

**PENGEMBANGAN APLIKASI BERGERAK SISTEM PARKIR  
MASJID (STUDI KASUS MASJID DAARUSSALAM  
GANJURAN SLEMAN)**



Disusun Oleh:

Nama : Reiky Aryanando Pratama  
NIM : 21523208

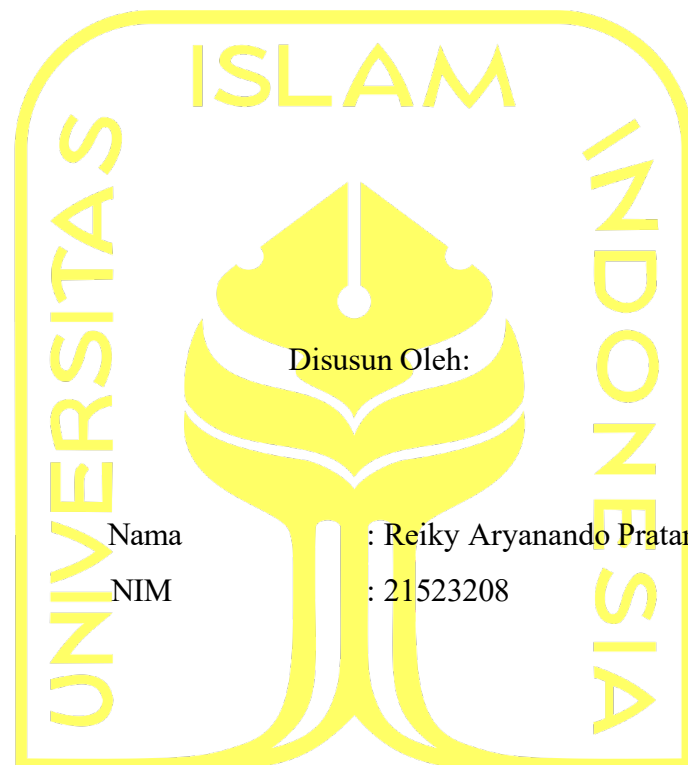
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2026**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI BERGERAK SISTEM PARKIR  
MASJID (STUDI KASUS MASJID DAARUSSALAM  
GANJURAN SLEMAN)**

**TUGAS AKHIR**



Nama : Reiky Aryanando Pratama  
NIM : 21523208

الجمعة المستد الاندو

Yogyakarta, 24 Desember 2025

Pembimbing,

( Kurniawan Dwi Irianto, S.T., M.Sc. )

## HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI BERGERAK SISTEM PARKIR  
MASJID (STUDI KASUS MASJID DAARUSSALAM  
GANJURAN SLEMAN)**

## 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, 10 Januari 2026

Tim Penguji

Kurniawan Dwi Irianto, S.T., M.Sc.

**Anggota 1**

Erika Ramadhani, S.T., M.Eng.

**Anggota 2**

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

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Reiky Aryanando Pratama

NIM : 21523208

Tugas akhir dengan judul:

**PENGEMBANGAN APLIKASI BERGERAK SISTEM PARKIR  
MASJID (STUDI KASUS MASJID DAARUSSALAM  
GANJURAN SLEMAN)**

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, 24 Desember 2025



( Reiky Aryanando Pratama )

## HALAMAN PERSEMBAHAN

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala, karena atas limpahan rahmat, hidayah, dan karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi ini dengan baik. Karya ini penulis persembahkan dengan segenap rasa bakti kepada orang tua tercinta. Terimakasih atas setiap untaian doa yang tulus, serta dukungan penuh yang menjadi sumber kekuatan bagi penulis untuk sampai di titik ini.

Penulis berharap penelitian ini tidak hanya menjadi pemenuhan syarat akademis semata, tetapi juga dapat memberikan manfaat, khususnya bagi kemaslahatan di lingkungan Masjid Daarussalam Ganjuran. Semoga apa yang tertuang dalam penelitian ini dapat menjadi kontribusi positif bagi pengembangan ilmu pengetahuan dan menjadi langkah awal bagi masa depan penulis.

**HALAMAN MOTO**

“Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan”

*(QS. Al-Insyirah: 5-6)*

“Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya, dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya)”

*(QS. An-Najm: 39-40)*

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Subhnanu wa Ta'ala, atas segala rahmat, karunia, dan hidayah-Nya, sehingga skripsi ini dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Selama proses penyusunan skripsi ini, penulis menyadari adanya berbagai keterbatasan serta kendala, baik dari segi teknis maupun nonteknis. Namun, berkat bantuan, bimbingan, dan dukungan dari berbagai pihak, skripsi ini dapat diselesaikan dengan baik. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Allah Subhanahu wa Ta'ala, atas segala limpahan rahmat dan kemudahan yang diberikan selama proses penyusunan skripsi.
2. Bapak Kurniawan Dwi Irianto S.T., M.Sc., selaku dosen pembimbing yang telah memberikan bimbingan, serta arahan dalam penyusunan skripsi ini.
3. Bapak Jumardi, selaku Ketua Takmir Masjid, serta Mas Nafis selaku Takmir Masjid, yang telah memberikan izin, dukungan, dan kerja sama selama pelaksanaan penelitian.
4. Keluarga penulis, yang senantiasa memberikan dukungan, dan motivasi sehingga penulis dapat menyelesaikan skripsi ini.
5. Teman-teman serta seluruh pihak yang telah membantu dan memberikan dukungan selama proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan dan belum sepenuhnya sempurna. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan untuk penyempurnaan di masa mendatang. Penulis berharap skripsi ini dapat memberikan manfaat serta menjadi kontribusi yang positif bagi pengembangan ilmu pengetahuan dan penelitian selanjutnya.

Yogyakarta, 24 Desember 2025



( Reiky Aryanando Pratama )

## SARI

Pengelolaan parkir masjid masih menjadi permasalahan yang sering ditemui, khususnya pada pelaksanaan kegiatan ibadah dengan jumlah jemaah yang tinggi. Keterbatasan sistem parkir yang masih bersifat konvensional menyebabkan kesulitan dalam memantau ketersediaan slot parkir yang berdampak pada ketidaknyamanan jemaah serta potensi kemacetan, sehingga mendorong perlunya pemanfaatan teknologi yang lebih efektif. Penelitian ini berfokus pada pengembangan pada pengembangan aplikasi sistem parkir masjid berbasis *mobile* yang memanfaatkan teknologi basis data *real-time* untuk menampilkan status ketersediaan area parkir, jadwal salat, serta kegiatan masjid yang dapat dipantau langsung oleh jemaah. Aplikasi dibangun menggunakan Android Studio dengan bahasa Kotlin dan mengintegrasikan Firebase sebagai media penyimpanan data, dengan studi kasus di Masjid Daarussalam Ganjuran. Pengujian sistem menggunakan *Black-box Testing* menunjukkan seluruh fitur telah berjalan sesuai skenario kebutuhan, sementara evaluasi *usability* menggunakan *System Usability Scale* (SUS) dengan 25 responden menghasilkan skor akhir 81,30 yang menempatkan aplikasi dalam kategori *excellent*, hal ini menunjukkan bahwa aplikasi memiliki tingkat kemudahan penggunaan yang baik dan dapat diterima oleh pengguna.

Kata kunci: Aplikasi *Mobile*, Sistem Parkir Masjid, *Black-box Testing*, *System Usability Testing* (SUS), Firebase.

## GLOSARIUM

Activity Diagram	Diagram yang memodelkan alur kerja suatu sistem.
AlarmManager	Layanan sistem pada Android untuk menjadwalkan operasi agar berjalan pada waktu tertentu.
Android Studio	Lingkungan pengembangan untuk merancang dan membangun aplikasi berbasis Android.
Black-box Testing	Metode pengujian yang berfokus pada penilaian fungsionalitas sistem tanpa memeriksa kode internal.
Firebase	Platform pengembangan aplikasi dari Google untuk mendukung pembuatan aplikasi dan web.
Firebase Authentication	Layanan keamanan dari Firebase untuk memverifikasi identitas dan mengelola proses login.
Firebase Realtime Database	Basis data cloud yang menyimpan dan menyinkronkan data secara langsung.
Internet of Things (IoT)	Konsep teknologi objek fisik terhubung ke jaringan internet untuk bertukar data.
Kotlin	Bahasa pemrograman dalam pengembangan aplikasi android.
Real-time	Kemampuan untuk memproses dan menyajikan data secara langsung tanpa penundaan yang signifikan.
System Usability Scale (SUS)	Instrumen kuesioner standar yang digunakan untuk mengukur tingkat kegunaan dan kepuasan pengguna terhadap suatu sistem.
Use Case Diagram	Diagram yang menggambarkan interaksi fungsional antara pengguna dan sistem.
Waterfall	Metode pengembangan perangkat lunak yang dilakukan secara berurutan dan sistematis.
WorkManager	Komponen pada android untuk menjadwalkan dan menjalankan tugas latar belakang aplikasi.

## 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 .....	vi
KATA PENGANTAR .....	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR.....	xiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian.....	4
1.7 Sistematika Penulisan .....	4
1.7.1 Pendahuluan.....	5
1.7.2 Tinjauan Pustaka.....	5
1.7.3 Metodologi Penelitian.....	5
1.7.4 Hasil dan Pembahasan .....	5
1.7.5 Kesimpulan dan Saran .....	5
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1 Parkir .....	6
2.2 Kajian Pustaka Terdahulu.....	6
2.2.1 Kajian Metode Pengembangan Aplikasi.....	7
2.2.2 Review Hasil Penelitian Terdahulu.....	8
2.3 Android Studio .....	12
2.4 Firebase.....	13
2.5 <i>Black-Box Testing</i> .....	13
2.6 <i>System Usability Scale (SUS)</i> .....	14
2.7 <i>Use Case Diagram</i> .....	14
2.8 <i>Activity Diagram</i> .....	15
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>16</b>
3.1 Analisis Kebutuhan.....	16
3.1.1 Ringkasan Hasil Wawancara .....	17
3.1.2 Analisis Kebutuhan Fungsional .....	22
3.1.3 Analisis Kebutuhan Non-Fungsional .....	23
3.2 Perancangan Sistem .....	23
3.2.1 Deskripsi Aktor/Pengguna .....	23
3.2.2 <i>Diagram Use Case</i> .....	24
3.2.3 <i>Activity Diagram</i> .....	25
3.2.4 Desain Aplikasi.....	32
3.3 Implementasi.....	38

	xi
3.4 Pengujian .....	38
3.4.1 Pengujian Fungsional ( <i>Black-box Testing</i> ) .....	38
3.4.2 Evaluasi Usability ( <i>System Usability Scale</i> ) .....	42
BAB IV HASIL DAN PEMBAHASAN .....	45
4.1 Gambaran Umum Sistem.....	45
4.1.1 Tujuan Sistem .....	45
4.1.2 Deskripsi Fitur Sistem .....	46
4.2 Implementasi Sistem.....	47
4.2.1 Integrasi Firebase .....	47
4.2.2 Mekanisme Notifikasi .....	50
4.2.3 Halaman <i>Slash Screen</i> .....	56
4.2.4 Halaman Login Takmir Masjid.....	57
4.2.5 Halaman Beranda Takmir Masjid.....	58
4.2.6 Halaman Jadwal Salat Takmir Masjid .....	59
4.2.7 Halaman Acara Masjid Takmir Masjid.....	59
4.2.8 Halaman Beranda Jemaah.....	60
4.2.9 Halaman Jadwal Salat Jemaah .....	61
4.2.10 Halaman Acara Masjid Jemaah .....	62
4.3 Pengujian Sistem .....	62
4.3.1 Pengujian Fungsional ( <i>Black-Box Testing</i> ).....	62
4.3.2 Pengujian Pengguna Menggunakan <i>System Usability Scale (SUS)</i> .....	68
4.3.3 Evaluasi dan <i>Feedback</i> Pengguna.....	69
4.4 Pembahasan .....	71
BAB V KESIMPULAN DAN SARAN.....	75
5.1 Kesimpulan .....	75
5.2 Saran .....	75
DAFTAR PUSTAKA .....	77
LAMPIRAN .....	81

**DAFTAR TABEL**

Tabel 2.1 Tabel Perbandingan Metode Penelitian Terdahulu .....	7
Tabel 2.2 Penelitian Terdahulu .....	9
Tabel 3.1 Ringkasan Hasil Wawancara Jemaah.....	18
Tabel 3.2 Ringkasan Hasil Wawancara Takmir Masjid.....	20
Tabel 3.3 Deskripsi Aktor .....	24
Tabel 3.4 Struktur Data Akun .....	36
Tabel 3.5 Struktur Data Parkir .....	37
Tabel 3.6 Struktur Data Salat .....	37
Tabel 3.7 Struktur Data Acara Masjid .....	37
Tabel 3.8 Tabel Skenario Pengujian Admin .....	39
Tabel 3.9 Tabel Skenario Pengujian Jemaah.....	41
Tabel 3.10 Daftar pertanyaan SUS.....	42
Tabel 4.1 Hasil pengujian <i>black-box</i> fitur admin .....	63
Tabel 4.2 Hasil Pengujian <i>black-box</i> fitur jemaah .....	65
Tabel 4.3 Rekapitulasi Jawaban <i>System Usability Scale</i> (SUS).....	68

