3, 5, 7, 9) bernada positif, sementara pertanyaan bernomor genap (2, 4, 6, 8, 10) bernada negatif. Bentuk pertanyaan dalam pengujian SUS ditunjukkan dalam tabel 2.1:

Tabel 2. 1 Pertanyaan Pengujian *System Usability Scale* (SUS)

Sumber: (Susilo, 2019)

No	Pertanyaan
1	Saya berfikir saya akan sering menggunakan sistem ini
2	Saya merasa sistem ini terlalu rumit padahal dapat dibuat lebih sederhana
3	Saya rasa sistem ini mudah untuk digunakan
4	Saya pikir saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini
5	Saya menemukan bahwa terdapat berbagai macam fitur yang terintegrasi dengan baik dalam sistem
6	Saya rasa banyak hal yang tidak konsisten terdapat pada sistem ini
7	Saya rasa mayoritas pengguna akan dapat memahami sistem ini dengan cepat
8	Saya menemukan bahwa sistem ini sangat tidak praktis ketika digunakan / membingungkan
9	Saya merasa tidak ada hambatan dan sangat yakin dapat menggunakan sistem ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Adapun pilihan jawaban untuk pertanyaan pengujian dan skala nilai pada *System Usability Scale* (SUS) ditunjukkan pada tabel 2.2.

Tabel 2. 2 Pilihan Jawaban Pengujian *System Usability Scale* (SUS)

Sumber: (Susilo, 2019)

Jawaban	Skor Nilai
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu-ragu	3
Setuju	4
Sangat Setuju	5

Untuk menentukan kesimpulan penilaian kinerja sistem, metode *System Usability Scale* (SUS) menggunakan perhitungan skor berdasarkan rumus yang telah ditetapkan. Untuk pertanyaan bernomor ganjil, skor dihitung dengan cara skor nilai yang diberikan

responden (X) dikurangi dengan 1, seperti ditunjukkan pada persamaan 2.1 (Putra, 2023):

$$\text{Skor Pertanyaan Ganjil} = X - 1 \quad (2.1)$$

Sedangkan untuk pertanyaan bernomor genap, skor dihitung dengan cara nilai 5 dikurangi dengan skor nilai yang diberikan oleh responden (X) (Putra, 2023). Ditunjukkan pada persamaan (2.2).

$$\text{Skor Pertanyaan Genap} = 5 - X \quad (2.2)$$

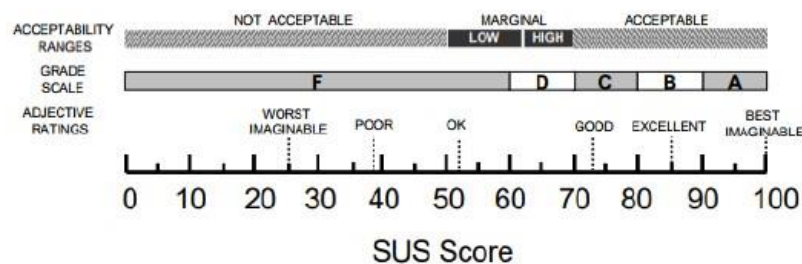
Dari hasil tersebut dilakukan perhitungan sub total yang didapatkan dari hasil penjumlahan skor setiap soal pertanyaan ganjil dan genap (Putra, 2023). Ditunjukkan pada persamaan (2.3).

$$\text{Skor sub total} = \text{Total Skor Ganjil} + \text{Total Skor Genap} \quad (2.3)$$

Setelah mendapatkan skor sub total, dilakukan perhitungan skor akhir responden dengan cara mengalikan skor sub total dengan nilai 2,5 (Putra, 2023). Ditunjukkan pada persamaan (2.4).

$$\text{Skor akhir responden} = \text{Sub Total} \times 2,5 \quad (2.4)$$

Setelah menghitung semua skor akhir dari tiap responden, langkah selanjutnya adalah melakukan perhitungan skor akhir *System Usability Scale* (SUS). Caranya adalah dengan mencari rata-rata dari penjumlahan seluruh skor akhir tiap responden, kemudian membaginya dengan jumlah responden (Putra, 2023). Dengan skor akhir SUS ini, dapat diperoleh kesimpulan tentang kinerja sistem dengan menyesuaikan skor akhir tersebut dengan kategori penilaian SUS yang ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Skala Penilaian *System Usability Scale*

Sumber: (Susilo, 2019)

2. *Black Box Testing*

Black Box Testing adalah metode pengujian yang berfokus pada spesifikasi fungsionalitas dari perangkat lunak tanpa mengacu pada struktur kode pemrogramannya (Febiharsa et al., 2018). Pengujian dilakukan dengan memberikan masukan sebagai pengujian spesifikasi fungsional aplikasi untuk mengetahui apakah perangkat lunak sudah berjalan seperti yang diharapkan (Putra, 2023). Penggunaan *Black Box Testing* bertujuan untuk menemukan beberapa kategori bug, antara lain:

- a. Fungsionalitas yang hilang atau salah
- b. Kesalahan pada desain antarmuka tampilan perangkat lunak
- c. Kesalahan pada struktur basis data
- d. Kesalahan pada kinerja perangkat lunak

3. *Lighthouse Testing*

Lighthouse adalah alat yang disediakan oleh Chrome untuk membantu meningkatkan kualitas aplikasi web (Fikri et al., 2023). Pengujian menggunakan *Lighthouse* akan secara otomatis melakukan serangkaian tes pada halaman yang ditentukan, kemudian menghasilkan laporan mengenai hasil pengujian halaman tersebut (Fikri et al., 2023). Berikut merupakan metode perhitungan lighthouse yang terdiri dalam 4 kategori :

a. *Performance*

Kategori ini mengukur seberapa cepat halaman web memuat dan merespons interaksi pengguna. Beberapa metrik yang digunakan untuk menilai *performance* meliputi :

1. *First Contentful Paint (FCP)*

FCP mengukur durasi sejak pengguna pertama kali membuka halaman hingga konten apa pun dari halaman tersebut tampil di layar (Walton, 2023). Nilai *FCP* yang baik adalah 1,8 detik atau kurang. Nilai buruk lebih besar dari 3,0 detik (Walton, 2023). Dalam versi lighthouse 10 yang digunakan penulis, *FCP* memiliki bobot 10%.

2. *Largest Contentful Paint (LCP)*

LCP melaporkan waktu render gambar, blok teks, atau video terbesar yang terlihat di area pandang, dibandingkan saat pengguna pertama kali membuka halaman (Walton & Pollard, 2023). Nilai *LCP* yang baik

adalah 2,5 detik atau kurang (Walton & Pollard, 2023). Dalam versi lighthouse 10 yang digunakan penulis, FCP memiliki bobot 25%.

3. *Speed Index*

Skor Indeks Kecepatan adalah perbandingan indeks kecepatan halaman dan indeks kecepatan situs sebenarnya, berdasarkan data dari Arsip HTTP (Walton, 2023). 0–3,4 = Hijau (cepat), 3,4–5,8 = Oranye (sedang), Di atas 5,8 = Merah (lambat) (Walton, 2023). Dalam versi lighthouse 8 yang digunakan penulis, FCP memiliki bobot 10%.

4. *Total Blocking Time* (TBT)

Untuk memberikan pengalaman pengguna yang baik, situs harus mengusahakan agar TBT kurang dari 200 milidetik saat diuji di *hardware* seluler rata-rata (Walton & Pollard, 2023). Dalam versi lighthouse 8 yang digunakan penulis, FCP memiliki bobot 30%.

5. *Cumulative Layout Shift* (CLS)

CLS adalah ukuran burst terbesar dari skor pergeseran tata letak untuk setiap pergeseran tata letak yang tidak terduga yang terjadi selama seluruh siklus proses halaman (Mihajlija & Walton, 2023). Pergeseran tata letak terjadi setiap kali elemen yang terlihat mengubah posisinya dari satu *frame* yang dirender ke *frame* berikutnya (Mihajlija & Walton, 2023). Nilai CLS yang baik adalah 0,1 atau kurang. Nilai buruk lebih besar dari 0,25. Dalam versi lighthouse 8 yang digunakan penulis, FCP memiliki bobot 25%.

b. *Accessibility*

Skor Aksesibilitas lighthouse adalah rata-rata tertimbang dari semua audit aksesibilitas (Phillip, 2019). Bobot didasarkan pada penilaian dampak pengguna. Nilai aksesibilitas didapatkan dari pengujian aplikasi terhadap tabel audit dan bobot yang didapatkan sesuai dengan bobot tersedia (Phillip, 2019).

Tabel 2. 3 Tabel Audit Lighthouse Aksesibilitas

Audit	Berat
Nilai [accesskey] bersifat unik	7
Atribut [aria-*] cocok dengan perannya	10
Nilai yang ditetapkan untuk role="" adalah peran ARIA yang valid	1

Audit	Berat
Elemen button, link, dan menuitem memiliki label aksesibilitas	7
Elemen dengan role="dialog" atau role="alertdialog" memiliki label aksesibilitas	7
[aria-hidden="true"] tidak ada di dokumen <body>	10
Elemen [aria-hidden="true"] tidak berisi turunan yang dapat difokuskan	7
Kolom input ARIA memiliki nama yang dapat diakses	7
Elemen meter ARIA memiliki label aksesibilitas	7
Elemen progressbar ARIA memiliki label aksesibilitas	7
[role] memiliki semua atribut [aria-*] yang diperlukan	10
Elemen dengan [role] ARIA yang mewajibkan turunan berisi [role] tertentu memiliki semua turunan yang diperlukan	10
[role] dimuat oleh elemen induk wajibnya	10
Nilai [role] valid	7
Elemen dengan atribut role=text tidak memiliki turunan yang dapat difokuskan	7
Kolom beralih ARIA memiliki nama yang dapat diakses	7
Elemen tooltip ARIA memiliki nama yang dapat diakses	7
Elemen treeitem ARIA memiliki label aksesibilitas	7
Atribut [aria-*] memiliki nilai yang valid	10
Atribut [aria-*] valid dan tidak salah eja	10
Tombol memiliki label aksesibilitas	10
Halaman berisi judul, link lewati, atau wilayah tempat terkenal	7
Warna latar belakang dan latar depan memiliki rasio kontras yang memadai	7
<dl> hanya berisi grup <dt> dan <dd>, elemen <script>, <template>, atau <div> yang diurutkan dengan tepat	7
Item daftar definisi digabungkan dalam elemen <dl>	7
Dokumen memiliki elemen <title>	7
Atribut [id] pada elemen aktif yang dapat difokuskan unik	7
ID ARIA unik	10
Tidak ada kolom formulir yang memiliki beberapa label	3
Elemen <frame> atau <iframe> memiliki judul	7
Elemen judul muncul dalam urutan menurun secara berurutan	3
Elemen <html> memiliki atribut [lang]	7
Elemen <html> memiliki nilai yang valid untuk atribut [lang] miliknya	7
Elemen <html> memiliki atribut [xml:lang] dengan bahasa dasar yang sama dengan atribut [lang]	3
Elemen gambar memiliki atribut [alt]	10

Audit	Berat
Elemen gambar tidak memiliki atribut [alt] yang merupakan teks berlebihan	1
Tombol input memiliki teks yang jelas	10
Elemen <code><input type="image"></code> memiliki teks [alt]	10
Elemen dengan label teks yang terlihat memiliki label aksesibilitas yang cocok	7
Elemen formulir memiliki label terkait	7
Link dapat dibedakan tanpa bergantung pada warna	7
Link memiliki nama yang dapat dikenali	7
Daftar hanya berisi elemen <code></code> dan elemen pendukung skrip (<code><script></code> dan <code><template></code>)	7
Item daftar (<code></code>) dimuat dalam elemen induk <code></code> , <code></code> , atau <code><menu></code>	7
Dokumen tidak menggunakan <code><meta http-equiv="refresh"></code>	10
[<code>pengguna-scalable="no"</code>] tidak digunakan dalam elemen <code><meta name="viewport"></code> dan atribut [<code>maximum-scale</code>] tidak kurang dari 5	10
Elemen <code><object></code> memiliki teks alternatif	7
Elemen "select" memiliki elemen label terkait	7
Link lewati dapat difokuskan	3
Tidak ada elemen yang memiliki nilai [<code>tabindex</code>] lebih besar dari 0	7
Tabel memiliki konten yang berbeda di atribut ringkasan dan <code><caption></code>	1
Tabel menggunakan <code><caption></code> , bukan sel dengan atribut [<code>colspan</code>] untuk menunjukkan teks	7
Elemen <code><td></code> dalam <code><table></code> besar memiliki satu atau beberapa header tabel	10
Sel dalam elemen <code><table></code> yang menggunakan atribut [<code>headers</code>] merujuk ke sel tabel dalam tabel yang sama	7
Elemen <code><th></code> dan elemen dengan [<code>role="columnheader"/"rowheader"</code>] memiliki sel data yang dideskripsikan	7
Atribut [<code>lang</code>] memiliki nilai yang valid	7
Elemen <code><video></code> berisi elemen <code><track></code> dengan [<code>kind="captions"</code>]	10

c. *Best Practices*

Best practices dalam Google Lighthouse menganalisis penerapan praktik terbaik dalam pembangunan situs web yang mencakup beberapa aspek teknis penting (Rahmalia, 2022). Salah satunya adalah penggunaan HTTPS

untuk mengamankan komunikasi data antara pengguna dan server, melindungi dari potensi serangan. Selain itu, Lighthouse juga menilai keamanan *outbound links* atau tautan keluar untuk memastikan bahwa tautan yang digunakan mengarah ke sumber yang tepercaya dan aman. Penggunaan JavaScript terbaru juga diuji untuk memastikan bahwa situs tidak mengandung pustaka atau skrip yang usang dan rentan terhadap masalah keamanan. Secara keseluruhan, Lighthouse mengidentifikasi dan mengevaluasi berbagai praktik terbaik yang penting untuk meningkatkan kinerja, keamanan, dan keandalan situs web.

d. *SEO (Search Engine Optimization)*

Pemeriksaan SEO memastikan bahwa halaman *website* mengikuti saran pengoptimalan mesin telusur dasar. Seperti halaman tidak diblokir dari pengindeksan, dokumen memiliki elemen <title>, dokumen memiliki deskripsi meta, halaman memiliki kode status HTTP yang berhasil, link dapat di-crawl robots.txt valid, elemen halaman memiliki atribut [alt], dokumen memiliki hreflang yang valid, dokumen memiliki rel=canonical yang valid.

2.5 Kajian Pustaka

Penulis menggunakan beberapa penelitian sejenis sebagai referensi dalam mempelajari dan memperluas wawasan dalam melakukan penelitian ini. Telah terdapat penelitian sebelumnya yang membahas tentang implementasi dan pengembangan aplikasi menggunakan teknologi Progressive Web App, yang menjadi landasan dalam penelitian ini.

Pada penelitian yang dilakukan oleh Ari Yudha Prawira, Ikhwan Ruslianto, dan Dian Prawira tahun 2023 dengan judul "Rancang Bangun Sistem Informasi Manajemen Masjid Menggunakan *Progressive Web Apps* (Studi Kasus: Masjid Di Kota Pontianak)." Hasil penelitian menunjukkan bahwa sistem telah berhasil memberikan kontribusi yang signifikan terhadap masjid. Pengujian terhadap pengurus masjid menunjukkan bahwa sistem berjalan sesuai dengan rancangan yang telah ditetapkan, serta memberikan tingkat transparansi yang tinggi dalam informasi pengelolaan masjid, mencakup kegiatan, keuangan, fasilitas, kepengurusan, inventaris, dan pelaporan.

Pada penelitian yang dilakukan oleh Hendriyana dan Somantri pada tahun 2021 dengan judul "Implementasi *Progressive Web Apps (PWA)* Pada Aplikasi keuangan SMK Yapan". Hasil penelitian ini menunjukkan bahwasanya penggunaan aplikasi keuangan ini terbukti cukup

efektif dan sangat membantu dalam mengelola keuangan di sekolah tersebut. Selain itu, aplikasi keuangan ini juga dapat diakses oleh siswa untuk melihat cicilan yang harus mereka bayar. Implementasi *progressive web apps* pada aplikasi keuangan ini juga berjalan dengan baik, memungkinkan aplikasi tetap dapat diakses secara *offline* dan berjalan secara responsif pada berbagai perangkat, seperti komputer, laptop, atau perangkat *mobile*. Hasil dari *Black Box Testing* menunjukkan bahwa semua fitur aplikasi berjalan dengan baik dan normal.

Pada penelitian yang dilakukan oleh Gharizi Matiini, Rahmat Setiyadi, Adi Setiawan, dan M. Ramli pada tahun 2021 dengan judul "Pengembangan *Aplikasi Progressive Web Application (PWA)* Untuk Pembelajaran dan Evaluasi Kelas *English Grammar Online Course*". Hasil penelitian ini menunjukkan bahwasanya PWA telah berhasil terpasang dengan baik di ponsel cerdas berbasis *Android* maupun *iOS*, dan telah digunakan untuk mengakses konten pembelajaran Bahasa Inggris pada *website* Pusat Bahasa ITI ketika terhubung dengan jaringan *online*. Evaluasi pembelajaran *English Grammar eCourse* menunjukkan bahwa sistem ini merupakan alternatif pembelajaran yang lebih inovatif dan interaktif dibandingkan dengan metode konvensional.

Tabel 2. 4 Kajian Pustaka Penelitian Terdahulu

No	Penelitian	Teknologi	Metode Pengujian	Hasil Penelitian
1	Rancang Bangun Sistem Informasi Manajemen Masjid Menggunakan <i>Progressive Web Apps</i> (Studi Kasus: Masjid Di Kota Pontianak) (Prawira et al., 2023)	<i>Progressive Web App, Framework Laravel</i>	<i>Lighthouse Testing, Black Box Testing</i>	Aplikasi dapat berjalan pada jaringan yang tidak stabil maupun <i>offline</i> serta bisa dipasang secara langsung pada <i>homescreen</i> perangkat tanpa harus mendownloadnya terlebih dahulu. Hasil pengujian fungsional sistem menunjukkan bahwasanya sistem ini mampu memberikan tingkat transparansi yang tinggi dalam menyajikan informasi terkait pengelolaan masjid, mencakup aspek kegiatan, keuangan,

No	Penelitian	Teknologi	Metode Pengujian	Hasil Penelitian
				fasilitas, kepengurusan, inventaris, dan pelaporan.
2	Implementasi <i>Progressive Web Aps (PWA)</i> Pada Aplikasi keuangan SMK Yapan. (Hendriyana & Somantri, 2021)	<i>Progressive Web App, Framework Laravel</i>	<i>Black Box Testing</i>	Implementasi <i>progressive web apps</i> pada aplikasi keuangan ini berjalan dengan baik, memungkinkan aplikasi tetap dapat diakses secara <i>offline</i> dan berjalan secara responsif pada berbagai perangkat, seperti komputer, laptop, atau perangkat <i>mobile</i> . Hasil dari <i>Black Box Testing</i> menunjukkan bahwa semua fitur aplikasi berjalan dengan baik dan normal.
3	Pengembangan Aplikasi <i>Progressive Web Application (PWA)</i> Untuk Pembelajaran dan Evaluasi Kelas <i>English Grammar Online Course</i> . (Matiini et al., 2021)	<i>Progressive Web App,</i>	<i>PWA Builder Testing</i>	Dari hasil pengujian <i>PWA Builder</i> yang telah dilakukan maka dapat diambil kesimpulan bahwa performansi untuk wilayah di Indonesia adalah baik dan <i>mobile app</i> telah siap. <i>PWA</i> untuk <i>website</i> Pusat Bahasa ITI (https://pb.iti.ac.id) sudah diimplementasikan, namun masih terdapat kekurangan pada ketersediaan <i>service worker</i> untuk mendapatkan layanan ketika <i>website</i> dalam kondisi <i>offline</i> .

BAB III

METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu *Design Thinking*. *Design thinking* adalah pendekatan yang dikenal sebagai proses berpikir menyeluruh, yang mengutamakan penciptaan solusi dengan memulai dari pemahaman mendalam terhadap kebutuhan spesifik yang berfokus pada manusia (*human-centered*) dan terus berlanjut untuk menghasilkan inovasi yang berkelanjutan sesuai dengan kebutuhan pengguna (Nadhif et al., 2021). *Design Thinking* terdiri dari 5 tahapan dimulai dari *Empathize* untuk memahami kebutuhan dan masalah pengguna, dilanjutkan dengan tahapan *Define* untuk mendefinisikan permasalahan pengguna, *Ideate* untuk merumuskan gagasan atau solusi dari permasalahan pengguna, *Prototype* untuk merancang produk yang akan dikembangkan, hingga tahap *Testing* untuk menguji produk yang dibuat (Shirvanadi & Idris, 2021).

3.1 Empathize

Langkah pertama dalam pendekatan *design thinking* adalah tahap *empathize*, yang mengharuskan pengumpulan data berdasarkan preferensi dan pengalaman pengguna. Tujuannya adalah untuk memahami secara mendalam dan merasakan masalah yang dihadapi oleh pengguna dengan fokus pada penggalian informasi yang relevan. Untuk mendapatkan informasi tentang kebutuhan pengguna, penulis melakukan beberapa proses seperti wawancara, observasi, dan pembuatan *empathy map*.

3.1.1 Observasi

Dalam tahap ini, penulis melakukan observasi dengan mendatangi langsung lokasi objek penelitian, yaitu Masjid Al-Ikhlas Sorosutan, serta menggali informasi melalui internet yang berkaitan dengan masjid tersebut. Observasi langsung memungkinkan penulis untuk melihat secara nyata kondisi dan aktivitas di masjid, termasuk interaksi antar pengguna dan fasilitas yang tersedia. Pengumpulan informasi melalui internet memberikan data tambahan tentang masjid lokasi penelitian. Kombinasi dari kedua sumber ini memberikan gambaran yang komprehensif tentang kebutuhan dan pengalaman pengguna masjid. Selanjutnya, hasil dari observasi ini akan dianalisis untuk mengidentifikasi masalah utama dan area yang memerlukan perbaikan atau inovasi. Hasil dari observasi yang telah dilakukan dibuat ke dalam bentuk tabel yang terdapat pada tabel 3.1.

Tabel 3. 1 Hasil Observasi

No	Hasil Observasi
1	Lokasi Masjid Al-Ikhlas Sorosutan berada didalam gang dan cukup dekat dengan pusat kota, yakni sekitar empat kilometer.
2	Berdasarkan data kantor kelurahan Sorosutan, mayoritas masyarakat sekitar Sorosutan menganut agama Islam, yakni lebih dari 93,63%.
3	Masjid Al-Ikhlas Sorosutan pernah mengalami pencurian uang infak dengan estimasi kerugian mencapai 50 juta rupiah pada 25 Agustus 2021 (Leon, 2021).
4	Masjid Al-Ikhlas Sorosutan sudah menggunakan CCTV dalam hal keamanan.
5	Para takmir tidak menetap di masjid Al-Ikhlas Sorosutan.
6	Masyarakat sekitar masjid aktif melakukan kegiatan salat berjamaah.

Tabel 3.1 menunjukkan hasil observasi mengenai Masjid Al-Ikhlas Sorosutan. Lokasi masjid ini berada di dalam gang yang cukup dekat dengan pusat kota, sekitar empat kilometer dari pusat kota. Berdasarkan data dari kantor kelurahan setempat, mayoritas masyarakat di sekitar masjid beragama Islam, dengan persentase mencapai 93,63%. Masjid ini juga pernah mengalami pencurian uang infak dengan estimasi kerugian mencapai 50 juta rupiah pada 25 Agustus 2021. Untuk meningkatkan keamanan, Masjid Al-Ikhlas sudah memasang CCTV. Selain itu, para takmir masjid tidak menetap di sekitar masjid, namun masyarakat setempat aktif melakukan kegiatan salat berjamaah.

3.1.2 Wawancara

Wawancara merupakan sebuah interaksi antara dua pihak atau lebih di mana pewawancara bertujuan untuk mengajukan pertanyaan-pertanyaan guna mengumpulkan informasi dari responden. Tujuan utama dari wawancara adalah untuk mendapatkan pemahaman yang lebih mendalam tentang topik yang sedang dibahas melalui perspektif dan pengetahuan yang dimiliki oleh responden (Sakti, 2023). Untuk memastikan bahwa proses wawancara dilakukan secara sistematis dan sesuai dengan tujuan penelitian, penulis menyusun sebuah laporan wawancara. Laporan ini bertujuan agar wawancara dapat dilakukan dengan terarah dan hasilnya bermanfaat dalam menyusun solusi. Berikut adalah laporan wawancara yang digunakan oleh penulis sebagai acuan dalam melakukan wawancara dengan pengguna.

a. Tujuan

Tujuan dari wawancara ini adalah untuk mendapatkan pemahaman tentang aktivitas yang terjadi di Masjid Al-Ikhlas Sorosutan serta untuk mengeksplorasi permasalahan

yang mungkin timbul di masjid tersebut. Penulis perlu memahami kedua aspek ini agar dapat mengidentifikasi fitur-fitur yang relevan dan dapat membantu pihak takmir masjid dalam pengelolaannya.

b. Metode

Penulis menggunakan teknik wawancara semi-terstruktur untuk mendapatkan pemahaman yang lebih mendalam. Dalam jenis wawancara ini, fokusnya adalah untuk memungkinkan narasumber berbicara secara lebih terbuka, mengemukakan pendapat serta ide-ide mereka. (Sugiyono, 2007) menjelaskan bahwa wawancara semi-terstruktur termasuk dalam kategori *in-depth interview*, di mana pendekatannya lebih fleksibel daripada wawancara terstruktur karena memungkinkan pengembangan topik serta pertanyaan yang lebih spontan.

c. Topik Wawancara

Topik wawancara yaitu terkait permasalahan yang terjadi di Masjid Al-Ikhlas Sorosutan. Termasuk aktivitas dan upaya yang pernah dilakukan oleh pihak takmir dalam mengatasi masalah tersebut.

d. Narasumber

Target narasumber yang akan diwawancarai adalah pihak takmir Masjid Al-Ikhlas Sorosutan. Nama yang akan diwawancarai ialah Bapak Nuri Subandi. Bapak Nuri Subandi bertugas sebagai ketua takmir Masjid Al-Ikhlas Sorosutan. Pihak tersebut merupakan pihak Takmir yang akan membantu penulis dalam menggali dan mengumpulkan informasi penelitian ini.

e. Lokasi wawancara

Wawancara dilakukan di Masjid Al-Ikhlas Sorosutan, karena narasumber biasanya berada di lokasi ketika waktu salat.

f. Waktu wawancara

Wawancara berlangsung pada pukul 16:00-17:00 WIB pada tanggal 13 Januari 2024.

3.1.3 Empathy Map

Empathy map adalah sebuah metode yang berpusat pada pengguna, yang bertujuan untuk memahami perspektif individu lain melalui pengamatan dan interaksi langsung (Bratsberg, 2012). Pendekatan ini bermanfaat ketika seseorang ingin mendalami pengalaman pengguna dalam konteks mereka. Tujuan utama dari pembuatan *empathy map* adalah untuk menggantikan asumsi dengan fakta yang diperoleh melalui wawancara dan observasi, sehingga memungkinkan perbaikan yang lebih baik terhadap pengalaman pengguna.

Dalam proses membuat *empathy map*, penulis memilih topik wawancara yang relevan serta mempertimbangkan narasumber yang tepat. Setelah menyelesaikan proses wawancara, langkah selanjutnya adalah menyusun laporan hasil wawancara. Laporan ini menjadi pijakan utama dalam pembuatan *empathy map* karena membantu penulis mengorganisir dan menggambarkan data atau informasi yang terkumpul secara sistematis. Fokus wawancara ini terarah pada aktivitas dan tantangan yang dialami di Masjid Al-Ikhlas Sorosutan. Berikut adalah *empathy map* yang dibuat berdasarkan hasil observasi dan wawancara dengan pengguna:



Gambar 3. 1 *Empathize Map* Takmir Masjid Al-Ikhlas Sorosutan.

Dari *Empathy Map* takmir Masjid Al-Ikhlas, kita dapat mengetahui bahwa masjid ini dikelola oleh 15 takmir dengan mayoritas berusia antara 30 hingga 60 tahun. Beberapa dari mereka bekerja atau berdagang, sedangkan yang lain tinggal di rumah. Struktur organisasi masjid terdiri dari ketua, sekretaris, bendahara, dan anggota lainnya yang bertugas mengelola kegiatan masjid. Mereka secara rutin mengadakan rapat untuk koordinasi dan perencanaan kegiatan masjid. Takmir juga menyebutkan bahwa keamanan kotak infak menjadi prioritas utama, dengan penggunaan gembok beralarm dan CCTV untuk memonitor aktivitas di sekitar kotak infak tersebut.

Dalam kesehariannya, takmir Masjid Al-Ikhlas aktif mengadakan rapat rutin setiap dua minggu atau sebulan sekali untuk membahas berbagai kegiatan dan isu-isu terkait masjid. Mereka bertanggung jawab untuk memonitor kotak infak melalui CCTV dan melakukan pencatatan keuangan setiap minggu setelah salat Jumat. Azan dikumandangkan sesuai waktu salat, meskipun tidak ada jadwal tetap untuk siapa yang akan melakukannya. Selain itu, takmir juga mengelola kegiatan Taman Pendidikan Al-Quran (TPA) dengan menggunakan modul yang terbatas dan jumlah pengajar yang sedikit. Untuk menjaga keamanan masjid, mereka menggunakan CCTV dan melakukan pengawasan berkala.

Takmir memiliki beberapa pemikiran untuk meningkatkan efektivitas dan keamanan masjid. Mereka mempertimbangkan untuk meningkatkan keamanan kotak infak dengan cara yang lebih modern dan efektif. Mereka juga ingin memperbaiki sistem pencatatan keuangan agar lebih transparan dan efisien, serta mengatasi masalah keterlambatan dalam pengumandangan azan. Takmir berencana untuk mengaktifkan kembali TPA dengan modul yang lebih baik guna meningkatkan kualitas pendidikan agama anak-anak. Mereka juga berpikir untuk meningkatkan partisipasi jamaah dalam kegiatan masjid dan mempertimbangkan penggunaan teknologi untuk manajemen masjid yang lebih baik.

Secara emosional, takmir Masjid Al-Ikhlas merasa sangat bertanggung jawab terhadap keamanan dan pengelolaan masjid. Mereka sering merasa frustrasi karena sistem pencatatan keuangan yang ada saat ini sering rusak atau hilang, yang menghambat transparansi dan efisiensi. Takmir juga merasa khawatir dengan keterlambatan pengumandangan azan dan merasa terbebani dengan jumlah pengajar TPA yang sedikit, yang berdampak pada kualitas pendidikan agama. Mereka merasa perlu adanya solusi modern untuk meningkatkan keamanan dan manajemen masjid secara keseluruhan. Takmir menginginkan cara-cara baru yang lebih efisien dan efektif dalam mengelola masjid dan memastikan semua aspek operasional berjalan dengan lancar.

Agar lebih mudah memahami *empathy map* takmir Masjid Al-Ikhlas Sorosutan, berikut penulis akan menyajikan dalam bentuk tabel yang ditunjukkan pada tabel 3.2.

Tabel 3. 2 *Empathize Map* Takmir Masjid Al-Ikhlas Sorosutan.

Bagian dari Emapthize Map	<i>Empathize Map</i>
<i>Says</i>	Ada 15 takmir.
	Berusia 30-60 tahun.
	Sebagian besar takmir bekerja atau berdagang, ada yang berdiam di rumah.
	Struktur organisasi terdiri dari ketua, sekretaris, bendahara, dan anggota lainnya.
	Rapat rutin untuk koordinasi kegiatan.
	Pengamanan kotak infak menggunakan gembok yang dilengkapi alarm, dan CCTV.
<i>Does</i>	Mengadakan rapat rutin dua minggu/satu bulan sekali.
	Memonitor kotak infak dengan CCTV.
	Melakukan pencatatan keuangan dengan buku setiap minggu, setelah salat Jumat.
	Azan dikumandangkan sesuai waktu salat, tidak ada jadwal pasti siapa yang mengumandangkan.
	Mengelola kegiatan TPA dengan modul yang terbatas dan jumlah pengajar yang sedikit.
	Menjaga keamanan masjid dengan CCTV dan pengawasan berkala.
<i>Thinks</i>	Meningkatkan keamanan kotak infak.
	Memperbaiki pencatatan keuangan yang transparan dan efisien
	Mengatasi keterlambatan pengumandangan azan.
	Mengaktifkan kembali TPA dengan modul yang lebih baik.
	Mempertimbangkan penggunaan teknologi untuk keterlambatan azan salat
	Meningkatkan partisipasi jamaah dalam kegiatan masjid.

Bagian dari Emapthize Map	<i>Empathize Map</i>
<i>Feels</i>	Merasa bertanggung jawab terhadap keamanan masjid.
	Frustrasi dengan pencatatan keuangan yang sering rusak atau hilang.
	Khawatir tentang keterlambatan azan.
	Terbebani dengan sedikitnya jumlah pengajar TPA.
	Perlu solusi modern untuk manajemen masjid.
	Merasa perlu cara yang modern untuk meningkatkan keamanan dan pengelolaan masjid.

3.2 Define

Pada tahap *define*, penulis fokus untuk mengidentifikasi kebutuhan pengguna serta melakukan analisis dan sintesis informasi yang diperoleh. Informasi tersebut sebelumnya telah dikumpulkan selama tahap *empathize* melalui observasi dan wawancara semi-terstruktur. Tahap *define* ini bertujuan untuk memahami kebutuhan dan permasalahan pengguna yang telah diidentifikasi dari tahap empati. Penulis akan merumuskan pernyataan masalah yang bermakna dan *actionable*, sehingga dapat dijadikan tantangan dalam menciptakan solusi yang efektif.

Dalam pendekatan *design thinking*, terdapat proses iterasi yang memungkinkan untuk meninjau kembali tahap-tahap sebelumnya. Iterasi pada tahap *define* dilakukan oleh penulis dengan tujuan untuk memperoleh ide yang lebih jelas mengenai kebutuhan dan permasalahan yang dialami oleh takmir Masjid Al-Ikhlas Sorosutan. Proses iterasi ini melibatkan peninjauan kembali tahap *empathize*, yang sebelumnya difokuskan melalui peta empati (*empathy map*).

Pernyataan masalah yang jelas dan tepat sangat penting pada tahap ini, karena hal tersebut akan membantu penulis untuk fokus pada kebutuhan spesifik pengguna yang telah diidentifikasi. Selain itu, penentuan masalah yang akurat akan mendukung penulis dalam menciptakan ide-ide berkelanjutan pada tahap *ideate* berikutnya.

3.2.1 Pernyataan Masalah

Berdasarkan hasil dari pembuatan *empathy map* yang telah didesain, yaitu *empathy map* takmir Masjid Al-Ikhlas Sorosutan, maka dapat diambil kesimpulan bahwa terdapat 4 permasalahan inti yang dialami oleh takmir.

Tabel 3. 3 Pernyataan Masalah Takmir Masjid Al-Ikhlas Sorosutan

Kode	Pernyataan Masalah	Keterangan
PM_01	Takmir Masjid Al-Ikhlas menghadapi kesulitan dalam memastikan keamanan kotak infak karena metode pengamanan manual yang masih terbatas.	Terdapat pada bagian <i>Empathy Map</i> di bagian <i>Thinks</i> dan <i>Feels</i> mengenai kebutuhan untuk meningkatkan keamanan kotak infak.
PM_02	Pencatatan keuangan masjid yang dilakukan secara manual dengan buku sering mengalami kerusakan atau hilang, menghambat transparansi dan efisiensi.	Terdapat pada bagian <i>Empathy Map</i> di bagian <i>Feels</i> dan <i>Does</i> terkait frustrasi dan kerepotan dalam pencatatan keuangan.
PM_03	Azan salat sering kali terlambat dikumandangkan karena tidak ada jadwal tetap untuk petugas azan.	Terdapat pada bagian <i>Empathy Map</i> di bagian <i>Does</i> , <i>Thinks</i> , dan <i>Feels</i> tentang keterlambatan azan dan kurangnya jadwal tetap.
PM_04	Kegiatan Taman Pendidikan Al-Qur'an (TPA) terbatas pada modul yang tidak terstruktur dan jumlah pengajar yang sedikit, yang mengakibatkan terbatasnya aktifitas pendidikan agama di masjid.	Terdapat pada bagian <i>Empathy Map</i> di bagian <i>Does</i> , <i>Thinks</i> , dan <i>Feels</i> terkait keterbatasan modul TPA dan jumlah pengajar.

Tabel 3.3 menggambarkan masalah yang dihadapi Takmir Masjid Al-Ikhlas Sorosutan. Masalah pertama terkait keamanan kotak infak yang menggunakan metode manual terbatas. Masalah kedua adalah pencatatan keuangan manual yang sering mengalami kerusakan, menghambat transparansi. Masalah ketiga adalah keterlambatan azan salat akibat tidak adanya jadwal tetap untuk petugas azan. Masalah keempat terkait terbatasnya kegiatan TPA akibat modul yang tidak terstruktur dan jumlah pengajar yang sedikit.

3.2.2 Pengguna Persona

Pengguna persona dibuat berdasarkan data pengguna yang dikumpulkan pada saat wawancara. Data pengguna yang dibutuhkan dalam pembuatan *pengguna persona* terdiri dari biografi persona, *quote* persona (kalimat yang relevan dan sesuai dengan latar belakang pengguna), *goals* persona (tujuan yang ingin dicapai pengguna), *frustations* persona (masalah yang membuat pengguna frustrasi), dan informasi tambahan yang diperlukan. Selanjutnya adalah membuat sebuah *pengguna persona* sebagai aktor pengguna sistem aplikasi seperti pada Gambar 3.2.

“Saya ingin masjid ini menjadi tempat yang aman dan terkelola dengan baik, agar jamaah merasa nyaman dan lebih aktif dalam kegiatan masjid.”



Nuri Subandi

Umur: 53 Tahun
Pendidikan: SMA
Domisili: Sorosutan,
 Kota Yogyakarta
Status Menikah: Menikah
Pekerjaan: Wiraswasta

Goals

- Meningkatkan keamanan kotak infak di masjid untuk mencegah kehilangan atau kecurangan.
- Meningkatkan efisiensi dan transparansi dalam pencatatan keuangan masjid.
- Mengatasi masalah keterlambatan adzan dengan cara yang lebih terstruktur dan terjadwal.
- Mengaktifkan kembali dan meningkatkan kualitas kegiatan TPA untuk anak-anak di lingkungan masjid.

Frustrations

- Pencatatan keuangan manual yang memakan waktu dan sering kali rusak atau hilang.
- Keamanan kotak infak yang kurang memadai, sering kali membuatnya khawatir.
- Keterlambatan adzan yang mengganggu kegiatan sholat berjamaah di masjid.
- Kesulitan mengelola kegiatan TPA dengan sumber daya yang terbatas.

Nuri Subandi adalah ketua takmir Masjid Al-Ikhlas Sorosutan. Dia sering kali merasa frustrasi dengan keamanan kotak infak yang rentan, pengelolaan keuangan manual yang merepotkan, dan keterlambatan adzan karena tidak adanya jadwal pasti. Selain itu, Ahmad ingin menghidupkan kembali kegiatan TPA yang kurang efektif. Dia berharap aplikasi baru dapat meningkatkan keamanan, efisiensi keuangan, ketepatan adzan, dan kualitas TPA.

Gambar 3. 2 *Pengguna Persona*

Gambar 3.2 menunjukkan *Pengguna Persona* Nuri Subandi, seorang ketua takmir Masjid Al-Ikhlas Sorosutan. Nuri Subandi berusia 53 tahun, berpendidikan SMA, dan bekerja sebagai wiraswasta. Dia memiliki tujuan untuk meningkatkan keamanan kotak infak, efisiensi pencatatan keuangan, serta mengatasi masalah keterlambatan azan dan keterbatasan kegiatan TPA. Namun, dia juga menghadapi frustrasi akibat masalah seperti kotak infak yang mudah hilang, pencatatan keuangan manual yang rentan rusak, serta kurangnya jadwal tetap petugas azan. Nuri berharap aplikasi dapat membantu meningkatkan keamanan, efisiensi keuangan, ketepatan azan, dan kualitas kegiatan TPA di masjid.

3.2.3 *The “How Might We” Method*

Metode *How Might We* digunakan untuk merumuskan tantangan desain yang akan membantu dalam menciptakan solusi yang inovatif dan *actionable*. Berdasarkan pernyataan masalah yang telah diidentifikasi, berikut adalah beberapa *How Might We statements* dari aplikasi ini:

1. *How Might We Keamanan Kotak Infak*

Masalah:

Takmir Masjid Al-Ikhlas menghadapi kesulitan dalam memastikan keamanan kotak infak karena metode pengamanan manual yang masih terbatas.

How Might We Statements:

- *How might we* meningkatkan keamanan kotak infak di Masjid Al-Ikhlas untuk mencegah kehilangan dan kerusakan?
- *How might we* menggunakan teknologi untuk membuat sistem pengamanan kotak infak yang lebih efisien dan efektif?

2. **How Might We Pencatatan Keuangan**

Masalah:

Pencatatan keuangan masjid yang dilakukan secara manual dengan buku sering mengalami kerusakan atau hilang, menghambat transparansi dan efisiensi.

How Might We Statements:

- *How might we* menciptakan sistem pencatatan keuangan digital yang mudah digunakan oleh takmir masjid untuk meningkatkan transparansi dan efisiensi?
- *How might we* memastikan data keuangan masjid tetap aman dan terorganisir dengan baik?

3. **How Might We Jadwal Azan Salat**

Masalah:

Azan salat sering kali terlambat dikumandangkan karena tidak ada jadwal tetap untuk petugas azan.

How Might We Statements:

- *How might we* menyusun jadwal tetap untuk petugas azan agar waktu salat bisa dilaksanakan tepat waktu?
- *How might we* membuat sistem pengingat otomatis untuk petugas azan?

4. **How Might We Kegiatan Taman Pendidikan Al-Qur'an (TPA)**

Masalah:

Kegiatan Taman Pendidikan Al-Qur'an (TPA) terbatas pada modul yang tidak terstruktur dan jumlah pengajar yang sedikit, yang mengakibatkan terbatasnya aktivitas pendidikan agama di masjid.

How Might We Statements:

- *How might we* menyusun modul pendidikan Al-Qur'an yang lebih terstruktur dan menarik untuk meningkatkan kualitas pembelajaran di TPA?

3.3 Ideate

Setelah melalui tahap *define*, penulis kini fokus pada solusi yang ditemukan untuk mengatasi masalah pengguna dan menganalisis kebutuhan perangkat lunak yang akan digunakan dalam perancangan sistem. Pada analisis perancangan sistem, salah satu langkah awal pengembangan adalah menetapkan karakteristik sistem yang mampu menyelesaikan masalah tersebut. Sistem ini akan dirancang menggunakan metode UML (*Unified Modeling Language*), yang akan menyediakan informasi penting dalam proses pemrograman. Metode ini mencakup identifikasi aktor dan diagram-diagram seperti *Use case Diagram*, *Activity Diagram*, dan *Class diagram*.

3.3.1 Identifikasi Aktor

Sesuai dengan rancangan alur sistem, aplikasi ini memiliki dua aktor yaitu pengguna dan admin. Pengguna merupakan takmir Masjid Al-Ikhlas Sorosutan dapat melakukan pemantauan kotak infak, menambah dan mengelola catatan keuangan masjid, melihat jadwal waktu salat, dan melihat modul pembelajaran TPA. Sedangkan admin dapat melakukan penghapusan data pengguna.

3.3.2 Identifikasi Use Case

Untuk menggambarkan berbagai proses yang terjadi dalam aplikasi ini, setiap fungsi aplikasi diidentifikasi dan dikelompokkan ke dalam beberapa *Use Case* beserta deskripsinya, yang dapat dilihat pada Tabel 3.4.

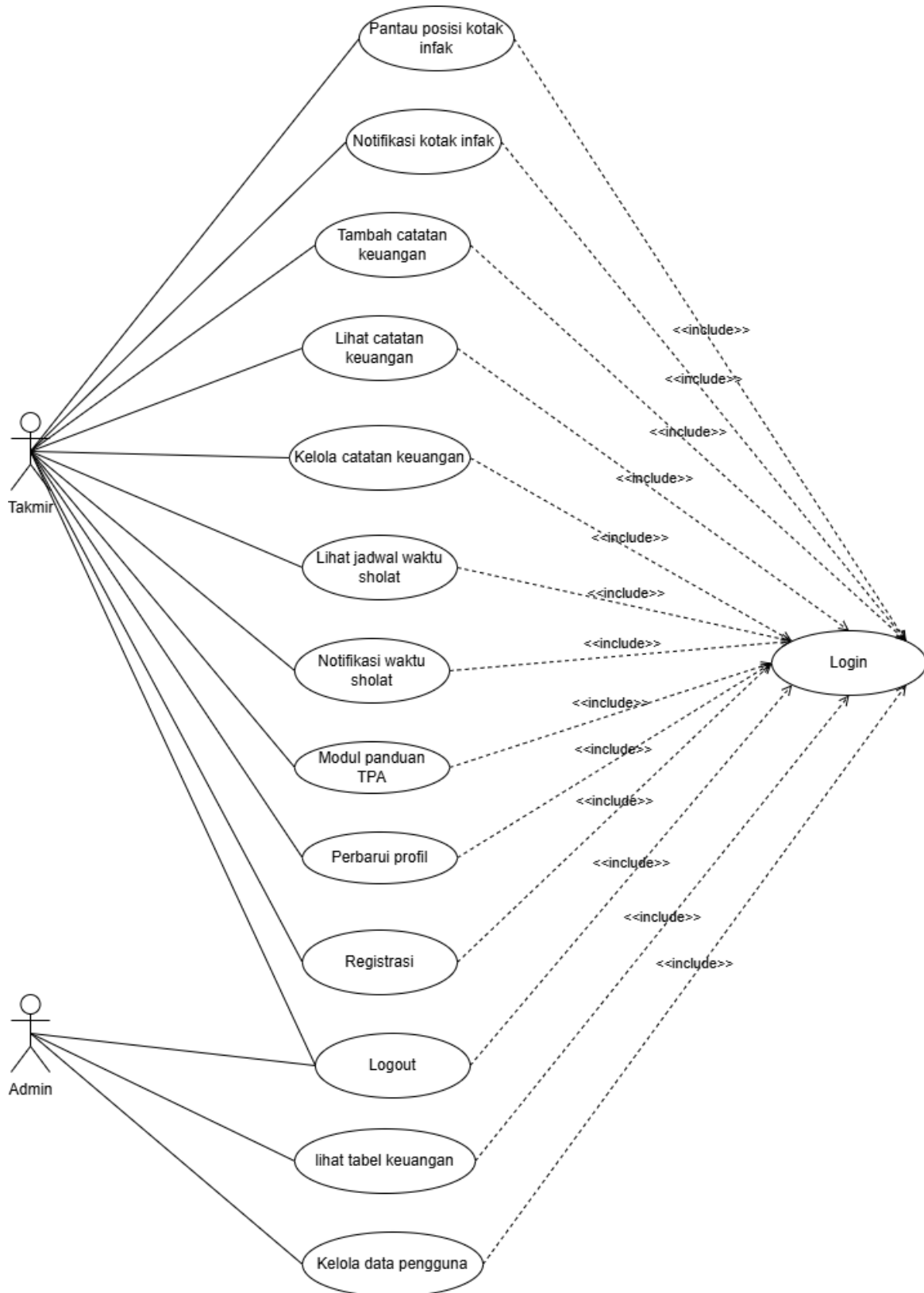
Tabel 3. 4 Identifikasi *Use case*

No	Aktor	<i>Use case</i>	Deskripsi
1.	Takmir	Pantau posisi kotak infak	Menjelaskan aktivitas aktor dalam memantau dimana posisi kotak infak masjid berada
2.	Takmir	Notifikasi kotak infak	Menjelaskan bagaimana aktor menerima notifikasi apabila kotak infak dicuri
3.	Takmir	Tambah catatan keuangan	Menjelaskan aktivitas aktor dalam melakukan pencatatan keuangan masjid baik yang masuk (pemasukan) maupun yang keluar (pengeluaran).
4.	Takmir	Lihat catatan keuangan	Menjelaskan aktivitas aktor dalam melihat catatan keuangan (pemasukan dan pengeluaran) masjid yang telah ditambahkan oleh aktor dalam bentuk tabel atau grafik.

No	Aktor	<i>Use case</i>	Deskripsi
5.	Takmir	Kelola catatan keuangan	Menjelaskan bagaimana aktor dapat mengubah dan menghapus catatan keuangan masjid yang mereka tambahkan.
6.	Takmir	Lihat jadwal waktu salat	Menjelaskan aktivitas aktor dalam melihat jadwal waktu salat
7.	Takmir	Notifikasi azan salat	Menjelaskan bagaimana aktor akan menerima notifikasi apabila akan memasuki waktu salat.
8.	Takmir	Modul panduan mengajar TPA	Menjelaskan aktivitas aktor dalam melihat modul panduan mengajar TPA.
9.	Takmir	Perbarui profil pengguna	Menjelaskan aktivitas aktor dalam melakukan perubahan terhadap <i>profile</i> mereka.
10.	Takmir	Registrasi	Menjelaskan aktivitas aktor dalam mendaftarkan diri untuk bisa mengakses aplikasi.
11.	Takmir	<i>Login Takmir</i>	menjelaskan aktivitas aktor dalam masuk ke dalam sistem aplikasi
12.	Takmir	<i>Logout Takmir</i>	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari aplikasi
13.	Admin	Mengelola data pengguna aplikasi	Menjelaskan bagaimana admin dapat menghapus data pengguna aplikasi
14.	Admin	Mengelola data keuangan	Menjelaskan bagaimana admin dapat melihat data keuangan termasuk siapa <i>pengguna</i> yang menambahkan
15.	Admin	<i>Login Admin</i>	Menjelaskan aktivitas aktor dalam masuk ke dalam menu admin
16.	Admin	<i>Logout Admin</i>	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari menu admin

3.3.3 Diagram *Use case*

Untuk mengilustrasikan hubungan interaksi antara pengguna dan sistem, dibuatlah sebuah diagram *Use case* yang ditampilkan pada Gambar 3.4. Diagram ini menjelaskan rancangan aplikasi berdasarkan identifikasi kebutuhan dan *Use case* dari tahap sebelumnya.



Gambar 3. 3 Diagram *Use case*

3.3.4 Skenario *Use case*

Untuk menggambarkan alur aktivitas pengguna pada setiap *Use case*, dibuatlah skenario penggunaan yang ditampilkan pada Tabel 3.5 hingga Tabel 3.19.

1. *Use case* pantau posisi kotak infak

Tabel 3. 5 *Use case* Pantau Posisi Kotak Infak

<i>Use case</i> Pantau Posisi Kotak Infak	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam memantau dimana posisi kotak infak masjid berada
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan tombol " <i>Monitoring</i> Kotak Infak" pada beranda aplikasi.	2. Mengambil data posisi kotak infak dari <i>database</i>
	3. menampilkan titik posisi kotak infak pada <i>maps</i>
Kondisi Akhir	Sistem menampilkan titik posisi kotak infak pada peta

Tabel 3.5 menggambarkan *use case* "Pantau Posisi Kotak Infak" yang bertujuan untuk menjelaskan aktivitas takmir dalam memantau lokasi kotak infak masjid menggunakan aplikasi. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aksi aktor meliputi menekan tombol "*Monitoring* Kotak Infak" pada beranda aplikasi. Sistem kemudian merespons dengan mengambil data posisi kotak infak dari *database* dan menampilkannya dalam bentuk titik lokasi pada peta menggunakan fitur *maps*. Kondisi akhir dari skenario ini adalah sistem berhasil menampilkan posisi kotak infak pada peta.

2. *Use case* notifikasi kotak infakTabel 3. 6 *Use case* Notifikasi Kotak Infak

Use case Notifikasi Kotak Infak	
Aktor	Takmir
Deskripsi	Menjelaskan bagaimana takmir akan menerima notifikasi apabila kotak infak berada di luar jarak yang telah ditentukan
Skenario Utama	
Kondisi Awal	Kotak infak berada di luar lokasi masjid.
Aksi Aktor	Reaksi Sistem
	1. Sistem secara periodik mengambil <i>database</i> lokasi kotak infak.
	2. Sistem membandingkan lokasi kotak infak dengan radius yang telah ditentukan dalam program.
	3. Sistem mengirimkan notifikasi ke <i>handphone</i> takmir.
Kondisi Akhir	Aktor menerima notifikasi bahwasanya kotak infak telah berada di luar lokasi masjid.

Tabel 3.6 menggambarkan *use case* “Notifikasi Kotak Infak” yang bertujuan untuk menjelaskan bagaimana takmir menerima notifikasi apabila kotak infak berada di luar radius jarak yang telah ditentukan. Skenario utama dimulai dengan kondisi awal, yaitu kotak infak terdeteksi berada di luar lokasi masjid. Sistem secara periodik mengambil data lokasi kotak infak dari *database* dan membandingkannya dengan radius yang telah ditentukan dalam program. Jika kotak infak berada di luar radius tersebut, sistem akan mengirimkan notifikasi ke *handphone* takmir. Kondisi akhir dari skenario ini adalah takmir menerima pemberitahuan bahwa kotak infak telah keluar dari lokasi masjid.

3. *Use case* tambah catatan keuanganTabel 3. 7 *Use case* Tambah Catatan Keuangan

Use case Tambah Catatan Keuangan	
Aktor	Takmir
Deskripsi	Menjelaskan bagaimana aktor akan menambahkan catatan keuangan
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Catatan Keuangan” pada beranda aplikasi.	2. Sistem mengambil data keuangan masjid dari <i>database</i> keuangan.
	3. Sistem menampilkan Saldo Keuangan Masjid, Total Pemasukan, dan Total Pengeluaran.
4. Aktor menekan tombol “Tambah Catatan Keuangan”	
5. Aktor memilih tombol “Tambah Pemasukan” atau “Tambah Pengeluaran”	
	6. Sistem menampilkan formulir “Catatan Pemasukan” atau “Catatan Pengeluaran”
7. Aktor mengisi formulir “Catatan Pemasukan” atau “Catatan Pengeluaran”	
8. Aktor menekan tombol “Masukan”	9. Sistem memperbarui data keuangan pada <i>database</i> .
	10. Sistem mengirimkan <i>alert</i> “Success, Berhasil Membuat Catatan”
Kondisi Akhir	Aktor berhasil membuat catatan keuangan dan sistem berhasil memperbarui data

Tabel 3.7 menggambarkan *use case* “Tambah Catatan Keuangan” yang bertujuan untuk menjelaskan bagaimana takmir menambahkan catatan keuangan masjid menggunakan aplikasi. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “Catatan Keuangan” pada beranda aplikasi, lalu sistem mengambil data keuangan masjid dari *database* dan menampilkan informasi Saldo Keuangan Masjid, Total Pemasukan, dan Total Pengeluaran. Aktor kemudian menekan tombol “Tambah Catatan Keuangan” dan memilih opsi “Tambah Pemasukan” atau “Tambah Pengeluaran”. Setelah itu, sistem menampilkan formulir sesuai pilihan aktor. Aktor mengisi formulir tersebut dan

menekan tombol “Masukan”. Sistem memperbarui data keuangan pada *database* dan mengirimkan *alert* “*Success, Berhasil Membuat Catatan*”. Kondisi akhir adalah aktor berhasil membuat catatan keuangan dan sistem berhasil memperbarui data keuangan.

4. *Use case* lihat catatan keuangan

Tabel 3. 8 *Use Case* Lihat Catatan Keuangan

Use case Lihat Catatan Keuangan	
Aktor	Takmir
Deskripsi	Menampilkan catatan keuangan (pemasukan dan pengeluaran) kas masjid yang telah ditambahkan oleh aktor dalam bentuk tabel atau grafik.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Catatan Keuangan” pada beranda aplikasi.	2. Sistem mengambil data keuangan masjid dari <i>database</i> keuangan.
	3. Sistem menampilkan Saldo Keuangan Masjid, Total Pemasukan, dan Total Pengeluaran.
4. Aktor menekan tombol “Lihat Catatan Keuangan”	5. Sistem menampilkan halaman “Laporan Keuangan Masjid”
6. Aktor menekan tombol “Tabel Keuangan” atau “Grafik Keuangan”	7. Sistem mengambil data keuangan dari <i>database</i> keuangan.
	8. Sistem menampilkan tabel atau grafik keuangan.
6. Aktor mensortir tabel sesuai dengan tanggal, kategori tabel yang ingin ditampilkan (semua, pemasukan, dan pengeluaran), dan kriteria urutan tabel (urutan berdasarkan kategori, nominal terbesar, dan nominal terkecil). Atau aktor mensortir grafik sesuai dengan tahun dan grafik yang ingin ditampilkan (semua, pemasukan, dan pengeluaran).	7. Sistem menampilkan tabel atau grafik keuangan yang telah disortir.
Kondisi Akhir	Aktor berhasil melihat tabel atau grafik keuangan yang telah disortir.

. Tabel 3.8 menggambarkan *use case* “Lihat Catatan Keuangan” yang bertujuan untuk menampilkan catatan keuangan masjid, termasuk pemasukan dan pengeluaran, dalam bentuk tabel atau grafik. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “Catatan Keuangan”

pada beranda aplikasi, dan sistem mengambil data keuangan masjid dari *database* untuk menampilkan Saldo Keuangan Masjid, Total Pemasukan, dan Total Pengeluaran. Aktor kemudian menekan tombol “Lihat Catatan Keuangan,” dan sistem menampilkan halaman “Laporan Keuangan Masjid.” Aktor memilih tampilan dalam bentuk tabel atau grafik dengan menekan tombol yang sesuai, dan sistem mengambil data dari *database* keuangan untuk menampilkan hasil tersebut. Aktor dapat menyortir tabel berdasarkan tanggal, kategori (semua, pemasukan, pengeluaran), atau kriteria urutan (kategori, nominal terbesar, nominal terkecil). Untuk grafik, aktor dapat menyortir berdasarkan tahun atau kategori yang ingin ditampilkan. Sistem kemudian menampilkan tabel atau grafik keuangan yang telah disortir. Kondisi akhir adalah aktor berhasil melihat tabel atau grafik keuangan sesuai dengan preferensi yang telah disortir.

5. *Use case* mengelola data catatan keuangan

Tabel 3. 9 *Use Case* Mengelola Data Catatan Keuangan

<i>Use case</i> Mengelola Data Catatan Keuangan	
Aktor	Takmir
Deskripsi	Menjelaskan bagaimana aktor dapat mengubah dan menghapus data catatan keuangan masjid yang telah mereka buat.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Catatan Keuangan” pada beranda aplikasi.	2. Sistem mengambil data keuangan masjid dari <i>database</i> keuangan.
	3. Sistem menampilkan Saldo Keuangan Masjid, Total Pemasukan, dan Total Pengeluaran.
4. Aktor menekan tombol “Lihat Catatan Keuangan”	5. Sistem menampilkan halaman “Laporan Keuangan Masjid”
6. Aktor menekan tombol “Tabel Keuangan”	7. Sistem mengambil data keuangan dari <i>database</i> dan ditampilkan dalam tabel.
8. Aktor menekan tombol “edit” pada kolom aksi.	9. Sistem mengambil data pada baris yang diklik edit, lalu ditampilkan ke dalam formulir edit data keuangan.
10. Aktor melakukan pembaruan data dan menekan tombol “perbarui”.	11. memperbarui data pengguna pada <i>database</i>
12. Aktor menekan tombol “hapus” pada kolom aksi.	13. menghapus catatan keuangan dari <i>database</i> .

	14. sistem menyimpan perubahan data keuangan pada <i>database</i> .
Kondisi Akhir	Perubahan data catatan keuangan tersimpan di <i>database</i> .

Tabel 3.9 menggambarkan *use case* “Mengelola Data Catatan Keuangan” yang bertujuan untuk menjelaskan bagaimana takmir dapat mengubah dan menghapus data catatan keuangan masjid yang telah dibuat. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “Catatan Keuangan” pada beranda aplikasi, kemudian sistem mengambil data keuangan masjid dari *database* dan menampilkan Saldo Keuangan Masjid, Total Pemasukan, serta Total Pengeluaran. Aktor melanjutkan dengan menekan tombol “Lihat Catatan Keuangan” sehingga sistem menampilkan halaman “Laporan Keuangan Masjid.” Aktor memilih opsi “Tabel Keuangan” dan sistem mengambil data dari *database* untuk ditampilkan dalam bentuk tabel. Untuk mengubah data, aktor menekan tombol “edit” pada kolom aksi, lalu sistem menampilkan formulir edit data keuangan berdasarkan data pada baris yang dipilih. Aktor memperbarui data dan menekan tombol “perbarui” kemudian sistem menyimpan perubahan tersebut ke dalam *database*. Untuk menghapus data, aktor menekan tombol “hapus” pada kolom aksi, dan sistem menghapus catatan tersebut dari *database*. Kondisi akhir adalah perubahan atau penghapusan data catatan keuangan berhasil disimpan di *database*.

6. *Use case* lihat jadwal waktu salat

Tabel 3. 10 *Use Case* Lihat Jadwal Waktu Salat

Use case Lihat Jadwal Waktu Salat	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam melihat jadwal waktu salat.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol “Jadwal Waktu Salat”	2. Sistem mengambil jadwal waktu salat dari API aladhan.com
	3. Sistem menampilkan halaman “Jadwal Waktu Salat”
Kondisi Akhir	Aktor berhasil melihat jadwal waktu salat.

Tabel 3.10 menggambarkan *use case* Lihat Jadwal Waktu Salat yang bertujuan untuk menjelaskan aktivitas takmir dalam melihat jadwal waktu salat melalui aplikasi.

Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “Jadwal Waktu Salat,” kemudian sistem mengambil data jadwal waktu salat dari API aladhan.com. Setelah itu, sistem menampilkan halaman “Jadwal Waktu Salat” yang berisi informasi waktu salat. Kondisi akhir adalah aktor berhasil melihat jadwal waktu salat yang ditampilkan oleh sistem.

7. *Use case* notifikasi azan salat

Tabel 3. 11 *Use Case* Notifikasi Azan Salat

<i>Use case</i> Notifikasi Azan Salat	
Aktor	Takmir
Deskripsi	Menjelaskan bagaimana aktor akan menerima notifikasi apabila akan memasuki waktu salat.
Skenario Utama	
Kondisi Awal	<i>Handphone</i> aktor dalam keadaan mati / terkunci.
Aksi Aktor	Reaksi Sistem
	1. Sistem mengambil jadwal waktu salat dari <i>virtual private server</i> (VPS).
	2. Sistem mengirimkan notifikasi 10 menit sebelum waktu salat.
Kondisi Akhir	Aktor menerima notifikasi “10 menit lagi akan datang waktu salat, segera kumandangkan azan”

Tabel 3.11 menggambarkan *use case* Notifikasi Azan Salat yang bertujuan untuk menjelaskan bagaimana takmir menerima notifikasi ketika waktu salat akan tiba. Skenario utama dimulai dengan kondisi awal, yaitu *handphone* aktor dalam keadaan mati atau terkunci. Sistem secara otomatis mengambil jadwal waktu salat dari *virtual private server* (VPS) dan mengirimkan notifikasi 10 menit sebelum waktu salat tiba. Kondisi akhir adalah aktor menerima notifikasi yang berbunyi “10 menit lagi akan datang waktu salat, segera kumandangkan azan.”

8. *Use case* modul panduan mengajar TPATabel 3. 12 *Use Case* Modul Panduan Mengajar TPA

Use case Modul Panduan Mengajar TPA	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam melihat modul panduan mengajar TPA.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol “Panduan Mengajar TPA”	2. Sistem menampilkan halaman beranda panduan mengajar TPA.
3. Aktor memilih bab modul panduan mengajar TPA.	4. Sistem menampilkan beranda bab modul panduan mengajar TPA.
5. Aktor memilih materi panduan mengajar TPA.	6. Sistem menampilkan materi panduan mengajar TPA.
Kondisi Akhir	Aktor melihat materi panduan mengajar TPA,

Tabel 3.12 menggambarkan *use case* Modul Panduan Mengajar TPA yang bertujuan untuk menjelaskan aktivitas takmir dalam melihat modul panduan mengajar TPA melalui aplikasi. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “Panduan Mengajar TPA” dan sistem menampilkan halaman beranda panduan mengajar TPA. Aktor kemudian memilih bab modul yang diinginkan, dan sistem menampilkan halaman bab tersebut. Selanjutnya, aktor memilih materi panduan mengajar yang spesifik, dan sistem menampilkan isi materi tersebut. Kondisi akhir adalah aktor berhasil melihat materi panduan mengajar TPA.

9. *Use case* perbarui profil penggunaTabel 3. 13 *Use Case* Perbarui Profil Pengguna

Use case Perbarui Kata Sandi	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam melakukan perubahan terhadap profil pengguna.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan icon “ <i>profile pengguna</i> ” pada <i>navbar</i> aplikasi	2. Sistem mengambil data <i>pengguna</i> dari <i>database</i> .
	3. Sistem menampilkan data pengguna (nama pengguna, email, alamat, nomor telepon).
4. Aktor menekan tombol “perbarui profil pengguna”.	5. Sistem mengambil uid pengguna dan menampilkan data pengguna pada form perbarui data pengguna
6. Aktor melakukan perubahan pada form perbarui data pengguna.	
7. Aktor menekan tombol perbarui	8. Sistem menyimpan perubahan data pengguna ke dalam <i>database</i> .
	9. Sistem menampilkan <i>alert</i> , “Success, Data Berhasil Diperbarui”
Kondisi Akhir	Aktor berhasil melakukan perubahan pada profil mereka.

Tabel 3.13 menggambarkan *use case* Perbarui Profil Pengguna yang bertujuan untuk menjelaskan aktivitas takmir dalam melakukan perubahan pada profil pengguna melalui aplikasi. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan ikon “*profile pengguna*” pada *navbar* aplikasi, kemudian sistem mengambil data pengguna dari *database* dan menampilkan informasi seperti nama pengguna, email, alamat, dan nomor telepon. Aktor selanjutnya menekan tombol “perbarui profil pengguna” lalu sistem menampilkan data pengguna pada formulir perbarui data. Aktor melakukan perubahan pada formulir tersebut dan menekan tombol “perbarui” Sistem menyimpan perubahan data ke dalam *database* dan memberikan notifikasi berupa *alert*, “Success, Data Berhasil Diperbarui” Kondisi akhir adalah aktor berhasil melakukan perubahan pada profil mereka.

10. *Use case* Registrasi takmirTabel 3. 14 *Use Case* Registrasi Takmir

Use case Registrasi	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam mendaftarkan diri untuk bisa mengakses aplikasi.
Skenario Utama	
Kondisi Awal	Halaman <i>login</i> aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan “bikin akun”.	2. Sistem menampilkan halaman registrasi.
3. Aktor mengisi form registrasi.	
4. Aktor menekan tombol “daftar”.	5. Sistem akan mengecek apabila seluruh <i>input</i> telah terisi dan <i>email</i> aktor belum terdapat dalam <i>database</i> .
	6. Sistem menambahkan data pengguna ke <i>database</i> .
	7. Sistem menampilkan <i>alert</i> “ <i>Success</i> , berhasil membuat akun”
Kondisi Akhir	Aktor berhasil membuat akun pengguna.

Tabel 3.14 menggambarkan *use case* Registrasi Takmir yang bertujuan untuk menjelaskan aktivitas takmir dalam mendaftarkan diri agar dapat mengakses aplikasi. Skenario utama dimulai dengan kondisi awal, yaitu halaman *login* aplikasi ditampilkan. Aktor menekan tombol “bikin akun” kemudian sistem menampilkan halaman registrasi. Aktor mengisi formulir registrasi dan menekan tombol “daftar” Sistem memverifikasi apakah seluruh *input* telah terisi dengan benar dan memastikan email aktor belum terdaftar dalam *database*. Jika valid, sistem menambahkan data pengguna ke *database* dan menampilkan notifikasi berupa *alert* “*Success*, berhasil membuat akun” Kondisi akhir adalah aktor berhasil membuat akun pengguna.

11. *Use case login takmir*Tabel 3. 15 *Use Case Login Takmir*

<i>Use case Login Takmir</i>	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor dalam mendaftarkan diri untuk bisa mengakses aplikasi.
Skenario Utama	
Kondisi Awal	Halaman <i>login</i> aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor mengisi form <i>login</i> .	
2. Aktor menekan tombol “Masuk”	3. Sistem mengautentikasi kecocokan <i>role</i> dan data <i>pengguna</i> dengan <i>database</i>
	4. Sistem menyimpan data <i>session</i> pengguna ke dalam peramban.
	5. Sistem menampilkan halaman beranda aplikasi.
Kondisi Akhir	Aktor berhasil <i>login</i> ke dalam aplikasi dan masuk kehalaman beranda.

Tabel 3.15 menggambarkan *use case Login Takmir* yang bertujuan untuk menjelaskan aktivitas takmir dalam mengakses aplikasi dengan akun yang telah terdaftar. Skenario utama dimulai dengan kondisi awal, yaitu halaman *login* aplikasi ditampilkan. Aktor mengisi formulir *login* dengan data akun yang dimiliki, kemudian menekan tombol “Masuk” Sistem melakukan autentikasi dengan mencocokkan peran (*role*) dan data pengguna dengan informasi yang ada di *database*. Setelah berhasil diverifikasi, sistem menyimpan data sesi pengguna ke dalam peramban dan menampilkan halaman beranda aplikasi. Kondisi akhir adalah aktor berhasil login ke dalam aplikasi dan diarahkan ke halaman beranda.

12. Use case logout takmir

Tabel 3. 16 Use Case Logout Takmir

Use case Logout Takmir	
Aktor	Takmir
Deskripsi	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari aplikasi.
Skenario Utama	
Kondisi Awal	Halaman beranda aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol “logout” pada <i>navbar</i> aplikasi.	
	2. Sistem menghapus <i>data session</i> pengguna dari peramban.
	3. Sistem mengeluarkan pengguna dari aplikasi.
	4. Sistem menampilkan halaman <i>login</i> aplikasi.
Kondisi Akhir	Aktor berhasil keluar dari aplikasi dan masuk kehalaman <i>login</i> .

Tabel 3.16 menggambarkan *use case Logout Takmir* yang bertujuan untuk menjelaskan aktivitas takmir dalam keluar dari aplikasi dan menghapus data sesi pengguna. Skenario utama dimulai dengan kondisi awal, yaitu halaman beranda aplikasi ditampilkan. Aktor menekan tombol “logout” pada *navbar* aplikasi. Sistem kemudian menghapus data sesi pengguna dari peramban, mengeluarkan pengguna dari aplikasi, dan menampilkan halaman *login*. Kondisi akhir adalah aktor berhasil keluar dari aplikasi dan diarahkan ke halaman *login*.

13. Use case mengelola data pengguna

Tabel 3. 17 Use Case Mengelola Data Pengguna

Use case Mengelola Data Pengguna Aplikasi	
Aktor	Admin
Deskripsi	Menjelaskan bagaimana admin dapat mengubah dan menghapus data pengguna aplikasi.
Skenario Utama	
Kondisi Awal	Halaman <i>login</i> aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor mengisi form <i>login</i> .	
2. Aktor menekan tombol "Masuk"	3. Sistem mengautentikasi kecocokan data <i>pengguna</i> dengan <i>database</i> . Disini juga sistem akan mengecek <i>role pengguna</i> , apabila <i>role</i> "admin" maka akan diarahkan ke halaman <i>dashboard</i> admin.
	4. Sistem menyimpan data <i>session</i> aktor ke dalam peramban.
	5. Sistem menampilkan halaman beranda admin.
6. Aktor menekan tombol "Kelola Data Pengguna".	7. Sistem mengambil data pengguna dari <i>database</i> .
	8. Sistem menampilkan halaman admin tabel pengguna.
9. Aktor menekan tombol "hapus" pada tabel.	10. Sistem menghapus data pengguna dari <i>database</i> .
	14. sistem menyimpan perubahan data pengguna pada <i>database</i> .
Kondisi Akhir	Perubahan data pengguna tersimpan di <i>database</i> .