## DAFTAR GAMBAR

Gambar 3.1 Fase Tahapan Metodologi Waterfall (Pressman, 2010).....	16
Gambar 3.2 Lokasi Penelitian Masjid Daarussalam Ganjuran.....	17
Gambar 3.3 <i>Use Case</i> Aplikasi Sistem Parkir Masjid.....	25
Gambar 3.4 <i>Activity Diagram</i> Akses Pengguna .....	26
Gambar 3.5 <i>Activity Diagram</i> Menu Jemaah .....	27
Gambar 3.6 <i>Activity Diagram</i> Melihat Ketersediaan Ruang Parkir .....	27
Gambar 3.7 <i>Activity Diagram</i> Melihat Jadwal Salat .....	28
Gambar 3.8 <i>Activity Diagram</i> Melihat Acara Masjid.....	29
Gambar 3.9 <i>Activity Diagram</i> Menu Takmir Masjid .....	30
Gambar 3.10 <i>Activity Diagram</i> Memperbarui Kapasitas Parkir.....	30
Gambar 3.11 <i>Activity Diagram</i> Memperbarui Jadwal Salat.....	31
Gambar 3.12 <i>Activity Diagram</i> Mengelola Acara Masjid.....	32
Gambar 3.13 Desain Halaman Ketersediaan Parkir .....	33
Gambar 3.14 Desain Halaman Jadwal Salat .....	33
Gambar 3.15 Desain Halaman Acara .....	34
Gambar 3.16 Desain Halaman Login.....	34
Gambar 3.17 Desain Tampilan Pengelolaan Parkir .....	35
Gambar 3.18 Desain Halaman Pengelolaan Jadwal Salat .....	35
Gambar 3.19 Desain Halaman Pengelolaan Acara Masjid.....	36
Gambar 3.20 Interpretasi Skor SUS (Bangor et al., 2008).....	44
Gambar 4.1 Tampilan integrasi Firebase dengan Android Studio .....	47
Gambar 4.2 Struktur Data Akun .....	48
Gambar 4.3 Struktur Data Parkir .....	48
Gambar 4.4 Struktur Data Salat .....	49
Gambar 4.5 Struktur Data Acara.....	49
Gambar 4.6 Kode program penjadwalan WorkManager pengecekan acara .....	50
Gambar 4.7 Kode program konfigurasi waktu notifikasi acara baru dan <i>reminder</i> .....	51
Gambar 4.8 Kode program pencegahan notifikasi ganda berdasarkan ID acara.....	51
Gambar 4.9 Kode program penjadwalan notifikasi dengan AlarmManager .....	51
Gambar 4.10 Kode program pemilihan notifikasi (acara baru/ <i>reminder</i> ) .....	51
Gambar 4.11 Kode program teks pesan notifikasi acara baru dan <i>reminder</i> .....	53
Gambar 4.12 Kode program aktivasi notifikasi adzan .....	54

Gambar 4.13 Kode program penyimpanan jadwal salat ke SharedPreferences .....	54
Gambar 4.14 Kode program penjadwalan AlarmManager .....	54
Gambar 4.15 Kode program tampilan notifikasi adzan .....	55
Gambar 4.16 Kode program penjadwalan ulang notifikasi .....	55
Gambar 4.17 Tampilan <i>Splash Screen</i> Aplikasi .....	56
Gambar 4.18 Tampilan Halaman Login Takmir Masjid .....	57
Gambar 4.19 Tampilan Halaman Beranda Takmir Masjid .....	58
Gambar 4.20 Tampilan Halaman Jadwal Salat Takmir Masjid .....	59
Gambar 4.21 Tampilan Halaman Acara Masjid Takmir Masjid .....	59
Gambar 4.22 Tampilan Halaman Beranda Jemaah .....	60
Gambar 4.23 Tampilan Halaman Jadwal Salat Jemaah .....	61
Gambar 4.24 Tampilan Halaman Acara Masjid Jemaah .....	62

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Pengelolaan parkir di masjid, khususnya pada hari-hari besar Islam seperti Idul Fitri dan Idul Adha, sering kali menjadi salah satu tantangan besar yang di alami oleh masjid-masjid dengan kawasan padat penduduk. Pada momen-momen tersebut, jumlah jemaah yang hadir untuk melaksanakan ibadah akan meningkat secara signifikan, sehingga menciptakan kebutuhan lahan parkir yang jauh lebih luas dibandingkan dengan hari-hari biasa. Hal ini akan berdampak pada terganggunya kenyamanan jemaah dan pengguna jalan di sekitar area masjid, khususnya ketika kendaraan diparkir di bahu jalan atau lokasi yang tidak semestinya.

Permasalahan ini menjadi semakin relevan di wilayah seperti Yogyakarta, yang memiliki tingkat populasi tinggi serta mobilitas penduduk yang terus meningkat. Berdasarkan data dari Badan Pusat (BPS) tahun 2023, tercatat ada 382.480 unit mobil dan 2.779.524 unit sepeda motor di wilayah ini (Badan Pusat Statistik, 2023). Dengan pertumbuhan jumlah kendaraan yang terus meningkat setiap tahunnya, kebutuhan akan sistem pengelolaan parkir yang lebih efektif dan efisien menjadi sangat dibutuhkan.

Meskipun demikian, upaya pengelolaan parkir di beberapa masjid masih banyak bergantung pada keterbatasan sistem tradisional yang cenderung tidak efisien. Guluh dan Sulki (2023) mencatat bahwa penggunaan sistem tradisional pada pengelolaan parkir sering kali menyulitkan petugas dalam memantau ketersediaan slot parkir secara akurat. Kondisi ini menyebabkan kendaraan harus berputar-putar dahulu untuk mencari slot parkir yang masih tersedia, sehingga tidak hanya menghabiskan waktu tetapi juga akan menimbulkan kemacetan di area parkir. Oleh karena itu, diperlukan solusi yang mampu menyediakan informasi ketersediaan slot parkir secara akurat untuk mengatasi permasalahan tersebut.

Seiring dengan perkembangan teknologi, *Internet of Thing* (IoT) menjadi solusi yang semakin relevan untuk mengatasi berbagai permasalahan, termasuk dalam pengelolaan parkir. Menurut Alfikri dan Irianto (2024), penerapan IoT pada sistem parkir masjid memungkinkan pemantauan ketersediaan slot secara *real-time* dan menyajikan informasi tersebut kepada jemaah melalui layar tampilan di area parkir. Dengan adanya teknologi ini, jemaah dapat mengetahui slot parkir yang tersedia sebelum tiba di lokasi, sehingga mengurangi waktu pencarian dan potensi kemacetan.

Berbagai penelitian sebelumnya juga telah menunjukkan keberhasilan penerapan IoT dalam sistem parkir, namun sebagian besar fokusnya masih pada area parkir komersial seperti pusat perbelanjaan atau fasilitas umum lainnya. Sebagai contoh, penelitian yang dilakukan oleh Rahmatillah (2022) telah mengembangkan sistem monitoring berbasis IoT yang mampu menyediakan informasi ketersediaan slot parkir secara *real-time* melalui sensor *ultrasonic*. Penelitian lain yang relevan juga telah dilakukan Marwita (2023), yang telah mengimplementasikan IoT untuk mendukung sistem parkir dengan fitur reservasi yang efektif dalam mengatur lalu lintas kendaraan di area parkir.

Sementara itu, penelitian oleh Alfikri dan Irianto (2024) juga telah melakukan pengembangan pada perangkat keras untuk pengelolaan sistem parkir masjid, berfungsi untuk menampilkan hasil pemantauan ketersediaan slot parkir. Namun, sistem yang dikembangkan masih terbatas pada pemrosesan dan penyajian data di tingkat perangkat keras, tanpa menyediakan antarmuka yang dapat diakses langsung oleh pengguna. Dalam saran penelitiannya, Alfikri dan Irianto (2024) merekomendasikan untuk melakukan pengembangan aplikasi berbasis *web* atau *mobile* sebagai pengganti platform Blynk yang digunakan sebelumnya, sehingga informasi ketersediaan slot parkir dapat dipantau secara *real-time* dari jarak jauh. Keterbatasan ini membuka peluang untuk melanjutkan penelitian mereka dengan mengembangkan aplikasi yang lebih terintegrasi dan sesuai dengan kebutuhan pengelolaan parkir di masjid secara lebih efisien dan praktis.

Penggunaan aplikasi berbasis IoT dalam sistem parkir masjid diharapkan dapat memberikan manfaat yang signifikan, baik bagi jemaah maupun takmir masjid. Menurut Tannujaya et al., (2021), integrasi IoT dengan aplikasi *mobile* tidak hanya meningkatkan efisiensi operasional tetapi juga memberikan pengalaman pengguna untuk memantau kapasitas parkir dengan lebih cepat dan akurat, sehingga memastikan pengelolaan lahan parkir yang optimal selama waktu-waktu ibadah.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan aplikasi *mobile* sebagai media pendukung pengelolaan parkir masjid secara lebih efisien, melalui penyediaan informasi ketersediaan slot parkir yang disajikan secara sistematis melalui aplikasi. Dengan aplikasi ini, diharapkan permasalahan parkir di masjid, terutama pada hari-hari besar Islam, dapat diantisipasi dengan lebih baik. Selain itu, aplikasi ini juga dilengkapi dengan fitur pendukung yang relevan bagi operasional masjid, sehingga dapat membantu jemaah dalam memperoleh akses informasi kegiatan masjid dengan lebih mudah.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah yang diangkat pada penelitian ini adalah bagaimana mengembangkan aplikasi *mobile* yang mampu menyediakan informasi ketersediaan parkir masjid, serta mendukung efisiensi pengelolaan informasi di masjid.

## 1.3 Batasan Masalah

1. Penelitian ini hanya mencakup pengembangan aplikasi *mobile* yang menyediakan informasi ketersediaan parkir masjid, yang dilengkapi dengan fitur pendukung yang relevan untuk membantu operasional masjid.
2. Penelitian tidak mencakup pengembangan perangkat keras berbasis IoT, melainkan hanya berfokus pada pengembangan aplikasi yang terhubung dengan perangkat keras IoT.

## 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan aplikasi yang berfungsi sebagai media penyajian informasi ketersediaan parkir masjid, serta informasi pendukung untuk memudahkan jemaah dalam mengakses informasi melalui perangkat *mobile*.

## 1.5 Manfaat Penelitian

1. Bagi Penulis
  - a. Menambah pengetahuan dan keterampilan dalam pengembangan aplikasi *mobile*.
  - b. Memperluas pemahaman tentang proses perancangan sistem yang sesuai dengan kebutuhan pengguna berdasarkan hasil analisis dan wawancara.
2. Bagi Jemaah
  - a. Memberikan kemudahan bagi jemaah dalam memperoleh informasi ketersediaan parkir masjid melalui aplikasi.
  - b. Membantu jemaah dalam memperoleh informasi kegiatan masjid.
  - c. Membantu mengurangi waktu dalam mencari tempat parkir masjid yang masih tersedia, terutama saat acara besar keagamaan.
3. Bagi Takmir Masjid
  - a. Membantu takmir masjid dalam mengelola informasi parkir melalui aplikasi,
  - b. Mendukung penyampaian informasi kegiatan masjid kepada jemaah.

- c. Menjadi sarana pendukung dalam menciptakan pengelolaan parkir masjid yang lebih tertib dan teratur.

## 1.6 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan. Berikut adalah uraian singkat dari masing-masing tahapan yang dilakukan:

- a. Analisis Kebutuhan

Tahapan ini bertujuan untuk mengidentifikasi kebutuhan sistem yang akan dikembangkan. Informasi dikumpulkan melalui wawancara dengan jemaah dan takmir masjid untuk menentukan kebutuhan utama aplikasi.

- b. Perancangan Sistem

Pada tahapan ini dilakukan perancangan skema aplikasi untuk menggambarkan bagaimana sistem akan bekerja dan berinteraksi dengan pengguna. Tahapan ini mencakup deskripsi aktor, diagram *use case*, *activity diagram*, serta desain antarmuka dan basis data aplikasi.

- c. Implementasi

Pada tahapan ini, aplikasi mulai diimplementasikan sesuai dengan desain yang telah dirancang sebelumnya. Proses implementasi dilakukan menggunakan platform Android Studio dengan bahasa pemrograman Kotlin, serta Firebase sebagai basis data untuk mendukung penyimpanan dan pengelolaan data.

- d. Pengujian

Tahapan pengujian dilakukan melalui dua pendekatan, yaitu *black-box testing* untuk memverifikasi fungsionalitas aplikasi serta *System Usability Scale (SUS)* untuk menilai kemudahan penggunaan aplikasi dari sudut pandang pengguna.

## 1.7 Sistematika Penulisan

Sistematika penulisan bertujuan untuk memberikan gambaran dari struktur penelitian agar lebih terarah dan sistematis. Adapun sistematika penulisan dalam penelitian ini adalah sebagai berikut:

### **1.7.1 Pendahuluan**

Pada bab ini akan dibahas pendahuluan yang terdiri atas latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

### **1.7.2 Tinjauan Pustaka**

Bab ini akan membahas berbagai teori dan referensi yang berkaitan dengan topik penelitian ini, meliputi definisi parkir, tools dan metode yang digunakan dalam pengembangan aplikasi, serta kajian pustaka terdahulu yang relevan dengan penelitian.

### **1.7.3 Metodologi Penelitian**

Pada bab ini akan membahas metode penelitian yang dilakukan dalam pengembangan aplikasi, mulai dari tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian.

### **1.7.4 Hasil dan Pembahasan**

Pada bab ini menyajikan pembahasan mengenai hasil implementasi aplikasi, tampilan antarmuka, serta hasil pengujian fungsional dan evaluasi *usability* menggunakan metode *System Usability Scale* (SUS).

### **1.7.5 Kesimpulan dan Saran**

Pada bab ini berisi kesimpulan dari hasil penelitian serta saran untuk pengembangan aplikasi di masa mendatang.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Parkir**

Parkir adalah kondisi ketika sebuah kendaraan dihentikan sementara dan ditinggalkan oleh pengemudinya (Kusumaningtyas (2016); Oktaviani et al. (2020)). Menurut pasal 1 angka 15 Undang-Undang Nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, parkir diartikan sebagai aktivitas berhenti di suatu tempat tertentu, baik yang ditandai dengan rambu maupun tidak, dan tidak bertujuan untuk menaikkan atau menurunkan barang atau penumpang (UU No. 22 Tahun 2009).

Parkir adalah salah satu komponen penting dalam sistem transportasi, karena dapat membantu menjaga kelancaran arus lalu lintas. Manajemen parkir yang efektif dapat meningkatkan efisiensi jaringan transportasi. Di wilayah perkotaan dengan tingkat kepadatan penduduk yang tinggi dan pertumbuhan ekonomi yang pesat, jumlah kendaraan pribadi cenderung meningkat. Kondisi tersebut menciptakan kebutuhan akan lahan parkir, baik yang resmi maupun tidak. Secara umum, lahan parkir dibagi menjadi dua jenis, yaitu parkir di badan jalan, yang biasanya terletak di sepanjang pinggir jalan, dan parkir di luar badan jalan, yang mencakup fasilitas parkir di area khusus seperti pusat perbelanjaan atau lokasi khusus lainnya (Gerungan et al., 2022).

Dalam konteks masjid, fasilitas parkir biasanya berupa kombinasi antara area parkir halaman sebagai bentuk *off-street* dan pemanfaatan tepi jalan sebagai bentuk *on-street*, dengan pengaturan yang disesuaikan dengan kapasitas lahan dan kondisi tata ruang di sekitar lingkungan masjid (Setlight et al., 2022). Penggunaan fasilitas parkir pada masjid juga memiliki karakteristik tersendiri karena pola kedatangan jemaah mengikuti waktu ibadah yang berlangsung secara periodik, sehingga kebutuhan ruang parkir cenderung mengikuti jadwal dan intensitas aktivitas keagamaan (Alfikri & Irianto, 2024).