Tabel 3.17 menjelaskan *use case* Mengelola Data Pengguna Aplikasi, di mana admin dapat mengubah dan menghapus data pengguna. Proses dimulai dari *login*, di mana sistem memverifikasi data dan *role* admin, lalu mengarahkan ke *dashboard* admin. Admin menekan tombol "Kelola Data Pengguna" sistem mengambil dan menampilkan data pengguna. Untuk menghapus data, admin menekan tombol "hapus" dan sistem menghapus data tersebut dari *database*. Kondisi akhir adalah perubahan atau penghapusan data pengguna tersimpan di *database*.

14. Use case lihat data keuangan

Tabel 3. 18 Use case Melihat Data Keuangan

Use case Melihat Data Keuangan	
Aktor	Admin
Deskripsi	Menjelaskan bagaimana admin dapat melihat data keuangan termasuk siapa pengguna yang menambahkan.
Skenario Utama	
Kondisi Awal	Halaman <i>login</i> aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor mengisi form <i>login</i> .	
2. Aktor menekan tombol "Masuk"	3. Sistem mengautentikasi kecocokan data <i>pengguna</i> dengan <i>database</i> . Disini juga sistem akan mengecek <i>role pengguna</i> , apabila <i>role</i> "admin" maka akan diarahkan ke halaman <i>dashboard</i> admin.
	4. Sistem menyimpan data <i>session</i> aktor ke dalam peramban.
	5. Sistem menampilkan halaman beranda admin.
6. Aktor menekan tombol "Kelola Data Keuangan".	7. Sistem mengambil data keuangan dari <i>database</i> .
	8. Sistem menampilkan halaman admin tabel keuangan.
Kondisi Akhir	Data keuangan termasuk siapa pengguna yang melakukan pencatatan ditampilkan.

Tabel 3.18 menjelaskan *use case* Mengelola Data Keuangan, di mana admin dapat melihat data keuangan termasuk informasi pengguna yang menambahkan data tersebut. Proses dimulai dengan admin *login*, sistem memverifikasi data dan *role* sebagai admin, lalu mengarahkan ke *dashboard* admin. Admin menekan tombol "Kelola Data Keuangan" dan sistem mengambil data keuangan dari *database* serta menampilkannya dalam tabel pada halaman admin. Kondisi akhir adalah admin dapat melihat data keuangan lengkap dengan informasi pengguna yang melakukan pencatatan.

15. Use case login admin

Tabel 3. 19 Use Case Login Admin

Use case Login Admin	
Aktor	Admin
Deskripsi	Menjelaskan aktivitas aktor dalam masuk ke dalam menu admin
Skenario Utama	
Kondisi Awal	Halaman <i>login</i> aplikasi ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor mengisi form <i>login</i> .	
2. Aktor menekan tombol “Masuk”	3. Sistem mengautentikasi kecocokan data admin dengan <i>database</i> .
	4. Sistem menyimpan data <i>session</i> admin ke dalam peramban.
	5. Sistem menampilkan halaman beranda admin.
Kondisi Akhir	Aktor berhasil <i>login</i> ke dalam aplikasi dan masuk kehalaman beranda admin.

Tabel 3.19 menjelaskan *use case Login Admin* yang menggambarkan aktivitas admin dalam mengakses menu admin pada aplikasi. Proses dimulai dengan admin mengisi formulir *login* di halaman *login* aplikasi, lalu menekan tombol “Masuk” Sistem memverifikasi data admin dengan *database*, menyimpan data sesi ke dalam peramban, dan menampilkan halaman beranda admin. Kondisi akhir adalah admin berhasil *login* dan diarahkan ke halaman beranda admin.

16. Use case logout admin

Tabel 3. 20 Use Case Logout Admin

Use case Logout Admin	
Aktor	Admin
Deskripsi	Menjelaskan aktivitas aktor keluar dari aplikasi dan menghapus data <i>session</i> dari menu admin
Skenario Utama	
Kondisi Awal	Halaman beranda admin ditampilkan.
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol “ <i>logout</i> ” pada <i>navbar</i> aplikasi.	
	2. Sistem menghapus <i>data session</i> pengguna dari peramban.
	3. Sistem mengeluarkan admin dari aplikasi.
	4. Sistem menampilkan halaman <i>login</i> aplikasi.
Kondisi Akhir	Aktor berhasil keluar dari aplikasi dan masuk kehalaman <i>login</i> .

Tabel 3.20 menjelaskan *use case Logout Admin* yang menggambarkan aktivitas admin dalam keluar dari aplikasi dan menghapus data sesi dari menu admin. Proses dimulai dengan admin berada di halaman beranda admin, kemudian menekan tombol “*logout*” pada *navbar* aplikasi. Sistem menghapus data sesi dari peramban, mengeluarkan admin dari aplikasi, dan menampilkan halaman *login*. Kondisi akhir adalah admin berhasil keluar dari aplikasi dan diarahkan ke halaman *login*.

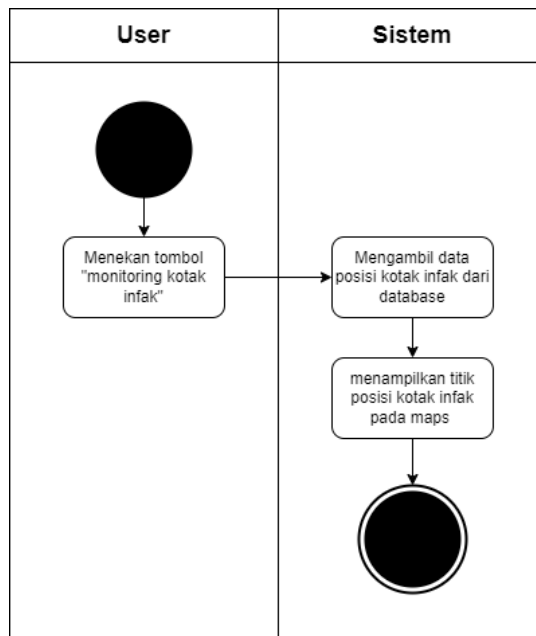
3.3.5 Activity Diagram

Untuk menggambarkan lebih lanjut alur aktivitas pengguna pada skenario *Use case*, dibuat sebuah *Activity Diagram* yang ditunjukkan pada Gambar 3.4 hingga 3.19.

1. Activity Diagram pantau posisi kotak infak

Pada bagian ini pengguna dapat memantau posisi kotak infak, pertama kali pengguna akan masuk dengan menekan tombol “*monitoring* kotak infak”. Selanjutnya sistem akan mengambil data posisi kotak infak dari *database* Firebase. Lalu sistem akan menampilkan titik koordinat lokasi kotak infak pada peta dan

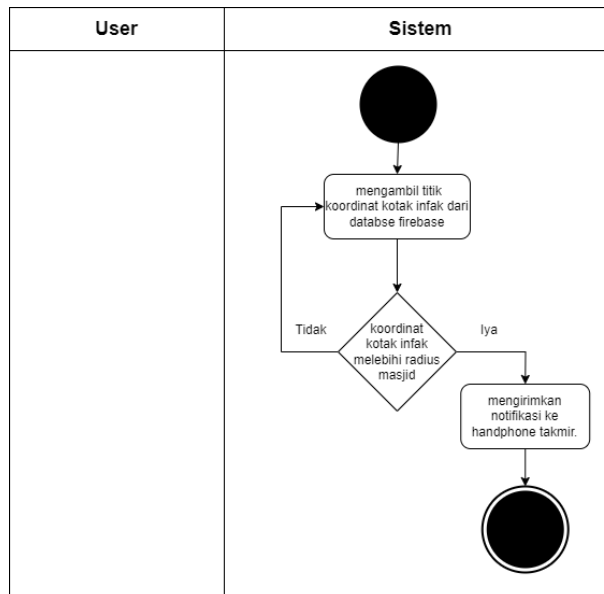
pengguna dapat memantau posisi kotak infak. *Activity diagram* pantau posisi kotak infak digambarkan pada gambar 3.4.



Gambar 3. 4 *Activity Diagram* Pantau Posisi Kotak Infak

2. *Activity Diagram* notifikasi kotak infak

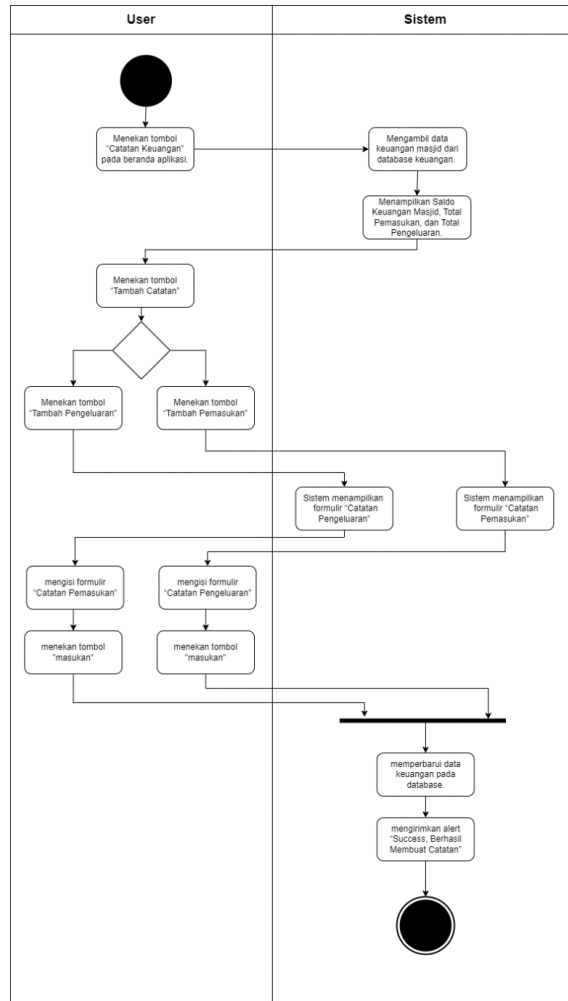
Pada bagian ini pengguna akan menerima notifikasi kotak infak. Pertama kali sistem akan mengambil titik koordinat kotak infak dari *database* firebase. Dimana *hardware Internet of Things (IoT)* akan mengirimkan titik koordinat kotak infak secara periodik setiap 30 detik. Selanjutnya sistem akan melakukan *looping* secara terus menerus, apabila posisi kotak infak berada di luar radius masjid yaitu 47 meter persegi maka akan mengirimkan notifikasi ke pengguna bahwasanya “Kotak Infak Berada Di luar Lokasi Masjid”. *Activity diagram* pantau notifikasi kotak infak digambarkan pada gambar 3.5.



Gambar 3. 5 *Activity Diagram* Notifikasi Kotak Infak

3. *Activity Diagram* tambah catatan keuangan

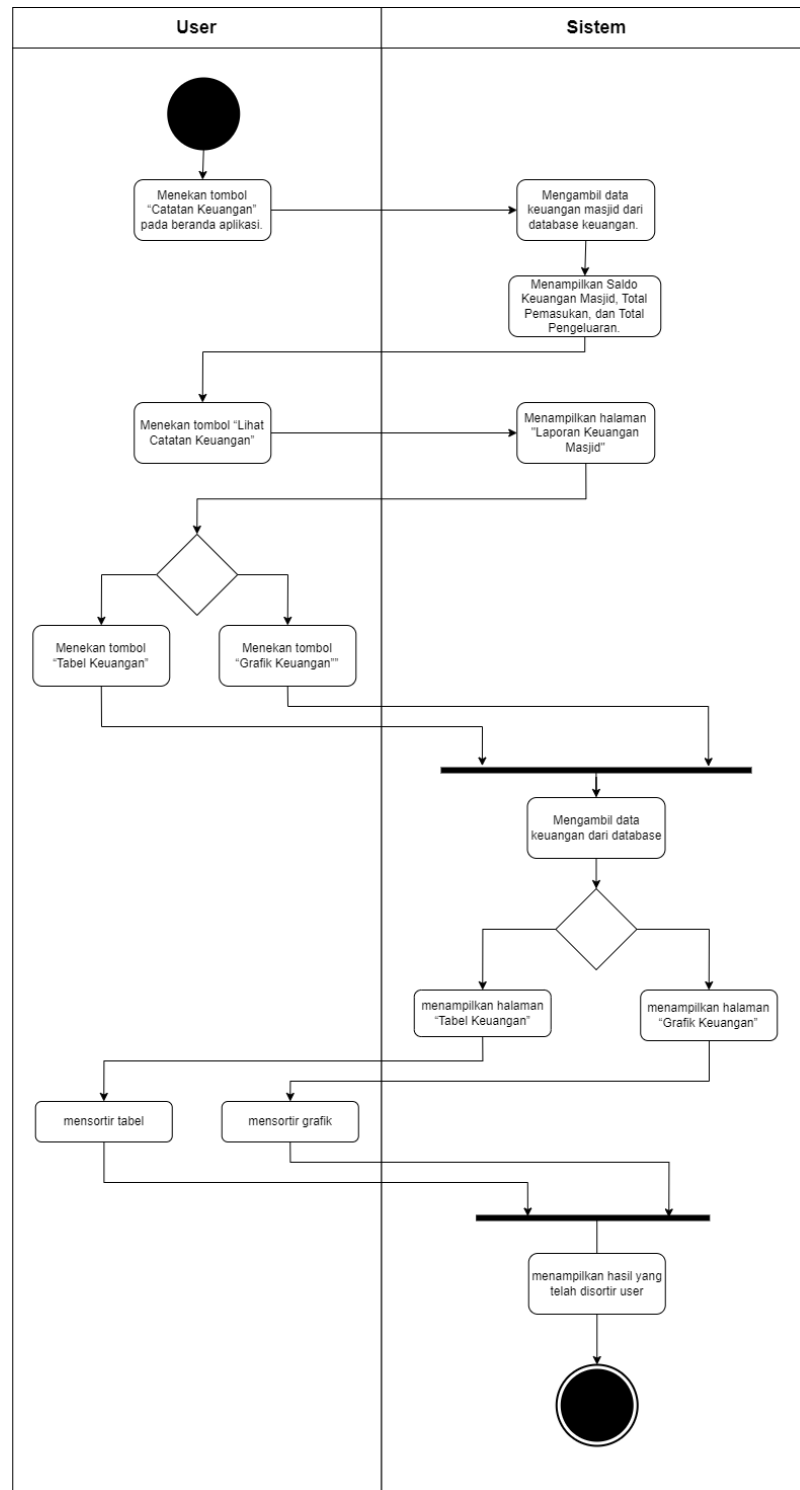
Pada bagian ini pengguna akan menambah catatan keuangan. Dimulai ketika pengguna berada di beranda aplikasi, pengguna menekan tombol “catatan keuangan”. Lalu sistem akan mengambil data keuangan dari *database* dan menampilkannya pada total saldo, saldo pemasukan, dan saldo pengeluaran. Selanjutnya pengguna menekan tombol “Tambah Catatan Keuangan” dan akan di arahkan ke halaman selanjutnya yaitu halaman “Beranda Pilih Catatan”. Pengguna dapat memilih tombol “Tambah Pemasukan” atau “Tambah Pengeluaran”. Lalu sistem akan menampilkan formulir catatan keuangan dan pengguna dapat mengisi formulir tersebut. Setelah selesai, pengguna dapat menekan tombol “masukan” dan sistem akan menyimpan catatan keuangan tersebut ke dalam *database* lalu menampilkan *alert* “Sukses, berhasil membuat catatan”. *Activity diagram* tambah catatan keuangan digambarkan pada gambar 3.6.



Gambar 3. 6 Activity Diagram Tambah Catatan Keuangan

4. *Activity Diagram* lihat catatan keuangan

Pada bagian ini, pengguna akan melihat catatan keuangan baik itu dalam bentuk tabel maupun grafik. Pertama pengguna dapat menekan tombol “Catatan Keuangan” pada beranda aplikasi. Sistem akan mengambil data keuangan dari *database* dan menampilkannya pada total saldo, saldo pemasukan, dan saldo pengeluaran. Setelah itu, pengguna dapat menekan tombol “Lihat Catatan Keuangan” dan sistem akan menampilkan halaman beranda “Laporan Keuangan Masjid”. Selanjutnya, pengguna dapat memilih ingin melihat laporan keuangan dalam bentuk tabel atau grafik dengan menekan tombol “Tabel Keuangan” atau “Grafik Keuangan”. Sistem akan mengambil data keuangan dari *database* dan akan ditampilkan pada halaman tabel atau grafik keuangan. Selanjutnya pengguna dapat mensortir tabel keuangan berdasarkan rentang tanggal, jenis tabel (semua, pemasukan, dan pengeluaran), dan kriteria urutan tabel (nominal terbesar, nominal terkecil, dan kategori) atau mensortir grafik keuangan berdasarkan tahun grafik dan jenis grafik (semua, pemasukan, dan pengeluaran). Sistem akan menampilkan tabel atau grafik keuangan yang telah disortir oleh pengguna. *Acitivity diagram* lihat catatan keuangan digambarkan pada gambar 3.7.

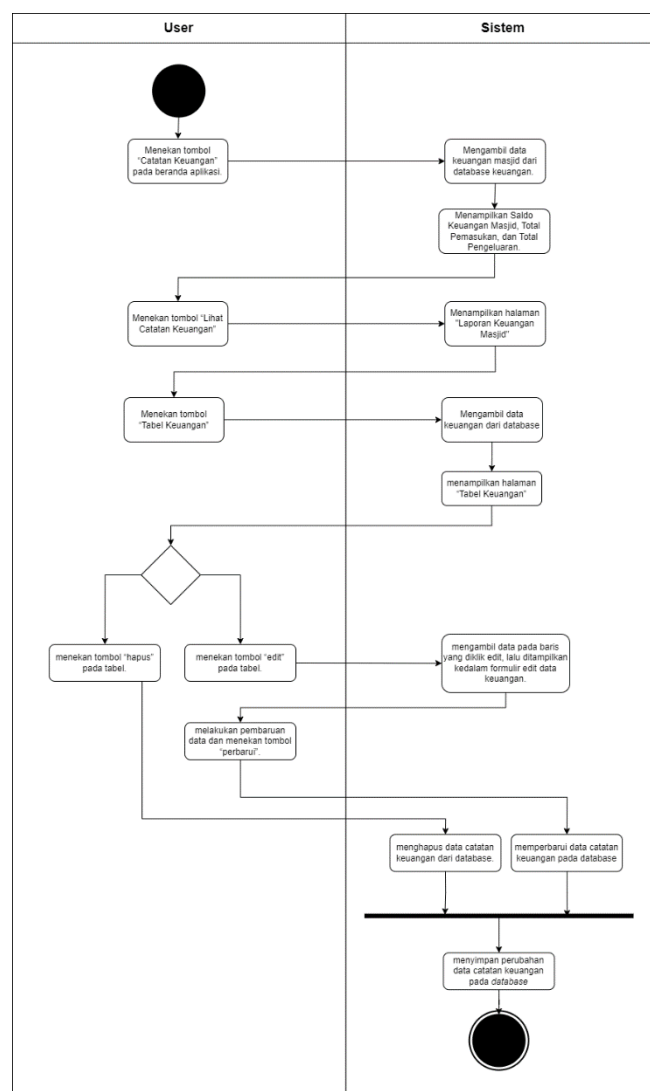


Gambar 3. 7 Activity Diagram Lihat Catatan Keuangan

5. Activity Diagram kelola catatan keuangan

Menu mengelola data catatan keuangan adalah menu bagi takmir untuk melakukan kelola catatan keuangan baik itu perubahan maupun penghapusan. Pertama takmir dapat menekan tombol “Catatan Keuangan” pada beranda aplikasi. Sistem akan mengambil data keuangan dari *database* dan menampilkannya pada total saldo,

saldo pemasukan, dan saldo pengeluaran. Setelah itu, takmir dapat menekan tombol “Lihat Catatan Keuangan” dan sistem akan menampilkan halaman beranda “Laporan Keuangan Masjid”. Selanjutnya takmir dapat menekan tombol “Tabel Keuangan” dan sistem akan mengambil data keuangan dari *database* untuk kemudian ditampilkan dalam tabel. Lalu pada kolom “aksi”, pengguna dapat mengubah data keuangan dan menghapus data keuangan. Semua perubahan yang telah selesai akan disimpan oleh sistem ke dalam *database*. *Acitivity diagram* kelola catatan keuangan digambarkan pada gambar 3.8.

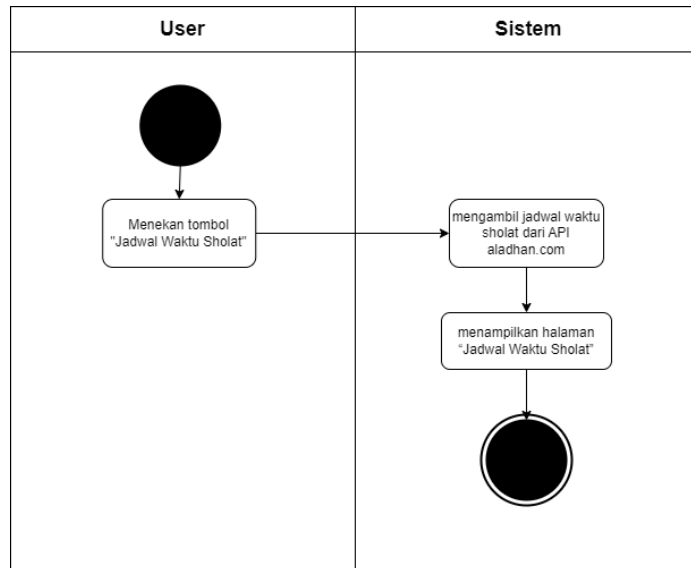


Gambar 3. 8 *Activity Diagram* Kelola Catatan Keuangan

6. *Activity Diagram* lihat jadwal waktu salat

Pada bagian ini pengguna akan melihat jadwal waktu salat. Dimulai pada halaman beranda, pengguna dapat menekan tombol “Jadwal Waktu Salat” lalu sistem akan

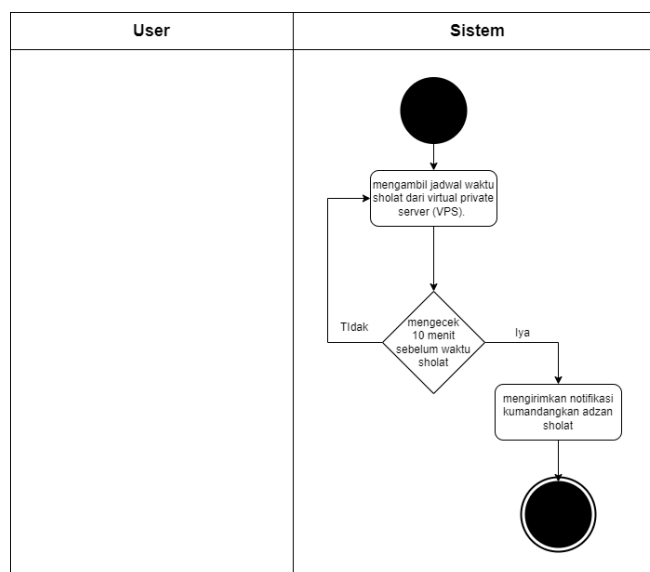
mengambil data waktu salat dengan API dari aladhan.com. Selanjutnya sistem akan menampilkan 5 waktu salat yaitu subuh, zuhur, asar, maghrib, dan isya. *Activity diagram* lihat jadwal waktu salat digambarkan pada gambar 3.9.



Gambar 3. 9 *Activity Diagram* Lihat Jadwal Waktu Salat

7. *Activity Diagram* notifikasi azan salat

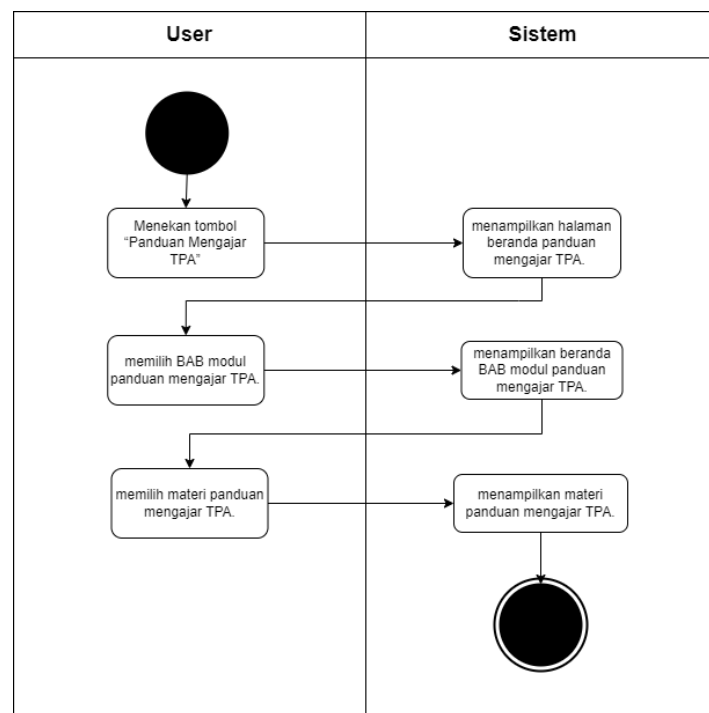
Pada bagian ini sistem akan mengirimkan notifikasi waktu salat. Sistem akan mengambil jadwal waktu salat dari API dan melakukan *looping* untuk mencari 10 menit sebelum waktu salat terdekat. Apabila sudah memasuki 10 menit sebelum waktu salat, maka sistem akan mengirimkan *push notification* ke *handphone* takmir agar takmir segera mengumandangkan azan salat. *Activity diagram* notifikasi azan salat digambarkan pada gambar 3.10.



Gambar 3. 10 *Activity Diagram* Notifikasi Azan Salat

8. *Activity Diagram* modul panduan mengajar TPA

Pada bagian ini pengguna akan melihat modul panduan mengajar TPA. Pada halaman beranda aplikasi pengguna dapat menekan tombol “Panduan Mengajar TPA” lalu sistem akan menampilkan halaman beranda panduan mengajar TPA. Selanjutnya pengguna dapat memilih salah satu bab dari panduan mengajar TPA dimana bab tersebut terdiri dari Tajwid dan Makhraj, Hafalan Surat Pendek, Doa Sehari-hari, Rukun Islam dan Iman, Wudu dan Salat, Kisah Nabi-nabi. Setelah pengguna memilih salah satu bab, maka sistem akan menampilkan materi-materi dari bab tersebut. Setelah pengguna memilih materi dari bab tersebut, maka sistem akan menampilkan materi secara lengkap disertai dengan gambar. *Activity diagram* Modul Panduan Mengajar TPA digambarkan pada gambar 3.11.

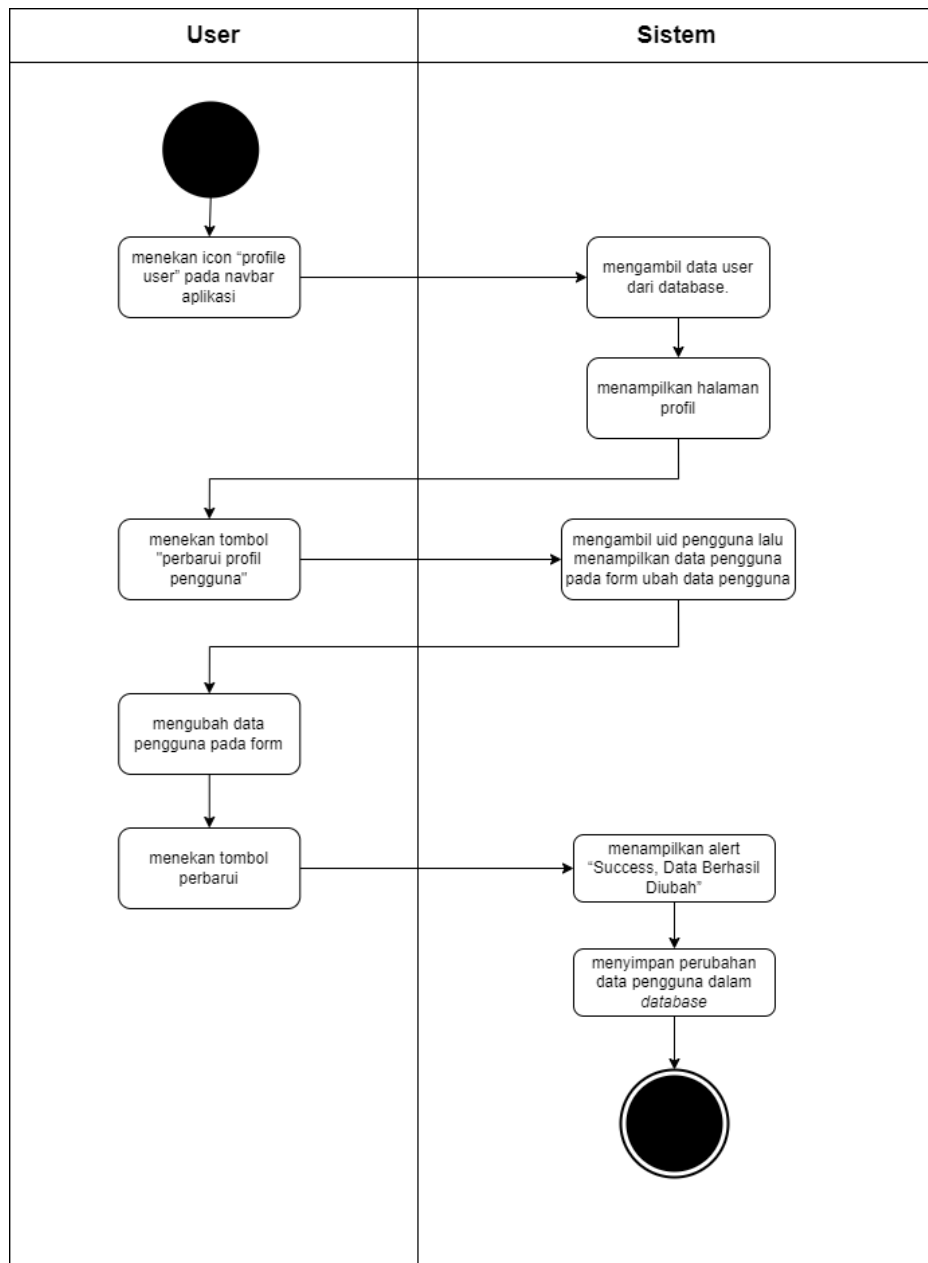


Gambar 3. 11 *Activity Diagram* Modul Panduan Mengajar TPA

9. *Activity Diagram* perbarui profil pengguna

Pada bagian ini, pengguna akan memperbaiki profil. Pengguna dapat menekan icon “*profile pengguna*” pada *navbar* aplikasi lalu sistem akan mengambil data *pengguna* dari *database* dan menampilkannya pada halaman *profile*. Selanjutnya pengguna dapat menekan tombol “perbarui *profile* pengguna” dan sistem akan mengambil uid pengguna untuk menampilkan data pengguna pada halaman form ubah data pengguna. Selanjutnya pengguna dapat mengubah data dengan memilih bagian yang ingin diubah pada salah satu form ubah data pengguna sesuai dengan

kebutuhan. Lalu pengguna dapat menekan tombol perbarui dan sistem akan menampilkan *alert* “Success, Data berhasil diubah” kemudian perubahan data pengguna disimpan ke dalam *database* firebase. *Acitivity diagram* perbarui profil pengguna digambarkan pada gambar 3.12.

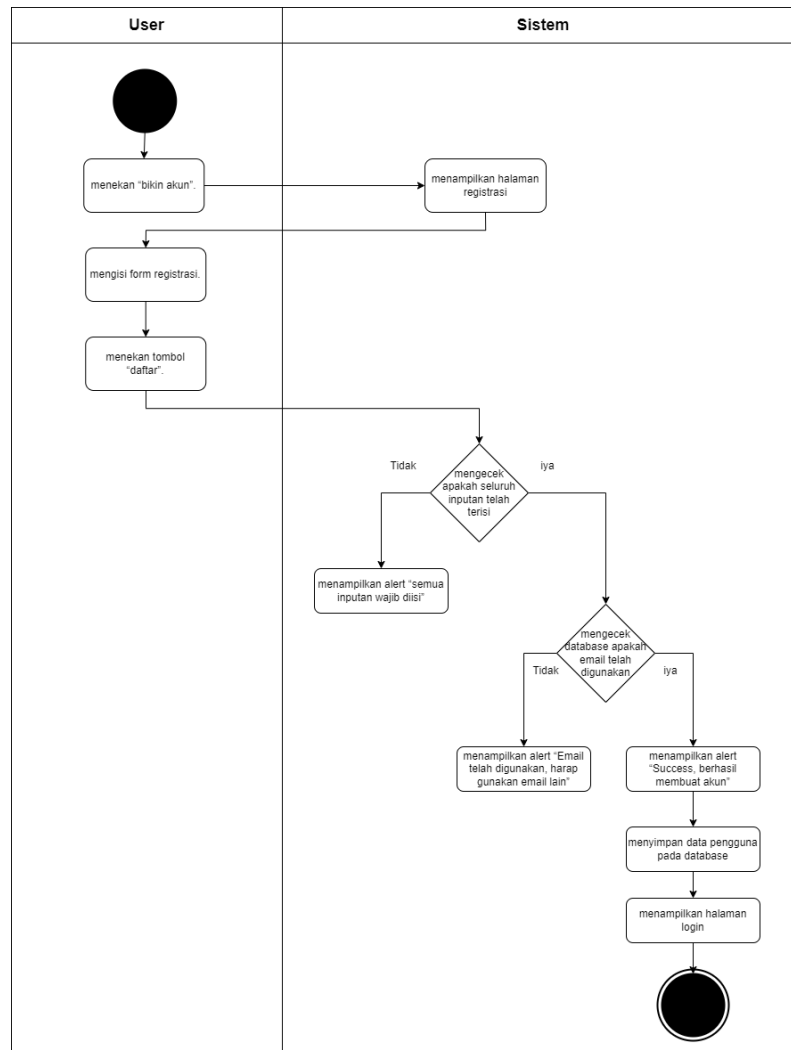


Gambar 3. 12 *Activity Diagram* Perbarui Profil Pengguna

10. *Activity Diagram* registrasi takmir

Pada bagian ini pengguna akan melakukan registrasi. Pada halaman *login*, pengguna dapat menekan “bikin akun” lalu sistem akan menampilkan halaman registrasi. Selanjutnya pengguna dapat mengisi form registrasi yang meliputi nama, nomor telepon, *email*, sandi, dan alamat. Setelah semua terisi pengguna

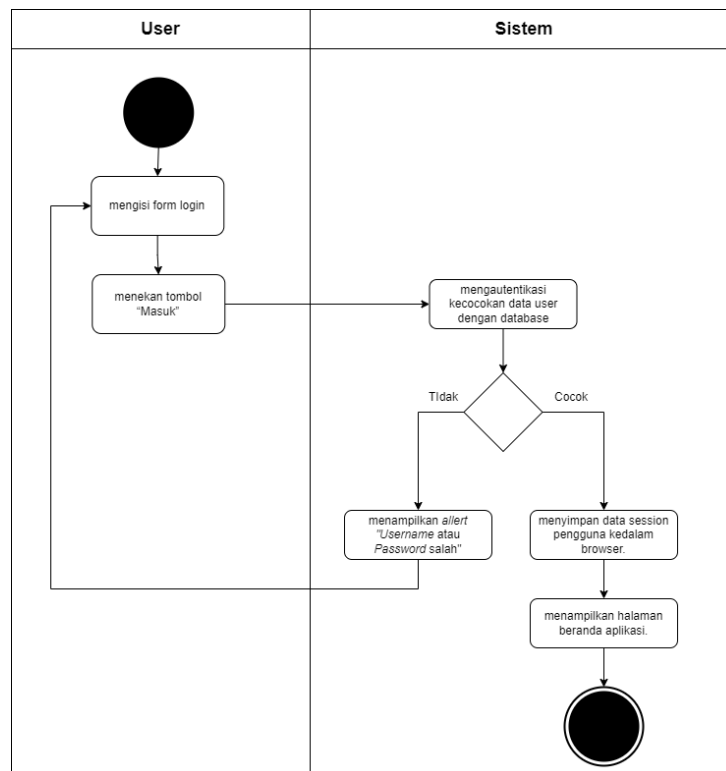
dapat menekan tombol daftar. Sistem akan mengecek apakah seluruh form telah terisi, apabila ada salah satu form belum terisi maka sistem akan menampilkan *alert* “semua form wajib diisi” dan pengguna diminta melengkapi form yang belum terisi. Selanjutnya apabila semua form telah terisi maka sistem akan mengecek dalam *database* apakah *email* tersebut sudah digunakan atau masih tersedia. Apabila *email* tersebut telah digunakan maka sistem akan menampilkan *alert* “*email* telah digunakan, harap gunakan *email* lain” dan pengguna diminta untuk menggunakan *email* yang belum digunakan. Selanjutnya apabila *email* belum digunakan maka proses registrasi berhasil dan sistem akan memunculkan *alert* “berhasil membuat akun” lalu data pengguna akan disimpan dalam *database* dan pengguna akan diarahkan ke halaman *login*. *Acitivity diagram* registrasi takmir digambarkan pada gambar 3.13.