#### **2.2 Kajian Pustaka Terdahulu**

Kajian pustaka terdahulu bertujuan untuk menganalisis dan membandingkan berbagai penelitian sebelumnya yang relevan dengan penelitian ini, bertujuan untuk menambah wawasan, memperkaya referensi, serta menghindari kesalahan yang sama seperti pada penelitian terdahulu.

### 2.2.1 Kajian Metode Pengembangan Aplikasi

Bagian ini bertujuan untuk menganalisis dan membandingkan berbagai metode pengembangan perangkat lunak yang telah digunakan dalam penelitian sebelumnya terkait sistem parkir. Perbandingan ini dilakukan berdasarkan pada kelebihan dan kekurangan masing-masing metode, serta relevansinya dengan kebutuhan aplikasi sistem parkir. Rincian lengkap dari perbandingan setiap metode dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Tabel Perbandingan Metode Penelitian Terdahulu

No	Metode	Kelebihan	Kekurangan	Referensi
1	Waterfall	<ul style="list-style-type: none"> <li>Alur perkembangan terstruktur, sistematis, dan mudah dimengerti.</li> <li>Sangat efektif untuk proyek dengan kebutuhan yang sudah jelas dan baku.</li> </ul>	<ul style="list-style-type: none"> <li>Tidak fleksibel terhadap perubahan selama pengembangan.</li> <li>Waktu pengerjaan relatif lama karena tahapan tidak bisa dilakukan secara bersamaan.</li> </ul>	Rahardjo et al. (2023); Purniawan & Nanang (2024)
2	RAD	<ul style="list-style-type: none"> <li>Waktu pengembangan sangat singkat karena fokus pada penyelesaian fitur utama.</li> <li>Melibatkan pengguna secara aktif dalam setiap iterasi desain</li> </ul>	<ul style="list-style-type: none"> <li>Kurang optimal untuk sistem kompleks yang membutuhkan analisis mendalam.</li> <li>Ketergantungan tinggi pada masukan pengguna selama pengembangan.</li> </ul>	Yulianggara (2022); (Ramadhan & Angelia, 2023)
3	Prototype	<ul style="list-style-type: none"> <li>Adanya rancangan sistem di awal, memudahkan pengguna memahami alur aplikasi.</li> </ul>	<ul style="list-style-type: none"> <li>Berisiko terjebak dalam siklus revisi tanpa henti jika target tidak dibatasi.</li> <li>Kualitas arsitektur sistem sering</li> </ul>	Aura Arasy et al. (2024); Devani et al. (2024)

No	Metode	Kelebihan	Kekurangan	Referensi
		<ul style="list-style-type: none"> <li>Sangat fleksibel dalam mengakomodasi revisi antarmuka dari pengguna.</li> </ul>	terabaikan demi mengejar tampilan visual semata.	

Berdasarkan Tabel 2.1, metode Waterfall dipilih sebagai pendekatan pengembangan perangkat lunak dalam penelitian ini karena memiliki alur kerja yang sistematis dan terdokumentasi dengan baik, sehingga memudahkan pengendalian setiap tahapan pengembangan (Rahardjo et al., 2023). Metode ini dinilai sesuai untuk sistem dengan kebutuhan yang telah terdefinisi secara jelas dan relatif stabil sejak tahap awal pengembangan (Purniawan & Nanang, 2024).

Dibandingkan dengan metode lain seperti Rapid Application Development (RAD) dan Prototype, metode Waterfall lebih tepat digunakan pada proyek yang tidak memerlukan perubahan kebutuhan secara signifikan selama proses pengembangan (Ilham & Naziro, 2019). Metode Prototype cenderung lebih sesuai untuk proyek dengan ketidakpastian kebutuhan pada tahap awal karena menekankan iterasi berbasis umpan balik pengguna (Aura Arasy et al., 2024), sedangkan metode RAD lebih efektif untuk proyek dengan batas waktu pengembangan yang singkat dan keterlibatan pengguna yang intensif (Yulianggara, 2022).

Sejalan dengan penelitian terdahulu, penerapan metode Waterfall terbukti mampu menghasilkan sistem parkir yang teruji serta terdokumentasi dengan baik, baik pada platform *web* maupun aplikasi *mobile* (Rahardjo et al., 2022; Purniawan & Nanang, 2024). Oleh karena itu, metode Waterfall dipandang sebagai pendekatan yang sesuai untuk pengembangan aplikasi informasi ketersediaan parkir berbasis *mobile* dalam penelitian ini, mengingat kebutuhan sistem telah terdefinisi dengan jelas dan tidak memerlukan perubahan yang signifikan selama pengembangan.

## 2.2.2 Review Hasil Penelitian Terdahulu

Subbab ini menyajikan tinjauan terhadap hasil penelitian terdahulu yang relevan dengan pengembangan sistem parkir. Penelitian-penelitian tersebut dianalisis untuk memperoleh gambaran mengenai pendekatan, metode, serta teknologi yang telah digunakan, sehingga dapat menjadi dasar dalam penentuan arah dan fokus penelitian yang dilakukan.

Tabel 2.2 Penelitian Terdahulu

Judul penelitian	Tujuan penelitian	Metode penelitian	Platform hasil penelitian	Hasil penelitian
Perancangan UI/UX Aplikasi GoMasjid Menggunakan Pendekatan <i>Human Centereed Design</i> (Adiswara & Irianto, 2024)	Merancang antarmuka aplikasi monitoring jemaah dan kapasitas parkir di Masjid Baiturrahman untuk mengatasi kecemasan jemaah terkait ketersediaan tempat.	<i>Human Centered Design</i> (HCD) dan <i>Usability Testing</i> (Maze <i>Design</i> )	Desain Prototipe	Penelitian ini menghasilkan desain <i>high-fidelity</i> yang teruji. Berdasarkan <i>usability testing</i> terhadap 10 responden, didapatkan tingkat keberhasilan sebesar 96% dan rata-rata waktu pengerjaan yang normal.
Implementasi <i>Internet of Things</i> pada Sistem Parkir Masjid (Alfikri & Irianto, 2024)	Mengembangkan sistem parkir berbasis IoT untuk memantau ketersediaan slot secara <i>real-time</i> guna mengurangi kemacetan saat salat Jumat.	<i>Black-box Testing</i> dan Prototipe	<i>Hardware</i> (ESP32) dan Aplikasi Blynk	Sensor ultrasonik HC-SR04 terbukti memiliki akurasi 100% dalam mendeteksi keberadaan dan jenis kendaraan (mobil/motor). Integrasi dengan Blynk berhasil menampilkan data secara <i>real-time</i> dengan

Judul penelitian	Tujuan penelitian	Metode penelitian	Platform hasil penelitian	Hasil penelitian
				<i>delay</i> yang minim.
<i>Mosque Parking System: An Implementation Using Camera and IoT</i> (Gahara & Irianto, 2025)	Menciptakan sistem perhitungan kapasitas parkir otomatis menggunakan kamera untuk mengatasi ketidakteraturan parkir di Masjid Daarussalam Ganjuran.	<i>Human Centered Design</i> (HCD) dan <i>Black-box Testing</i>	<i>Website</i> dan IoT (EsSP32-CAM)	Sistem mampu mendeteksi kendaraan dan menampilkan sisa slot parkir di <i>website</i> serta LCD. Namun, akurasi deteksi kamera tercatat sebesar 66,67% dan mengalami penurunan performa yang signifikan pada kondisi pencahayaan yang rendah, serta kesulitan melacak kendaraan berkecepatan tinggi.
Perancangan Sistem Informasi Parkir Berbasis <i>Web</i> di Universitas Islam Balitar	Membangun sistem informasi berbasis <i>web</i> untuk mengatasi minimnya informasi ketersediaan lahan	Prototipe	<i>Website</i>	Penelitian menghasilkan <i>e-parkir</i> yang mampu mengelola basis data kendaraan dan menyajikan

Judul penelitian	Tujuan penelitian	Metode penelitian	Platform hasil penelitian	Hasil penelitian
(Devani et al., 2024)	parkir dan mempermudah pengawasan kendaraan yang masuk dan keluar.			informasi lokasi parkir secara <i>digital</i> . Implementasi sistem ini terbukti meningkatkan efisiensi manajemen parkir dan memudahkan pengguna dalam mengetahui status area parkir.
Prototipe Sistem <i>Smart Car Parking</i> Berbasis IoT dengan Monitoring Melalui <i>Web</i> dan Android (Romdoni & Ramadhan, 2022)	Mengembangkan sistem parkir cerdas yang memungkinkan pengguna memantau ketersediaan slot parkir secara langsung lewat perangkat seluler.	Waterfall	Aplikasi Android dan <i>Web</i> (Firebase)	Sistem yang dibangun mampu mendeteksi kendaraan dan mengirimkan data dengan pembaruan status setiap 2-3 detik. Aplikasi android berjalan stabil dalam menampilkan informasi slot kosong, sehingga efektif membantu pengendara menghindari

Judul penelitian	Tujuan penelitian	Metode penelitian	Platform hasil penelitian	Hasil penelitian
				pencarian parkir secara manual.

Berdasarkan Tabel 2.2, penelitian terkait sistem parkir masjid telah dikembangkan dengan beragam pendekatan teknologi, mulai dari perancangan antarmuka hingga pemanfaatan IoT serta integrasi dengan platform *web* dan aplikasi. Meskipun mampu menyajikan informasi ketersediaan parkir secara *real-time*, sebagian penelitian masih belum sepenuhnya terintegrasi dalam aplikasi *mobile* yang mudah diakses oleh jemaah.

Sehubungan dengan hal tersebut, penelitian ini difokuskan pada pengembangan aplikasi sistem parkir masjid berbasis *mobile* yang terintegrasi dengan layanan Firebase secara *real-time*. Aplikasi dirancang dengan fitur utama informasi parkir serta dilengkapi fitur pendukung yang relevan bagi kebutuhan masjid, sehingga dapat memberikan kemudahan akses informasi bagi jemaah dan membantu pengelolaan oleh takmir masjid.

### 2.3 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi yang direkomendasikan oleh Google untuk pengembangan aplikasi android (Android Studio, 2024). Berbasis pada kode editor dan alat pengembangan dari IntelliJ IDEA, Android Studio menawarkan berbagai fitur yang dapat meningkatkan efisiensi dalam pengembangan aplikasi Android, seperti berikut:

- Sistem *build* berbasis *Gradle* yang fleksibel
- *Emulator* yang cepat dan kayak fitur
- Lingkungan terintegrasi untuk pengembangan di semua perangkat Android
- Fitur edit langsung untuk pembaruan *real-time* pada emulator dan perangkat fisik

Setiap proyek di Android Studio terdiri dari beberapa modul, termasuk modul aplikasi Android, modul perpustakaan, dan modul *Google App Engine*. Masing-masing modul ini memiliki folder berikut:

- Folder Manifest yang berisi file dengan ekstensi xml untuk mendefinisikan struktur dan metadata aplikasi.
- Folder Java yang berisi file kode sumber Kotlin dan Java. Termasuk kode pengujian Junit.

- Folder res yang berisi sumber daya non-kode seperti *string* UI dan gambar bitmap.

Dalam penelitian ini, pengembangan aplikasi akan menggunakan bahasa pemrograman Kotlin. Penulisan kode dengan Kotlin lebih ringkas dan mudah dibaca dibandingkan dengan menggunakan bahasa Java (Bose et al., 2018). Selain itu, Kotlin juga mudah untuk dipelajari dan memiliki dukungan penuh dari Android Studio sebagai IDE, yang mempermudah dalam pemrograman (Sibarani et al., 2018).

## 2.4 Firebase

Firebase adalah platform yang dikembangkan oleh Google untuk membantu pengembang dalam membangun aplikasi *mobile* dan web. Salah satu fitur utamanya adalah *Realtime Database*, yang memungkinkan penyimpanan dan sinkronisasi data secara *real-time*, sehingga sangat berguna untuk aplikasi yang memerlukan pembaruan data secara instan, seperti pada aplikasi sistem parkir yang memberikan informasi tentang ketersediaan slot parkir kepada pengguna secara *real-time*.

Firebase tidak hanya menawarkan fitur *Realtime Database*, tetapi juga fitur lainnya, seperti *Cloud Firestore*, *Authentication*, dan *Cloud Storage*. *Cloud Firestore* dirancang untuk mengelola data dengan struktur yang lebih kompleks, sehingga sangat sesuai untuk aplikasi dalam skala besar. Untuk mendukung aspek keamanan, Firebase menyediakan layanan *Authentication*, yang dapat mempermudah pengelolaan login melalui berbagai metode otentikasi. Sementara itu, fitur *Cloud Storage* memungkinkan aplikasi untuk dapat menyimpan file media secara efisien dalam infrastruktur berbasis *cloud* (Chougale et al., 2021).

## 2.5 Black-Box Testing

*Black Box Testing* adalah metode yang digunakan untuk menguji perangkat lunak dengan menilai cara kerja sistem dari sudut pandang pengguna, tanpa melihat ke dalam detail struktur kode internalnya. Metode ini bertujuan untuk memastikan bahwa fungsi perangkat lunak berjalan sesuai dengan spesifikasi yang telah ditetapkan. Metode ini dapat digunakan untuk mendeteksi kesalahan fungsi yang tidak berjalan, hasil yang tidak sesuai, atau masalah pada antarmuka tanpa perlu memahami kode sumbernya (Nugraha et al., 2020).

Keunggulan utama *Black Box Testing* adalah kemudahannya untuk diimplementasikan, karena pengujian tidak perlu memahami kode sumber perangkat lunak yang diuji. Namun, metode ini juga memiliki kekurangan, yaitu tidak mampu mengevaluasi keseluruhan aspek

pada perangkat lunak secara mendalam, sehingga beberapa kesalahan tersembunyi dalam kode mungkin tidak terdeteksi (Praniffa et al., 2023).

## 2.6 *System Usability Scale (SUS)*

*System Usability Scale (SUS)* merupakan instrumen yang digunakan untuk mengukur persepsi pengguna terhadap tingkat kemudahan penggunaan (*usability*) suatu sistem. Instrumen ini pertama kali dikembangkan oleh John Brooke pada tahun 1986 dan selanjutnya dianalisis secara lebih komprehensif oleh Bangor, Kortum, dan Miller (2008), yang memperkuat validitas, reliabilitas, serta konsistensi hasil penilaiannya. Literatur internasional menyebutkan bahwa SUS termasuk alat ukur yang sederhana namun efektif, sehingga banyak digunakan dalam evaluasi antarmuka perangkat lunak dan aplikasi digital (Hyzy et al., 2022). Dalam berbagai studi sistem informasi, SUS diakui sebagai salah satu instrumen standar yang dapat memberikan gambaran umum mengenai kualitas pengalaman pengguna (Andriyani & Sari, 2023).