Gambar 3. 13 *Activity Diagram* Registrasi Takmir

11. Activity Diagram login takmir

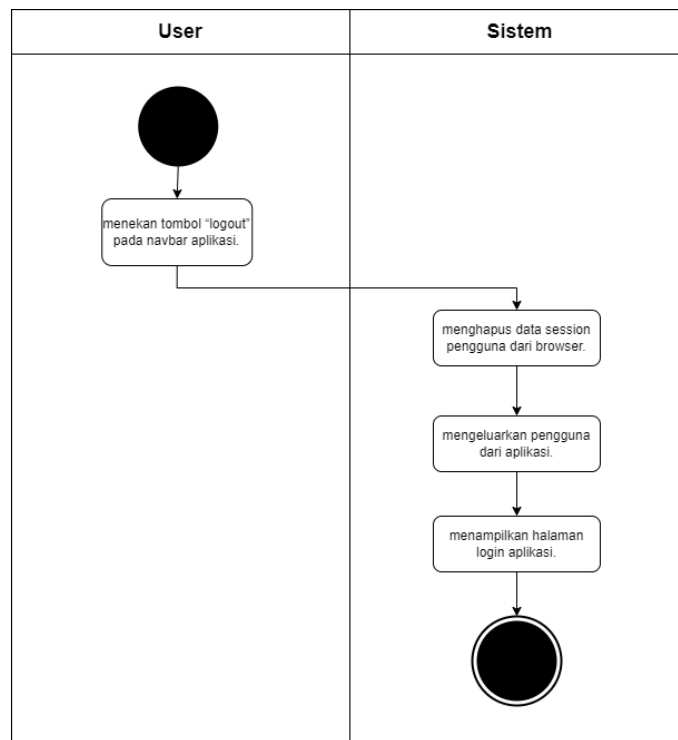
Pada bagian ini pengguna akan melakukan *login*. Pada halaman *login* pengguna dapat mengisi *email* dan *password* lalu menekan tombol masuk. Sistem akan melakukan proses autentikasi kecocokan antara *input* pengguna dengan *database*. Apabila data tersebut tidak cocok maka sistem akan menampilkan *alert* “*username* atau *password* salah” dan pengguna diminta untuk mengisi ulang form *login*. Apabila data *input* pengguna cocok dengan *database* maka sistem akan menyimpan *data session* pengguna ke dalam peramban dan pengguna akan diarahkan ke halaman beranda aplikasi. *Acitivity diagram login* takmir digambarkan pada gambar 3.14.



Gambar 3. 14 Activity Diagram Login Takmir

12. Activity Diagram logout takmir

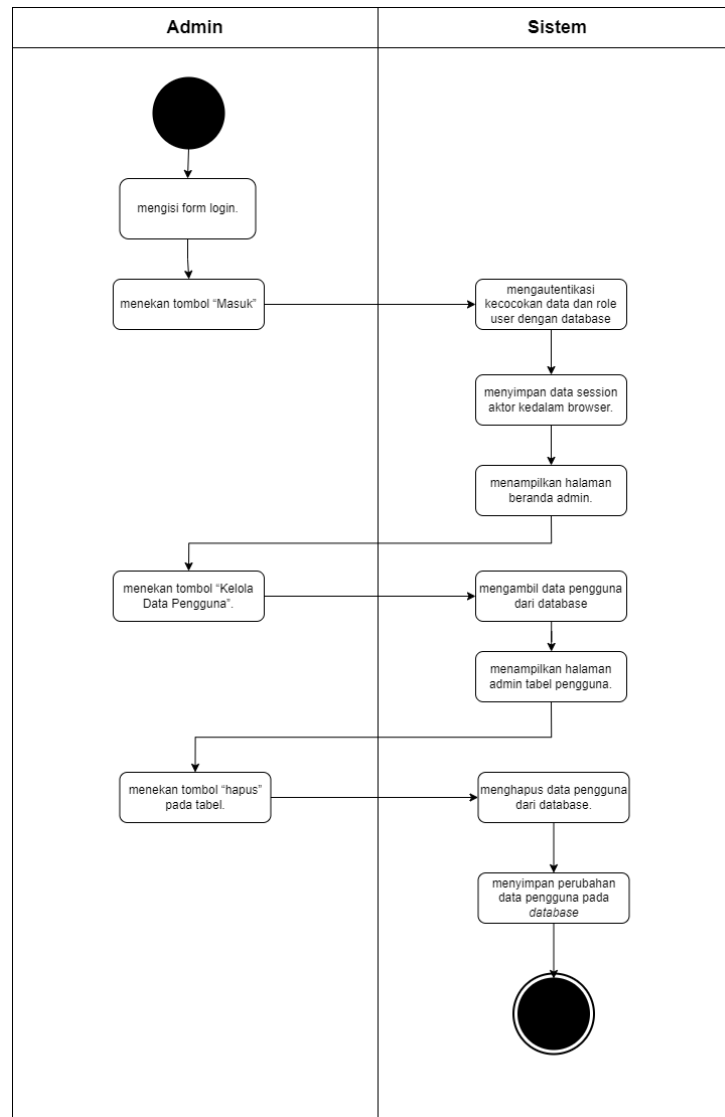
Pada bagian ini pengguna akan melakukan *logout*. Pengguna dapat menekan *icon* “*logout*” pada *navbar* aplikasi. Selanjutnya sistem akan menghapus data *session* pengguna dari peramban dan mengeluarkan pengguna dari aplikasi. Lalu pengguna akan diarahkan ke halaman *login* aplikasi. *Acitivity diagram logout* takmir digambarkan pada gambar 3.15.



Gambar 3. 15 Activity Diagram Logout Takmir

13. Activity Diagram mengelola data pengguna

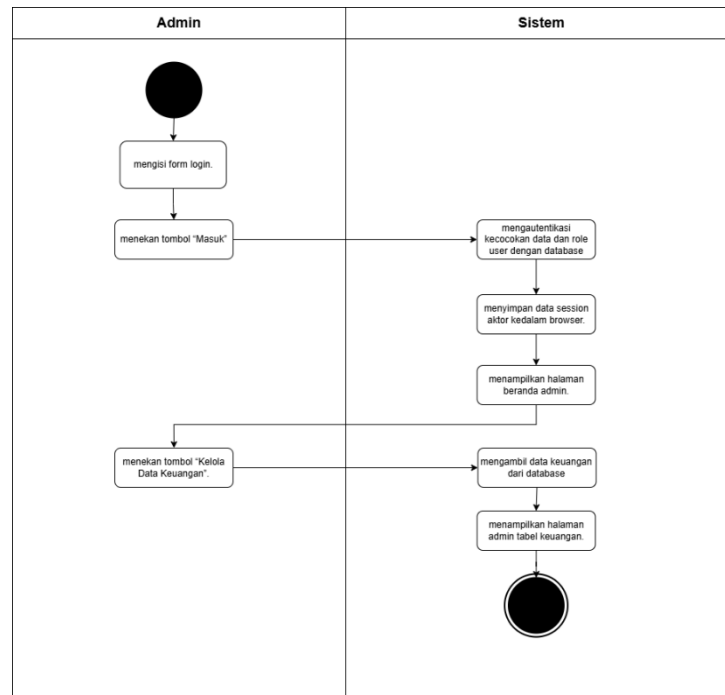
Menu mengelola data pengguna adalah menu yang dikhususkan untuk admin untuk melakukan hapus data pengguna. Pada halaman *login*, admin mengisi form *login* lalu sistem akan mengecek *role* pengguna. Apabila pengguna tersebut memiliki *role* admin maka sistem akan menyimpan data *session* dan mengarahkan pengguna ke halaman beranda admin. Selanjutnya admin dapat menekan tombol kelola data pengguna dan sistem akan mengambil data pengguna dari *database* dan menampilkan tabel pengguna. Lalu admin dapat melakukan perubahan dengan menekan tombol hapus pada kolom aksi di tabel pengguna. Semua perubahan yang telah selesai akan disimpan oleh sistem ke dalam *database*. *Acitivity diagram* mengelola data pengguna digambarkan pada gambar 3.16.



Gambar 3. 16 Activity Diagram Mengelola Data Pengguna

14. Activity Diagram mengelola data keuangan

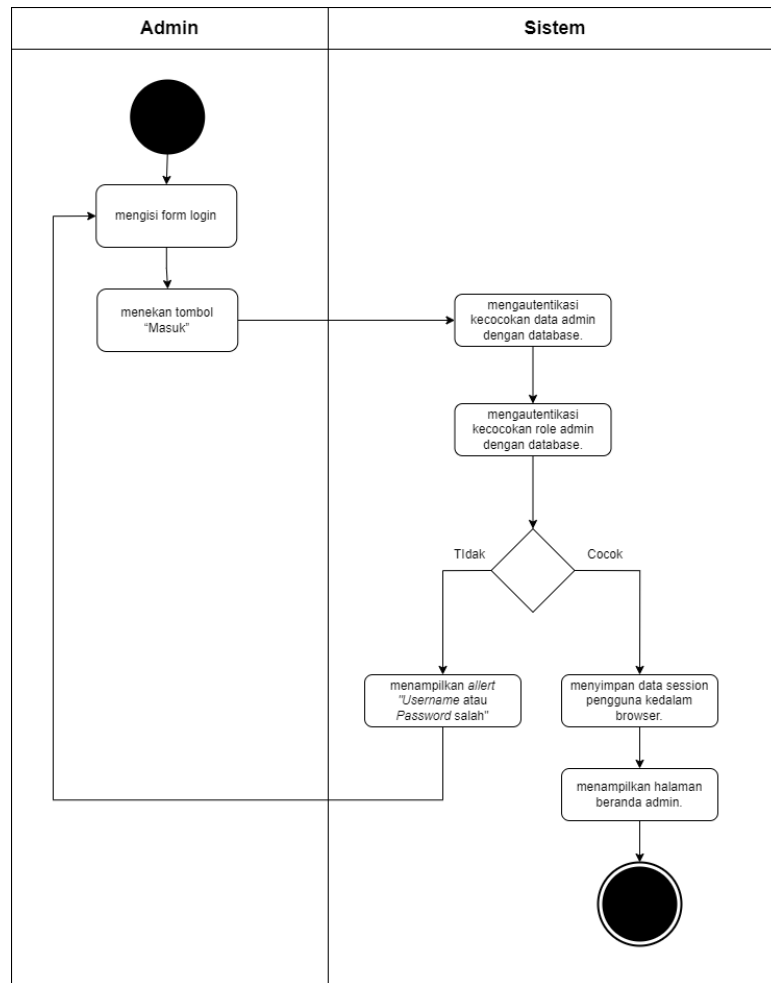
Menu mengelola data keuangan adalah menu yang dikhususkan untuk admin untuk melihat laporan keuangan dalam bentuk tabel termasuk siapa yang membuat catatan keuangan tersebut. Pada halaman *login*, admin mengisi form *login* lalu sistem akan mengecek *role* pengguna. Apabila pengguna tersebut memiliki *role* admin maka sistem akan menyimpan data *session* dan mengarahkan pengguna ke halaman beranda admin. Selanjutnya admin dapat menekan tombol kelola data keuangan dan sistem akan mengambil data keuangan secara lengkap dari *database* dan menampilkan tabel keuangan. *Acitivity diagram* mengelola data keuangan digambarkan pada gambar 3.17.



Gambar 3. 17 Activity Diagram Kelola Data Keuangan

15. Activity Diagram login admin

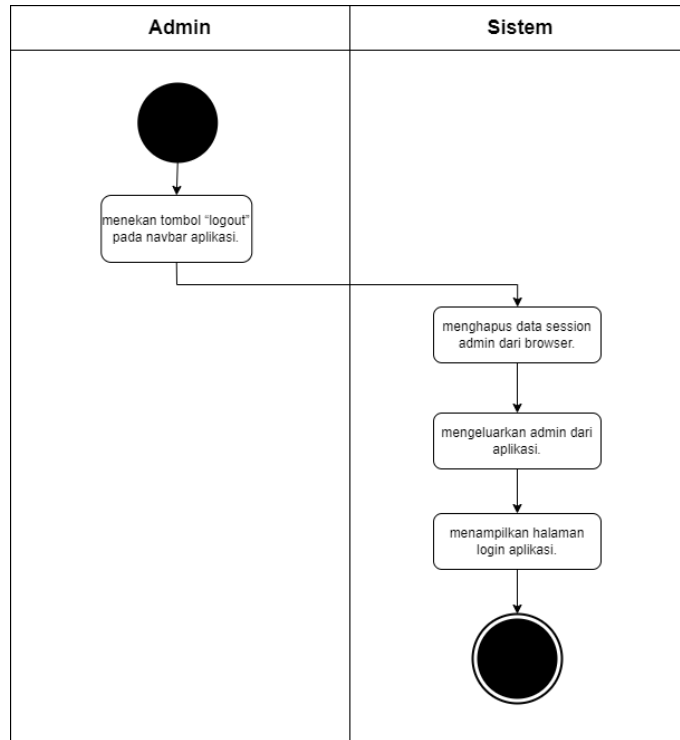
Pada bagian ini, admin akan melakukan proses *login*. Pada halaman *login* admin dapat mengisi form *login* lalu menekan tombol masuk. Sistem akan mengautentikasi kecocokan data dan *role* admin dengan *database*. Apabila data tersebut tidak cocok maka akan terdapat *alert* “*username* atau *password* salah” dan apabila cocok maka sistem akan menyimpan data *session* admin ke dalam peramban. Selanjutnya pengguna akan diarahkan ke halaman admin. *Acitivity diagram login* admin digambarkan pada gambar 3.18.



Gambar 3. 18 Activity Diagram Login Admin

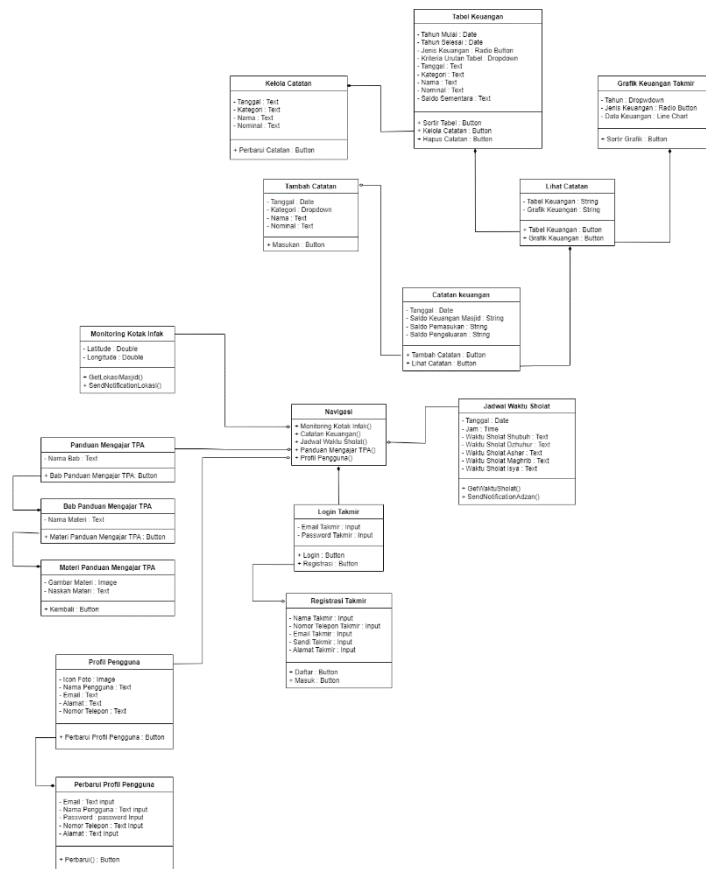
16. Activity Diagram logout admin

Pada bagian ini admin akan melakukan *logout*. Admin dapat menekan *icon "logout"* pada *navbar* aplikasi. Selanjutnya sistem akan menghapus data *session* admin dari peramban dan mengeluarkan admin dari aplikasi. Lalu admin akan diarahkan ke halaman *login* aplikasi. *Activity diagram logout* admin digambarkan pada gambar 3.19.



Gambar 3. 19 Activity Diagram Logout Admin

3.3.6 Class diagram



Gambar 3. 20 Class diagram Aplikasi Masjid Al-Ikhlas Sorosutan

Class diagram ini menggambarkan struktur aplikasi yang berfokus pada pengelolaan keuangan masjid, *monitoring* kotak infak, jadwal waktu salat, dan panduan mengajar TPA, yang digunakan oleh dua aktor utama, yaitu takmir dan admin. Diagram ini terdiri dari beberapa kelas utama seperti Kelola Catatan, Tambah Catatan, Lihat Catatan, *Monitoring* Kotak Infak, Jadwal Waktu Salat, dan Panduan Mengajar TPA. Setiap kelas memiliki atribut dan operasi yang spesifik untuk mendukung fungsi yang sesuai. Contohnya, kelas Kelola Catatan memiliki atribut seperti tanggal, nominal, dan keterangan, serta operasi seperti "Perbarui Catatan" untuk memodifikasi data keuangan.

Diagram ini juga menunjukkan hubungan antar kelas, seperti dependensi antara kelas Kelola Catatan, Lihat Catatan, dan Tambah Catatan. Selain itu, terdapat kelas-kelas seperti *Login* Takmir, Profil Pengguna, dan Perbarui Profil Pengguna, yang mendukung fitur autentikasi dan manajemen profil pengguna. Atribut dan operasi pada setiap kelas dirancang untuk mendukung interaksi pengguna dengan aplikasi, misalnya tombol "Masuk" pada kelas *Login* Takmir dan tombol "Sortir" pada kelas Grafik Keuangan Takmir untuk mempermudah penyortiran data visual. *Class diagram* ini memberikan gambaran lengkap tentang arsitektur sistem yang modular dan fokus pada kemudahan penggunaan bagi takmir dan admin.

3.4 Prototype

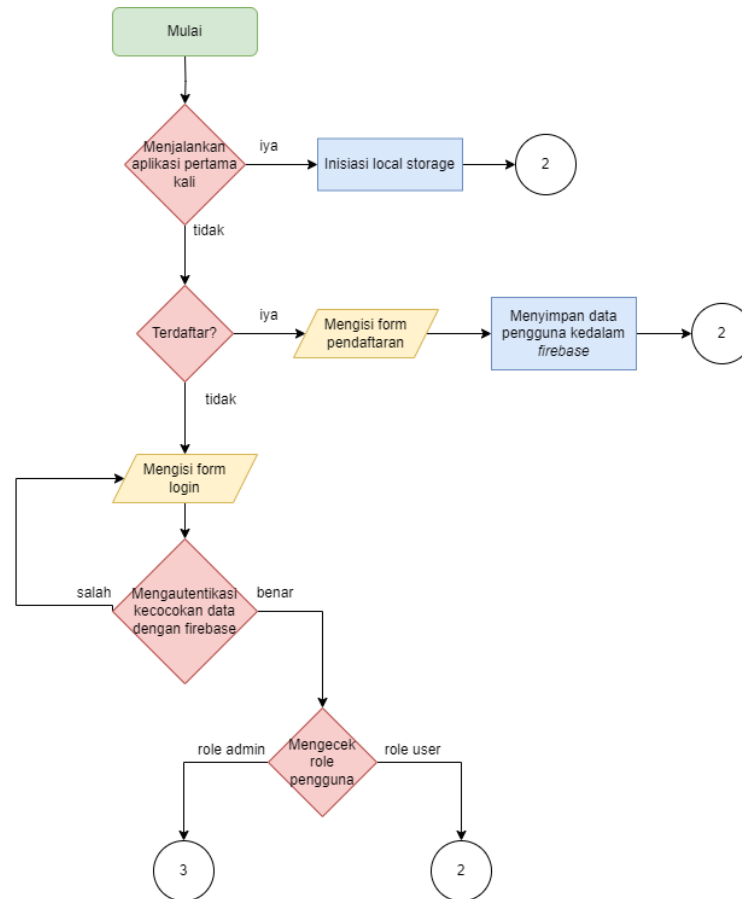
3.4.1 Alur Proses Bisnis Aplikasi

Sesuai hasil identifikasi kebutuhan sistem, aplikasi ini diharapkan dapat menyediakan fitur *monitoring* kotak infak, notifikasi kotak infak, pencatatan dan tampilan data keuangan masjid, jadwal waktu salat, notifikasi azan salat, dan modul panduan mengajar TPA. Untuk membuat fitur-fitur tersebut, aplikasi menerapkan fungsi dasar *create, read, update, delete* (CRUD) dengan alur sistem sebagai berikut:

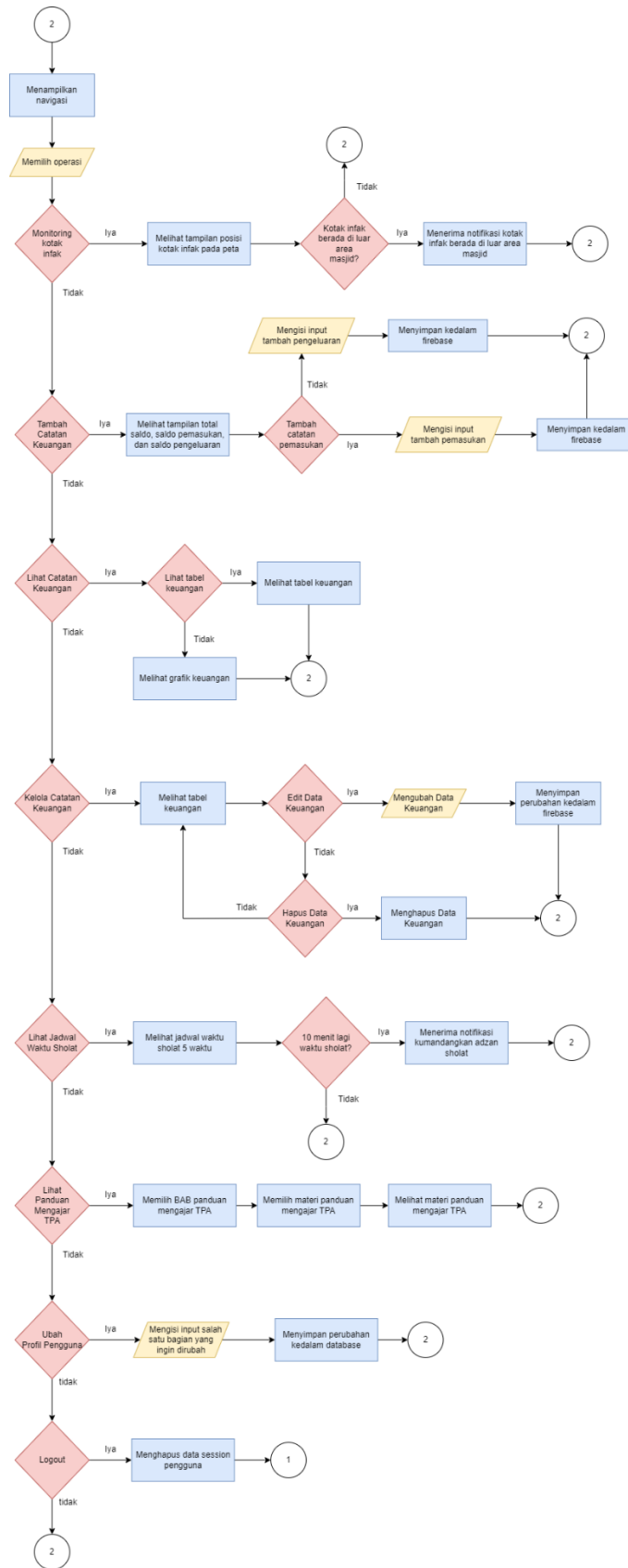
1. Pengguna (takmir) dan admin sebagai aktor aplikasi.
2. Sistem melakukan inisiasi basis data *Local Storage* dalam peramban ketika aplikasi dijalankan pertama kali untuk menyimpan UID (*Unique Identifier*) pengguna.
3. Pengguna harus *login* ke dalam aplikasi terlebih dahulu untuk mengakses semua fitur yang dibutuhkan.
4. Sistem menampilkan halaman beranda yang terdiri dari 4 fitur utama (*monitoring* kotak infak, pencatatan keuangan, jadwal waktu salat, dan panduan mengajar TPA) sebagai halaman pertama aplikasi.
5. Sistem menampilkan titik koordinat lokasi kotak infak pada peta dan mengirimkan notifikasi kepada pengguna apabila kotak infak berada di luar area masjid.

6. Sistem menampilkan halaman yang menunjukkan total saldo keuangan masjid, total pemasukan, dan total pengeluaran. Pengguna dapat melakukan pencatatan keuangan dengan memasukkan *input* berupa tanggal, kategori, nama, dan nominal pemasukan atau pengeluaran.
7. Sistem menerima *input* pengguna dan menyimpan informasi tersebut ke penyimpanan *database* Firebase.
8. Sistem menyediakan halaman tabel catatan keuangan yang menampilkan daftar seluruh data pencatatan keuangan pengguna diurutkan berdasarkan tanggal pencatatan terbaru dan dapat disortir sesuai dengan kebutuhan pengguna.
9. Pengguna dapat mengubah dan menghapus data pencatatan keuangan. Setiap perubahan yang dilakukan oleh admin akan disimpan ke dalam Firebase.
10. Sistem menyediakan halaman grafik catatan keuangan yang menampilkan grafik keuangan masjid setiap bulan dalam rentang satu tahun.
11. Sistem menyediakan halaman untuk menampilkan jadwal salat 5 waktu dan mengirimkan notifikasi kepada pengguna apabila 10 menit lagi akan memasuki waktu azan salat.
12. Sistem menyediakan halaman untuk menampilkan modul panduan mengajar TPA. Pengguna dapat memilih bab dan materi panduan yang sesuai dengan kebutuhan mereka.
13. Pengguna dapat melakukan perubahan data pada halaman profil pengguna. Sistem akan menyimpan perubahan data pengguna pada *database* firebase.
14. Admin dapat melihat dan menghapus data pengguna. Setiap perubahan yang dilakukan oleh admin akan disimpan ke dalam Firebase.
15. Takmir dapat mendaftarkan diri untuk memiliki akun dan *login* menggunakan akun yang telah didaftarkan.
16. Pengguna dapat *login* menggunakan akun yang telah didaftarkan.
17. Sistem menampilkan halaman *login* yang terintegrasi dengan Firebase.
18. Pengguna dapat *logout* dari aplikasi, dan sistem akan menghapus data *session* dari *Local Storage*.

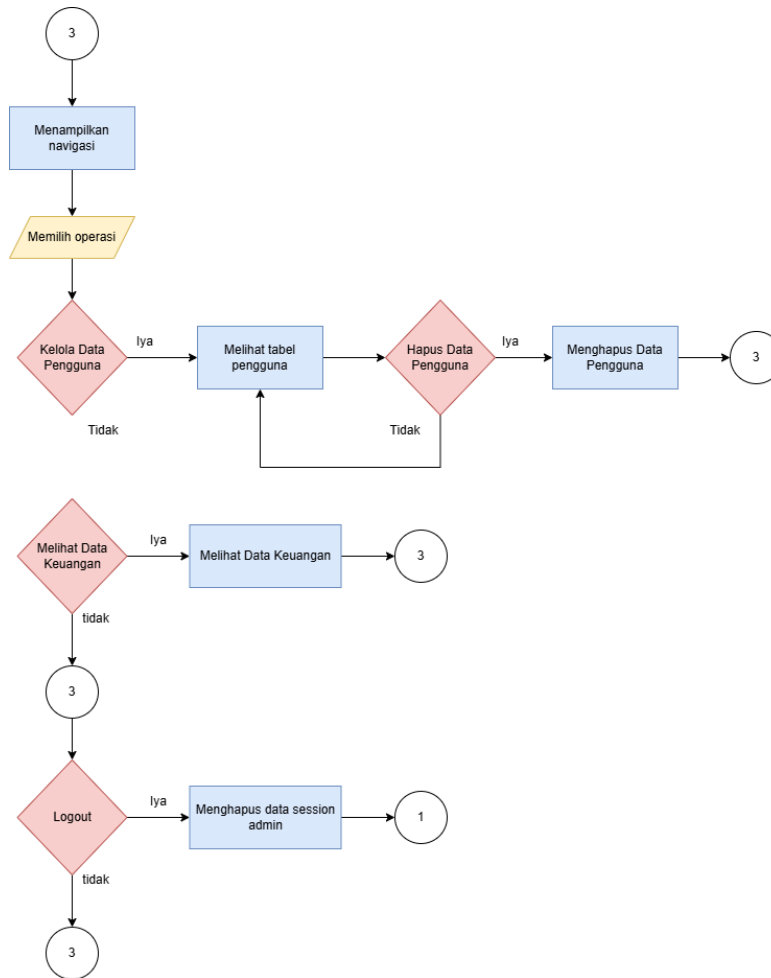
Tahapan alur sistem divisualisasikan pada gambar 3.21 hingga gambar 3.23.



Gambar 3. 21 Alur aplikasi Masjid Al-Ikhlas Sorosutan



Gambar 3. 22 Sub alur fitur aplikasi pengguna (takmir)



Gambar 3. 23 Sub alur fitur aplikasi admin

Pada Gambar 3.21, dijelaskan alur proses bisnis sistem ketika aplikasi pertama kali dijalankan. Dalam skenario ini, sistem akan melakukan inisiasi basis data *pada Local Storage* peramban. Proses ini penting untuk menyimpan UID (*Unique Identifier*) pengguna secara lokal agar dapat digunakan pada interaksi selanjutnya. Pengguna yang baru pertama kali menggunakan aplikasi harus mendaftarkan diri untuk membuat akun pengguna. Setelah pendaftaran berhasil, pengguna dapat *login* ke dalam aplikasi untuk mengakses fitur-fitur yang tersedia.

Pada Gambar 3.23, menjelaskan alur proses bisnis fitur-fitur utama aplikasi. Diagram ini menggambarkan bagaimana pengguna dapat melakukan *monitoring* kotak infak, menambahkan dan melihat catatan keuangan, mengelola catatan keuangan, melihat jadwal waktu salat, dan mengakses panduan mengajar TPA. Alur diagram ini juga menjelaskan bagaimana pengguna menerima notifikasi apabila kotak infak berada di luar lokasi masjid atau apabila waktu salat akan segera tiba.

Pada Gambar 3.23, dijelaskan alur proses bisnis yang terkait dengan pengelolaan data pengguna oleh admin. Admin memiliki kemampuan untuk menghapus data pengguna. Alur ini juga mencakup proses *logout* admin, di mana data *session* akan dihapus dari *Local Storage* untuk menjaga keamanan.

3.4.2 Perancangan Antarmuka Pengguna

1. Rancangan halaman Beranda

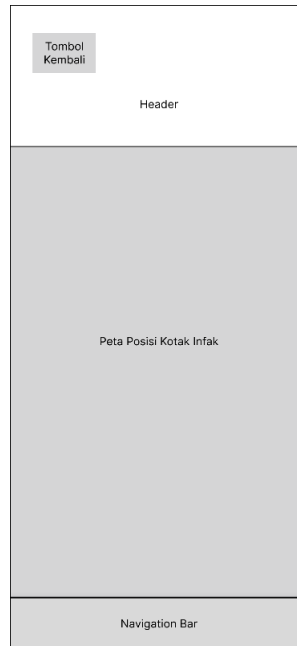
Halaman beranda menampilkan empat tombol untuk mengakses fitur-fitur yang tersedia dalam aplikasi ini, yaitu *monitoring* kotak infak, penambahan catatan keuangan, jadwal waktu salat, dan panduan mengajar TPA. Selain itu, halaman ini juga menampilkan informasi waktu salat terdekat dan saldo keuangan masjid yang tersisa. Pada bagian bawah halaman ini terdapat menu *navigation bar*. Rancang halaman beranda pengguna dapat dilihat pada gambar 3.24.



Gambar 3. 24 Rancang Halaman Beranda Pengguna

2. Rancangan halaman fitur *monitoring* kotak infak

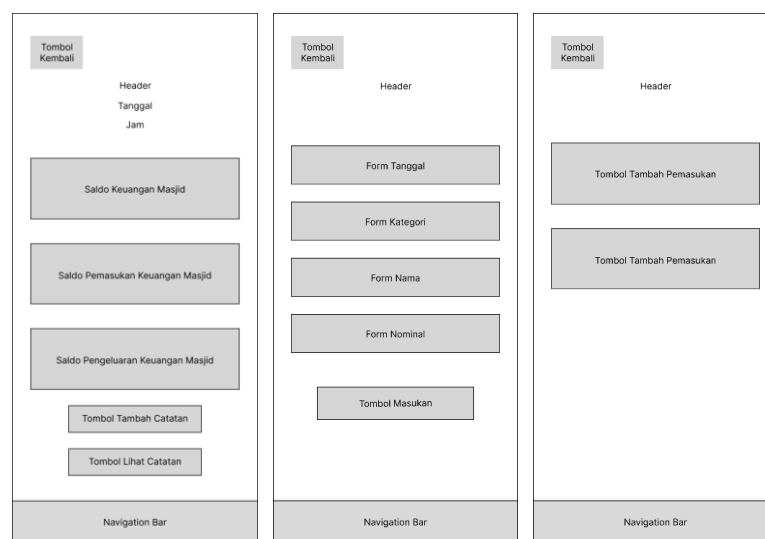
Halaman *monitoring* kotak infak adalah halaman yang digunakan oleh pengguna untuk memantau posisi kotak infak berada. Halaman ini menampilkan peta / maps beserta titik koordinat dari lokasi kotak infak itu sendiri. Rancangan halaman *monitoring* kotak infak dapat dilihat pada gambar 3.25.



Gambar 3. 25 Rancangan halaman fitur *monitoring* kotak infak

3. Rancangan halaman fitur tambah catatan keuangan

Halaman fitur tambah catatan keuangan adalah halaman yang digunakan oleh pengguna untuk menambahkan catatan keuangan. Pada halaman awal catatan keuangan akan menampilkan total saldo keuangan masjid, total pemasukan keuangan masjid, total pengeluaran keuangan masjid, serta tombol untuk menambah dan melihat catatan keuangan. Selanjutnya pengguna akan masuk ke halaman pemasukan dan pengeluaran lalu akan mengisi formulir catatan keuangan. Rancangan halaman fitur tambah catatan keuangan dapat dilihat pada gambar 3.26.



Gambar 3. 26 Rancangan halaman fitur tambah catatan keuangan

4. Rancangan halaman fitur jadwal waktu salat

Halaman fitur jadwal waktu salat adalah halaman yang digunakan pengguna untuk melihat jadwal waktu salat. Halaman ini menampilkan waktu salat wajib 5 waktu. Selain itu, di bagian *header* halaman terdapat hari dan tanggal beserta jam. Rancangan halaman fitur jadwal waktu salat dapat dilihat pada gambar 3.27.



Gambar 3. 27 Rancangan halaman fitur jadwal waktu salat

5. Rancangan halaman fitur panduan mengajar TPA

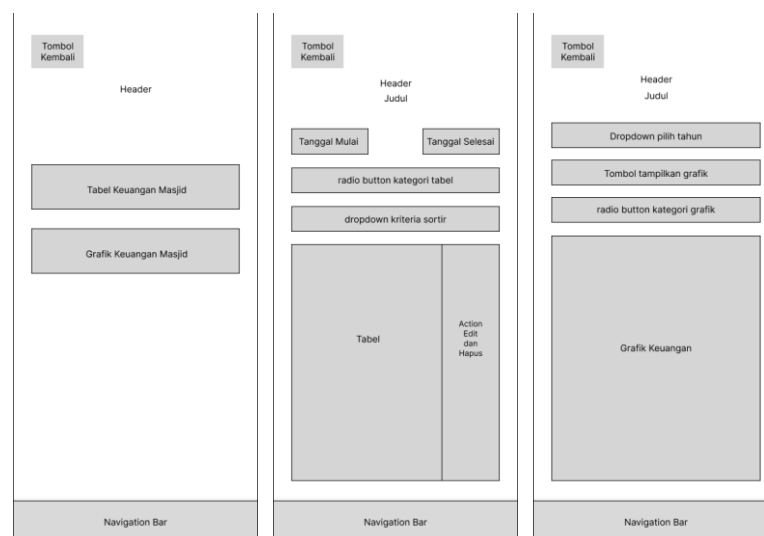
Halaman fitur panduan mengajar TPA adalah halaman yang digunakan oleh pengguna untuk melihat modul panduan mengajar TPA. Halaman ini menampilkan materi-materi mengenai modul pembelajaran TPA yang diambil dari buku Kementerian Agama (Kemenag) RI. Fitur panduan mengajar TPA berisi bab, materi, lalu detail pembelajaran yang sudah dilengkapi gambar. Rancangan halaman fitur panduan mengajar TPA dapat dilihat pada gambar 3.28.



Gambar 3. 28 Rancangan halaman fitur panduan mengajar TPA

6. Rancangan halaman fitur lihat laporan keuangan

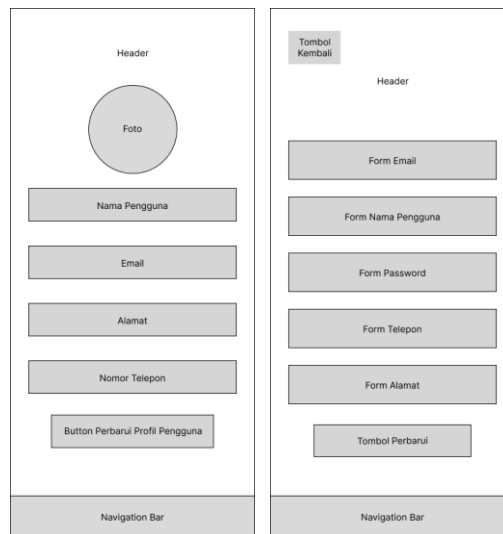
Halaman fitur lihat laporan keuangan adalah halaman yang digunakan oleh pengguna untuk melihat data dari catatan keuangan yang telah masuk ke dalam *database*. Data dari catatan keuangan tersebut dapat dilihat dalam bentuk tabel atau dapat divisualisasikan dalam bentuk grafik. Dalam halaman tabel terdapat sortir tanggal, jenis tabel yang akan ditampilkan, dan kriteria dari catatan keuangan. Selain itu juga terdapat menu *action* bagi pengguna untuk melakukan olah data baik itu edit maupun hapus data catatan keuangan. Pada halaman grafik keuangan terdapat dropdown pilihan tahun dan tampilan grafik pemasukan dan pengeluaran. Rancangan halaman fitur lihat laporan keuangan dapat dilihat pada gambar 3.29.



Gambar 3. 29 Rancangan halaman fitur lihat laporan keuangan

7. Rancangan halaman profil pengguna

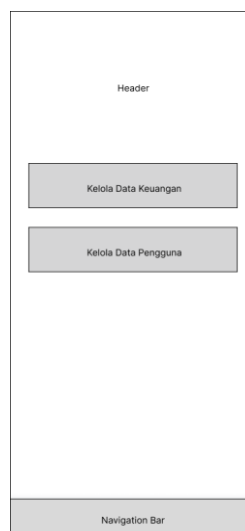
Halaman profil pengguna adalah halaman digunakan oleh pengguna untuk melihat data pengguna serta memperbarui profil mereka. Halaman menampilkan nama pengguna, *email* alamat, nomor telepon, dan tombol perbarui *profile* pengguna untuk memperbarui profil mereka. Lalu setelah itu terdapat halaman form untuk memperbarui profil pengguna. Rancangan halaman fitur profil pengguna dapat dilihat pada gambar 3.30.



Gambar 3. 30 Rancangan halaman profil pengguna

8. Rancang Halaman Beranda Admin

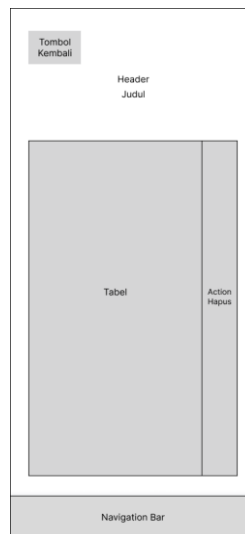
Halaman beranda admin adalah halaman awal yang digunakan oleh admin untuk melakukan olah data. Pada halaman beranda menampilkan *button* kelola data keuangan dan kelola data pengguna. Rancangan halaman beranda admin dapat dilihat pada gambar 3.31.



Gambar 3. 31 Rancangan halaman beranda admin

9. Rancang Halaman Edit Data Pengguna

Halaman edit data pengguna adalah halaman yang digunakan untuk melakukan olah data pengguna dalam bentuk penghapusan. Pada tabel halaman edit data pengguna akan ditampilkan tabel dan terdapat satu tabel tambahan yaitu tabel *action* untuk melakukan hapus data. Rancangan halaman edit data pengguna dapat dilihat pada gambar 3.32.



Gambar 3. 32 Rancangan halaman edit data pengguna

10. Rancang halaman registrasi

Halaman registrasi adalah halaman formulir yang digunakan oleh pengguna untuk melakukan proses pendaftaran atau membuat akun. Formulir pada halaman ini terdiri dari form *email*, form nama pengguna, form *password*, form telepon, form alamat, dan tombol perbarui. Lalu di bawah juga terdapat tombol "masuk" untuk mengarah ke halaman *login* apabila pengguna telah mempunyai akun. Rancangan halaman fitur registrasi dapat dilihat pada gambar 3.33.



Header

Nama Aplikasi

Form Email

Form Nama Pengguna

Form Password

Form Telepon

Form Alamat

Tombol Perbarui

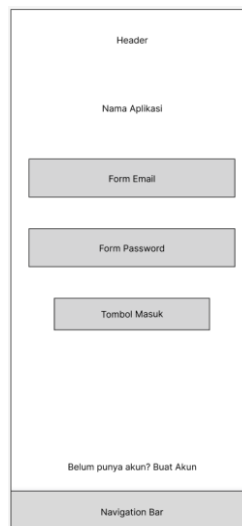
Sudah punya akun? Masuk

Navigation Bar

Gambar 3. 33 Rancangan halaman registrasi pengguna

11. Rancang halaman *login*

Halaman *login* adalah halaman formulir yang digunakan oleh pengguna untuk melakukan proses masuk ke dalam beranda aplikasi. Formulir halaman ini terdiri dari form *email*, form *password*, dan button masuk. Rancangan halaman *login* pengguna dapat dilihat pada gambar 3.34.



Header

Nama Aplikasi

Form Email

Form Password

Tombol Masuk

Belum punya akun? Buat Akun

Navigation Bar

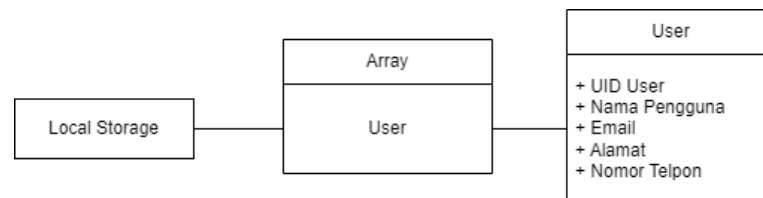
Gambar 3. 34 Rancangan halaman *login*

3.4.3 Perancangan *Database*

Pada pengembangan aplikasi Masjid Al-Ikhlas, digunakan dua jenis penyimpanan data, yaitu *Local Storage* dan *Firestore*. *Local Storage* berfungsi untuk menyimpan sementara data pengguna (uid) setelah proses *login*, sehingga pengguna dapat tetap masuk tanpa perlu autentikasi ulang. Sementara itu, *Firestore* dari *Firestore* digunakan sebagai basis data utama untuk menyimpan data sistem secara menyeluruh ke dalam cloud.

1. Struktur Basis Data *Local Storage*

Pada basis data *Local Storage*, data disimpan dalam bentuk array yang berisi objek pengguna. Setiap objek pengguna memiliki UID (Pengguna ID), Nama Pengguna, Email, Alamat, dan Nomor Telepon. Data ini disimpan dalam array "Pengguna" dan digunakan untuk menyimpan informasi *login* sementara. Struktur basis data *local storage* dapat dilihat pada gambar 3.35.

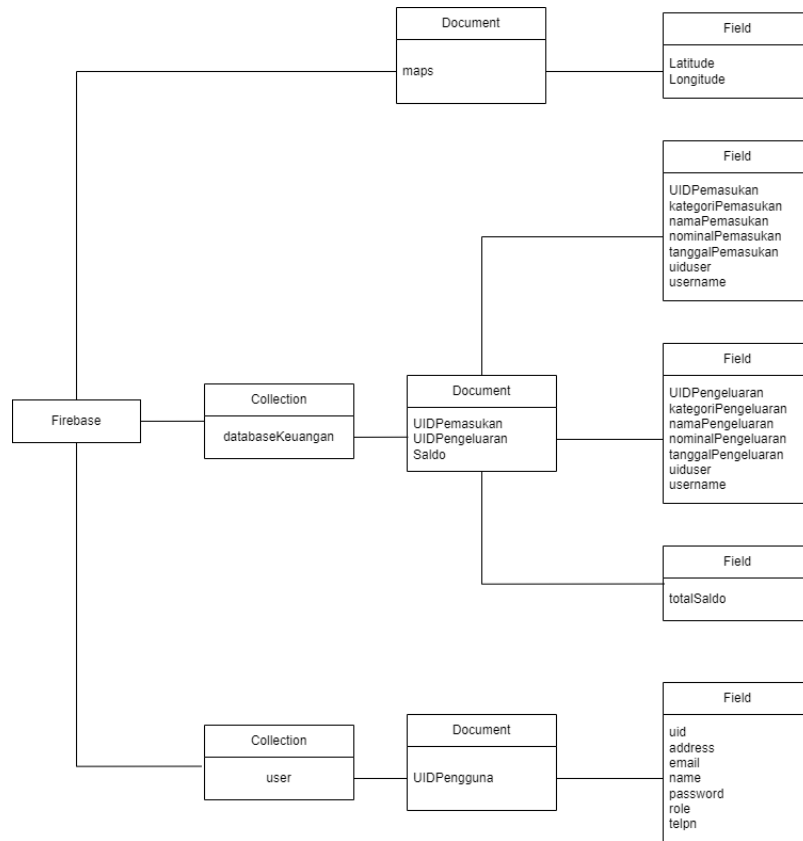


Gambar 3. 35 Struktur Basis Data *Local Storage*

2. Struktur Basis Data Firebase

Struktur basis data Firebase yang digunakan pada aplikasi ini terdiri dari beberapa *collection*. *Collection* "databaseKeuangan" digunakan untuk menyimpan data keuangan pengguna, mencatat pemasukan, pengeluaran, dan saldo dalam setiap dokumen dengan *field-field* seperti UIDPemasukan, UIDPengeluaran, dan Saldo. Setiap transaksi pemasukan dan pengeluaran disertai dengan rincian berupa kategori, nama, nominal, dan tanggal transaksi.

Di sisi lain, *Collection* "Pengguna" menyimpan informasi pengguna, seperti uid, alamat, email, nama, *password*, *role*, dan nomor telepon. Selain itu, terdapat *Collection* "maps" yang berfungsi untuk menyimpan data lokasi pengguna berdasarkan *latitude* dan *longitude*. Struktur basis data firebase dapat dilihat pada gambar 3.36.



Gambar 3. 36 Struktur Basis Data Firebase

3.4.4 Implementasi Proyek

1. Memulai Proyek

Aplikasi Masjid Al-Ikhlas Sorosutan dikembangkan dengan menggunakan *framework* ReactJS dan memanfaatkan npm sebagai pengelola paket. Sebelum memulai pengembangan, perlu dilakukan instalasi ReactJS terlebih dahulu dengan menjalankan perintah melalui terminal atau *command line* di komputer. Selain itu, dilakukan juga penginstallan PWA menggunakan *template* yang telah disediakan. Perintah instalasi ReactJS beserta *template* PWA yang digunakan dapat dilihat pada gambar 3.37 di bawah ini.

```
npx create-react-app infakpro --template cra-template-pwa
```

Gambar 3. 37 Perintah Installasi React JS Ke dalam Komputer

Setelah ReactJS berhasil diinstal, tahap selanjutnya adalah memulai proyek dengan masuk ke direktori proyek yang telah dibuat. Perintah yang digunakan untuk memulai proyek ReactJS ditunjukkan pada gambar 3.38.

```
cd infakpro
npm run start
```

Gambar 3. 38 Perintah untuk memulai proyek ReactJS

2. Mengecek file *service worker* dari PWA yang telah terinstall

Setelah berhasil menginstal PWA yang dilakukan diawal, ReactJS akan secara otomatis membuat file konfigurasi untuk *service worker*. File ini terletak di direktori `/src` dengan nama `serviceWorkerRegistration.js`. Isi file `serviceWorkerRegistration.js` ditunjukkan pada gambar 3.39.

```
const isLocalhost = Boolean(window.location.hostname === "localhost"
|| window.location.hostname === "[::1]" ||
window.location.hostname.match(/^127(?:\.[0-9]+){0,2}\.[0-9]+$/));

export function register(config) {
  if (process.env.NODE_ENV === "production" && "serviceWorker" in
navigator) {
    const publicUrl = new URL(process.env.PUBLIC_URL,
window.location.href);
    if (publicUrl.origin !== window.location.origin) {
      return;
    }

    window.addEventListener("load", () => {
      const swUrl = `${process.env.PUBLIC_URL}/service-worker.js`;

      if (isLocalhost) {
        checkValidServiceWorker(swUrl, config);
        navigator.serviceWorker.ready.then(() => {
          console.log("This web app is being served cache-first by
a service worker.");
        });
      } else {
        registerValidSW(swUrl, config);
      }
    });
  }
}

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then((registration) => {
      registration.onupdatefound = () => {
        const installingWorker = registration.installing;
        if (installingWorker == null) {
          return;
        }
        installingWorker.onstatechange = () => {
          if (installingWorker.state === "installed") {
            if (navigator.serviceWorker.controller) {
              console.log("New content is available and will be used
when all tabs for this page are closed.");
            }
          }
        }
      }
    });
}
```

```

        if (config && config.onUpdate) {
            config.onUpdate(registration);
        }
    } else {
        console.log("Content is cached for offline use.");
        if (config && config.onSuccess) {
            config.onSuccess(registration);
        }
    }
}
};
});
}
}

function checkValidServiceWorker(swUrl, config) {
    fetch(swUrl, {
        headers: { "Service-Worker": "script" },
    })
        .then((response) => {
            const contentType = response.headers.get("content-type");
            if (response.status === 404 || (contentType !== null &&
contentType.indexOf("javascript") === -1)) {
                navigator.serviceWorker.ready.then((registration) => {
                    registration.unregister().then(() => {
                        window.location.reload();
                    });
                });
            } else {
                registerValidSW(swUrl, config);
            }
        })
        .catch(() => {
            console.log("No internet connection found. App is running in
offline mode.");
        });
    }

export function unregister() {
    if ("serviceWorker" in navigator) {
        navigator.serviceWorker.ready
            .then((registration) => {
                registration.unregister();
            })
            .catch((error) => {
                console.error(error.message);
            });
    }
}
}

```

Gambar 3. 39 Isi File *ServiceWorker* Aplikasi Masjid Al-Ikhlas Sorosutan

Gambar 3.39 menunjukkan isi file `serviceWorkerRegistration.js` yang berfungsi untuk mengelola *service worker* dalam aplikasi PWA Masjid Al-Ikhlas Sorosutan. Secara bawaan, implementasi file ini memungkinkan peramban untuk mendaftarkan

dan mengaktifkan *service worker*, yang berperan penting dalam mendukung fitur *offline-first*. Hal ini dilakukan dengan mekanisme *caching* konten sehingga pengguna tetap dapat mengakses aplikasi meskipun tidak terhubung ke internet.

Proses ini dimulai dengan fungsi `register()` yang memastikan *service worker* hanya diaktifkan pada lingkungan produksi dan memuat file `service-worker.js`. Fungsi ini juga memvalidasi *service worker* melalui `checkValidServiceWorker()` untuk memastikan bahwa file yang digunakan valid dan sesuai. Selain itu, melalui fungsi `registerValidSW()`, file ini mengelola status instalasi *service worker*, termasuk pemberitahuan jika ada konten baru yang tersedia atau memastikan konten yang *di-cache* siap digunakan. Dengan demikian, implementasi ini memberikan pengalaman pengguna yang lebih baik melalui aksesibilitas aplikasi secara *offline*.

3. Konfigurasi Manifest File

Konfigurasi File Manifest digunakan untuk mendefinisikan perilaku aplikasi PWA saat diinstal pada perangkat pengguna. Pada aplikasi Masjid Al-Ikhlas Sorosutan, konfigurasi ini dilakukan dengan mengedit file `manifest.json`, yang terletak di dalam folder `public`. File ini mendefinisikan atribut seperti nama aplikasi, ikon yang digunakan, serta warna tema aplikasi ketika diinstal di perangkat. Berikut adalah contoh konfigurasi yang terdapat dalam file `manifest.json` yang ditunjukkan pada gambar 3.40.

```
{
  "short_name": "Ikhlas App",
  "name": "Mobile Ikhlas App",
  "icons": [
    {
      "src": "logoikhlas192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logoikhlas512.png",
      "type": "image/png",
      "sizes": "512x512"
    },
    {
      "src": "logoikhlas32.png",
      "type": "image/png",
      "sizes": "32x32"
    },
    {
      "src": "logoikhlas16.png",
      "type": "image/png",
      "sizes": "16x16"
    }
  ],
}
```

```

"start_url": ".",
"display": "standalone",
"theme_color": "#000000",
"background_color": "#000000"
}

```

Gambar 3. 40 Konfigurasi Manifest File PWA pada aplikasi

Penjelasan tiap objek konfigurasi antara lain:

a. *Name*

Menyatakan nama lengkap aplikasi ketika diinstal sebagai PWA di perangkat pengguna. Nama ini akan ditampilkan pada layar perangkat. Dalam konfigurasi ini, nama aplikasi yang ditampilkan adalah "*Mobile Ikhlas App*".

b. *Short-name*

Memberikan nama singkat untuk aplikasi, yang biasanya digunakan pada ikon di layar utama atau tempat lain yang membutuhkan nama yang lebih pendek. Pada konfigurasi ini, short-name yang digunakan adalah "*Ikhlas App*".

c. *Theme_color*

Menentukan warna tema yang digunakan oleh aplikasi. Warna ini biasanya muncul di bagian atas atau area navigasi perangkat saat aplikasi dibuka.

d. *Icons*

Menyediakan daftar ikon yang digunakan oleh aplikasi dalam berbagai ukuran, yang akan muncul ketika pengguna menambahkan aplikasi ke layar utama perangkat mereka.

e. *Start_url*

Menentukan halaman awal yang akan ditampilkan ketika aplikasi dijalankan. Dalam konfigurasi ini, "." berarti aplikasi akan dimulai dari root atau halaman utama.

f. *Display:*

Mengatur bagaimana aplikasi akan ditampilkan pada perangkat pengguna. Mode tampilan yang tersedia termasuk "fullscreen", "standalone", dan "minimal-ui".

Pada contoh ini, mode "standalone" digunakan, yang berarti aplikasi akan tampil seperti aplikasi asli tanpa bilah alamat peramban.

g. *Background_color:*

Menetapkan warna latar belakang yang digunakan saat aplikasi memuat atau saat splash screen muncul.

4. Implementasi React Router

Langkah selanjutnya adalah menambahkan fungsi React Router untuk navigasi dalam aplikasi. Pertama, instal React Router dengan menggunakan perintah berikut di terminal:

```
npm install react-router-dom
```

Gambar 3. 41 Perintah memasang plugin React Router pada proyek ReactJS

Setelah plugin React Router berhasil diinstal, tahap berikutnya adalah membuat konfigurasi router di dalam file App.js. Di dalam file App.js, dilakukan impor fungsi PerambanRouter, Routes, dan Route yang berasal dari plugin react-router-dom. Isi kodingan file App.js ditunjukkan pada Gambar 3.42.

```
import { PerambanRouter, Navigate, Route, Routes } from "react-
router-dom";
.....
return (
  <PerambanRouter>
    <Routes>
      <Route
        path="/"
        element={
          <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <Home />
          </PrivateRoute>
        }
      />
      <Route
        path="/landing-page"
        element={
          <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <LandingPage />
          </PrivateRoute>
        }
      />
      <Route
        path="/login"
        element={
          <Private                                    isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <Login />
          </Private>
        }
      />
      <Route
        path="/register"
        element={
          <Private                                    isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <Register />
          </Private>
        }
      />
    </Routes>
  </PerambanRouter>
)
```

```

        </Private>
    }
/>
<Route
    path="/maps"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <Maps />
        </PrivateRoute>
    }
/>
<Route
    path="/secret"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <Secret />
        </PrivateRoute>
    }
/>
<Route
    path="/laporanKeuangan"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <BerandaLaporanKeuangan data={data} />
        </PrivateRoute>
    }
/>
<Route
    path="/profile"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <Profile dataUser={dataUser} />
        </PrivateRoute>
    }
/>
<Route
    path="/notification"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <Notifcations />
        </PrivateRoute>
    }
/>
<Route
    path="/pray"
    element={
        <PrivateRoute
            isAuthenticated={pengguna}
            userRole={dataUser?.role}>
            <PrayPage />
        </PrivateRoute>
    }
/>
<Route
    path="/keuangan"
    element={

```

```

        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <KelolaKeuangan
dataKeuanganPemasukan={dataKeuanganPemasukan}
dataKeuanganPengeluaran={dataKeuanganPengeluaran} />
        </PrivateRoute>
    }
/>
<Route
    path="/pemasukan"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <FormPemasukan dataUser={dataUser} />
        </PrivateRoute>
    }
/>
<Route
    path="/pengeluaran"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <FormPengeluaran dataUser={dataUser} />
        </PrivateRoute>
    }
/>
<Route
    path="/TabelPemasukan"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <TabelPemasukan />
        </PrivateRoute>
    }
/>
<Route
    path="/TabelPengeluaran"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <TabelPengeluaran />
        </PrivateRoute>
    }
/>
<Route
    path="/TabelKeseluruhan"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <TabelKeseluruhan />
        </PrivateRoute>
    }
/>
<Route
    path="/TabelKeuangan"
    element={
        <PrivateRoute                                isAuthenticated={pengguna}
userRole={dataUser?.role}>
            <TabelKeuangan />
        </PrivateRoute>
    }
/>

```

```

    }
  />
  <Route
    path="/GrafikPemasukan"
    element={
      <PrivateRoute
        isAuthenticated={pengguna}
userRole={dataUser?.role}>
        <GrafikPemasukan />
      </PrivateRoute>
    }
  />
  <Route
    path="/GrafikPengeluaran"
    element={
      <PrivateRoute
        isAuthenticated={pengguna}
userRole={dataUser?.role}>
        <GrafikPengeluaran />
      </PrivateRoute>
    }
  />
  <Route
    path="/BerandaPanduanMengajar"
    element={
      <PrivateRoute
        isAuthenticated={pengguna}
userRole={dataUser?.role}>
        <BerandaPanduanMengajar />
      </PrivateRoute>
    }
  />
.....
</Routes>
</PerambanRouter>
);
};

export default App;
const PrivateRoute = ({ children, isAuthenticated, userRole })
=> {
  if (!isAuthenticated) {
    return <Navigate to="/login" />;
  }

  if (isAuthenticated && userRole === "admin") {
    return <Navigate to="/admin/dashboard" />;
  }

  return children; // Akses untuk pengguna non-admin
};

const PrivateRouteAdmin = ({ children, isAuthenticated,
userRole }) => {
  if (!isAuthenticated) {
    return <Navigate to="/login" />;
  }

  if (isAuthenticated && userRole !== "admin") {
    return <Navigate to="/" />;
  }

  return children; // Akses untuk admin

```

```

};

const Private = ({ children, isAuthenticated, userRole }) => {
  if (!isAuthenticated) {
    return children; // Akses ke halaman login atau halaman
    umum
  }

  if (userRole === "admin") {
    return <Navigate to="/admin/dashboard" />;
  }

  if (userRole === "pengguna") {
    return <Navigate to="/" />;
  }

  return children;
};

```

Gambar 3. 42 Contoh isi file App.js sebagai route

Kode ini adalah bagian dari aplikasi ReactJS yang menggunakan React Router untuk navigasi, serta mengimplementasikan rute privat (Private Route) yang mengontrol akses halaman berdasarkan status autentikasi dan peran pengguna (*userRole*). Berikut adalah beberapa penjelasan dari cuplikan code pada file App.js:

- a. *PerambanRouter*: Ini adalah komponen yang membungkus seluruh aplikasi dan memungkinkan penggunaan navigasi berbasis URL dalam React. *PerambanRouter* menangani URL di peramban dan mengarahkan ke rute yang sesuai berdasarkan konfigurasi.
- b. *Routes*: Berisi kumpulan rute yang didefinisikan dalam aplikasi, di mana setiap Route menentukan path URL dan komponen yang akan dirender ketika URL tersebut diakses.
- c. Komponen *PrivateRoute* dan *Private* digunakan untuk melindungi rute tertentu dalam aplikasi, memastikan hanya pengguna yang telah diautentikasi dapat mengakses halaman tersebut. *PrivateRoute* juga memeriksa peran pengguna (*userRole*) dan menentukan apakah pengguna bisa mengakses halaman atau harus diarahkan ke halaman lain.

5. Implementasi Firebase

Pada pengembangan aplikasi Masjid Al-Ikhlas, Firebase digunakan sebagai *backend* untuk mengelola autentikasi pengguna serta menyimpan data ke penyimpanan *cloud*. Penggunaan Firebase memungkinkan pengguna untuk *login* dan

menyimpan data secara *real-time*, sehingga perubahan yang dilakukan pada data dapat diakses dari berbagai perangkat.

Untuk menggunakan layanan Firebase, langkah pertama yang dilakukan adalah melakukan inisialisasi proyek di *website* Firebase. Setelah proyek berhasil didaftarkan, Firebase akan menyediakan konfigurasi SDK yang perlu ditambahkan ke dalam proyek pengembangan ReactJS. Berikut adalah konfigurasi SDK Firebase pada file `Fire.js`:

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
import { getStorage } from "firebase/storage";
import { getDatabase } from "firebase/database";
import { getMessaging } from "firebase/messaging";

const firebaseConfig = {
  apiKey: "AIzaSyDjzl2GOkgu[REDACTED]",
  authDomain: "lockinfak.firebaseio.com",
  databaseURL: "https://lockinfak-default-rtdb.asia-southeast1.firebaseio.com",
  projectId: "lockinfak",
  storageBucket: "lockinfak.appspot.com",
  messagingSenderId: "57204[REDACTED]",
  appId: "1:57204572[REDACTED]:7f219[REDACTED]",
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
export const auth = getAuth(app);
export const storage = getStorage(app);
export const database = getDatabase(app);
export const messaging = getMessaging(app);
```

Gambar 3. 43 Konfigurasi Firebase Fire.js

Setelah konfigurasi SDK berhasil ditambahkan ke dalam proyek, pengembang bisa memanfaatkan berbagai layanan Firebase dengan mengacu pada file konfigurasi Firebase di setiap fitur yang dikembangkan dalam aplikasi.

6. Implmentasi Aplikasi

Setelah semua fungsi yang direncanakan berhasil diimplementasikan melalui tahapan-tahapan persiapan yang telah dilakukan sebelumnya, pengembangan proyek PWA ini kini memasuki tahap yang lebih lanjut. Pada fase ini, aplikasi Masjid Al-Ikhlas Sorosutan sudah siap untuk diintegrasikan dengan fitur-fitur tambahan yang akan meningkatkan fungsionalitas dan memenuhi kebutuhan pengguna secara lebih efektif. Pembahasan lebih rinci mengenai hasil dari proses ini, serta pengaruhnya terhadap kinerja aplikasi, akan disampaikan pada bab berikutnya.

7. Implementasi Hosting Aplikasi

Aplikasi Masjid Al-Ikhlas Sorosutan yang telah dikembangkan perlu di-hosting di internet agar dapat diakses oleh pengguna dari berbagai perangkat. Pada tahap ini, dilakukan proses build untuk menggabungkan semua file dan modul aplikasi ke dalam satu paket yang lebih ringan dan siap didistribusikan. Proses build ini menghasilkan folder bernama "build" yang akan terletak di root directory proyek. Perintah untuk melakukan build aplikasi menggunakan npm ditunjukkan pada Gambar 3.44.