Instrumen SUS terdiri atas sepuluh pernyataan yang dijawab menggunakan skala *Likert* lima tingkat. Skala ini digunakan untuk mengubah persepsi subjektif pengguna menjadi data kuantitatif yang dapat dianalisis untuk menilai dimensi efektivitas, efisiensi, serta tingkat kenyamanan penggunaan sistem. Penggunaan skala *Likert* dalam SUS juga banyak dibahas dalam literatur evaluasi *usability*, karena memberikan dasar yang terstruktur untuk mengukur respon pengguna terhadap antarmuka sistem (Suria, 2024). Dalam kajian sistem informasi, penggunaan sepuluh pernyataan yang konsisten dan format penilaian lima tingkat menjadikan SUS sebagai instrumen yang mudah diterapkan pada berbagai jenis aplikasi (Putera et al., 2022).

## 2.7 *Use Case Diagram*

*Use Case Diagram* merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan fungsi-fungsi yang harus disediakan oleh sebuah sistem dari sudut pandang pengguna. Fokus utama diagram ini adalah menjelaskan apa yang dilakukan sistem, bukan bagaimana cara sistem tersebut bekerja. Sebagaimana dijelaskan oleh (Dharwiyanti & Wahono, 2003), sebuah *use case* merepresentasikan bentuk interaksi antara seorang aktor dan sistem dalam rangka menyelesaikan suatu tujuan tertentu, seperti proses masuk ke sistem, pembuatan data, atau pelaksanaan tugas lainnya. Aktor dalam konteks

ini dapat berupa manusia maupun perangkat eksternal yang berhubungan langsung dengan sistem.

*Use Case Diagram* memiliki peran penting dalam tahap analisis kebutuhan karena membantu mengidentifikasi fungsi utama sistem, mengomunikasikan rancangan dengan pengguna atau klien, serta menjadi acuan dalam penyusunan skenario pengujian. Selain itu, hubungan antar *use case* dapat berupa *include*, yaitu ketika suatu *use case* memanggil fungsi lain yang selalu dijalankan dalam alurnya; *extend*, ketika sebuah *use case* memperluas perilaku *use case* lain; serta *generalization*, yang menunjukkan adanya spesialisasi dari suatu *use case* ke *use case* lain. Melalui struktur ini, diagram dapat membantu mengurangi duplikasi fungsi dan memperjelas batasan fungsional sebuah sistem (Dharwiyanti & Wahono, 2003).

## **2.8 Activity Diagram**

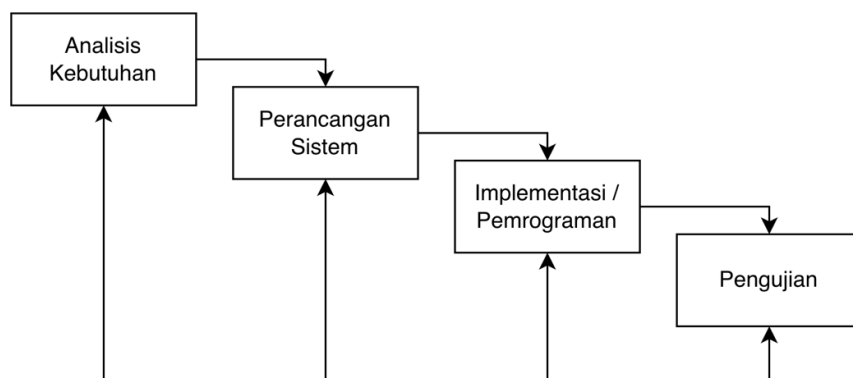
*Activity Diagram* merupakan jenis diagram UML yang digunakan untuk memodelkan aliran aktivitas dalam suatu sistem secara lebih terstruktur. Diagram ini menggambarkan bagaimana suatu proses dimulai, kemungkinan keputusan yang terjadi, aktivitas yang berjalan secara sekuensial maupun *paralel*, hingga proses tersebut berakhir. Menurut Dharwiyanti dan Wahono (2003), *Activity Diagram* berfungsi menampilkan jalur aktivitas dari tingkat yang lebih tinggi sehingga mempermudah pemahaman alur kerja sistem secara menyeluruh.

*Activity Diagram* termasuk dalam kategori *specialized state diagram*, di mana setiap aktivitas direpresentasikan sebagai aksi, dan transisi antarkegiatan umumnya terjadi karena selesainya proses sebelumnya. Diagram ini tidak menggambarkan perilaku detail internal sistem, namun memberikan representasi umum mengenai bagaimana pekerjaan berlangsung dalam sistem. Elemen-elemen seperti *decision*, *fork*, *join*, serta simbol aktivitas berbentuk persegi panjang bersudut membulat digunakan untuk menunjukkan kondisi percabangan, proses *paralel*, dan urutan tindakan. Selain itu, penggunaan *swimlane* memungkinkan pembagian tanggung jawab antar objek atau aktor yang terlibat dalam proses, sehingga alur kerja sistem dapat dipahami dengan lebih jelas (Dharwiyanti & Wahono, 2003).

## BAB III METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk mengembangkan aplikasi sistem parkir berbasis *mobile* di Masjid Daarussalam Ganjuran. Aplikasi ini dirancang untuk menyediakan informasi ketersediaan ruang parkir sehingga jemaah dapat mengetahui kondisi parkir sebelum tiba di lokasi. Proses pengembangan aplikasi ini menggunakan metodologi Waterfall, yaitu pendekatan pengembangan perangkat lunak yang berjalan secara linear dan berurutan, di mana setiap tahap perlu diselesaikan sebelum melanjutkan ke tahap berikutnya (Ramadhan & Angelia, 2023). Model ini dipilih karena memberikan struktur kerja yang jelas dan sesuai untuk proyek dengan kebutuhan yang telah terdefinisi. Pada bab ini dijelaskan tahapan pengembangan yang diterapkan dalam penyusunan aplikasi, diikuti dengan diagram alur metodologi Waterfall yang digunakan.

Berikut adalah diagram alur dari tahapan-tahapan dalam metodologi Waterfall yang digunakan dalam penelitian ini.



Gambar 3.1 Fase Tahapan Metodologi Waterfall (Pressman, 2010)

### 3.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk memahami apa saja yang benar-benar dibutuhkan oleh jemaah dan takmir masjid dari aplikasi sistem parkir yang akan dikembangkan. Proses ini dilakukan melalui wawancara dengan dua kelompok responden, yaitu jemaah dan takmir masjid. Dalam penelitian ini, proses wawancara dilaksanakan di Masjid Daarussalam Ganjuran yang beralamat di Jalan Kinanti, Condongcatur, Kecamatan Depok, Kabupaten Sleman. Temuan dari proses ini menjadi dasar dalam menyusun kebutuhan

fungsional dan non-fungsional sistem, sehingga aplikasi yang dibuat dapat sesuai dengan permasalahan yang terjadi di lapangan.



Gambar 3.2 Lokasi Penelitian Masjid Daarussalam Ganjuran

### 3.1.1 Ringkasan Hasil Wawancara

Wawancara dilakukan dengan dua kelompok responden, yaitu jemaah sebagai pengguna fasilitas parkir dan takmir masjid sebagai pihak yang mengelola operasional masjid. Hasil wawancara menunjukkan adanya kesamaan pandangan antara kedua pihak terkait permasalahan parkir di Masjid Daarussalam Ganjuran.

Dari sudut pandang jemaah, kondisi parkir dirasakan kurang tertata pada waktu kunjungan tinggi, khususnya saat salat Jumat dan kegiatan berskala besar. Situasi tersebut menyulitkan jemaah dalam mengetahui ketersediaan ruang parkir dan berdampak pada ketidakteraturan di sekitar area masjid. Sementara itu, takmir masjid menyampaikan bahwa pengelolaan parkir yang masih dilakukan secara manual menyebabkan pemanfaatan lahan belum berjalan secara optimal. Pola parkir kendaraan yang tidak teratur turut memengaruhi efektivitas penggunaan ruang parkir yang tersedia.

Secara umum, baik jemaah maupun takmir masjid menilai perlunya penerapan solusi berbasis teknologi untuk menyediakan informasi ketersediaan parkir yang lebih jelas serta membantu pengelolaan fasilitas agar aktivitas ibadah dapat berlangsung dengan lebih tertib dan

teratur. Ringkasan hasil wawancara selanjutnya disajikan dalam bentuk tabel berdasarkan kategori responden.

a. Ringkasan Wawancara Jemaah

Ringkasan wawancara dengan jemaah disusun untuk menggambarkan kebutuhan dan kendala yang mereka dalam penggunaan area parkir masjid. Hasil temuan tersebut dirangkum pada Tabel 3.1.

Tabel 3.1 Ringkasan Hasil Wawancara Jemaah

No	Inti Pertanyaan	Responden	Ringkasan Temuan
1	Pengalaman mencari parkir	Mas Gustiawan	Mengalami kesulitan memperoleh tempat parkir pada waktu salat Jumat karena keterbatasan lahan yang tidak sebanding dengan jumlah jemaah.
		Mas Fajri	Menyampaikan bahwa kondisi parkir sangat padat saat salat Jumat, sehingga menyulitkan pencarian ruang parkir.
		Mas Hafizh	Merasakan kesulitan parkir terutama saat acara besar, akibat tidak adanya arahan petugas dan penataan parkir yang kurang teratur.
2	Kesulitan mengetahui kondisi parkir	Mas Gustiawan	Tidak adanya informasi awal menyebabkan jemaah sering terlanjur masuk ke area parkir yang ternyata sudah penuh dan harus memutar balik.
		Mas Fajri	Harus berputar-putar mencari celah parkir karena tidak mengetahui kondisi ketersediaan sebelum masuk area masjid.
		Mas Hafizh	Dari luar terlihat tersedia, namun setelah masuk area parkir ternyata penuh, sehingga membuang waktu

No	Inti Pertanyaan	Responden	Ringkasan Temuan
			dan menyulitkan pengambilan keputusan
3	Kecukupan kapasitas parkir	Mas Gustiawan	Menilai kapasitas parkir tidak memadai saat kegiatan besar hingga kendaraan melebar ke pinggir jalan.
		Mas Fajri	Menganggap kapasitas cukup pada hari biasa, namun tidak mampu menampung lonjakan kendaraan saat salat Jumat.
		Mas Hafizh	Menyatakan bahwa secara lahan cukup luas, tapi karena tidak teratur membuat kapasitas menjadi tidak maksimal.
4	Manfaat sistem informasi parkir	Mas Gustiawan	Sistem dinilai akan membantu dalam pengelolaan parkir bagi jemaah.
		Mas Fajri	Sistem informasi parkir dinilai sangat membantu, karena memberikan kepastian kondisi parkir sebelum masuk area masjid.
		Mas Hafizh	Informasi kapasitas parkir membantu jemaah menghindari pemborosan waktu ketika area parkir sudah penuh.
5	Kebutuhan fitur tambahan	Mas Gustiawan	Fitur informasi jadwal salat dan kegiatan masjid diperlukan agar aplikasi tetap relevan.
		Mas Fajri	Informasi kegiatan masjid membantu meningkatkan partisipasi jemaah dalam aktivitas masjid.
		Mas Hafizh	Fitur tambahan menjadikan aplikasi lebih multifungsi bagi jemaah.

No	Inti Pertanyaan	Responden	Ringkasan Temuan
6	Preferensi UI/UX	Mas Gustiawan	Tampilan aplikasi harus sederhana dan mudah dipahami oleh seluruh kalangan.
		Mas Fajri	Antarmuka sederhana memudahkan pengguna dalam menggunakan aplikasi.
		Mas Hafizh	Kesederhanaan tampilan penting agar aplikasi dapat digunakan dengan tingkat literasi teknologi yang beragam.

b. Ringkasan Wawancara Takmir Masjid

Wawancara dengan takmir masjid dilakukan untuk mengetahui kendala pengelolaan parkir dan kebutuhan sistem dari sisi operasional. Ringkasan hasil wawancara disajikan pada Tabel Tabel 3.2.

Tabel 3.2 Ringkasan Hasil Wawancara Takmir Masjid

No	Inti Pertanyaan	Responden	Ringkasan Temuan
1	Tantangan pengelolaan parkir	Pak Jumardi (Ketua Takmir)	Ketidaktertiban perilaku parkir jemaah dan ketiadaan marka parkir yang menyeluruh menyebabkan pengelolaan parkir menjadi sulit.
		Mas Nafis (Takmir Masjid)	Lonjakan kendaraan saat salat Jumat menuntut pembukaan seluruh area parkir masjid agar tetap dapat menampung kendaraan jemaah.
2	Hambatan saat kendaraan meningkat	Pak Jumardi (Ketua Takmir)	Keterbatasan petugas parkir di luar hari Jumat menyebabkan kendaraan sering menghalangi akses keluar masuk jemaah.

No	Inti Pertanyaan	Responden	Ringkasan Temuan
		Mas Nafis (Takmir)	Perilaku parkir yang tidak presisi menghambat arus lalu lintas didalam area masjid.
3	Proses pengaturan parkir saat ini	Pak Jumardi (Ketua Takmir)	Pengelolaan parkir masih dilakukan secara manual, dan hanya dibantu petugas pada waktu tertentu.
		Mas Nafis (Takmir)	Pada hari biasa tidak ada petugas khusus, sedangkan saat salat Jumat terdapat petugas yang mengarahkan arus parkir.
4	Total kapasitas parkir	Pak Jumardi (Ketua Takmir)	Kapasitas diperkirakan 10-15 mobil, dan 100-150 motor dengan memanfaatkan seluruh area masjid.
		Mas Nafis (Takmir)	Estimasi kapasitas sekitar 10 mobil dan 50 motor, yang dapat meningkat hingga ratusan unit motor pada kondisi padat (seperti salat Jumat) dengan memanfaatkan area tambahan.
5	Kebutuhan sistem monitoring otomatis	Pak Jumardi (Ketua Takmir)	Sistem diperlukan untuk memberikan informasi kondisi parkir kepada jemaah sebelum tiba di masjid.
		Mas Nafis (Takmir)	Sistem sangat bermanfaat terutama pada momen kepadatan tinggi seperti saat salat Jumat dan hari besar islam.
6	Kebutuhan fitur pendukung tambahan	Pak Jumardi (Ketua Takmir)	Fitur informasi jadwal salat dan informasi acara masjid dinilai dapat membantu penyampaian informasi kepada jemaah secara lebih efektif.
		Mas Nafis (Takmir)	Integrasi notifikasi kegiatan masjid dinilai sangat membantu sebagai

No	Inti Pertanyaan	Responden	Ringkasan Temuan
			sarana penyampaian informasi dan pengingat kegiatan kepada jemaah.
7	Harapan terhadap sistem	Pak Jumardi (Ketua Takmir)	Sistem diharapkan informatif dan mampu memberikan edukasi tertib parkir kepada jemaah.
		Mas Nafis (Takmir)	Sistem diharapkan fleksibel dan mudah diakses oleh jemaah.