```
npm run build
```

Gambar 3. 44 perintah untuk build aplikasi react.js

3.4.5 Pengujian Aplikasi

Pada tahap ini, aplikasi yang telah dikembangkan akan diuji untuk memastikan bahwa sistem berfungsi sesuai harapan. Pengujian dilakukan dengan menggunakan metode *Black Box Testing* untuk menguji fungsi aplikasi, *Lighthouse Testing* yang disediakan oleh peramban Google Chrome untuk menilai kinerja aplikasi, serta *System Usability Scale* (SUS) untuk mengukur tingkat kepuasan pengguna terhadap antarmuka aplikasi.

1. *Black Box Testing*

Pengujian *Black Box* adalah metode pengujian yang menilai spesifikasi fungsional aplikasi tanpa memerlukan pengetahuan tentang struktur kode sumbernya. Dalam penelitian ini, pengujian dilakukan pada setiap fungsi aplikasi Masjid Al-Ikhlas Sorosutan.

2. *Lighthouse Testing*

Lighthouse Testing digunakan untuk mengevaluasi kinerja dan kelayakan aplikasi dalam menerapkan fungsi *Progressive Web App (PWA)*. Pengujian ini akan menghasilkan penilaian mengenai performa, aksesibilitas, dan *Search Engine Optimization* (SEO).

3. *System Usability Scale (SUS)*

Pengujian *System Usability Scale* (SUS) diterapkan untuk menilai apakah sistem berfungsi dengan baik sesuai harapan. Pengujian ini dilakukan dengan membagikan kuesioner yang berisi pertanyaan mengenai kegunaan sistem dan kenyamanan pengguna dalam menggunakannya.

4. *Prosedur Pengujian Black Box Testing dan Lighthouse Testing*

Pengujian aplikasi Masjid Al-Ikhlas Sorosutan dilakukan dengan persiapan sebagai berikut.

h. Perangkat Keras

Perangkat keras berupa laptop dengan spesifikasi sebagai berikut:

1. Processor Intel Core i5-9300H
2. 8GB RAM
3. 512GB SSD
4. GPU Nvidia GTX 1650

Perangkat keras lain berupa *smartphone* dengan spesifikasi sebagai berikut:

1. Processor Qualcomm Snapdragon 712
2. GPU Adreno 616
3. 4GB RAM

i. Perangkat Lunak

Perangkat lunak yang digunakan dalam perangkat laptop antara lain:

1. Peramban Google Chrome Dev Versi 110.0.5464.2 47
2. Sistem operasi Windows 10

Perangkat lunak yang digunakan dalam perangkat *smartphone* android antara lain:

1. Sistem operasi Android 10
2. Peramban Google Chrome Android Versi 100.0.4896.79

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Aplikasi

Pada bagian ini diuraikan hasil implementasi fitur aplikasi Masjid Al-Ikhlas Sorosutan yang dibuat berdasarkan perancangan kebutuhan dan desain sistem yang sudah dilakukan. Hasil implementasi ditunjukkan pada tangkapan layar tampilan setiap halaman beserta penjelasan fungsinya sebagai berikut:

4.1.1 Tampilan Halaman Beranda Aplikasi

Halaman beranda aplikasi merupakan halaman awal ketika aplikasi dibuka. Pada halaman ini, disajikan informasi mengenai waktu salat berikutnya yang terdekat serta total saldo keuangan masjid, yang ditampilkan secara jelas untuk memudahkan pengguna dalam memantau informasi penting tersebut. Selain itu, terdapat empat tombol utama yang dirancang untuk mengarahkan pengguna ke fitur-fitur inti aplikasi, yaitu *monitoring* kotak infak, pencatatan keuangan, jadwal waktu salat, dan panduan mengajar TPA.

Selain itu, terdapat juga *navbar* yang menyediakan navigasi cepat bagi pengguna. *Navbar* ini terdiri dari beberapa opsi, yaitu: Beranda untuk kembali ke halaman utama aplikasi, Profil untuk mengakses halaman Profil Pengguna, serta *Logout* untuk keluar dari aplikasi. Tampilan *desktop* dari halaman beranda aplikasi ditunjukkan pada Gambar 4.1. Tampilan *mobile* dari halaman beranda aplikasi ditunjukkan pada Gambar 4.2.



Gambar 4. 1 Tampilan *Desktop* Halaman Beranda Aplikasi



Gambar 4. 2 Tampilan *Mobile* Halaman Beranda Aplikasi

4.1.2 Tampilan *Monitoring* Posisi Kotak Infak

Halaman *Monitoring* Kotak Infak merupakan fitur yang memungkinkan pengguna untuk memantau lokasi kotak infak secara *real-time*. Data koordinat dari kotak infak dikirimkan secara otomatis ke aplikasi melalui *virtual private server*, yang kemudian diolah dan ditampilkan pada peta dalam aplikasi. Peta dalam aplikasi memanfaatkan layanan API dari MapTiler untuk memberikan tampilan yang akurat. Pengguna dapat melihat posisi kotak infak yang ditandai dengan ikon khusus di peta, sehingga memudahkan dalam pemantauan kotak infak.

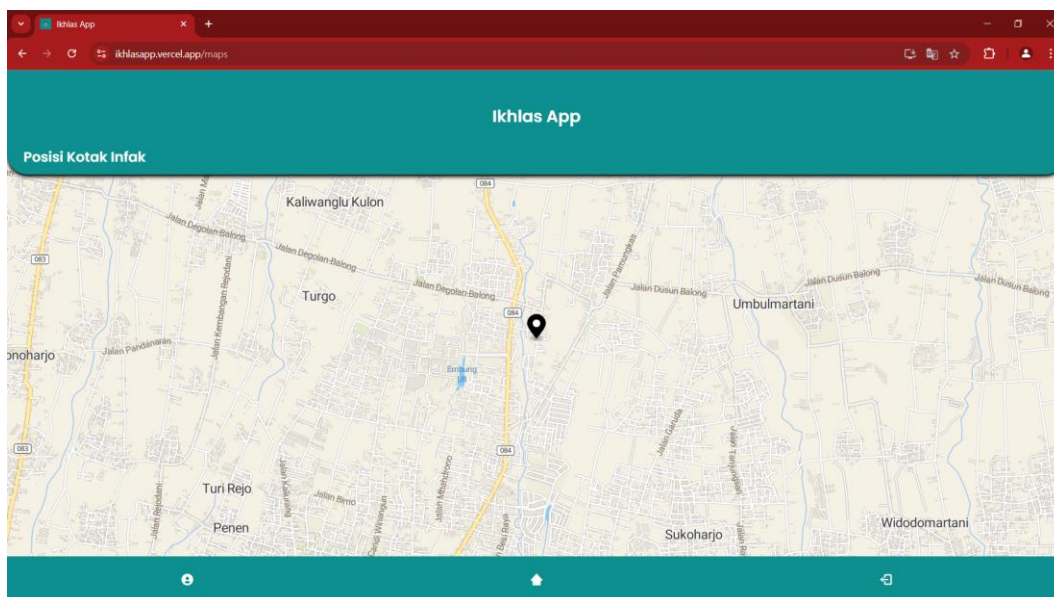
Perangkat kotak infak berbasis *Internet of Things* (IoT) yang digunakan pada Masjid Al-Ikhlas Sorosutan merupakan hasil dari Pekan Ilmiah Mahasiswa Nasional (PIMAS) ke-36 tahun 2023 yang diselenggarakan oleh Pusat Prestasi Nasional (Puspresnas) melalui Program Kreativitas Mahasiswa (PKM) skema Pengabdian Kepada Masyarakat. Proyek ini diciptakan untuk mengatasi permasalahan utama yang terjadi di masjid Al-Ikhlas Sorosutan melalui solusi pendekatan aplikasi berbasis *Internet of Things* (IoT).

Alat *Internet of Things* (IoT) yang diterapkan pada kotak infak di Masjid Al-Ikhlas Sorosutan dirancang untuk meningkatkan keamanan dengan menggunakan berbagai komponen teknologi. Lylygo, yang mengintegrasikan fungsi GPS dan ESP32, berperan sebagai mikrokontroler utama yang bertugas mengelola komunikasi, pengelolaan data, dan pelacakan

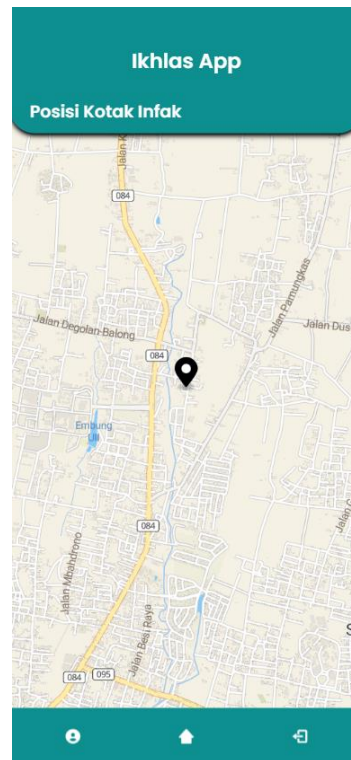
lokasi kotak infak secara *real-time*. *Radio Frequency Identification* (RFID) digunakan untuk memberikan akses aman ke kotak infak menggunakan kartu khusus takmir yang terdaftar, sehingga hanya takmir yang berwenang dapat membukanya.

Proses pengiriman data lokasi dari perangkat IoT ke aplikasi melalui Firebase dimulai ketika modul GPS pada Lylygo membaca koordinat *latitude* dan *longitude* kotak infak. Data ini kemudian diformat dalam struktur JSON oleh Lylygo dan dikirimkan ke Firebase melalui konektivitas seluler yang juga didukung oleh Lylygo. Firebase menyimpan data tersebut dalam *database* secara *real-time*, sehingga dapat diakses oleh aplikasi Masjid Al-Ikhlas untuk memonitor lokasi kotak infak secara akurat dan efisien.

Sistem akan secara terus-menerus melakukan perulangan untuk membandingkan data lokasi kotak infak yang tersimpan di *database* Firebase dengan koordinat lokasi Masjid Al-Ikhlas Sorosutan. Jika *latitude* dan *longitude* kotak infak yang tercatat di Firebase berada di luar radius 47 meter dari lokasi masjid, sistem akan secara otomatis mengirimkan notifikasi kepada takmir bahwa kotak infak berada di luar area masjid. Adapun hasil tampilan *desktop* halaman ini ditunjukkan pada gambar 4.3 dan tampilan *mobile* ditunjukkan pada gambar 4.4.



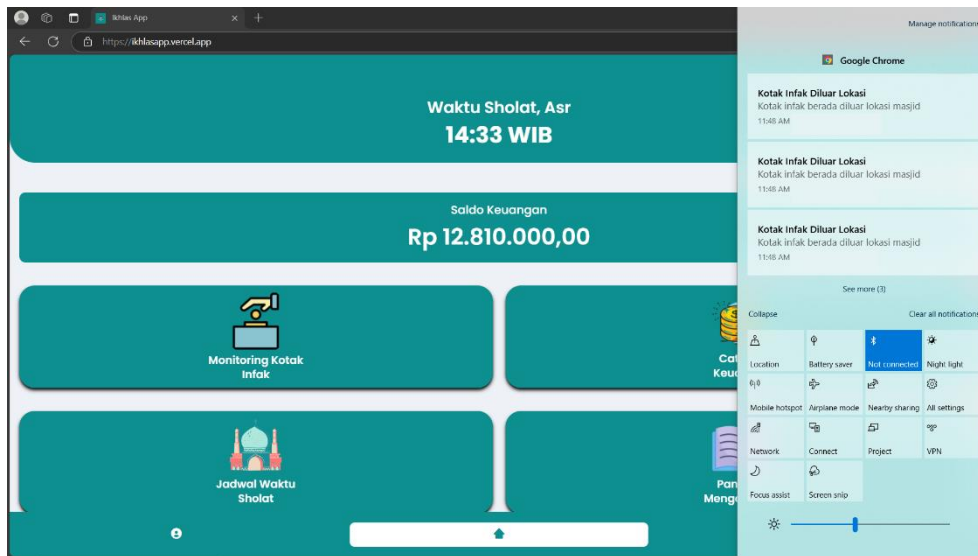
Gambar 4. 3 Tampilan *Desktop* Halaman Pantau Posisi Kotak Infak



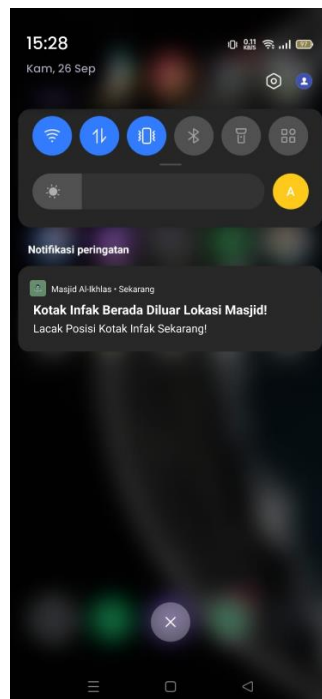
Gambar 4. 4 Tampilan *Mobile* Halaman Pantau Posisi Kotak Infak

4.1.3 Tampilan Notifikasi Kotak Infak

Notifikasi Kotak Infak adalah salah satu fitur penting dalam aplikasi ini yang memungkinkan pengguna untuk menerima notifikasi jika posisi kotak infak berada di luar area Masjid Al-Ikhlas Sorosutan. Sistem ini bekerja dengan memantau titik koordinat kotak infak yang tercatat dalam *database*. Jika titik koordinat tersebut berada di luar jangkauan area Masjid Al-Ikhlas Sorosutan, yang ditetapkan sebesar 47 meter persegi, maka sistem secara otomatis akan mengirimkan *push notification* kepada pengguna. Notifikasi ini menandakan bahwa kotak infak telah keluar dari area masjid, sehingga memungkinkan tindakan pencegahan segera. Fitur ini dirancang untuk meningkatkan keamanan dan memastikan bahwa kotak infak tetap berada di lokasi yang seharusnya.



Gambar 4. 5 Tampilan *Desktop* Notifikasi Kotak Infak



Gambar 4. 6 Tampilan *Mobile* Notifikasi Kotak Infak

4.1.4 Tampilan Halaman Catatan Keuangan

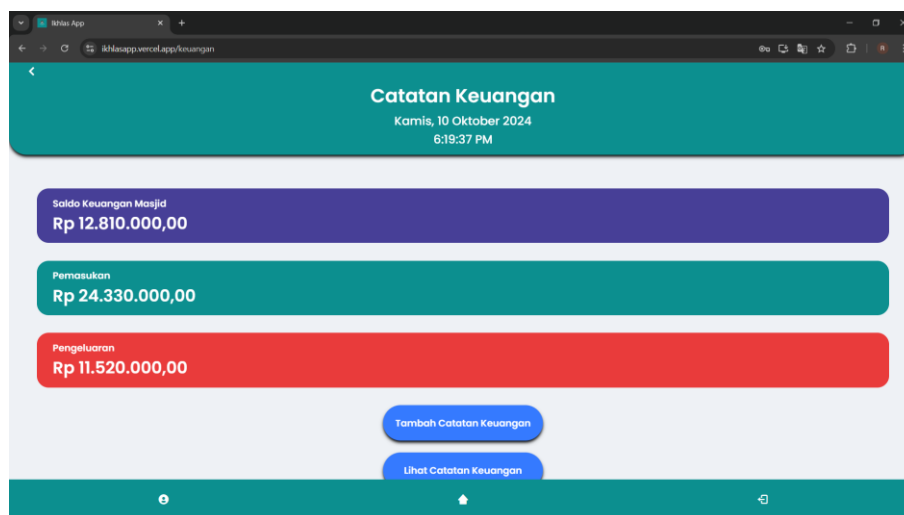
Halaman catatan keuangan merupakan halaman yang memungkinkan pengguna untuk menambahkan aktivitas catatan keuangan yang terjadi di Masjid Al-Ikhlâs Sorosutan. Pada awal halaman ini, sistem akan menampilkan informasi saldo keuangan masjid, total pemasukan, dan total pengeluaran. Pengguna dapat menekan tombol "Tambah Catatan" untuk menambahkan catatan keuangan, lalu memilih antara catatan pemasukan atau pengeluaran.

Halaman "Tambah Pemasukan" merupakan halaman yang membantu pengguna untuk menambahkan catatan keuangan yang masuk ke kas masjid. Pengguna diminta untuk mengisi

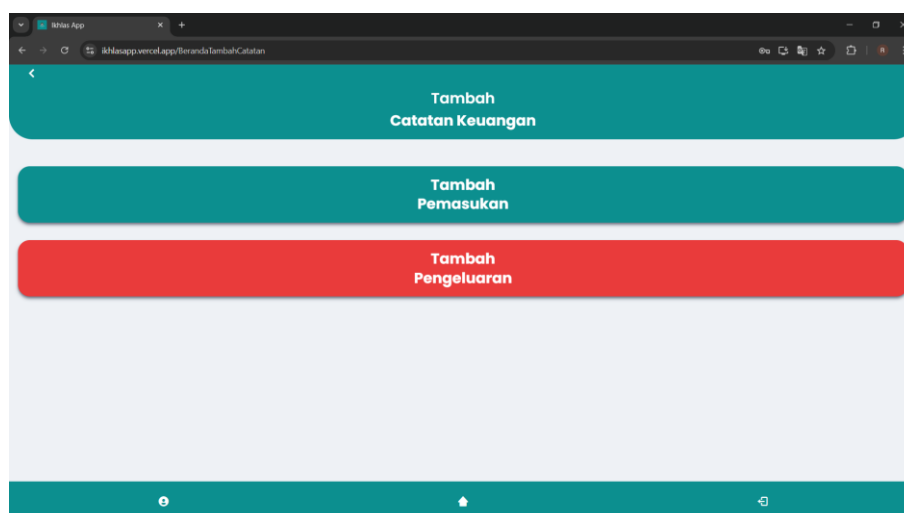
formulir yang meliputi tanggal pemasukan, kategori pemasukan (seperti Kotak Infak, Zakat, Shadaqah, Barang, dan Lainnya), nama pemasukan, dan nominal pemasukan. Setelah semua form terisi, pengguna harus menekan tombol "Masukkan," dan data pemasukan akan tersimpan dalam *database* Firebase.

Halaman "Tambah Pengeluaran" membantu pengguna mencatat pengeluaran dari kas masjid. Pengguna diminta untuk mengisi formulir yang mencakup tanggal pengeluaran, kategori pengeluaran (seperti Operasional Masjid, Inventaris Masjid, Shadaqah/Santunan, dan Lainnya), nama pengeluaran, dan nominal pengeluaran. Setelah semua form terisi, pengguna harus menekan tombol "Masukkan" dan data pengeluaran akan tersimpan dalam *database* Firebase.

Adapun hasil tampilan *desktop* halaman ini ditunjukkan pada gambar



Gambar 4. 7 Tampilan *Desktop* Beranda Catatan Keuangan



Gambar 4. 8 Tampilan *Desktop* Tambah Catatan Keuangan

The screenshot shows a desktop browser window displaying the 'Catatan Pemasukan' form. The form has a teal header with a back arrow and the title 'Catatan Pemasukan'. Below the header, there are four input fields: 'Tanggal Pemasukan' with a date picker showing 'hh/bb/tttt', 'Kategori Pemasukan' with a dropdown menu labeled 'Pilih Kategori', 'Nama Pemasukan' with a text input field containing 'Masukan Nama Pemasukan', and 'Nominal Pemasukan' with a text input field containing 'Rp. Masukan Nominal Pemasukan'. At the bottom center, there is a blue button labeled 'Masukan'.

Gambar 4. 9 Tampilan *Desktop* Formulir Pemasukan Keuangan

The screenshot shows a desktop browser window displaying the 'Catatan Pengeluaran' form. The form has a teal header with a back arrow and the title 'Catatan Pengeluaran'. Below the header, there are four input fields: 'Tanggal Pengeluaran' with a date picker showing 'hh/bb/tttt', 'Kategori Pengeluaran' with a dropdown menu labeled 'Pilih Kategori', 'Nama Pengeluaran' with a text input field containing 'Masukan Nama Pengeluaran', and 'Nominal Pengeluaran' with a text input field containing 'Rp. Masukan Nominal Pengeluaran'. At the bottom center, there is a blue button labeled 'Masukan'.

Gambar 4. 10 Tampilan *Desktop* Formulir Pengeluaran Keuangan

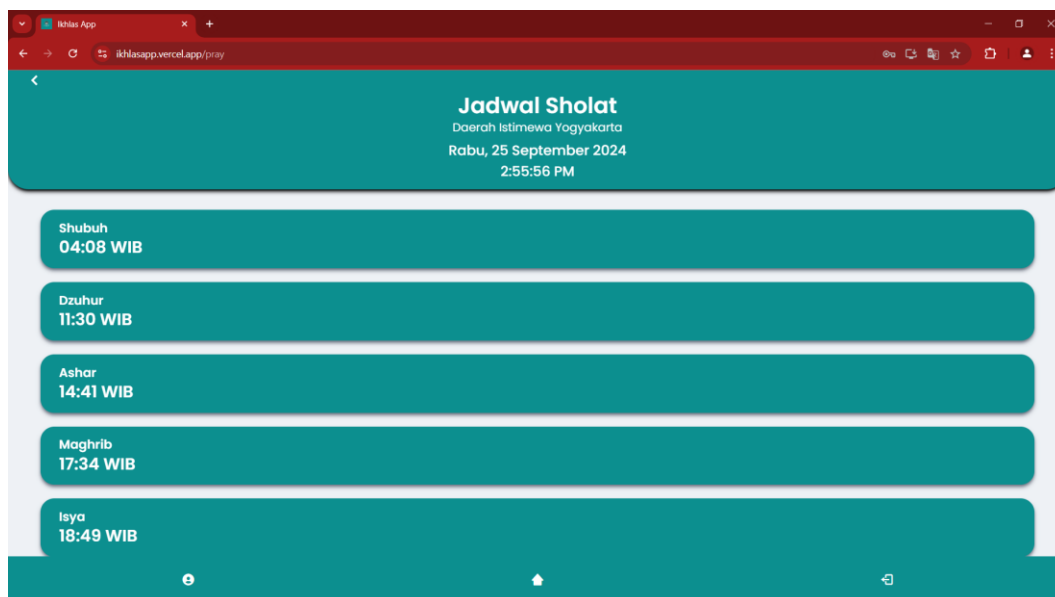
The screenshot displays four mobile app screens side-by-side. From left to right: 1. 'Catatan Keuangan' showing a summary for 'Selasa, 13 Agustus 2024 12:02:03 AM' with three cards: 'Saldo Keuangan Masjid Rp 10.790.000,00', 'Pemasukan Rp 28.625.000,00', and 'Pengeluaran Rp 17.835.000,00', with a 'Tambah Catatan' button. 2. 'Tambah Catatan Keuangan' with two buttons: 'Tambah Pemasukan' (teal) and 'Tambah Pengeluaran' (red). 3. 'Catatan Pengeluaran' form with fields for date, category, name, and nominal, and a 'Masukan' button. 4. 'Catatan Pemasukan' form with fields for date, category, name, and nominal, and a 'Masukan' button. All screens have a teal header and a bottom navigation bar.

Gambar 4. 11 Tampilan *Mobile* Halaman Catatan Keuangan

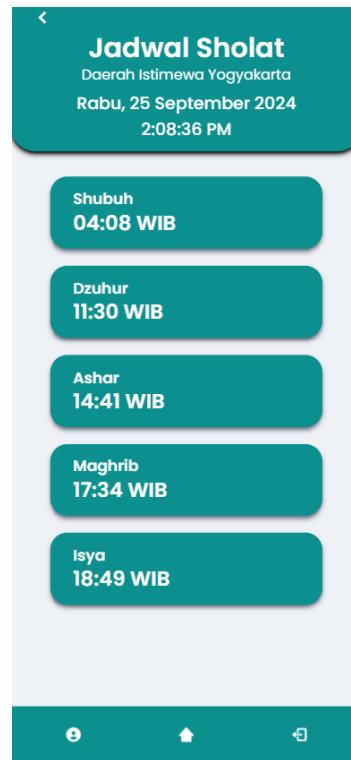
4.1.5 Tampilan Halaman Jadwal Waktu Salat

Halaman jadwal waktu salat adalah halaman yang digunakan oleh pengguna untuk melihat waktu salat harian. Pada halaman ini, ditampilkan informasi mengenai jadwal salat berdasarkan hari dan tanggal saat pengguna membuka halaman ini. Waktu salat yang ditampilkan mencakup *Subuh*, *Zuhur*, *Asar*, *Maghrib*, dan *Isya*. Data jadwal waktu salat ini diambil secara otomatis dari API Aladhan, sehingga waktu yang ditampilkan selalu akurat dan sesuai dengan lokasi pengguna.

Untuk mengintegrasikan API Aladhan dengan aplikasi Masjid Al-Ikhlash Sorosutan, pertama, sistem mengakses data jadwal salat harian dari API Aladhan berdasarkan lokasi geografis masjid, yang diperoleh melalui koordinat *latitude* dan *longitude*. Data tersebut diambil dengan mengirimkan permintaan (*request*) ke *endpoint* API Aladhan yang mencakup parameter tanggal, *latitude*, *longitude*, dan metode perhitungan waktu salat. Hasilnya berupa waktu salat *Subuh*, *Zuhur*, *Asar*, *Maghrib*, dan *Isya* yang dikirimkan dalam format JSON. Data ini kemudian diproses oleh aplikasi untuk ditampilkan secara *real-time* di halaman jadwal salat. Adapun tampilan *desktop* halaman ini ditujukan pada gambar 4.4.



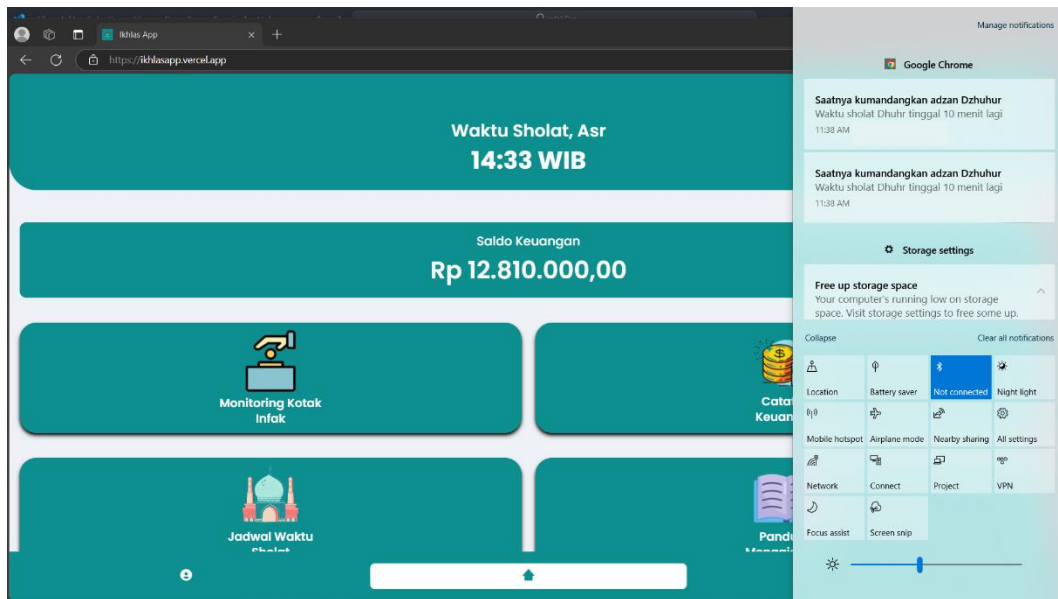
Gambar 4. 12 Tampilan *Desktop* Halaman Jadwal Waktu Salat



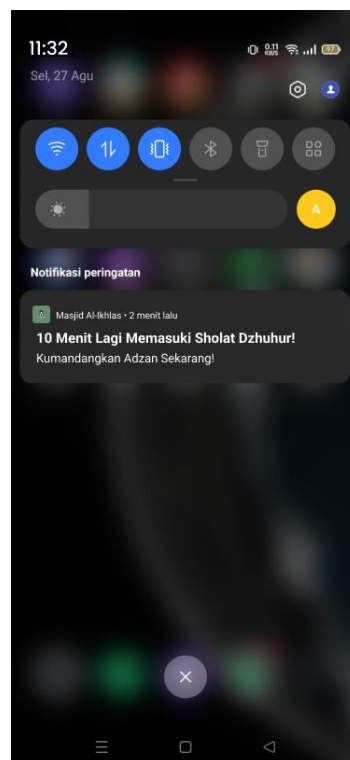
Gambar 4. 13 Tampilan *Mobile* Halaman Jadwal Waktu Salat

4.1.6 Tampilan Notifikasi Azan Salat

Notifikasi Azan Salat adalah salah satu fitur yang memungkinkan pengguna untuk menerima notifikasi ketika waktu salat akan segera tiba, yaitu 10 menit sebelum masuknya waktu salat. Sistem secara otomatis mencari data waktu salat berikutnya dan, ketika sudah mendekati 10 menit sebelum waktu salat tersebut, sistem akan mengirimkan *push notification* kepada pengguna. Notifikasi ini berfungsi sebagai pengingat bagi pengguna untuk segera mengumandangkan azan, memastikan bahwa waktu salat dijalankan tepat waktu. Fitur ini dirancang untuk membantu pengguna dalam menjaga ketepatan waktu dalam pelaksanaan azan dan salat.



Gambar 4. 14 Tampilan *Desktop* Notifikasi Azan Salat



Gambar 4. 15 Tampilan *Mobile* Notifikasi Azan Salat

4.1.7 Tampilan Halaman Panduan Mengajar TPA

Halaman Panduan Mengajar TPA adalah halaman yang dirancang untuk memudahkan pengguna dalam mengakses materi pengajaran Taman Pendidikan Al-Qur'an (TPA), yang terdiri dari enam bab: Tajwid dan Makhraj, Hafalan Surat Pendek, Doa Sehari-hari, Rukun Islam dan Iman, serta Kisah Nabi-nabi. Setiap bab berisi materi yang tersusun rapi dan diambil dari sumber-sumber terpercaya seperti Kementerian Agama (Kemenag) dan buku-buku

referensi yang diakui, sehingga dapat menjadi panduan yang jelas dan sistematis bagi para pengajar dalam menyampaikan pelajaran.

Bab Tajwid dan Makhraj terdiri dari enam materi utama, yaitu Makharijil Huruf, Hukum Bacaan Nun Sukun dan Tanwin, Hukum Bacaan Mim Sukun, Hukum Bacaan Alif Lam, Hukum Bacaan Mad Panjang, dan Hukum Bacaan Qalqalah. Materi-materi ini diambil dari sumber yang terpercaya, termasuk buku karya Dr. Abu Zakariya Sutrisno serta referensi resmi dari Kementerian Agama (Kemenag). Tampilan materi dilengkapi dengan gambar ilustratif dan contoh-contoh ayat Al-Qur'an, yang membantu pengguna dalam memahami dan menerapkan hukum-hukum tajwid.

Bab Hafalan Surat Pendek terdiri dari 13 surat pendek yang terdapat dalam Al-Qur'an juz 30, yaitu An-Nas, Al-Falaq, Al-Ikhlash, Al-Lahab, An-Nasr, Al-Kafirun, Al-Kautsar, Al-Maun, Al-Quraisy, Al-Fil, Al-Humazah, Al-Asr, dan At-Takasur. Sumber materi untuk bab ini diambil dari *website* Al-Qur'an resmi Kementerian Agama (Kemenag). Setiap materi dalam bab ini menampilkan ayat-ayat dalam bahasa Arab, transliterasi latin, serta terjemahan arti dari setiap ayat, sehingga memudahkan pengguna dalam menghafal dan memahami makna dari surat-surat tersebut.

Bab Doa Sehari-hari terdiri dari 15 doa yang sering digunakan dalam aktivitas sehari-hari, yaitu doa sebelum tidur, doa bangun tidur, doa sebelum makan, doa setelah makan, doa masuk WC, doa keluar WC, doa melepas pakaian, doa memakai pakaian, doa bercermin, doa keluar rumah, doa masuk rumah, doa sebelum wudu, doa setelah wudu, doa masuk masjid, dan doa keluar masjid. Sumber materi untuk bab ini berasal dari Kementerian Agama (Kemenag). Tampilan materi dilengkapi dengan gambar, doa dalam bahasa Arab dan Latin, serta terjemahan artinya, sehingga memudahkan pengguna dalam menghafal dan memahami makna dari doa-doa tersebut.

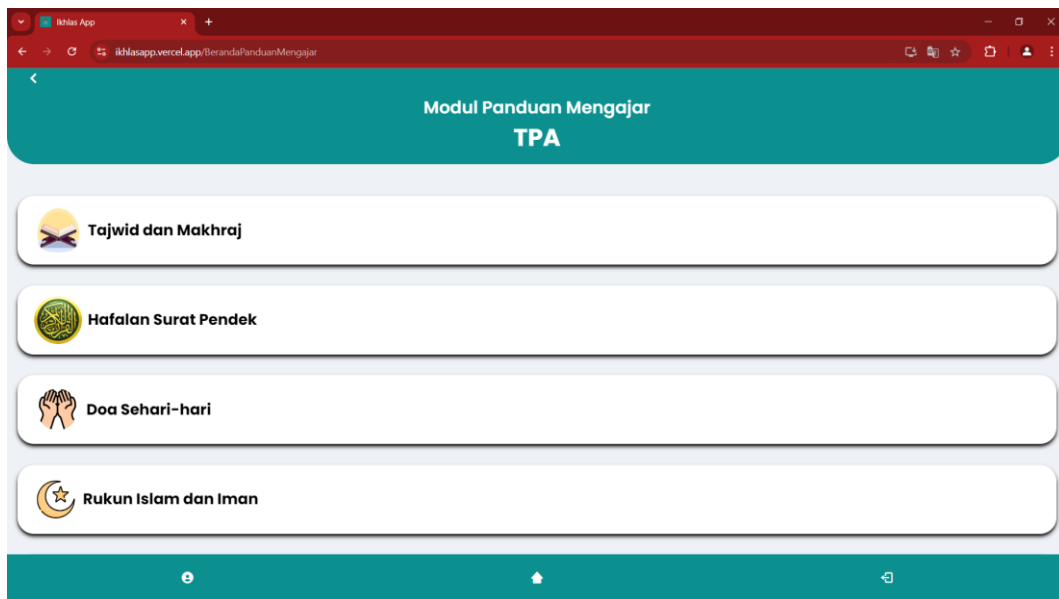
Bab Rukun Islam dan Iman terdiri dari dua materi, yaitu Rukun Islam dan Rukun Iman. Sumber materi ini berasal dari Kementerian Agama (Kemenag). Penjelasan untuk setiap materi dilengkapi dengan gambar ilustratif, teks dalam huruf Arab dan Latin, serta terjemahannya, sehingga memudahkan pengguna dalam memahami dan mengajarkan konsep dasar dalam Islam.

Bab Wudu dan Salat terdiri dari enam materi, yaitu tata cara wudu, hal-hal yang membatalkan wudu, syarat-syarat salat, rukun salat, tata cara salat, dan hal-hal yang membatalkan salat. Sumber materi dalam bab ini diambil dari referensi resmi Kementerian Agama (Kemenag). Setiap penjelasan dalam materi ini dilengkapi dengan gambar ilustratif,

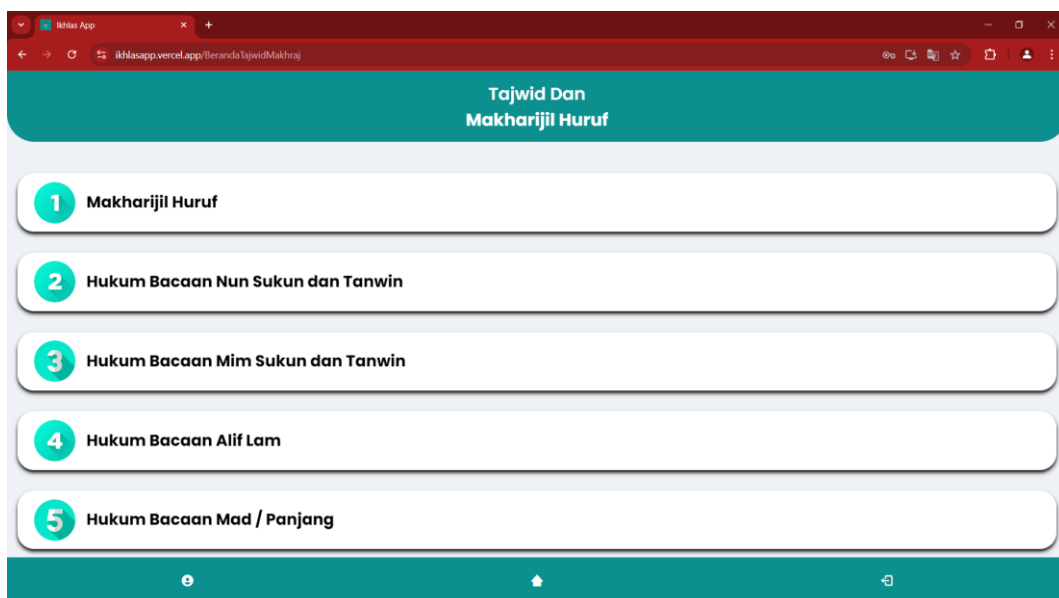
teks dalam huruf Arab dan Latin, serta terjemahannya, yang membantu pengguna untuk memahami dan mempraktikkan wudu dan salat dengan benar sesuai dengan tuntunan Islam.

BAB Kisah-kisah Nabi terdiri dari cerita singkat mengenai 25 nabi dan rasul, yang disusun berdasarkan sumber dari Gramedia. Materi ini dirancang untuk memberikan pemahaman dasar kepada pengguna tentang kehidupan dan ajaran para nabi dan rasul.

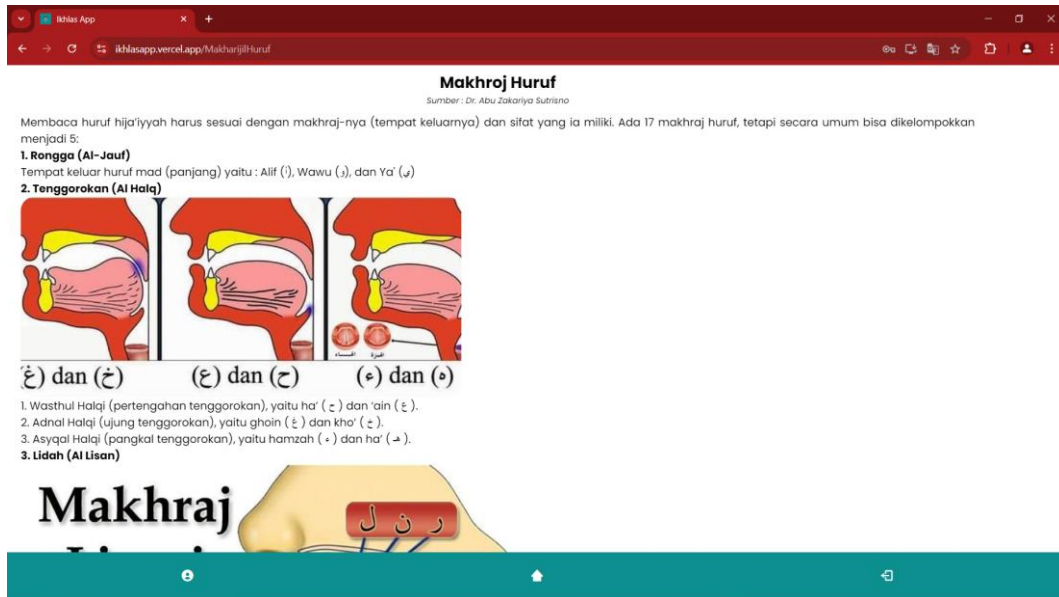
Adapun hasil tampilan *desktop* halaman panduan mengajar TPA dapat dilihat pada gambar 4.16 hingga 4.19.



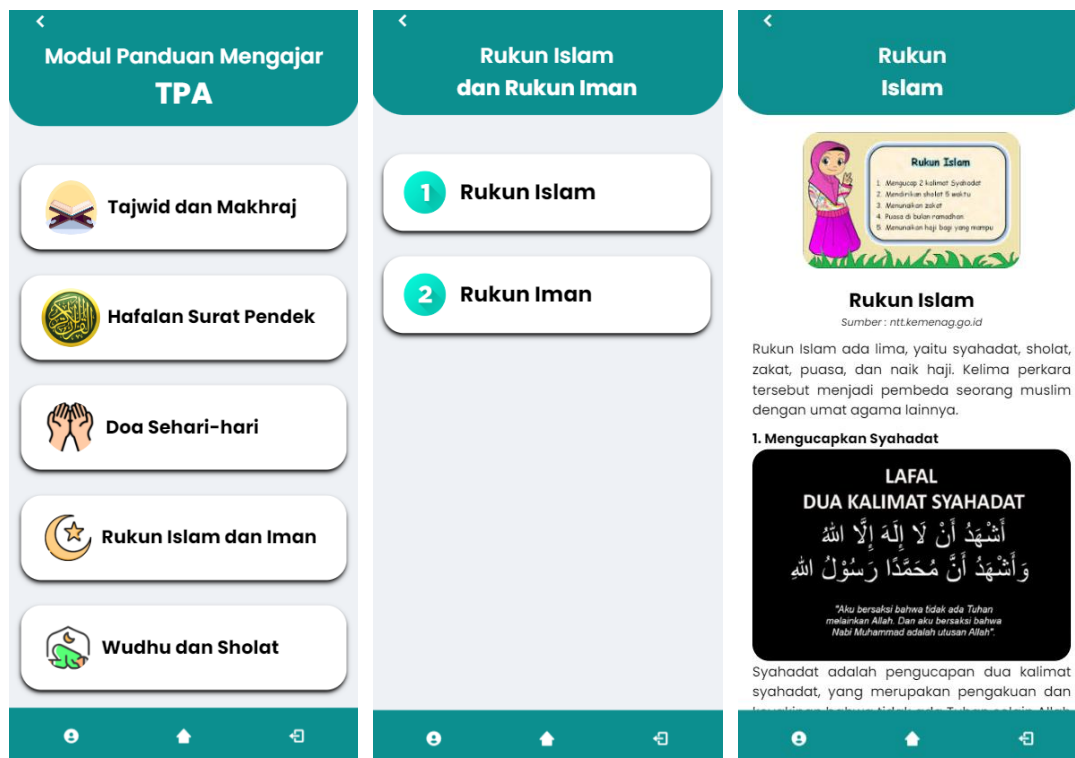
Gambar 4. 16 Tampilan *Desktop* Beranda Modul Panduan Mengajar TPA



Gambar 4. 17 Tampilan *Desktop* Beranda Salah Satu Modul Panduan Mengajar TPA



Gambar 4. 18 Tampilan *Desktop* Salah Satu Materi Modul Panduan Mengajar TPA



Gambar 4. 19 Tampilan *Mobile* Modul Panduan Mengajar TPA

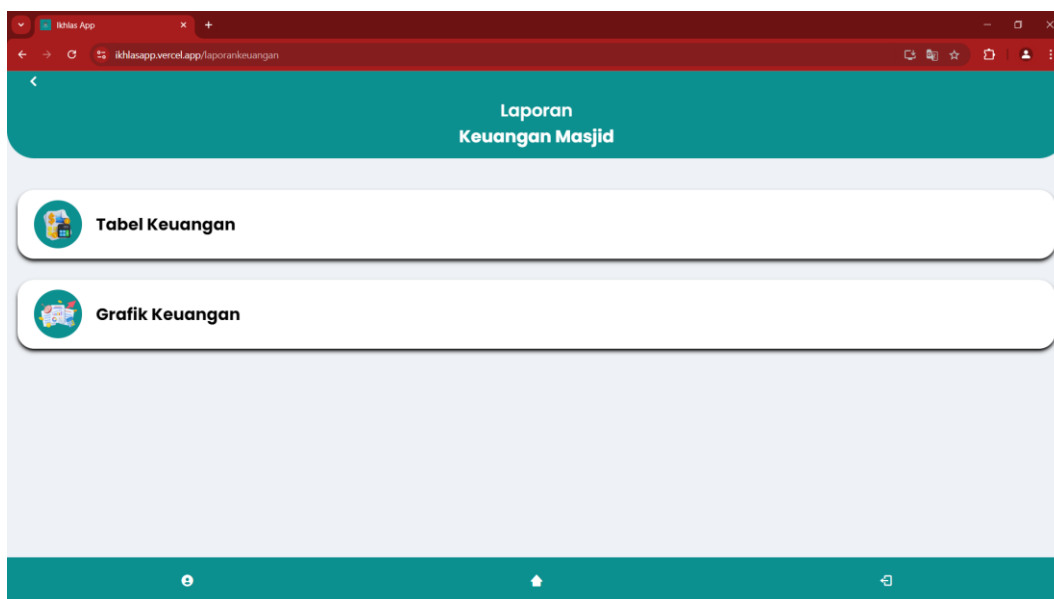
4.1.8 Tampilan Halaman Lihat Laporan Keuangan

Halaman Lihat Laporan Keuangan adalah halaman yang dirancang untuk memungkinkan pengguna melihat aktivitas catatan keuangan yang telah terjadi di Masjid Al-Ikhlash Sorosutan. Pengguna dapat memilih untuk melihat catatan keuangan dalam bentuk tabel, yang menampilkan rincian transaksi secara terstruktur, atau dalam bentuk grafik, yang memberikan visualisasi data keuangan untuk memudahkan analisis dan pemahaman tren keuangan masjid.

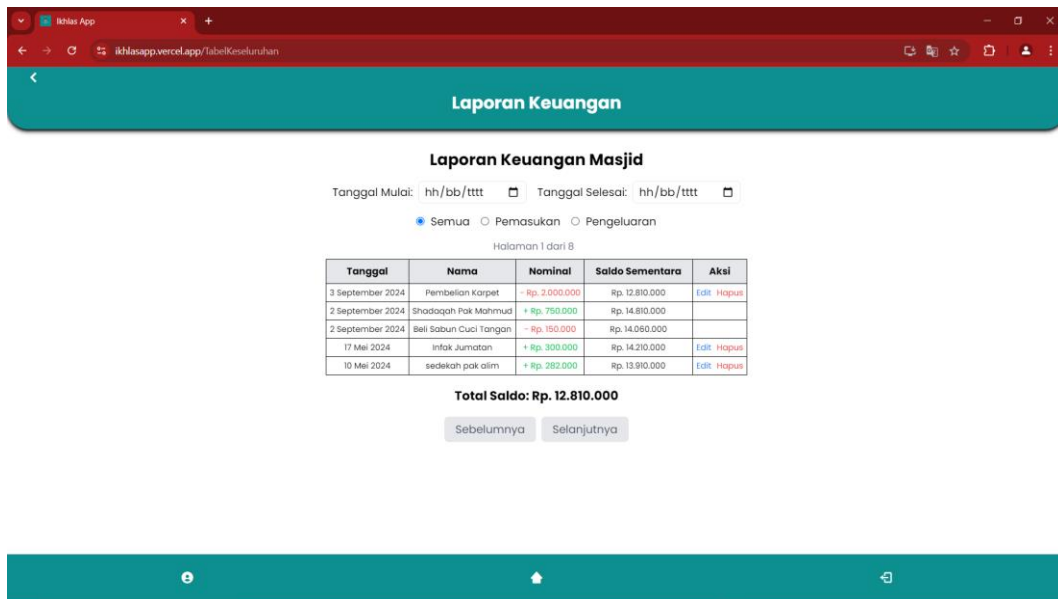
Pada halaman Tabel Keuangan, pengguna dapat melihat data keuangan yang disajikan dalam bentuk tabel yang diurutkan berdasarkan tanggal terbaru. Pengguna juga dapat memfilter tabel keuangan tersebut berdasarkan rentang tanggal, jenis catatan keuangan (semua, pemasukan, pengeluaran), serta kriteria urutan tabel seperti kategori, nominal terbesar, atau nominal terkecil. Selain itu, pengguna dapat melihat saldo keuangan, baik itu total saldo, total pemasukan, maupun total pengeluaran. Tabel tersebut juga menampilkan informasi seperti tanggal, kategori, nama, dan nominal. Data ditampilkan sebanyak 5 baris pada tiap halaman tabel, selanjutnya pengguna dapat menekan tombol selanjutnya ataupun sebelumnya untuk mengganti halaman tabel.

Pada halaman Grafik Keuangan, pengguna dapat melihat data keuangan yang disajikan dalam bentuk grafik *line chart* yang mencakup periode satu tahun. Pengguna dapat memilih jenis grafik yang ingin ditampilkan, baik itu semua transaksi, pemasukan, atau pengeluaran, serta memilih tahun yang ingin ditampilkan. Selain itu, pengguna juga memiliki opsi untuk menekan grafik pada tiap bulan untuk melihat nominal secara detail.

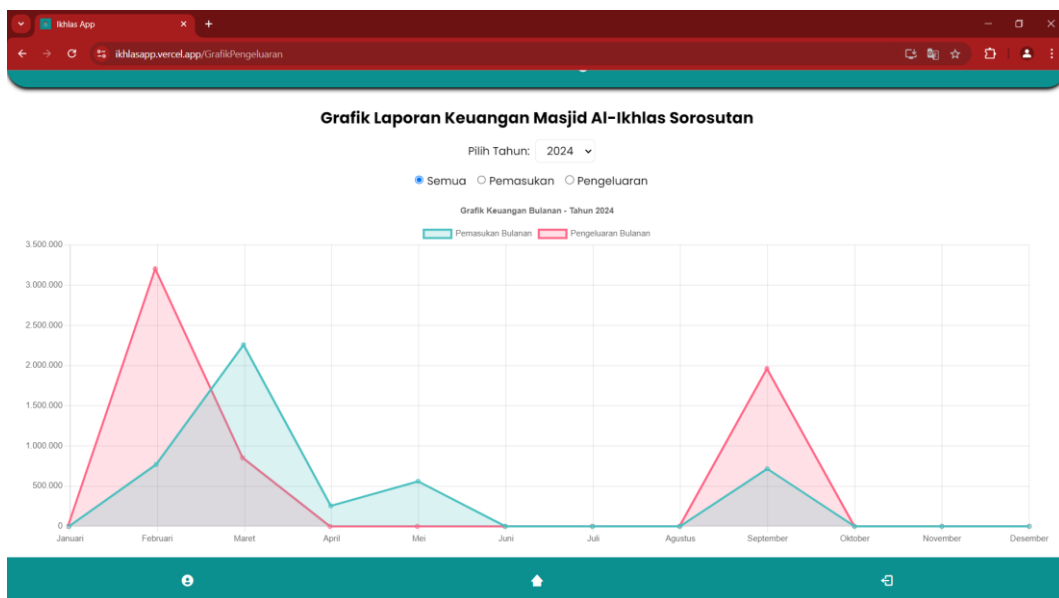
Adapun hasil tampilan *desktop* halaman lihat laporan keuangan dapat dilihat pada gambar 4.20 hingga 4.23.



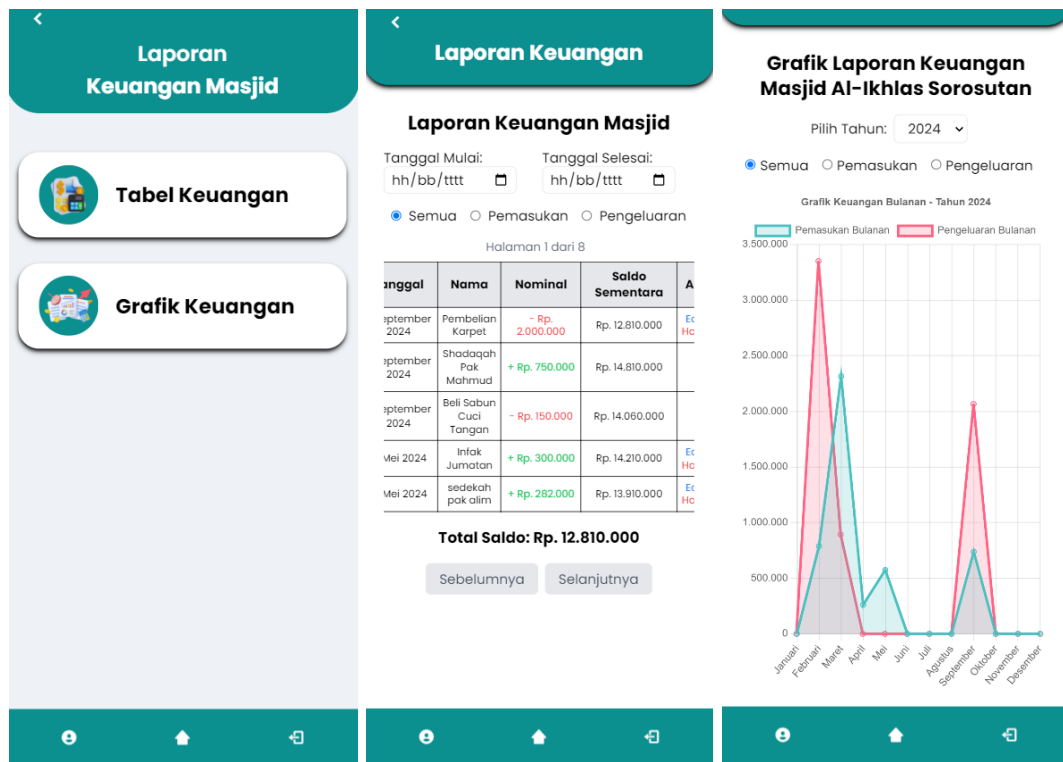
Gambar 4. 20 Tampilan *Desktop* Beranda Laporan Keuangan



Gambar 4. 21 Tampilan *Desktop* Tabel Laporan Keuangan

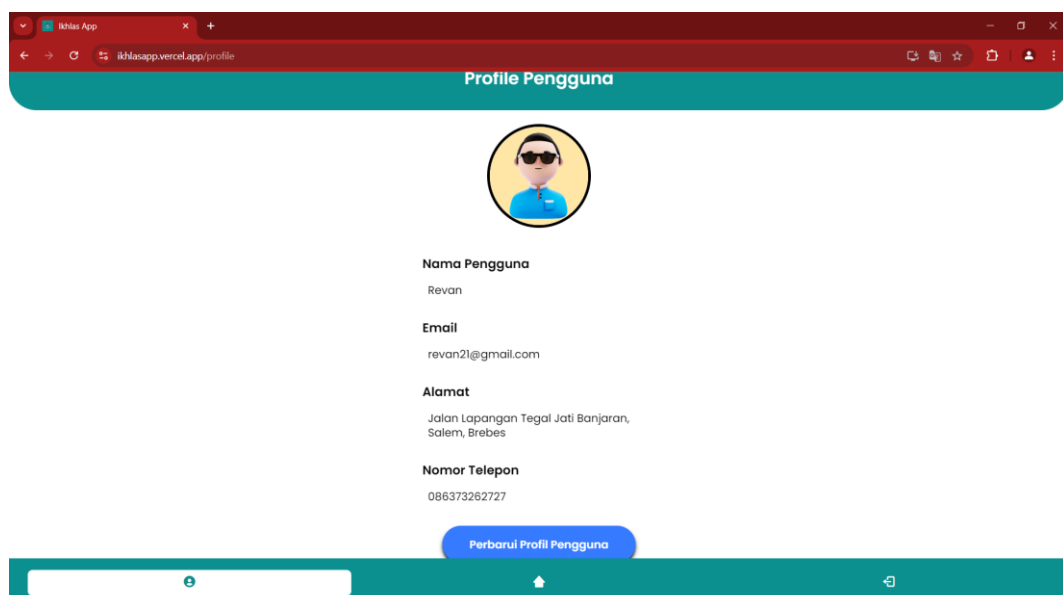


Gambar 4. 22 Tampilan *Desktop* Grafik Laporan Keuangan

Gambar 4. 23 Tampilan *Mobile* Laporan Keuangan

4.1.9 Tampilan Halaman *Profile* Pengguna

Halaman Profil Pengguna adalah halaman yang dirancang untuk memungkinkan pengguna melihat dan mengelola data identitas mereka. Pada halaman ini, pengguna dapat melakukan perubahan terhadap profil mereka dengan menekan tombol “Perbarui Profil Pengguna”. Lalu sistem akan mengambil UID pengguna untuk kemudian diarahkan dan ditampilkan pada halaman formulir ubah data pengguna. Selanjutnya pengguna dapat mengubah profil sesuai dengan kebutuhan mereka dan sistem akan menyimpan perubahan dalam *database* firebase.



Gambar 4. 24 Tampilan *Desktop* Halaman Profil Pengguna

Perbarui Data Pengguna

Email
revan21@gmail.com

Nama Pengguna
Revan

Password
12345678

Telepon
086373262727

Alamat
Jalan Lapangan Tegal Jati Banjaran, Salem, Brebes

Perbarui

Gambar 4. 25 Tampilan *Desktop* Halaman Formulir Ubah Data Pengguna

Profile Pengguna

Perbarui Data Pengguna

Email
revan21@gmail.com

Nama Pengguna
Revan

Password
12345678

Telepon
086373262727

Alamat
Jalan Lapangan Tegal Jati Banjaran

Perbarui

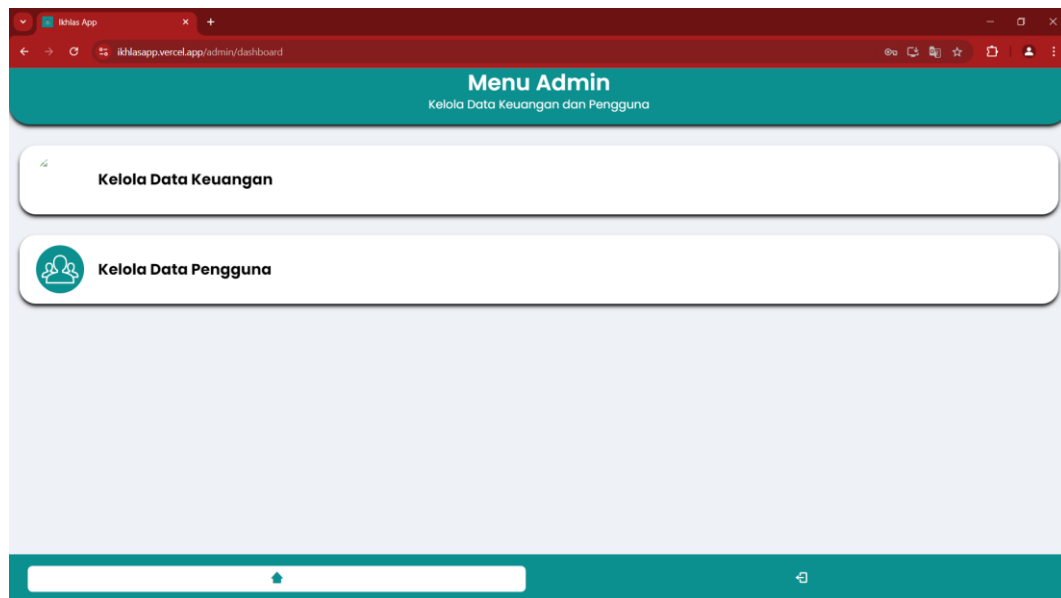
Perbarui Profil Pengguna

Gambar 4. 26 Tampilan *Mobile* Halaman Profil Pengguna dan Formulir Ubah Data Pengguna

4.1.10 Tampilan Halaman Beranda Admin

Halaman Beranda Admin merupakan halaman awal yang ditampilkan saat admin memasuki aplikasi. Halaman ini menyajikan dua tombol utama yang mengarahkan admin untuk masuk ke halaman Kelola Data Keuangan atau halaman Kelola Data Pengguna. Selain itu, di bagian bawah halaman beranda admin terdapat *navbar* yang berfungsi untuk navigasi

cepat, termasuk akses kembali ke halaman beranda dan opsi untuk *logout* dari aplikasi. Tampilan *desktop* halaman beranda admin dapat dilihat pada gambar 4.27 hingga 4.29.



Gambar 4. 27 Tampilan *Desktop* Halaman Beranda Admin

4.1.11 Tampilan Halaman Tabel Data Keuangan

Halaman Tabel Data Keuangan adalah halaman yang dirancang khusus untuk admin untuk dapat mengetahui siapa takmir yang menambahkan data catatan keuangan tersebut. Sehingga akan memudahkan admin dalam hal *tracking* terhadap riwayat transaksi yang telah dilakukan oleh masing-masing takmir, termasuk memantau pemasukan dan pengeluaran secara lebih terperinci. Dengan adanya identifikasi pengguna yang bertanggung jawab atas setiap input data, proses audit dan verifikasi bisa dilakukan lebih cepat dan akurat, menjaga transparansi keuangan masjid.

 A screenshot of the 'Tabel Admin Keuangan' page. The page title is 'Tabel Admin Keuangan'. Below the title, there is a section for 'Laporan Keuangan Masjid' with filters for 'Tanggal Mulai' and 'Tanggal Selesai', and radio buttons for 'Semua', 'Pemasukan', and 'Pengeluaran'. A table displays financial transactions with columns for 'Tanggal', 'Nama', 'Nominal', and 'User'. The table shows several entries with their respective dates, descriptions, and amounts. Below the table, there is a 'Total Saldo' of Rp. 12.810.000 and navigation buttons for 'Sebelumnya' and 'Selanjutnya'.

Tanggal	Nama	Nominal	User
3 September 2024	Pembelian Karpet	- Rp. 2.000.000	Revon
2 September 2024	Beli Sabun Cuci Tangan	- Rp. 150.000	dapa
2 September 2024	Shodaqah Pak Mahmud	+ Rp. 750.000	dapa
17 Mei 2024	Infak Jumaton	+ Rp. 300.000	Revon
10 Mei 2024	sedekah pak alim	+ Rp. 282.000	Revon

Total Saldo: Rp. 12.810.000

Gambar 4. 28 Tampilan Halaman *Desktop* Admin Tabel Laporan Keuangan Masjid

Tabel Admin Keuangan

Laporan Keuangan Masjid

Tanggal Mulai: hh/bb/tttt Tanggal Selesai: hh/bb/tttt

Semua Pemasukan Pengeluaran

Pilih kriteria sortir

Halaman 1 dari 8

Tanggal	Nama	Nominal	User
3 September 2024	Pembelian Karpet	- Rp. 2.000.000	Revan
2 September 2024	Beli Sabun Cuci Tangan	- Rp. 150.000	dapa
2 September 2024	Shadaqah Pak Mahmud	+ Rp. 750.000	dapa
17 Mei 2024	Infak Jumatatan	+ Rp. 300.000	Revan
10 Mei 2024	sedekah pak alim	+ Rp. 282.000	Revan

Total Saldo: Rp. 12.810.000

Sebelumnya Selanjutnya

Gambar 4. 29 Tampilan Halaman *Mobile* Admin Tabel Laporan Keuangan Masjid

4.1.12 Tampilan Halaman Kelola Data Pengguna

Halaman Kelola Data Pengguna adalah halaman yang dirancang khusus untuk admin dalam mengelola data pengguna, yaitu penghapusan data pengguna. Halaman ini menampilkan tabel yang berisi informasi seperti email, nama, nomor telepon, dan alamat pengguna, serta kolom aksi yang memungkinkan admin untuk menghapus data pengguna tersebut dari sistem.

Saat admin menekan tombol 'Hapus' pada kolom aksi, sistem akan menghapus data pengguna terkait dari *database* Firebase, memastikan bahwa informasi pengguna terhapus sepenuhnya dari *database*.



Gambar 4. 30 Tampilan *Desktop* Halaman Admin Tabel Daftar Pengguna



Gambar 4. 31 Tampilan *Mobile* Halaman Admin Tabel Daftar Pengguna

4.2 Hasil Pengujian Aplikasi

4.2.1 Pengujian *Black Box*

Pengujian *black box* dilakukan pada aplikasi yang telah dikembangkan untuk memastikan bahwa fungsionalitas dan tampilan aplikasi berjalan dengan baik dan sesuai dengan hasil yang diharapkan. Hasil pengujian fungsionalitas aplikasi menggunakan metode *black box* disajikan dalam Tabel 4.1, sementara hasil pengujian antarmuka aplikasi ditunjukkan pada Tabel 4.2.

Tabel 4. 1 Hasil Pengujian *Black Box* Fungsionalitas Aplikasi

No	Kegiatan	Tahapan Kegiatan	Hasil yang Diharapkan	Hasil
1	Monitoring Kotak Infak	1. Pengguna menekan tombol “Monitoring Kotak Infak” pada beranda aplikasi.	Sistem menampilkan lokasi titik posisi dimana kotak infak berada.	Berhasil
2	Notifikasi Posisi Kotak Infak	1. Kotak infak diletakan di luar area masjid. 2. Sistem membaca titik koordinat kotak infak. 3. Sistem mengirimkan <i>push notification</i> kepada pengguna.	Pengguna menerima notifikasi bahwasanya kotak infak berada di luar lokasi masjid.	Berhasil
3	Tambah Catatan Keuangan	1. Pengguna menekan tombol “Catatan Keuangan” pada beranda aplikasi. 2. Sistem menampilkan informasi saldo keuangan masjid, total pemasukan, dan total pengeluaran. 3. Pengguna menekan tombol ”Tambah Catatan” 4. Pengguna memilih tambah catatan keuangan pemasukan atau pengeluaran. 5. Pengguna mengisi form notifikasi tambah catatan keuangan pemasukan atau pengeluaran. 6. Pengguna menekan tombol ”Masukan”.	Pengguna dapat menambahkan catatan keuangan dan sistem dapat menyimpan catatan keuangan ke <i>database</i> firebase.	Berhasil
4	Lihat Laporan Keuangan	1. Pengguna menekan tombol “Catatan Keuangan” pada beranda aplikasi. 2. Sistem menampilkan informasi saldo keuangan masjid, total pemasukan, dan total pengeluaran. 3. Pengguna menekan tombol ”Lihat Catatan” 4. Pengguna memilih lihat tabel keuangan atau grafik keuangan. 5. Pengguna melakukan sortir pada tabel keuangan atau grafik keuangan sesuai dengan kebutuhan.	Sistem menampilkan data keuangan dalam bentuk tabel atau grafik, dan memungkinkan pengguna untuk melakukan sortir data.	Berhasil

No	Kegiatan	Tahapan Kegiatan	Hasil yang Diharapkan	Hasil
5	Kelola data keuangan	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Catatan Keuangan” pada beranda aplikasi. 2. Sistem menampilkan informasi saldo keuangan masjid, total pemasukan, dan total pengeluaran. 3. Pengguna menekan tombol ”Lihat Catatan”. 4. Pengguna memilih lihat tabel keuangan. 5. Pengguna mensortir tabel keuangan sesuai dengan kebutuhan. 3. Pengguna menekan tombol edit pada kolom <i>action</i>. 4. Pengguna mengisi form perubahan data keuangan. 5. Pengguna menekan tombol ”Perbarui”. 6. Pengguna menekan tombol hapus pada kolom <i>action</i>. 	<p>Sistem menyimpan perubahan data catatan keuangan ke dalam <i>database</i> firebase. Sistem menghapus catatan keuangan dari firebase.</p>	Berhasil
6	Lihat Jadwal Waktu Salat	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Jadwal Waktu Salat” pada beranda aplikasi. 	<p>Sistem mengambil jadwal waktu salat dari API dan menampilkannya kepada pengguna pada hari tersebut.</p>	Berhasil
7	Notifikasi Azan Salat	<ol style="list-style-type: none"> 1. Sistem membaca jadwal waktu salat. 2. Sistem mendeteksi waktu salat berikutnya. 3. 10 menit sebelum waktu salat, sistem mengirimkan <i>push notification</i> kepada pengguna. 	<p>Pengguna menerima notifikasi bahwasanya 10 menit lagi waktu salat akan segera dimulai, dan menyuruh pengguna untuk segera kumandangkan azan.</p>	Berhasil
8	Panduan Mengajar TPA	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Panduan Mengajar TPA” pada beranda aplikasi. 2. Pengguna memilih bab dari modul panduan mengajar TPA. 3. Pengguna memilih materi dari bab modul panduan mengajar TPA. 	<p>Sistem menampilkan materi yang dipilih pengguna. Pengguna dapat melihat modul panduan mengajar TPA.</p>	Berhasil

No	Kegiatan	Tahapan Kegiatan	Hasil yang Diharapkan	Hasil
9	Perbarui Profile Pengguna	<ol style="list-style-type: none"> 1. Pengguna menekan icon "profile" pada <i>navbar</i> aplikasi. 2. Sistem menampilkan data <i>profile</i> pengguna. 3. Pengguna menekan tombol edit profil. 4. Pengguna mengisi form perubahan data pengguna. 5. Pengguna menekan tombol "Perbarui". 	Pengguna dapat memperbarui profil dan sistem dapat menyimpan perubahan ke dalam <i>database</i> firebase.	Berhasil
10	Login pengguna	<ol style="list-style-type: none"> 1. Pengguna mengisi form email dan <i>password</i>. 2. Pengguna menekan tombol "Masuk". 	Sistem mengautentikasi kecocokan data yang diisikan pengguna dengan <i>database</i> firebase. Pengguna dapat masuk ke dalam aplikasi.	Berhasil
11	Registrasi	<ol style="list-style-type: none"> 1. Pengguna menekan tombol buat akun pada halaman <i>login</i>. 2. Pengguna mengisikan form registrasi. 3. Pengguna menekan tombol daftar. 	Pengguna dapat membuat akun dan sistem dapat menyimpan data pengguna ke dalam <i>database</i> firebase.	Berhasil
12	Logout pengguna	<ol style="list-style-type: none"> 1. Pengguna menekan icon "Logout" pada <i>navbar</i> aplikasi. 	Sistem menghapus data <i>session</i> pengguna dan mengeluarkan pengguna dari aplikasi	Berhasil
13	Kelola data pengguna	<ol style="list-style-type: none"> 1. Admin menekan tombol "Kelola Data Pengguna" pada halaman beranda admin. 2. Admin menekan tombol hapus pada kolom <i>action</i>. 	Admin dapat menghapus pengguna dan sistem dapat menyimpan perubahan dalam <i>database</i> firebase.	Berhasil
14	Login admin	<ol style="list-style-type: none"> 1. Admin mengisi form email dan <i>password</i>. 2. Admin menekan tombol "Masuk". 	Sistem mengautentikasi kecocokan data dan <i>role</i> yang diisikan admin dengan <i>database</i> firebase. Admin dapat masuk ke dalam halaman admin.	Berhasil
15	Logout admin	<ol style="list-style-type: none"> 1. Admin menekan icon "Logout" pada <i>navbar</i> aplikasi. 	Sistem menghapus data <i>session</i> admin dan mengeluarkan admin dari aplikasi	Berhasil
16	Offline	<ol style="list-style-type: none"> 1. Mematikan koneksi internet. 	Aplikasi tetap dapat dijalankan secara <i>offline</i>	Gagal

No	Kegiatan	Tahapan Kegiatan	Hasil yang Diharapkan	Hasil
		2. Melakukan uji kegiatan nomor 1-15.		
17	<i>Offline</i> Setelah <i>login</i>	1. Melakukan <i>login</i> ke aplikasi 2. Mematikan koneksi internet 3. Melakukan uji coba kegiatan nomor 1 – 15 4. Menyalakan koneksi internet kembali	Aplikasi tetap dapat dijalankan secara <i>offline</i> dan sistem mengirimkan perubahan pengguna ke penyimpanan firebase	Gagal Sebagian

Tabel 4. 2 Hasil Pengujian Black Box Antarmuka Aplikasi

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
1	<i>Navbar</i> Aplikasi	Menekan icon beranda	Pengguna masuk ke halaman beranda aplikasi	Berhasil
		Menekan icon <i>profile</i>	Pengguna masuk ke halaman <i>profile</i> pengguna	Berhasil
		Menekan icon <i>logout</i>	Pengguna keluar dari aplikasi dan sistem menghapus data <i>session</i> pengguna	Berhasil
2	Beranda	Melihat waktu salat terdekat berikutnya.	Sistem menampilkan waktu salat berikutnya.	Berhasil
		Melihat saldo keuangan kas masjid	Sistem menampilkan saldo keuangan kas masjid.	Berhasil
		Menekan tombol <i>monitoring</i> kotak infak	Pengguna masuk ke halaman <i>monitoring</i> kotak infak	Berhasil
		Menekan tombol catatan keuangan	Pengguna masuk ke halaman catatan keuangan	Berhasil
		Menekan tombol jadwal waktu salat	Pengguna masuk ke halaman jadwal waktu salat	Berhasil
		Menekan tombol panduan mengajar TPA	Pengguna masuk ke halaman panduan mengajar TPA	Berhasil
3	<i>Monitoring</i> Kotak Infak	Melihat peta dan titik kotak infak	Pengguna dapat melihat peta dan titik lokasi kotak infak	Berhasil
		Menggeser peta	Pengguna dapat menggeser peta ke berbagai arah	Berhasil
		Memperbesar peta	Pengguna dapat memperbesar ukuran peta	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Memperkecil peta	Pengguna dapat memperkecil ukuran peta	Berhasil
4	Beranda Catatan Keuangan	Melihat saldo keuangan, total pemasukan, dan total pengeluaran.	Sistem menampilkan saldo keuangan, total pemasukan, dan total pengeluaran kas masjid.	Berhasil
		Menekan tombol tambah catatan	Pengguna dapat masuk ke halaman beranda tambah catatan.	Berhasil
		Menekan tombol lihat catatan	Pengguna dapat masuk ke halaman beranda lihat catatan.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
5	Beranda Tambah Catatan	Menekan tombol Tambah Pemasukan	Pengguna dapat masuk ke halaman formulir pemasukan	Berhasil
		Menekan tombol Tambah Pengeluaran	Pengguna dapat masuk ke halaman formulir pengeluaran	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
6	Formulir Pemasukan	Menekan dan memilih <i>input</i> tanggal pemasukan	Sistem menampilkan antarmuka tanggal pemasukan	Berhasil
		Menekan dan memilih <i>input</i> dropdown kategori pemasukan	Sistem menampilkan dropdown kategori pemasukan yang terdiri dari kotak infak, zakat, shodaqoh, barang, lainnya.	Berhasil
		Menekan dan mengisi text <i>input</i> nama pemasukan	Pengguna dapat mengisi nama pemasukan pada text <i>input</i>	Berhasil
		Menekan dan mengisi text <i>input</i> nominal pemasukan	Pengguna dapat mengisi nominal pemasukan pada text <i>input</i>	Berhasil
		Menekan tombol masukan	Pengguna dapat menekan tombol masukan dan sistem dapat menerima dan menyimpan <i>inputan</i> form pemasukan. Sistem dapat menampilkan <i>alert</i> apabila <i>inputan</i> formulir berhasil atau gagal.	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
7	Formulir Pengeluaran	Menekan dan memilih <i>input</i> tanggal pengeluaran	Sistem menampilkan antarmuka tanggal pengeluaran	Berhasil
		Menekan dan memilih <i>input</i> dropdown kategori pengeluaran	Sistem menampilkan dropdown kategori pengeluaran yang terdiri dari operasional masjid, inventaris masjid, shodaqoh / santunan, dan lainnya.	Berhasil
		Menekan dan mengisi text <i>input</i> nama pengeluaran	Pengguna dapat mengisi nama pengeluaran pada text <i>input</i>	Berhasil
		Menekan dan mengisi text <i>input</i> nominal pengeluaran	Pengguna dapat mengisi nominal pengeluaran pada text <i>input</i>	Berhasil
		Menekan tombol masukan	Pengguna dapat menekan tombol masukan dan sistem dapat menerima dan menyimpan <i>inputan</i> form pengeluaran. Sistem dapat menampilkan <i>alert</i> apabila <i>inputan</i> formulir berhasil atau gagal.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
8	Beranda Laporan Keuangan	Menekan tombol Tabel Keuangan	Pengguna dapat masuk ke halaman tabel keuangan	Berhasil
		Menekan tombol Grafik Keuangan	Pengguna dapat masuk ke halaman grafik keuangan	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
9	Lihat Tabel Keuangan	Menekan dan memilih <i>input</i> tanggal mulai dan tanggal selesai	Pengguna dapat memilih rentang tanggal dan sistem dapat menampilkan tabel sesuai dengan rentang tanggal	Berhasil
		Memilih radio button kategori keuangan	Pengguna dapat memilih radio button kategori keuangan (semua, pemasukan, dan pengeluaran) dan sistem	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
			dapat menampilkan tabel sesuai dengan kategori keuangan.	
		Menekan dan memilih dropdown kriteria sortir	Pengguna dapat memilih dropdown kriteria sortir (kategori, nominal terbesar, dan nominal terkecil) dan sistem dapat menampilkan tabel sesuai dengan kriteria sortir.	Berhasil
		Melihat tabel keuangan	Sistem dapat menampilkan tabel keuangan yang terdiri dari kolom tanggal, kategori, nama, nominal, dan saldo sementara. Data keuangan akan ditampilkan pada tiap tabelnya sejumlah 5 data.	Berhasil
		Menekan tombol sebelumnya dan selanjutnya pada tabel	Sistem dapat menampilkan halaman tabel keuangan selanjutnya ataupun sebelumnya.	Berhasil
		Menekan tombol edit pada kolom aksi	Pengguna dapat menekan tombol edit dan sistem dapat mengarahkan pengguna ke halaman formulir edit data keuangan	Berhasil
		Menekan tombol hapus pada kolom aksi	Pengguna dapat menghapus baris data keuangan.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
10	Lihat Grafik Keuangan	Memilih radio button kategori keuangan	Pengguna dapat memilih radio button kategori keuangan (semua, pemasukan, dan pengeluaran) dan sistem dapat menampilkan grafik sesuai dengan kategori keuangan.	Berhasil
		Menekan dan memilih dropdown tahun	Pengguna dapat memilih dropdown tahun tampilan grafik.	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Memilih radio button kategori keuangan	Pengguna dapat memilih radio button kategori keuangan (semua. pemasukan, dan pengeluaran) dan sistem dapat menampilkan tabel sesuai dengan kategori keuangan.	Berhasil
		Melihat grafik keuangan	Sistem dapat menampilkan grafik keuangan dalam rentang 1 tahun yang terdiri dari 12 bulan.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
11	Jadwal Waktu Salat	Melihat tanggal pada hari tersebut	Sistem dapat menampilkan tanggal pada hari tersebut.	Berhasil
		Melihat jam pada jam tersebut	Sistem dapat menampilkan jam pada jam tersebut.	Berhasil
		Melihat waktu salat <i>subuh</i>	Sistem dapat menampilkan waktu salat <i>subuh</i> pada hari tersebut.	Berhasil
		Melihat waktu salat dzuhur	Sistem dapat menampilkan waktu salat dzuhur pada hari tersebut.	Berhasil
		Melihat waktu salat asar	Sistem dapat menampilkan waktu salat asar pada hari tersebut.	Berhasil
		Melihat waktu salat maghrib	Sistem dapat menampilkan waktu salat maghrib pada hari tersebut.	Berhasil
		Melihat waktu salat isya	Sistem dapat menampilkan waktu salat isya pada hari tersebut.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
12	Beranda Panduan Mengajar TPA	Menekan tombol salah satu bab	Pengguna dapat masuk ke salah satu halaman bab pada beranda panduan mengajar TPA	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
13	Beranda salah satu bab Panduan Mengajar TPA	Menekan tombol salah satu materi	Pengguna dapat masuk ke salah satu halaman materi pada beranda salah satu bab panduan mengajar TPA	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
14	Beranda salah satu materi Panduan Mengajar TPA	Melihat materi pada salah satu materi	Pengguna dapat melihat materi yang dilengkapi dengan gambar, ayat, latin, maupun terjemahannya.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
15	Profile Pengguna	Melihat nama pengguna	Sistem menampilkan nama pengguna dari <i>database</i> firebase	Berhasil
		Melihat email pengguna	Sistem menampilkan email pengguna dari <i>database</i> firebase	Berhasil
		Melihat alamat pengguna	Sistem menampilkan alamat pengguna dari <i>database</i> firebase	Berhasil
		Melihat nomor telepon pengguna	Sistem menampilkan alamat email pengguna dari <i>database</i> firebase	Berhasil
		Menekan tombol perbarui profil pengguna	Pengguna dapat menekan tombol perbarui profil pengguna dan sistem dapat mengambil uid pengguna serta mengarahkan pengguna ke dalam halaman for perbarui profil	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
16	Perbarui Data Pengguna	Mengubah email pengguna	Sistem dapat menampilkan email pengguna sebelumnya dan admin dapat mengubah email pengguna pada <i>text input</i>	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Mengubah nama pengguna	Sistem dapat menampilkan nama pengguna sebelumnya dan admin dapat mengubah nama pengguna pada <i>text input</i>	Berhasil
		Mengubah <i>password</i> pengguna	Sistem dapat menampilkan <i>password</i> pengguna sebelumnya dan admin dapat mengubah <i>password</i> pengguna pada <i>text input</i>	Berhasil
		Mengubah nomor telepon pengguna	Sistem dapat menampilkan nomor telepon pengguna sebelumnya dan admin dapat mengubah nomor telepon pengguna pada <i>text input</i>	Berhasil
		Mengubah alamat pengguna	Sistem dapat menampilkan alamat pengguna sebelumnya dan admin dapat mengubah alamat pengguna pada <i>text area</i>	Berhasil
		Menekan tombol perbarui	Admin dapat menekan tombol perbarui dan sistem dapat menyimpan perubahan data pengguna	Berhasil
		Menekan tombol kembali navigasi pada header	Admin dapat kembali ke halaman sebelumnya	Berhasil
17	<i>Login</i>	Menekan dan memasukkan email	Pengguna dapat mengisi email pada <i>text input</i>	Berhasil
		Menekan dan memasukkan <i>password</i>	Pengguna dan admin dapat mengisi <i>password</i> pada <i>text input</i>	Berhasil
		Menekan tombol masuk	Pengguna dan admin dapat menekan tombol masuk dan mengirimkan <i>inputan</i> pengguna ke <i>database</i> untuk diautentikasi.	Berhasil
		Menekan tombol buat akun	Pengguna dapat masuk ke halaman registrasi	Berhasil
18	Registrasi	Menekan dan memasukkan nama	Pengguna dapat mengisi nama pada <i>text input</i>	Berhasil

No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
		Menekan dan memasukkan nomor telepon	Pengguna dapat mengisi nomor telepon pada text <i>input</i>	Berhasil
		Menekan dan memasukkan email	Pengguna dapat mengisi email pada text <i>input</i>	Berhasil
		Menekan dan memasukkan kata sandi	Pengguna dapat mengisi kata sandi pada text <i>input</i>	Berhasil
		Menekan dan memasukkan alamat	Pengguna dapat mengisi alamat pada text area	Berhasil
		Menekan tombol daftar	Pengguna dapat menekan tombol daftar dan sistem memasukkan formulir pengguna ke dalam <i>database</i> firebase	Berhasil
		Menekan tombol masuk	Pengguna dapat masuk ke halaman <i>login</i> .	Berhasil
19	Navbar Admin	Menekan icon beranda	Admin dapat masuk ke halaman beranda admin	Berhasil
		Menekan icon <i>logout</i>	Admin dapat keluar dari aplikasi dan sistem dapat menghapus data <i>session</i> admin	Berhasil
20	Beranda Admin	Menekan tombol Kelola Data Keuangan	Admin dapat masuk ke halaman kelola data keuangan	Berhasil
		Menekan Tombol Kelola Data Pengguna	Admin dapat masuk ke halaman kelola data pengguna	Berhasil
21	Perbarui Data Keuangan	Mengubah <i>input</i> tanggal catatan keuangan	Sistem dapat menampilkan tanggal catatan keuangan sebelumnya dan pengguna dapat mengubah tanggal catatan keuangan pada text <i>input</i>	Berhasil
		Mengubah kategori catatan keuangan	Sistem dapat menampilkan kategori catatan keuangan sebelumnya dan pengguna dapat mengubah kategori catatan keuangan dropdown	Berhasil
		Mengubah nama catatan keuangan	Sistem dapat menampilkan nama catatan keuangan sebelumnya dan	Berhasil

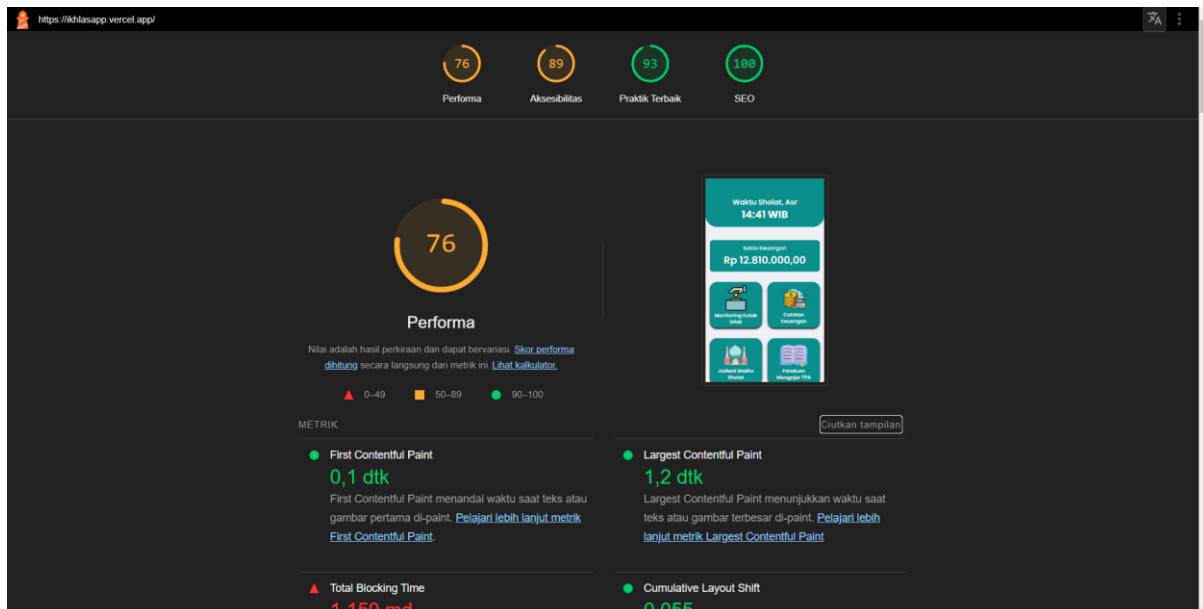
No	Halaman	Kegiatan	Hasil yang Diharapkan	Hasil
			pengguna dapat mengubah nama catatan keuangan pada text <i>input</i>	
		Mengubah nominal catatan keuangan	Sistem dapat menampilkan nominal catatan keuangan sebelumnya dan pengguna dapat mengubah nominal catatan keuangan pada text <i>input</i>	Berhasil
		Menekan tombol perbarui	Pengguna dapat menekan tombol perbarui dan sistem dapat menyimpan perubahan catatan keuangan.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil
22	Kelola Data Pengguna	Melihat tabel pengguna	Sistem dapat menampilkan tabel pengguna yang terdiri dari kolom email, nama, telepon, alamat dan aksi. Data pengguna akan ditampilkan pada tiap tabelnya sejumlah 5 data.	Berhasil