### 3.1.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional dirumuskan berdasarkan temuan dari proses wawancara dengan jemaah dan takmir masjid. *Insight* yang diperoleh mengenai permasalahan parkir, kebutuhan informasi, serta preferensi fitur menjadi dasar dalam menentukan fungsi yang harus tersedia pada aplikasi. Kebutuhan ini memastikan bahwa sistem yang dikembangkan relevan dengan kondisi lapangan dan mampu menjawab kebutuhan pengguna. Berdasarkan analisis tersebut, kebutuhan fungsional yang harus dipenuhi oleh sistem dapat diuraikan sebagai berikut.

#### A. Informasi ketersediaan parkir

Aplikasi menyediakan informasi mengenai kapasitas ketersediaan ruang parkir untuk setiap jenis kendaraan sebagai referensi awal bagi jemaah. Informasi tersebut bertujuan untuk membantu jemaah mengantisipasi kondisi parkir sebelum tiba ke masjid, sehingga dapat menyesuaikan waktu kedatangan dan meminimalisir potensi kepadatan kendaraan.

#### B. Informasi dan Pengelolaan Jadwal Salat

Aplikasi menampilkan jadwal salat harian agar jemaah dapat mengakses informasi waktu ibadah dengan lebih mudah. Takmir masjid dapat melakukan pembaruan jadwal secara berkala sehingga informasi yang tersedia pada aplikasi tetap sesuai dengan ketentuan waktu yang berlaku.

#### C. Informasi dan Pengelolaan Agenda Acara Masjid

Aplikasi menyediakan informasi kegiatan acara masjid yang dapat diakses jemaah secara terpusat dalam satu platform. Takmir masjid dapat menambahkan, memperbarui, maupun menghapus data kegiatan sesuai kebutuhan agar penyampaian informasi kepada jemaah tetap relevan.

#### D. Notifikasi Informasi untuk Jemaah

Aplikasi mendukung pengiriman notifikasi kepada jemaah sebagai media penyampaian informasi dan pengingat, seperti jadwal salat maupun agenda acara masjid. Notifikasi ini juga mendorong jemaah untuk membuka aplikasi sehingga pengguna dapat melihat informasi yang diperlukan, termasuk estimasi kondisi parkir sebelum berangkat.

### 3.1.3 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional disusun berdasarkan preferensi pengguna dan tantangan penggunaan yang disampaikan dalam wawancara. Aspek kualitas seperti kemudahan penggunaan, dan keandalan dibutuhkan agar aplikasi dapat berjalan optimal serta mudah dioperasikan oleh seluruh jemaah dan takmir masjid. Dengan mempertimbangkan aspek-aspek tersebut, kebutuhan non-fungsional yang diperlukan dalam pengembangan aplikasi diuraikan sebagai berikut.

#### A. Kemudahan Penggunaan (*Usability*)

Desain antarmuka aplikasi perlu dirancang dengan sederhana, dan mudah untuk dipahami oleh pengguna dari berbagai latar belakang, sehingga para pengguna dapat dengan mudah mengakses informasi ketika menggunakan aplikasi.

#### B. Keandalan (*Reliability*)

Aplikasi harus mampu menyajikan informasi yang diperbarui secara konsisten dan akurat sesuai dengan kondisi yang sebenarnya. Sistem diharapkan tetap stabil ketika terjadi pembaruan data oleh takmir.

## 3.2 Perancangan Sistem

Pada bagian ini, akan dijelaskan konsep skema perancangan aplikasi sistem berbasis *mobile*. Bagian ini akan mencakup deskripsi aktor, diagram *use case*, *activity diagram*, dan antarmuka desain aplikasi. Tujuannya adalah untuk memberikan pemahaman yang komprehensif mengenai cara kerja aplikasi dan bagaimana interaksi antar setiap aktor dengan fitur-fitur utama yang ada dalam aplikasi.

### 3.2.1 Deskripsi Aktor/Pengguna

Aktor dalam aplikasi terdiri dari dua entitas, yaitu jemaah dan takmir masjid. Setiap aktor memiliki hak akses yang berbeda-beda dalam aplikasi sesuai dengan peran dan kebutuhannya.

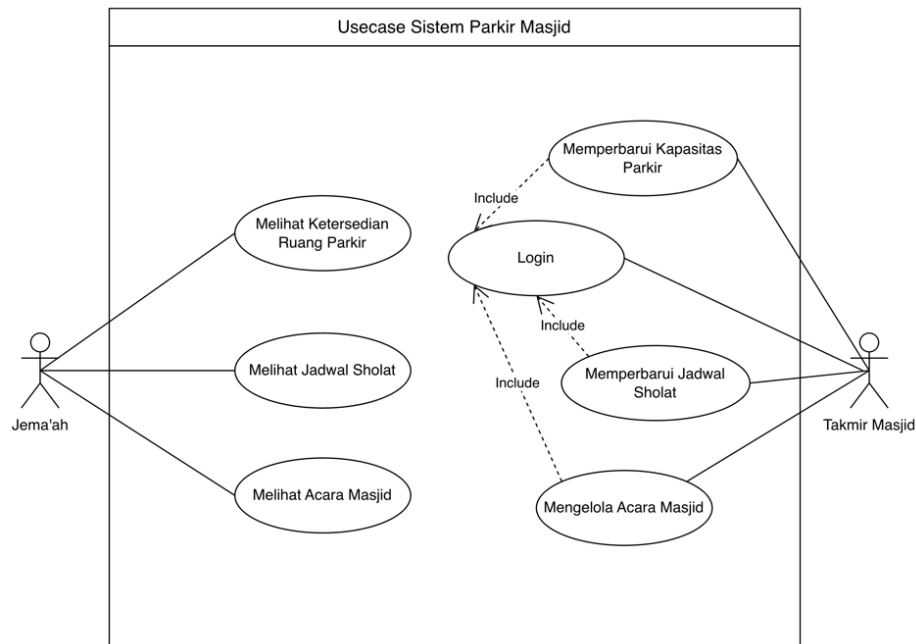
Perincian akses dan fitur-fitur yang dapat diakses oleh tiap aktor disajikan lebih lanjut pada Tabel 3.3.

Tabel 3.3 Deskripsi Aktor

No.	Aktor	Deskripsi
1.	Jemaah	Pengguna utama yang memiliki akses untuk melihat informasi ketersediaan ruang parkir, jadwal salat, serta daftar acara masjid. Selain itu, pengguna juga dapat mengaktifkan notifikasi untuk pengingat waktu salat dan pemberitahuan acara masjid.
2.	Takmir Masjid (admin)	Pengguna yang memiliki peran lebih luas dibandingkan jemaah. Selain dapat melihat informasi ketersediaan ruang parkir, jadwal salat, dan acara kegiatan masjid, takmir masjid juga dapat memperbarui kapasitas parkir dan jadwal salat, serta mengelola data acara kegiatan masjid.

### 3.2.2 *Diagram Use Case*

*Diagram usecase* digunakan untuk menggambarkan interaksi antara aktor (jemaah dan takmir masjid) dengan sistem aplikasi, serta memudahkan pemahaman tentang fungsi-fungsi yang tersedia bagi setiap aktor.



Gambar 3.3 Use Case Aplikasi Sistem Parkir Masjid

Gambar 3.3 menunjukkan diagram *use case* sistem parkir masjid yang menggambarkan interaksi antara antara aktor jemaah dan takmir masjid dengan sistem. Aktor jemaah berinteraksi dengan sistem untuk melihat informasi ketersediaan ruang parkir, jadwal salat, dan acara masjid sebagai bentuk pemanfaatan sistem dalam memperoleh informasi yang dibutuhkan. Sementara itu, aktor takmir masjid berperan dalam pengelolaan sistem, yang mencakup pembaruan kapasitas parkir, pembaruan jadwal salat, serta pengelolaan data acara masjid. Seluruh aktivitas pengelolaan data yang dilakukan oleh takmir masjid mensyaratkan proses login sebagai mekanisme autentikasi, yang dimodelkan menggunakan relasi *include*, di mana akses login hanya tersedia melalui halaman *splash screen*, sedangkan pengguna jemaah tidak memerlukan proses login dan langsung diarahkan ke halaman utama jemaah.

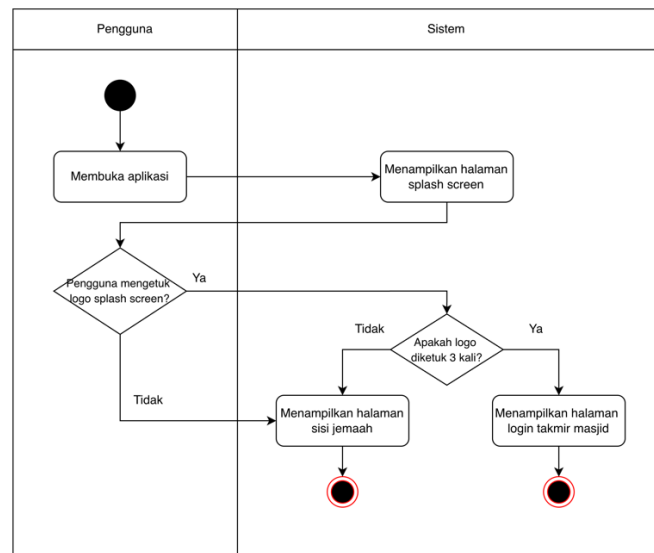
### 3.2.3 Activity Diagram

*Activity diagram* digunakan untuk memvisualisasikan alur proses dari tiap-tiap aktor dalam aplikasi. Diagram ini akan menggambarkan hubungan antara masing-masing aktor yaitu jemaah dan takmir masjid dengan sistem dalam menggunakan aplikasi sistem parkir. Berikut adalah penjelasan setiap *activity diagram* dari masing-masing aktor.

#### A. Activity Diagram Akses Pengguna

*Activity Diagram* akses pengguna menggambarkan proses awal ketika aplikasi dijalankan. Pada tahap ini, sistem membedakan akses berdasarkan interaksi pengguna pada

halaman *splash screen*. Apabila pengguna melakukan interaksi khusus berupa mengetuk logo sebanyak tiga kali, sistem akan menampilkan halaman login takmir masjid sebagai akses menuju fitur pengelolaan. Sebaliknya, apabila interaksi tersebut tidak dilakukan, sistem akan mengalihkan pengguna secara otomatis ke halaman sisi jemaah.



Gambar 3.4 *Activity Diagram* Akses Pengguna

## B. *Activity Diagram* Menu Jemaah

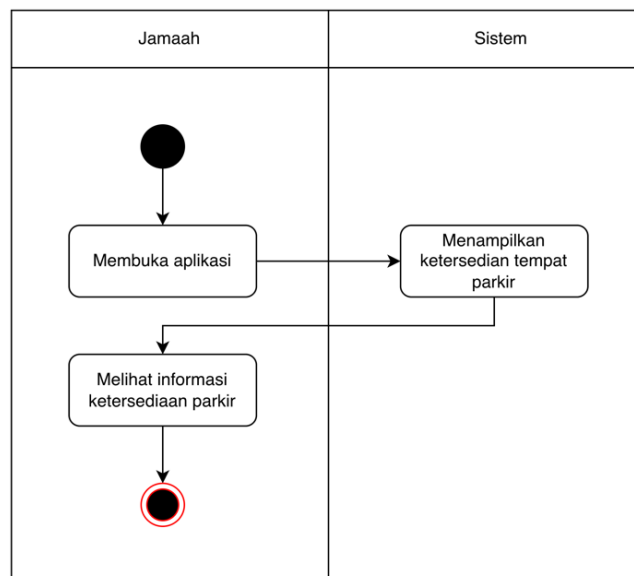
*Activity diagram* menu jemaah menggambarkan alur saat pengguna umum (jemaah) mengakses aplikasi untuk memilih fitur yang tersedia. Pada tahap ini, jemaah dapat memilih untuk melihat informasi ketersediaan ruang parkir, jadwal salat, atau agenda acara masjid. Diagram ini menunjukkan bagaimana sistem menampilkan menu utama setelah aplikasi dibuka, kemudian jemaah memilih salah satu fitur sesuai kebutuhan. Alur *activity diagram* menu jemaah ditunjukkan pada Gambar 3.5.



Gambar 3.5 *Activity Diagram* Menu Jemaah

### C. *Activity Diagram* Melihat Ketersediaan Ruang Parkir

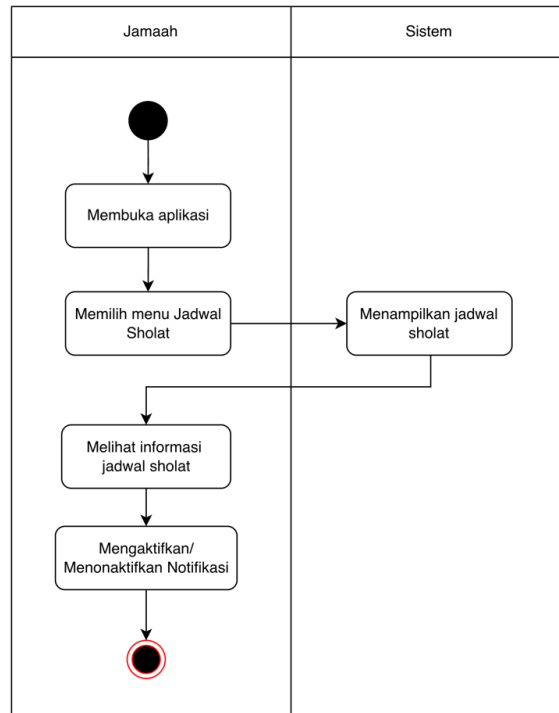
Aktivitas untuk melihat ketersediaan ruang parkir dapat dilakukan oleh jemaah maupun takmir masjid melalui aplikasi. Pada proses ini, pengguna dapat memeriksa informasi ketersediaan ruang parkir untuk kendaraan mobil dan motor. Sistem menampilkan jumlah ketersediaan berdasarkan data yang tersimpan pada Firebase, sehingga pengguna memperoleh informasi sesuai pembaruan data yang tersedia pada sistem. *Activity diagram* untuk melihat ketersediaan ruang parkir ditunjukkan pada Gambar 3.6.



Gambar 3.6 *Activity Diagram* Melihat Ketersediaan Ruang Parkir

#### D. *Activity Diagram* Melihat Jadwal Salat

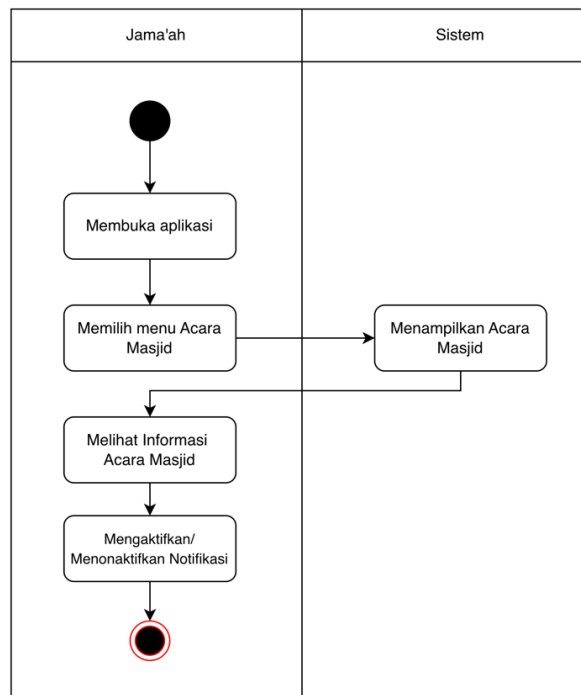
Aktivitas melihat jadwal salat dilakukan oleh jemaah maupun takmir masjid. Pengguna dapat melihat jadwal salat lima waktu yang telah disimpan dalam aplikasi. Selain itu, jemaah juga dapat mengaktifkan atau menonaktifkan notifikasi pengingat salat sesuai kebutuhan. Sistem akan menampilkan jadwal secara lengkap, dan notifikasi akan dikirim sesuai konfigurasi pengguna. *Activity diagram* melihat jadwal salat ditunjukkan pada Gambar 3.7.



Gambar 3.7 *Activity Diagram* Melihat Jadwal Salat

#### E. *Activity Diagram* Melihat Acara Masjid

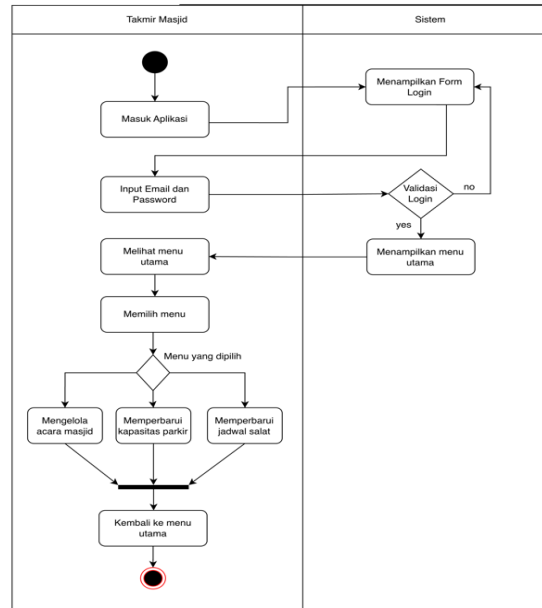
Aktivitas melihat acara masjid dapat dilakukan oleh jemaah maupun takmir masjid. Pengguna dapat melihat daftar kegiatan masjid yang sedang berlangsung maupun agenda yang akan datang. Jemaah juga dapat mengaktifkan atau menonaktifkan notifikasi acara agar menerima pengingat sesuai jadwal kegiatan. *Activity diagram* lengkap untuk melihat acara masjid ditampilkan pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Melihat Acara Masjid

#### F. *Activity Diagram* Menu Takmir Masjid

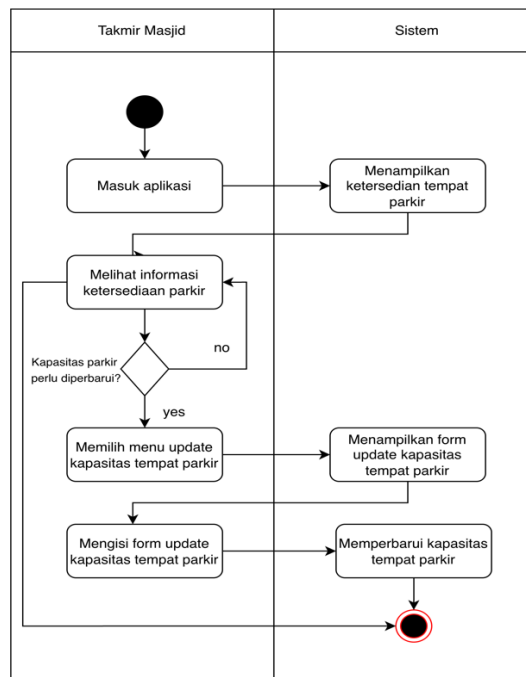
*Activity diagram* ini menjelaskan alur akses yang dilakukan oleh takmir masjid dalam menggunakan aplikasi sebelum melakukan pengelolaan data. Takmir masjid harus melakukan login dengan memasukkan email dan password terlebih dahulu. Sistem kemudian melakukan proses validasi terhadap data login yang dimasukkan. Apabila data login tidak valid, sistem akan menampilkan kembali form login. Namun apabila data login valid, sistem akan menampilkan menu utama yang menyediakan tiga pilihan menu, yaitu mengelola acara masjid, memperbarui kapasitas parkir, dan memperbarui jadwal salat. Alur lengkap dari proses tersebut ditampilkan pada Gambar 3.9.



Gambar 3.9 *Activity Diagram* Menu Takmir Masjid

### G. *Activity Diagram* Memperbarui Kapasitas Parkir

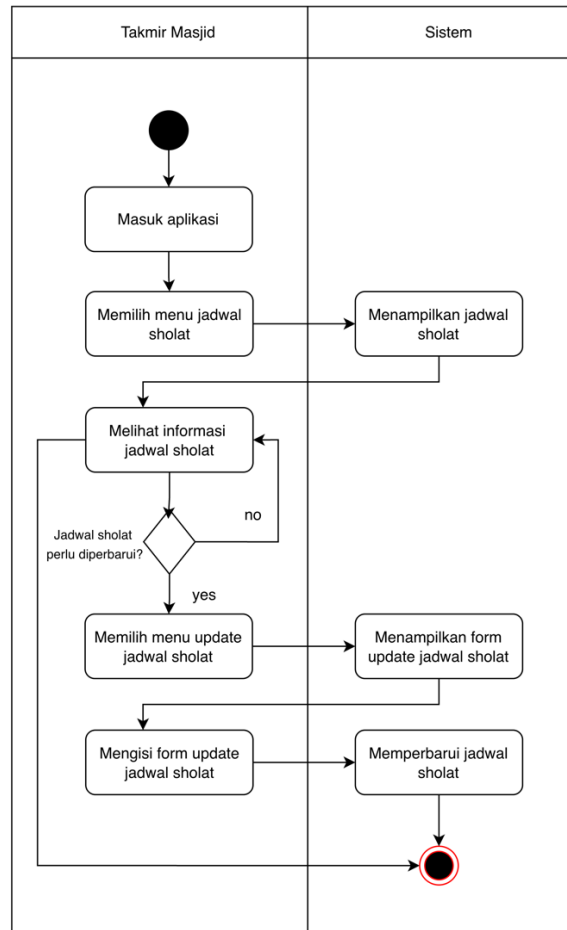
Aktivitas memperbarui kapasitas parkir hanya dapat dilakukan oleh takmir masjid. Takmir masjid dapat melakukan pembaruan pada kapasitas ketersediaan ruang parkir jika terdapat perubahan pada area parkir. Berikut adalah *activity diagram* memperbarui kapasitas parkir yang ditampilkan pada Gambar 3.10.



Gambar 3.10 *Activity Diagram* Memperbarui Kapasitas Parkir

## H. Activity Diagram Memperbarui Jadwal Sholat

Aktivitas memperbarui jadwal salat hanya dapat dilakukan oleh takmir masjid. Takmir masjid dapat mengelola dan memperbarui informasi terkait jadwal salat agar tetap relevan dan akurat. Berikut adalah *activity diagram* memperbarui jadwal salat yang ditampilkan pada Gambar 3.11.

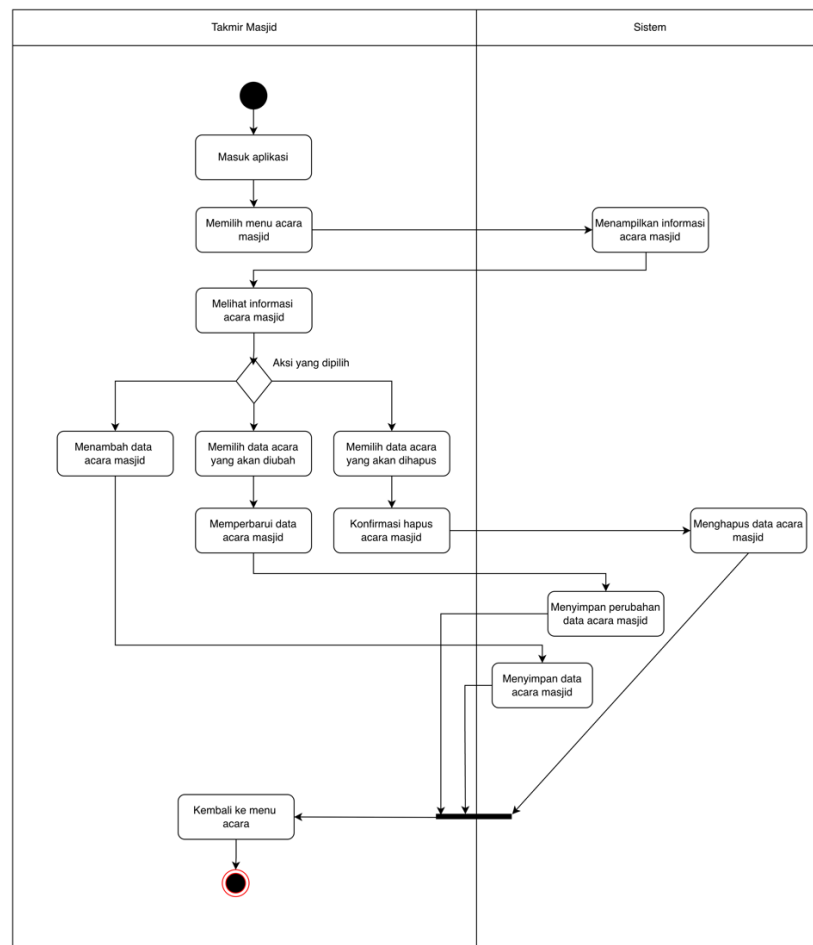


Gambar 3.11 Activity Diagram Memperbarui Jadwal Salat

## I. Activity Diagram Mengelola Acara Masjid

*Activity Diagram Mengelola Acara Masjid* menggambarkan alur aktivitas yang dilakukan oleh takmir masjid dalam melakukan pengelolaan data acara masjid. Setelah memilih menu acara masjid, sistem akan menampilkan daftar acara yang telah tersimpan. Takmir masjid kemudian dapat memilih aksi pengelolaan yang diinginkan, yaitu menambah data acara baru, memperbarui data acara yang sudah ada, atau menghapus data acara masjid. Setelah salah satu aksi pengelolaan dilakukan, sistem akan memproses dan menyimpan

perubahan data yang terjadi. *Activity diagram* mengelola acara masjid ditampilkan pada Gambar 3.12.



Gambar 3.12 *Activity Diagram* Mengelola Acara Masjid

### 3.2.4 Desain Aplikasi

Bagian ini memberikan gambaran mengenai rancangan antarmuka dan struktur database dari aplikasi sistem parkir. Tahapan ini akan berfungsi sebagai fondasi awal dalam proses pengembangan aplikasi. Berikut adalah penjelasan dari rancangan antarmuka dan desain database.

#### A. Rancangan Antarmuka

Perancangan antarmuka merupakan sebuah proses dalam mendefinisikan suatu tata letak, navigasi, dan elemen-elemen visual pada setiap halaman. Tahapan ini bertujuan untuk menghasilkan sketsa antarmuka dalam aplikasi. Berikut adalah rincian dari rancangan sketsa desain aplikasi.