		Menekan tombol sebelumnya dan selanjutnya pada tabel	Sistem dapat menampilkan halaman tabel pengguna selanjutnya ataupun sebelumnya.	Berhasil
		Menekan tombol hapus pada kolom aksi	Pengguna dapat menghapus baris data pengguna.	Berhasil
		Menekan tombol kembali navigasi pada header	Pengguna dapat kembali ke halaman sebelumnya	Berhasil

Pada pengujian *Black Box* fungsionalitas aplikasi Masjid Al-Ikhlas Sorosutan, semua fitur aplikasi dapat berjalan dengan baik kecuali untuk nomor 16 dan 17. Aplikasi tidak dapat sepenuhnya dijalankan secara *offline* karena beberapa fungsionalitas memerlukan koneksi internet untuk terhubung dengan *database* Firebase dan API. Meskipun demikian, beberapa fitur seperti modul panduan mengajar TPA dan catatan keuangan berhasil dijalankan secara *offline*.

Pada pengujian *Black Box* antarmuka aplikasi Masjid Al-Ikhlas Sorosutan, semua antarmuka berfungsi dengan baik. Pengujian ini menunjukkan bahwa setiap elemen antarmuka, termasuk tombol, menu, dan tampilan, telah bekerja sesuai dengan desain dan fungsinya. Tidak ada masalah yang ditemukan dalam hal interaksi pengguna atau tampilan antarmuka, memastikan pengalaman pengguna yang lancar dan efisien.

4.2.2 Pengujian *Lighthouse*

Pengujian terhadap aplikasi Masjid Al-Ikhlas yang telah dikembangkan dilakukan menggunakan *Lighthouse Testing*, sebuah ekstensi yang tersedia di Google Chrome Devtools. *Lighthouse* memberikan laporan yang mencakup performa, aksesibilitas, praktik terbaik, dan SEO dengan penilaian yang menggunakan skala 0-100. Dalam skala tersebut, skor 0 mengindikasikan performa yang buruk, sedangkan skor antara 50-89 menunjukkan performa yang memadai / cukup baik, dan skor 90-100 mencerminkan performa yang sangat baik.



Gambar 4. 32 Tampilan Hasil Pengujian *Lighthouse Testing*

Dari hasil pengujian *Lighthouse* yang telah dilakukan oleh penulis, didapati hasil bahwasanya aplikasi masjid Al-Ikhlas sorosutan memiliki skor performa 76, aksesibilitas 89, best practice 93, dan *Search Engine Optimazion* (SEO) 100. Penjelasan mengenai detail hasil dari pengujian *Lighthouse* dapat dilihat pada tabel 4.3.

Tabel 4. 3 Hasil Pengujian *Lighthouse*

Metrik	Hasil	Penjelasan
Performa	76/100	Pengujian performa kecepatan pada aplikasi Masjid Al-Ikhlas menunjukkan skor 76 dari 100. Nilai ini menempatkan performa aplikasi dalam kategori “cukup baik” atau “memadai”.

Metrik		Hasil	Penjelasan
1	<i>First Contentful Paint</i>	0,1 detik	<i>First Contentful Paint</i> (FCP) adalah metrik yang mengukur waktu ketika elemen pertama pada halaman, seperti teks atau gambar, mulai tampil di layar pengguna.
2	<i>Total Blocking Time</i>	1.150 md	Total durasi antara <i>First Contentful Paint</i> (FCP) dan <i>Time to Interactive</i> (TTI), saat tugas berlangsung lebih dari 50 milidetik, diukur dalam milidetik.
3	<i>Largest Contentful Paint</i>	1,2 detik	<i>Largest Contentful Paint</i> (LCP) mengukur waktu ketika elemen teks atau gambar terbesar pada halaman mulai ditampilkan.
4	<i>Speed Index</i>	2,3 detik	<i>Speed Index</i> mengukur seberapa cepat konten visual halaman ditampilkan sepenuhnya kepada pengguna.
5	<i>Cumulative Layout Shift</i>	0.055	<i>Cumulative Layout Shift</i> (CLS) mengukur jumlah perpindahan elemen-elemen yang terlihat pada layar selama proses pemuatan halaman.
Aksesibilitas		89	Metrik ini menunjukkan seberapa efektif aplikasi dalam mendukung interaksi bagi pengguna dengan keterbatasan kemampuan interaksi. Pengujian aksesibilitas pada aplikasi Masjid Al-Ikhlas Sorosutan memiliki nilai 89/100. Nilai ini menempatkan performa aplikasi dalam kategori “cukup baik” atau “memadai”.
<i>Best Practice</i>		93	Metrik ini menilai seberapa baik aplikasi menerapkan panduan standar untuk menghindari berbagai masalah umum pada <i>website</i> .
SEO (<i>Search Engine Optimization</i>)		100	Metrik ini menilai seberapa baik aplikasi dapat dicari dan ditemukan oleh mesin pencarian.

4.2.3 Pengujian System Usability Scale (SUS)

Pengujian pada bagian ini menggunakan metode *System Usability Scale* (SUS). Proses pengujian melibatkan pemberian tautan aplikasi Masjid Al-Ikhlas Sorosutan yang telah dihosting kepada Takmir, serta membagikan kuesioner berbentuk Google Form. Kuesioner tersebut terdiri dari 10 pertanyaan dengan pilihan jawaban berskala Likert 1-5, yang berfokus pada aspek kegunaan sistem dan kenyamanan pengguna dalam mengoperasikannya. Selain itu, terdapat juga pertanyaan dalam bentuk pilihan dan *essay* terkait dengan fungsionalitas sistem dalam memecahkan permasalahan yang terjadi di Masjid Al-Ikhlas Sorosutan. Pertanyaan SUS dalam kuesioner dapat dilihat pada Tabel 2.1 di bab II. Adapun terkait pertanyaan fungsionalitas sistem terhadap permasalahan dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Pertanyaan Fungsionalitas Sistem Terhadap Permasalahan

No	Pertanyaan
1	Menurut anda, apakah sistem ini sudah dapat membantu mengatasi permasalahan keamanan kotak infak masjid? Berikan alasannya!
2	Menurut anda, apakah sistem ini sudah dapat membantu mengatasi permasalahan terhadap pencatatan keuangan masjid? Berikan alasannya!
3	Menurut anda, apakah sistem ini sudah dapat membantu mengatasi permasalahan keterlambatan azan salat di masjid? Berikan alasannya!
4	Menurut anda, apakah sistem ini sudah dapat membantu mengatasi permasalahan kurang efektifnya kegiatan Taman Pendidikan Al-Qur'an (TPA) di masjid? Berikan alasannya!
5	Berikan saran atau masukan anda terkait rancangan sistem aplikasi Masjid Al-Ikhlas Sorosutan!

Kuesioner ini diisi oleh 7 responden, dan hasil penilaian dari para responden terhadap pertanyaan-pertanyaan pengujian tercantum pada Tabel 4.5

Tabel 4. 5 Hasil Kuisisioner *System Usability Scale*

No	Nama	Jabatan	Pertanyaan									
			1	2	3	4	5	6	7	8	9	10
P1	Nuri Subandi	Ketua Takmir	5	1	4	2	5	1	4	1	4	2
P2	Erna M	Sekretaris Takmir	4	2	5	3	4	3	3	2	5	1
P3	Handoko	Bendahara Takmir	5	3	4	2	4	2	4	1	4	1
P4	Kusmanto	Anggota Takmir	5	2	3	2	5	1	4	2	3	2
P5	Winarti	Anggota Takmir	3	1	5	1	3	2	5	1	4	2
P6	Ali Marzuki	Anggota Takmir	4	2	4	3	5	2	5	1	4	1
P7	Ahmad Kohar	Anggota Takmir	5	1	3	1	4	1	4	2	5	2

Berdasarkan data pada Tabel 4.5, dilakukan perhitungan nilai SUS menggunakan persamaan (2.1), (2.2), (2.3), dan (2.4) yang telah dijelaskan di bab II. Hasil dari perhitungan tersebut kemudian digunakan untuk menyimpulkan hasil pengujian SUS. Hasil perhitungan metode SUS disajikan pada Tabel 4.6.

Tabel 4. 6 Hasil Jawaban SUS yang Telah Dilakukan Perhitungan

No	Nama	Jabatan	Pertanyaan Yang Telah Dilakukan Perhitungan Nilai SUS										Sub Total	Sub Total x 2.5
			1	2	3	4	5	6	7	8	9	10		
P1	Nuri Subandi	Ketua Takmir	4	4	3	3	4	4	3	4	3	3	35	87.5
P2	Erna M	Sekretaris Takmir	3	3	4	2	3	2	2	3	4	4	30	75
P3	Handoko	Bendahara Takmir	4	2	3	3	3	3	3	4	3	4	32	80
P4	Kusmanto	Anggota Takmir	4	3	2	3	4	4	3	3	2	3	31	77.5
P5	Winarti	Anggota Takmir	2	4	4	4	2	3	4	4	3	3	33	82.5
P6	Ali Marzuki	Anggota Takmir	3	3	3	2	4	3	4	4	3	4	33	82.5
P7	Ahmad Kohar	Anggota Takmir	4	4	2	4	3	2	3	3	4	3	34	85
Rata-rata														81.43

Berdasarkan perhitungan SUS, nilai akhir keseluruhan mencapai 81,43%. Dalam *acceptability ranges*, nilai ini termasuk kategori *acceptable*, sedangkan menurut *adjective ratings*, berada dalam kategori *good*, dengan skor akhir B.

Tabel 4. 7 Jawaban Pertanyaan Fungsionalitas Sistem Terhadap Permasalahan

No	Nama	Jabatan	Nomor Pertanyaan	Jawaban	Alasan
1	Nuri Subandi	Ketua Takmir	1	Sudah	Efisiensi dan tingkat keamanan infaq menjadi membaik bahkan membantu dalam manajemen keuangan masjid
			2	Sudah	Proses manajemen keuangan masjid lebih rapi dan sudah digital
			3	Sudah	Para takmir bisa estimasi waktu azan dan diingatkan kalau waktu salat akan tiba
			4	Sudah	Membantu pengajar TPA saat proses belajar mengajar dengan anak-anak yang sedang belajar ngaji
			5	Saran	Pengembangan biar saling koneksi dengan CCTV untuk meningkatkan keamanan kotak infak masjid

No	Nama	Jabatan	Nomor Pertanyaan	Jawaban	Alasan
2	Erna M	Sekretaris Takmir	1	Sudah	Membuat kotak infak menjadi lebih aman dari pencurian yang dilakukan oleh seseorang
			2	Sudah	Pencatatan keuangan masjid lebih efisien karena tidak menggunakan sistem pembukuan fisik
			3	Sudah	Ada pesan pengingat bagi para takmir jika waktu salat telah tiba
			4	Sudah	Membantu pengajar TPA selama mengajar karena sudah ada panduan praktis
			5	Saran	Ditambahkan fitur supaya takmir bisa mengedit dan menambahkan modul panduan mengajar TPA
	Handoko	Bendahara Takmir	1	Sudah	Keamanan kotak infak menjadi lebih baik karena adanya teknologi pemantauan anti maling
			2	Sudah	Pembukuan sudah tidak perlu menggunakan kertas tetapi cukup menggunakan aplikasi saja
			3	Sudah	Notifikasi otomatis muncul saat menjelang azan yang sangat memudahkan takmir untuk pengingat waktu salat
			4	Sudah	Bahan ajar yang sudah disiapkan lewat HP membuat kemudahan terhadap materi yang mau diajarkan
			5	Saran	Notifikasi azan pakai suara khusus yang nyaring supaya dapat mengingatkan takmir
4	Kusmanto	Anggota Takmir	1	Sudah	Ada teknologi terkini yang membantu mengecek keamanan kotak infak di masjid
			2	Sudah	Tidak perlu repot lagi dalam mencatat keuangan secara tertulis karena sudah ada di aplikasi
			3	Sudah	Pengingat yang bisa otomatis sangat membantu takmir
			4	Sudah	Bahan ajar sudah tersedia dan siap dipakai untuk mengajar TPA
			5	Saran	Materi bisa ditambahkan oleh takmir sehingga bisa lebih beragam materi TPA-nya
5	Winarti	Anggota Takmir	1	Sudah	Karena kotak amal bisa diketahui lokasinya

No	Nama	Jabatan	Nomor Pertanyaan	Jawaban	Alasan
			2	Sudah	Karena uang yg masuk bisa di data dengan lebih mudah tapi tetap aman
			3	Sudah	Kegiatan shalat di masjid bisa dijalankan tepat waktu tergantung dari takmirnya. Apk hanya mengingatkan, tapi setidaknya ada yang bantu mengingatkan.
			4	Sudah	Sistem ini bagus malah karena materi ajar sudah siap tinggal dieksekusi
			5	Saran	Materi ajar tpa nya bisa lebih di rincikan aja. Bikin per tingkatan trus ada fitur buat persiapan ujian dll. Bagian jadwal shalat bagus lagi kalau dibikin kayak alarm
6	Ali Marzuki	Anggota Takmir	1	Sudah	Sudah karena dengan adanya fitur <i>monitoring</i> kotak infaq takmir dapat langsung mengetahui lokasi kotak infaq
			2	Sudah	Sudah, sangat terbantu karena takmir dapat mengetahui laporan keuangan dengan teliti dan transparan
			3	Sudah	Sudah, dengan adanya fitur jadwal solat takmir lebih sigap untuk mempersiapkan waktu solat tiba
			4	Sudah	Dengan ini takmir menjadi lebih mudah mengakses bahan ajar di TPA
			5	Saran	Untuk bagian jadwal bisa di tingkatkan dengan adanya suara azan pada waktu shalat tiba agar dapat terdengar
7	Ahmad Kohar	Anggota Takmir	1	Sudah	Karena teknologi sekarang takmir tidak perlu takut diambil orang karna ada GPS di dalamnya
			2	Sudah	Dengan ini laporan keuangan takmir lebih transparan dan dapat diketahui oleh semua takmir
			3	Sudah	Adanya jadwal solat membuat saya menjadi lebih tau kapan jadwal solat setelahnya tiba
			4	Sudah	Bahan ajar di TPA lebih praktis dan mudah di jangkau, bisa di

No	Nama	Jabatan	Nomor Pertanyaan	Jawaban	Alasan
					ajarkan dan tentu menjadi hal yang seru bagi anak anak
			5	Saran	Untuk bahan ajar TPA sangat bagus dan alangkah baiknya ditambah lagi materinya

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengembangan Aplikasi Masjid Al-Ikhlas Sorosutan berbasis *Progressive Web App* (PWA) menggunakan metode *design thinking*, merupakan kombinasi yang efektif untuk mengatasi permasalahan di Masjid Al-Ikhlas Sorosutan. Integrasi antara keunggulan PWA yang responsif dan mudah diakses dengan pendekatan iteratif dan berpusat pada pengguna dari *design thinking* memastikan aplikasi dapat memenuhi kebutuhan operasional masjid secara optimal.

Berdasarkan hasil penelitian yang telah dilakukan, aplikasi Masjid Al-Ikhlas Sorosutan berhasil diimplementasikan dengan baik, ditunjukkan melalui pengujian Black Box, Lighthouse dan *System Usability Scale* (SUS). Pada pengujian Black Box, semua fitur dan antarmuka aplikasi berfungsi dengan baik, kecuali untuk beberapa fitur yang memerlukan koneksi internet. Meskipun begitu, aplikasi tetap mampu menjalankan salah satu fitur utama secara *offline*, seperti fitur modul panduan mengajar TPA. Pengujian Lighthouse menunjukkan bahwa aplikasi memiliki skor performa sebesar 76, aksesibilitas 89, *best practice* 93, dan SEO 100, yang menempatkan aplikasi dalam rentang skor cukup baik hingga sangat baik. Sementara itu, hasil pengujian SUS dengan skor rata-rata 81,43% menunjukkan aplikasi berada dalam kategori "*acceptable*" menurut *acceptability ranges* dan "*good*" menurut *adjective ratings*, dengan skor akhir B.

Berdasarkan hasil kuesioner *System Usability Scale* (SUS) yang diberikan kepada para takmir Masjid Al-Ikhlas Sorosutan, aplikasi ini telah terbukti membantu menyelesaikan berbagai permasalahan utama masjid. Fitur monitoring kotak infak dinilai efektif dalam meningkatkan efisiensi dan keamanan, bahkan memberikan rasa aman terhadap potensi kehilangan kotak infak. Selain itu, sistem pencatatan keuangan berbasis digital dianggap memudahkan manajemen dan membuat laporan keuangan menjadi lebih transparan dan efisien. Fitur pengingat waktu salat membantu para takmir mempersiapkan azan tepat waktu, sementara modul panduan mengajar TPA memberikan kemudahan bagi pengajar dalam menyampaikan materi kepada anak-anak.

5.2 Saran

Dengan penerapan saran-saran ini, diharapkan aplikasi yang ada di Masjid Al-Ikhlas Sorosutan dapat lebih optimal dan memberikan manfaat yang lebih besar bagi manajemen

masjid serta masyarakat sekitar. Berikut beberapa saran terkait pengembangan aplikasi yang telah dilakukan:

1. Mengingat saran dari takmir tentang pengembangan fitur yang terhubung dengan CCTV, integrasi sistem pemantauan video dengan aplikasi PWA diperlukan untuk dapat lebih meningkatkan keamanan kotak infak.
2. Sebagaimana disarankan oleh beberapa anggota takmir, penambahan fitur yang memungkinkan takmir untuk mengedit dan menambahkan modul panduan mengajar TPA akan memperkaya konten pembelajaran. Dengan demikian, materi yang disajikan dapat lebih beragam dan sesuai dengan kebutuhan pengajaran.
3. Mengingat pentingnya pengingat waktu salat, fitur notifikasi azan menggunakan suara yang lebih jelas dan nyaring, seperti yang diusulkan oleh beberapa takmir, akan membantu takmir dalam mempersiapkan ibadah salat dengan lebih baik.
4. Menambahkan fitur tambahan terkait jadwal petugas azan masjid Al-Ikhlas Sorosutan supaya takmir memiliki acuan tetap dalam pengumandangan azan. Selain itu, ditambahkan juga notifikasi tambahan yang dikhususkan kepada takmir yang bertugas mengumandangan azan pada jadwal tersebut.
5. Menambahkan satu tambahan masukan pada formulir tambah catatan keuangan untuk mengunggah gambar sebagai bukti transaksi pada fitur tambah catatan keuangan supaya setiap pemasukan dan pengeluaran dapat didokumentasikan dengan lebih lengkap dan akan meningkatkan transparansi.
6. Menambahkan fitur untuk membuat laporan dari setiap kemajuan proses anak-anak yang sedang belajar di TPA supaya pengajar dapat memantau dan mengevaluasi perkembangan belajar dengan lebih terstruktur. Fitur ini juga dapat membantu orang tua dalam mengetahui progres anak secara berkala.

DAFTAR PUSTAKA

- Almayda, R. A. (2022). *PERANCANGAN UI/UX APLIKASI AYO BERAKSI (BELAWAN BERSIH ANTI KORUPSI) DENGAN METODE DESIGN THINKING*.
- Bratsberg, H. M. (2012). *Empathy Maps of the FourSight Preferences*. <http://creativity.buffalostate.edu/>.
- Connolly, T. M., & Bagg, C. E. (2005). *Database Systems: A Practical Approach to Design, Implementation, and Management* (illustrated). Addison-Wesley.
- Dicoding Team Intern. (2021, May 19). *Contoh Use Case Diagram Lengkap dengan Penjelasannya*. Dicoding.
- El Hakim, Y. (2021, February 19). *Memasang Web App Manifest untuk PWA*. El Creativity Academy.
- Febiharsa, D., Sudana, I. M., & Hudallah, N. (2018). Uji Fungsionalitas (Blackbox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan AppPerfect Web Test dan Uji Pengguna. *Joined Journal (Journal of Informatics Education)*, 1(2), 117. <https://doi.org/10.31331/joined.v1i2.752>
- Fikri, M. A., Aji Primajaya, & Mohamad Jajuli. (2023). PENERAPAN PROGRESSIVE WEB APP PADA PEMBUATAN WEBSITE MAGANG STUDI KASUS PRODI INFORMATIKA UNSIKA. *INFOTECH Journal*, 9(2), 563–578. <https://doi.org/10.31949/infotech.v9i2.7059>
- Ham, H. (2019, December 30). *Apa itu React.js?* School of Computer Science BINUS University.
- Hendriyana, H., & Somantri, S. (2021). Implementasi Progressive Web Aps (PWA) Pada Aplikasi keuangan SMK Yapan. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 267–274. <https://doi.org/10.54367/jtiust.v6i2.1507>
- Kurniawan, S. (2022, December 5). *Mengenal PWA – Progressive Web App untuk Website Lebih Cepat*. Niagahoster Blog.
- Leon, Y. (2021, September 11). *Pencurian Kotak Amal di Masjid Sorosutan Terekam CCTV, Pelaku Tertangkap*. Harian Jogja.
- Mahatmavidya, P. A. (2021, November 15). *Memahami Apa Itu Proses Bisnis Beserta Jenis, Fungsi dan Manfaatnya*. Mekari.Com.

- Maricar, M. A., Pramana, D., & Edwar, E. (2022). Design Thinking Problems in Determining Solution for Creative Gift Orders. *SISTEMASI*, 11(1), 148. <https://doi.org/10.32520/stmsi.v11i1.1666>
- Matiini, G., Setiyadi, R., Setiawan, A., & Ramli, M. (2021). Pengembangan Aplikasi Progressive Web Application (PWA) Untuk Pembelajaran dan Evaluasi Kelas English Grammar Online Course. *Jurnal Pendidikan Edutama*, 8(2), 163. <https://doi.org/10.30734/jpe.v8i2.984>
- Mihajlija, M., & Walton, P. (2023, November 21). *Cumulative Layout Shift (CLS)*. Web.Dev.
- Nadhif, A. K., Jati, D. T. W., Hussein, Muh. F., & Widiati, I. S. (2021). Perancangan UI/UX Aplikasi Penjualan Dengan Pendekatan Design Thinking. *Jurnal Ilmiah IT CIDA*, 7(1). <https://doi.org/10.55635/jic.v7i1.146>
- Naim, R. W., Fabroyir, H., & Akbar, R. J. (2021). Desain dan Evaluasi Antarmuka Pengguna Aplikasi Web Responsif myITS Marketplace Berdasarkan Design Thinking. *Jurnal Teknik ITS*, 10(2). <https://doi.org/10.12962/j23373539.v10i2.64072>
- Nursyifa, S., Ridwan, T., & Ridha, A. A. (2023). PERANCANGAN UI/UX PROTOTYPE SISTEM PENCATATAN PELANGGAN BERBASIS MOBILE APPLICATION MENGGUNAKAN METODE DESIGN THINKING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), 1939–1945. <https://doi.org/10.36040/jati.v7i3.6925>
- Panatagama, A. (2023, March 23). *Definisi Progressive Web App (PWA) dan Manfaatnya dalam Bisnis*. Terralogiq.
- Phillip, W. (2019, September 19). *Skor aksesibilitas Lighthouse*. Chrome For Developers.
- Plattner, H., Meinel, C., & Leifer, L. (2011). *Design Thinking Research*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-01303-9>
- Prawira, A. Y., Ruslianto, I., & Prawira, D. (2023). RANCANG BANGUN SISTEM INFORMASI MANAJEMEN MASJID MENGGUNAKAN PROGRESSIVE WEB APPS (Studi Kasus: Masjid Di Kota Pontianak). *Coding : Jurnal Komputer Dan Aplikasi*, 11, 313–324.
- Pujiyanti, Y. (2020, August 13). *MENGENAL MATERI ACTIVITY DIAGRAM*. Medium.
- Putra, R. W. (2023). *Pengembangan Aplikasi Jurnal Emosi Berbasis Progressive Web App*.
- Raharjo, A. (2018, January 3). *Apa itu UML?* Medium.
- Rahmalia, N. (2022, April 20). *Google Lighthouse: Apa Itu, Cara Menggunakan, dan Metrik-metriknya*. Glints Blog.

- Rahmalia, N. (2023, February 2). *Yuk, Kenalan dengan Wireframing untuk Desain UI/UX*. Glints.
- Rahman, H. (2022, November 21). *Progressive Web Apps (PWA): An Installable Web Application*. DOT Blog.
- Regita, N. (2022, April 29). *Firebase: Pengertian, Fungsi, Fitur, dan Kelebihannya*. Niagahoster Blog.
- Rusniko, V. (2022). *How Might We – Design Sprint*. Vriske.
- Sakti, R. Y. (2023). *PERANCANGAN DESAIN UI/UX MENGGUNAKAN PENDEKATAN DESIGN THINKING DALAM PENGEMBANGAN APLIKASI MECHA*.
- Saniyyah. (2023, November 8). *Apa Itu Class diagram dalam Pemrograman? Ini Penjelasan dan Fungsinya*. DetikEdu.
- Saputra, D., & Kania, R. (2022). Implementasi Design Thinking untuk Pengguna Experience Pada Penggunaan Aplikasi Digital. *Prosiding Industrial Research Workshop and National Seminar, 13*.
- Sasks, E. (2019). *JavaScript frameworks: Angular vs React vs Vue*.
- Shelli. (2014, March 13). *Apa itu react js? Bahas Tuntas Sejarahinya Disini*. Codepolitan.
- Sheppard, D. (2017). Beginning Progressive Web App Development: Creating a Native App Experience on the Web. In *Beginning Progressive Web App Development: Creating a Native App Experience on the Web*. Springer. <https://doi.org/10.1007/978-1-4842-3090-9>
- Shirvanadi, E. C., & Idris, M. (2021). Perancangan Ulang UI/UX Situs E-Learning Amikom Center Dengan Metode Design Thinking (Studi Kasus: Amikom Center). *Automata, 2*, 1–8.
- Siagian, A. L. (2020, February 3). *Use Case Diagram dan Activity Diagram*. Medium.
- Sojitra, R. (2024, March 22). *The Ultimate Guide to Client Side vs. Server Side in Modern Web Development*. Dhwisebeta.
- Sugiyono. (2007). *Memahami penelitian kualitatif*. Alfabeta.
- Susilo, E. (2019, March 7). *Cara Menggunakan System Usability Scale (SUS) Pada Evaluasi Usability*. Edi Susilo Blog.
- Walton, P. (2023, November 30). *First Contentful Paint (FCP)*. Web.Dev.
- Walton, P., & Pollard, B. (2023, November 23). *Largest Contentful Paint (LCP)*. Web.Dev.
- Wieruch, R. (2017). *The Road to Learn React*. Leanpub.

LAMPIRAN

Foto survey dan brainstorming di masjid Al-Ikhlâs Sorosutan



Foto kotak infak Masjid Al-Ikhlas Sorosutan



Foto pengenalan aplikasi dan penyerahan dokumen pertanyaan *System Usability Scale* (SUS)