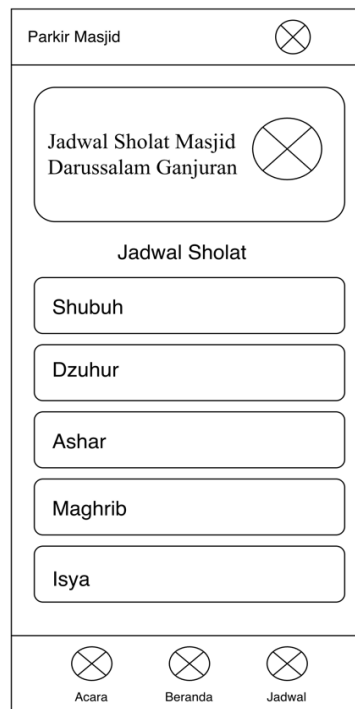
## 1) Antarmuka untuk Jemaah

## a. Halaman ketersediaan parkir



Gambar 3.13 Desain Halaman Ketersediaan Parkir

## b. Halaman jadwal sholat



Gambar 3.14 Desain Halaman Jadwal Salat

## c. Halaman Acara Masjid

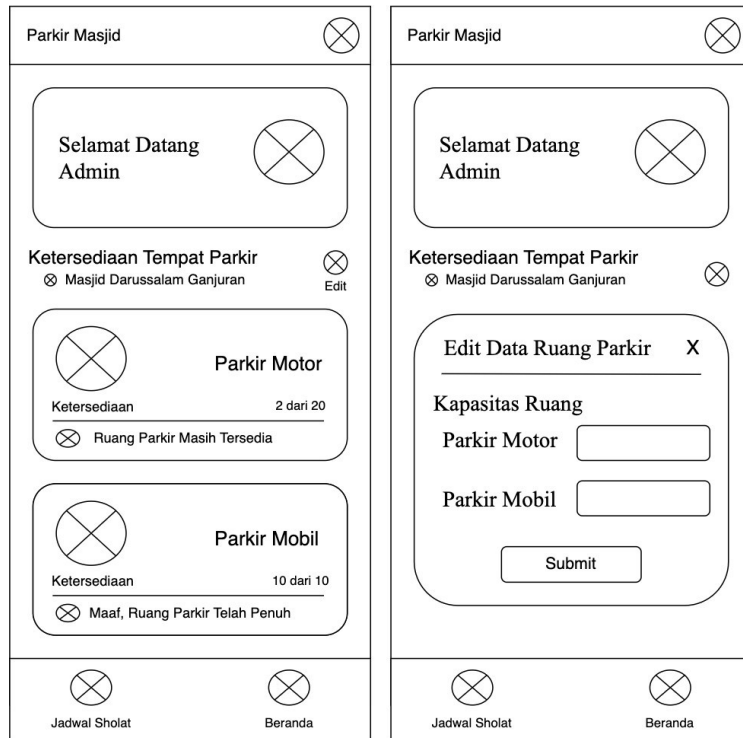
Gambar 3.15 Desain Halaman Acara

## 2) Antarmuka untuk Takmir Masjid

### a. Halaman Login

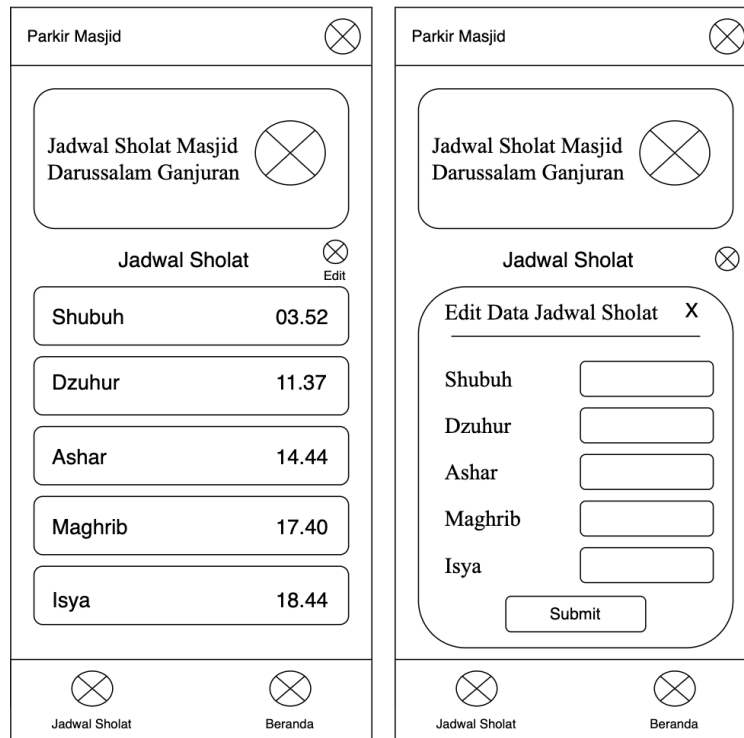
Gambar 3.16 Desain Halaman Login

### b. Halaman pengelolaan parkir



Gambar 3.17 Desain Tampilan Pengelolaan Parkir

c. Halaman pengelolaan jadwal salat



Gambar 3.18 Desain Halaman Pengelolaan Jadwal Salat

d. Halaman Pengelolaan Acara Masjid

The image shows two mobile application screens for managing mosque events. Both screens are titled 'Parkir Masjid (Admin)'.  
 The left screen, 'Informasi Acara Kegiatan Masjid', displays a list of events. Each event card contains a 'Nama Acara' field, a 'Deskripsi Acara' field, and a 'Tanggal & Jam Acara' field. Below each card are 'Hapus' and 'Edit' buttons. A 'Tambah' button is located at the bottom right of the list. At the very bottom, there are three navigation icons labeled 'Acara', 'Beranda', and 'Jadwal'.  
 The right screen, also titled 'Informasi Acara Kegiatan Masjid', is a form for adding or editing an event. It features a 'Tambah/Edit Acara' button at the top. Below it are input fields for 'Nama Acara', 'Deskripsi Acara', 'Tanggal Acara', and 'Jam Acara'. At the bottom of the form are 'Tidak' and 'Simpan' buttons. The same three navigation icons ('Acara', 'Beranda', 'Jadwal') are present at the bottom.

Gambar 3.19 Desain Halaman Pengelolaan Acara Masjid

## B. Desain Database

Perancangan *database* dilakukan untuk mengatur alur penyimpanan dan pengolahan data agar aplikasi dapat dijalankan secara efisien dan terorganisasi. Aplikasi ini memanfaatkan Firebase, dengan *Firestore Authentication* sebagai mekanisme login berbasis email dan kata sandi, serta *Firestore Realtime Database* sebagai penyimpanan data utama. Data yang digunakan dalam aplikasi terbagi ke dalam empat kategori, yaitu akun, parkir, salat, dan acara. Rincian struktur dari masing-masing kategori ditampilkan pada tabel berikut.

### 1) Data Akun

Data akun berfungsi untuk menyimpan informasi autentikasi pada proses login. Struktur tabel data akun dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4 Struktur Data Akun

No.	Nama Kolom	Tipe Data	Keterangan
1.	Email	<i>String</i>	Alamat email pengguna
2.	Password	<i>String</i>	Kata sandi pengguna

### 2) Data Parkir

Data parkir berfungsi untuk menyimpan informasi tentang kapasitas maksimum area parkir serta jumlah ruang parkir yang sedang terisi. Struktur data parkir dapat dilihat pada Tabel 3.5 berikut.

Tabel 3.5 Struktur Data Parkir

No.	Nama Kolom	Tipe Data	Keterangan
1.	Jenis	<i>String</i>	Id untuk setiap jenis kendaraan
2.	Kapasitas	<i>Integer</i>	Kapasitas maksimum area parkir
3.	Terisi	<i>Integer</i>	Jumlah kendaraan yang sedang terparkir

### 3) Data Salat

Tabel data salat berfungsi untuk menyimpan informasi dari keseluruhan jadwal salat. Struktur data salat dapat dilihat pada Tabel 3.6 berikut.

Tabel 3.6 Struktur Data Salat

No.	Nama Kolom	Tipe Data	Keterangan
1.	Nama	<i>String</i>	Id untuk setiap jadwal salat
2.	Jadwal	<i>String</i>	Waktu spesifik salat

### 4) Data Acara Masjid

Data acara masjid berfungsi untuk menyimpan informasi dari keseluruhan kegiatan masjid. Struktur data acara masjid dapat dilihat pada Tabel 3.7 berikut.

Tabel 3.7 Struktur Data Acara Masjid

No.	Nama Kolom	Tipe Data	Keterangan
1.	Id	<i>String</i>	Id untuk setiap acara masjid
2.	Nama	<i>String</i>	Nama kegiatan masjid
3.	Deskripsi	<i>String</i>	Deskripsi acara
4.	Jam	<i>String</i>	Waktu acara
5.	Tanggal	<i>String</i>	Tanggal acara

### 3.3 Implementasi

Pada tahap ini, proses implementasi dilakukan setelah analisis dan perancangan sistem diselesaikan. Implementasi berlangsung secara bertahap sesuai rancangan yang telah ditetapkan sebelumnya. Kegiatan pada tahap ini mencakup pengembangan fitur aplikasi, penyusunan antarmuka pengguna, serta konfigurasi basis data. Pengembangan aplikasi dilakukan menggunakan bahasa pemrograman Kotlin melalui platform Android Studio, sedangkan pengelolaan data dilakukan dengan memanfaatkan layanan Firebase.

### 3.4 Pengujian

Pada tahap ini dilakukan proses pengujian untuk memastikan kualitas dan kelayakan penggunaan aplikasi sistem parkir yang telah dikembangkan. Pengujian dilakukan untuk memverifikasi bahwa setiap fitur dapat berfungsi sesuai kebutuhan pengguna serta memberikan pengalaman penggunaan yang optimal. Metode yang digunakan meliputi *black-box testing*, yang berfokus pada kesesuaian keluaran terhadap masukan tanpa meninjau kode program, serta evaluasi *usability* menggunakan *System Usability Scale* (SUS) guna memperoleh gambaran persepsi pengguna terhadap kemudahan, dan kenyamanan aplikasi.

#### 3.4.1 Pengujian Fungsional (*Black-box Testing*)

Pengujian fungsional pada aplikasi ini dilakukan dengan pendekatan *black-box*, yaitu metode yang menilai apakah keluaran sistem sudah sesuai dengan masukan yang diberikan tanpa memeriksa proses logika di dalam program. Setiap skenario pengujian dirancang untuk merepresentasikan situasi tertentu yang perlu divalidasi agar fungsi aplikasi dapat dipastikan berjalan sebagaimana mestinya. Untuk menjaga alur pengujian tetap terstruktur, skenario disusun terpisah berdasarkan peran pengguna yang berbeda. Susunan skenario tersebut disajikan dalam dua tabel berikut sebagai dasar pelaksanaan pengujian pada tahap selanjutnya.

##### A. Pengujian *Black-box* untuk Takmir Masjid (Admin)

Pengujian *black-box* pada sisi takmir masjid (admin) dilakukan untuk memastikan bahwa fungsi-fungsi pengelolaan sistem dapat berjalan sesuai dengan kebutuhan yang telah dirancang. Pengujian ini mencakup proses autentikasi, pengelolaan kapasitas parkir, pembaruan jadwal salat, serta pengelolaan data acara masjid. Rincian skenario pengujian untuk sisi takmir masjid disajikan pada Tabel 3.8.

Tabel 3.8 Tabel Skenario Pengujian Admin

Halaman	Pengujian	Hasil yang diharapkan	Hasil
<i>Splash screen</i> Admin	Akses halaman admin (tap logo 3x)	Sistem membuka ke halaman login admin.	
Login admin	Login dengan kredensial valid	Admin berhasil masuk ke halaman <i>dashboard</i> admin.	
	Login dengan kredensial salah	Sistem menampilkan pesan <i>toast</i> gagal login.	
	Login dengan form kosong	Sistem menampilkan pesan <i>toast</i> “ <i>email</i> dan <i>password</i> harus diisi”.	
<i>Dashboard</i> admin	<i>Logout</i> admin	Sistem kembali ke halaman <i>splash screen</i> .	
Data parkir admin	Menampilkan data kapasitas dan ketersediaan parkir	Sistem menampilkan data terisi dan kapasitas total parkir untuk motor dan mobil.	
	Mengubah kapasitas total parkir	Sistem menyimpan perubahan kapasitas total parkir, dan memperbarui data tampilan parkir.	
	Sinkronisasi perubahan kapasitas ke sisi jemaah	Perubahan nilai kapasitas tercermin pada tampilan parkir sisi jemaah secara sinkron	
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> , dan gagal menyimpan perubahan.	
Jadwal salat admin	Menampilkan jadwal salat	Sistem menampilkan data jadwal salat.	
	Mengubah jadwal salat	Data jadwal salat tersimpan dan diperbarui.	

Halaman	Pengujian	Hasil yang diharapkan	Hasil
	Mengubah lebih dari satu jadwal salat	Sistem menyimpan seluruh perubahan secara bersamaan.	
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> , dan gagal menyimpan perubahan.	
Acara masjid admin	Menampilkan daftar acara	Sistem menampilkan daftar acara masjid.	
	Menambah acara	Sistem menyimpan data acara baru, dan menampilkannya pada daftar acara.	
	Validasi salah satu <i>field</i> ada yang kosong	Sistem gagal menyimpan dan menampilkan pesan <i>toast</i> “mohon lengkapi semua <i>field</i> ”.	
	Mengedit acara	Data acara pada salah satu item berhasil diperbarui.	
	Menghapus acara	Data acara pada salah satu item berhasil dihapus.	
	Perilaku saat <i>offline</i>	Sistem gagal menyimpan data acara.	

## B. Pengujian *Black-box* untuk Jemaah

Pengujian *black-box* pada sisi jemaah dilakukan untuk memastikan bahwa fitur-fitur aplikasi yang berkaitan dengan penyampaian informasi dapat digunakan dengan baik oleh pengguna. Pengujian ini meliputi akses informasi ketersediaan parkir, jadwal salat, agenda acara masjid, serta penerimaan notifikasi. Rincian skenario pengujian untuk sisi jemaah disajikan pada Tabel 3.9.

Tabel 3.9 Tabel Skenario Pengujian Jemaah

Halaman	Pengujian	Hasil yang diharapkan	Hasil
<i>Splash screen</i> jemaah	Jemaah membuka aplikasi	Sistem langsung menampilkan ke halaman sisi jemaah.	
Data parkir jemaah	Menampilkan data ketersediaan parkir	Sistem menampilkan data terisi dan kapasitas ketersediaan parkir.	
	Melakukan <i>refresh</i> data parkir	Sistem mengambil data parkir terbaru.	
	Perubahan data jumlah kapasitas parkir dari admin	Data kapasitas parkir berubah sesuai update dari admin.	
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> untuk data parkir.	
Jadwal salat jemaah	Menampilkan jadwal salat	Sistem menampilkan data jadwal salat.	
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai default pada jadwal salat.	
	Mengaktifkan notifikasi salat	Notifikasi salat diaktifkan.	
	Menonaktifkan notifikasi salat	Notifikasi salat dinonaktifkan.	
	Penerimaan notifikasi salat	Sistem menampilkan notifikasi azan sesuai jadwal salat.	
Acara masjid jemaah	Menampilkan daftar data acara masjid	Sistem menampilkan daftar acara masjid.	
	Mengaktifkan notifikasi acara masjid	Notifikasi acara masjid diaktifkan.	
	Menonaktifkan notifikasi acara masjid	Notifikasi acara masjid dinonaktifkan.	

Halaman	Pengujian	Hasil yang diharapkan	Hasil
	Notifikasi ketika acara masjid baru dibuat oleh admin	Sistem menampilkan notifikasi acara masjid.	
	Notifikasi pengingat 1 jam sebelum waktu acara	Sistem menampilkan notifikasi acara masjid sekitar 1 jam sebelum waktu acara.	
	Perilaku saat <i>offline</i>	Sistem tidak menampilkan daftar acara masjid.	

### 3.4.2 Evaluasi Usability (*System Usability Scale*)

Pengujian *usability* pada aplikasi dilakukan menggunakan instrumen *System Usability Scale* (SUS) sebagai alat ukur untuk mengetahui persepsi pengguna terhadap tingkat kemudahan dan kenyamanan penggunaan sistem. Pemilihan SUS didasarkan pada kemampuannya dalam memberikan gambaran umum mengenai kualitas *usability* suatu sistem melalui instrumen yang sederhana namun telah teruji secara luas dalam berbagai penelitian sistem informasi. Data persepsi pengguna diperoleh melalui penyebaran kuesioner SUS kepada responden setelah mereka mencoba menggunakan aplikasi.

Instrumen SUS terdiri atas sepuluh pernyataan yang disusun untuk merepresentasikan aspek-aspek utama *usability* sistem. Setiap pernyataan dijawab menggunakan skala *likert* lima tingkat, yaitu Sangat Tidak Setuju (STS), Tidak Setuju (TS), Netral (N), Setuju (S), dan Sangat Setuju (ST). Daftar pernyataan yang digunakan dalam penelitian ini disajikan pada Tabel 3.10 sebagai instrumen evaluasi yang diberikan kepada responden.

Tabel 3.10 Daftar pertanyaan SUS

No	Pertanyaan	STS	TS	N	S	ST
1	Saya merasa ingin menggunakan sistem ini secara berkelanjutan.					
2	Saya menilai sistem ini terasa terlalu rumit.					
3	Saya merasa sistem ini mudah untuk digunakan.					
4	Saya merasa memerlukan bantuan dari orang teknis untuk dapat menggunakan sistem ini.					

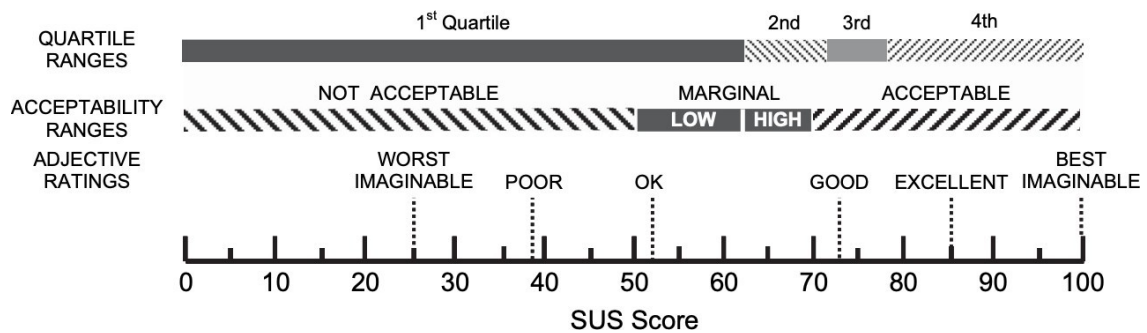
No	Pertanyaan	STS	TS	N	S	ST
5	Saya menilai fungsi-fungsi dalam sistem ini terintegrasi dengan baik.					
6	Saya merasa terdapat terlalu banyak ketidakkonsistenan dalam sistem ini.					
7	Saya membayangkan bahwa sebagian besar orang akan dapat mempelajari penggunaan sistem ini dengan cepat.					
8	Saya merasa sistem ini cukup membingungkan untuk digunakan.					
9	Saya merasa percaya diri saat menggunakan sistem ini.					
10	Saya merasa perlu mempelajari banyak hal terlebih dahulu sebelum dapat menggunakan sistem ini.					

Perhitungan skor *System Usability Scale* (SUS) dilakukan berdasarkan 10 pernyataan dengan skala *likert* 1–5. Setiap jawaban responden kemudian dikonversi menjadi skor kontribusi. Untuk pernyataan bernomor ganjil (p1, p3, p5, p7, dan p9), skor kontribusi diperoleh dari (jawaban – 1). Sementara itu, untuk pernyataan bernomor genap (p2, p4, p6, p8, dan p10), skor kontribusi diperoleh dari (5 – jawaban). Seluruh skor kontribusi tersebut dijumlahkan sehingga menghasilkan skor total dengan rentang 0–40. Selanjutnya, skor total dikalikan 2,5 untuk mendapatkan skor SUS dalam rentang 0–100.

Skor SUS yang diperoleh dari masing-masing responden selanjutnya dihitung nilai rata-ratanya untuk menghasilkan nilai akhir *System Usability Scale* penelitian. Nilai rata-rata ini diperoleh dengan menjumlahkan seluruh skor SUS responden kemudian dibagi dengan jumlah total responden, sehingga menghasilkan satu nilai representatif yang digunakan sebagai dasar dalam menilai tingkat *usability* aplikasi secara keseluruhan.

Nilai akhir SUS tersebut kemudian diinterpretasikan untuk menentukan tingkat *usability* aplikasi yang dikembangkan. Interpretasi skor mengacu pada *adjective rating scale* yang diperkenalkan oleh Bangor et al. (2008), yang mengelompokkan nilai SUS ke dalam kategori tertentu berdasarkan tingkat kemudahan penggunaan. Skala ini memberikan gambaran posisi kualitas *usability* sistem secara umum, mulai dari tingkat penerimaan yang rendah hingga

kategori dengan tingkat penerimaan yang tinggi. Visualisasi interpretasi skor *System Usability Scale* (SUS) berdasarkan *adjective rating scale* ditampilkan pada Gambar 3.20.



Gambar 3.20 Interpretasi Skor SUS (Bangor et al., 2008)

Dalam kajian literatur, nilai SUS di atas 70 umumnya dianggap sebagai indikator bahwa suatu sistem memiliki tingkat *usability* yang baik dan dapat diterima oleh pengguna, sehingga batas nilai tersebut sering digunakan sebagai acuan dalam menilai kualitas kemudahan penggunaan suatu aplikasi.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab ini memaparkan realisasi sistem parkir masjid yang telah dikembangkan dengan mengacu pada rancangan di bab sebelumnya. Fokus pembahasan meliputi gambaran umum implementasi antarmuka serta fitur-fitur utama aplikasi. Tahapan selanjutnya menguraikan hasil pengujian sistem yang terdiri dari uji fungsional, pengukuran tingkat usability menggunakan metode *System Usability Scale* (SUS), serta analisis evaluasi dan umpan balik pengguna. Hasil yang ditampilkan pada bab ini menjadi dasar untuk menilai sejauh mana aplikasi telah memenuhi tujuan pengembangan dan dapat digunakan secara efektif oleh jemaah maupun takmir masjid.

#### **4.1 Gambaran Umum Sistem**

Bagian ini menjelaskan gambaran umum dari sistem yang dikembangkan, mulai dari tujuan aplikasi hingga penjelasan fitur-fitur utama yang menjadi inti dari implementasi. Gambaran awal ini diperlukan untuk memberikan konteks sebelum memasuki penjelasan teknis pada bagian implementasi dan pengujian. Dengan memahami struktur dasar dan fitur yang tersedia, proses analisis pada tahap berikutnya dapat dilakukan secara lebih terarah.

##### **4.1.1 Tujuan Sistem**

Pengembangan Aplikasi Sistem Parkir Masjid bertujuan untuk menyediakan layanan *digital* yang menyajikan informasi kapasitas parkir serta aktivitas masjid kepada jemaah. Aplikasi ini memungkinkan pengguna memperoleh gambaran awal mengenai kondisi ketersediaan lahan parkir sebelum tiba di lokasi, sehingga dapat membantu jemaah dalam mempertimbangkan kondisi parkir, terutama pada waktu dengan tingkat kunjungan yang tinggi.

Selain sebagai sarana informasi bagi jemaah, aplikasi ini dikembangkan untuk membantu takmir masjid dalam mengelola dan menyampaikan informasi secara terpusat. Melalui pengelolaan data yang terintegrasi, aplikasi ini diharapkan dapat mendukung penyampaian informasi yang lebih teratur dan mudah diperbarui, sekaligus membantu mengurangi potensi kepadatan kendaraan. Informasi pendukung seperti jadwal salat dan acara kegiatan masjid juga disediakan untuk menunjang kebutuhan informasi jemaah serta mendukung operasional masjid secara lebih efektif.

#### 4.1.2 Deskripsi Fitur Sistem

Aplikasi sistem parkir masjid dilengkapi dengan sejumlah fitur yang dikembangkan untuk mendukung kebutuhan informasi jemaah maupun takmir masjid. Seluruh fitur disusun berdasarkan kebutuhan informasi di masjid serta dirancang agar dapat saling melengkapi satu sama lain, sehingga fungsi aplikasi dapat berjalan secara lebih selaras dan relevan dengan aktivitas jemaah. Pendekatan ini membantu memastikan bahwa setiap fitur tidak hanya berdiri sendiri, tetapi juga berkontribusi terhadap tercapainya tujuan aplikasi dalam menyampaikan informasi yang diperlukan oleh jemaah. Berikut penjelasan dari masing-masing fitur yang tersedia pada aplikasi.

##### 1. Fitur Informasi Kapasitas Parkir

Fitur ini menampilkan jumlah ketersediaan lahan parkir untuk kendaraan mobil dan motor. Melalui fitur ini, jemaah dapat mengetahui apakah area parkir masih tersedia atau sudah penuh sebelum tiba di masjid. Hal ini dapat membantu mereka menyesuaikan waktu keberangkatan dan mengurangi risiko penumpukan kendaraan pada waktu-waktu tertentu, seperti salat Jumat atau ketika ada acara besar di masjid.

##### 2. Fitur Jadwal Salat

Fitur jadwal salat menyediakan informasi waktu salat lima waktu yang dapat diperbarui langsung oleh takmir masjid. Aplikasi ini juga menyediakan notifikasi pengingat yang dapat diaktifkan oleh jemaah untuk memberi tanda menjelang waktu salat. Kehadiran pengingat ini tidak hanya untuk membantu jemaah dalam mempersiapkan ibadah, tetapi juga menjadi pengingat bahwa aplikasi dapat dibuka untuk melihat fitur lain seperti informasi parkir atau kegiatan masjid.

##### 3. Fitur Informasi Acara Masjid

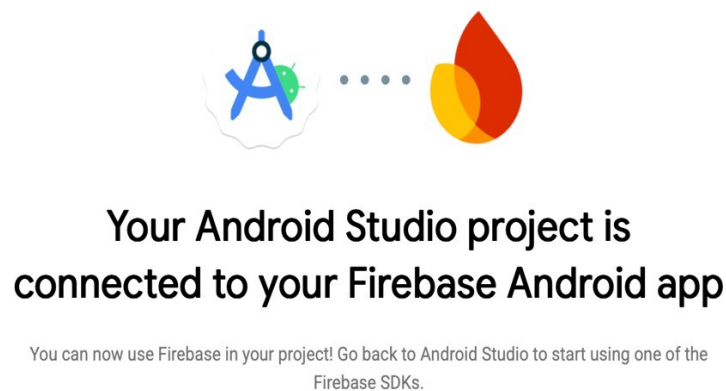
Fitur acara masjid berfungsi untuk menampilkan informasi acara yang akan atau sedang berlangsung. Setiap acara ditampilkan dengan nama, tanggal dan waktu, serta deskripsi singkat agar jemaah dapat memahami isi acaranya dengan mudah. Fitur ini juga dilengkapi dua jenis notifikasi, yaitu notifikasi ketika acara baru ditambahkan oleh takmir dan pengingat yang muncul satu jam sebelum acara dimulai. Kedua jenis notifikasi ini dapat membantu jemaah agar tidak melewatkan kegiatan penting di masjid, sekaligus mengingatkan mereka untuk membuka aplikasi kembali guna memeriksa kondisi parkir sebelum berangkat ke lokasi acara.

## 4.2 Implementasi Sistem

Bagian ini menjelaskan hasil implementasi dari aplikasi sistem parkir masjid yang telah dikembangkan berdasarkan rancangan pada bab sebelumnya. Implementasi dilakukan dengan menerapkan fitur utama yang digunakan oleh takmir masjid untuk melakukan pengelolaan data, serta jemaah sebagai penerima informasi. Setiap halaman aplikasi ditampilkan melalui antarmuka dan fungsi yang telah dibangun untuk menunjukkan bagaimana rancangan yang dibuat sebelumnya diwujudkan dalam bentuk aplikasi yang sebenarnya. Pembahasan dimulai dari integrasi layanan Firebase, kemudian dilanjutkan dengan uraian implementasi pada masing-masing halaman aplikasi.

### 4.2.1 Integrasi Firebase

Pada tahap ini, aplikasi diintegrasikan dengan layanan Firebase sebagai *database* utama untuk mendukung proses autentikasi dan penyimpanan data secara *real-time*. Integrasi dilakukan melalui konfigurasi proyek di *Firebase Console* dan pengaturan pada Android Studio, sehingga aplikasi dapat berkomunikasi langsung dengan layanan Firebase. Setelah proses konfigurasi selesai, status keberhasilan integrasi ditunjukkan melalui tampilan pada Android Studio, seperti terlihat pada gambar berikut.

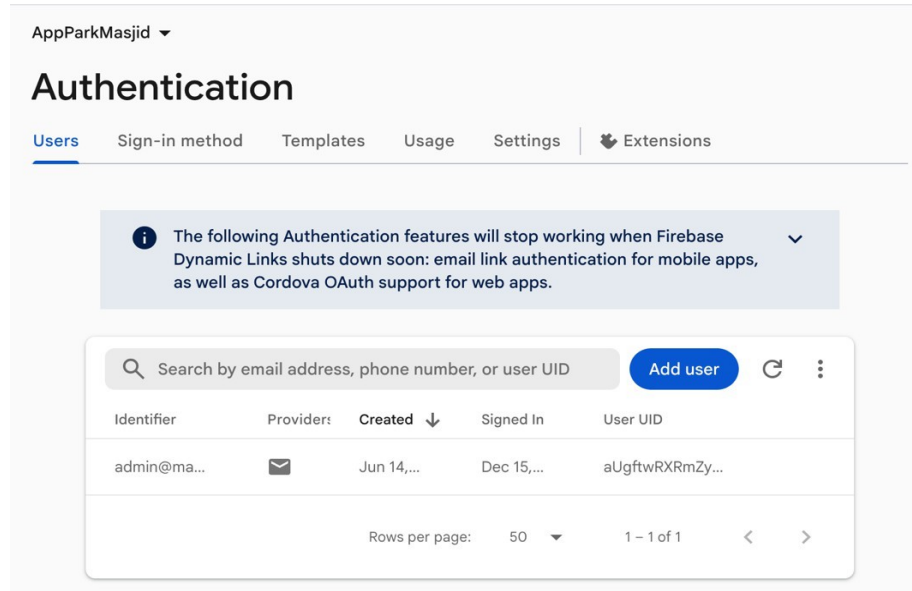


Gambar 4.1 Tampilan integrasi Firebase dengan Android Studio

Tampilan tersebut menunjukkan bahwa *project* Android telah terdaftar di Firebase dan seluruh layanan yang dibutuhkan sudah dapat digunakan. Setelah integrasi berhasil, langkah selanjutnya adalah menerapkan struktur data yang digunakan aplikasi ke dalam Firebase. Bagian berikut menampilkan hasil implementasi struktur database yang menyimpan data operasional aplikasi.

#### A. Struktur Data Akun

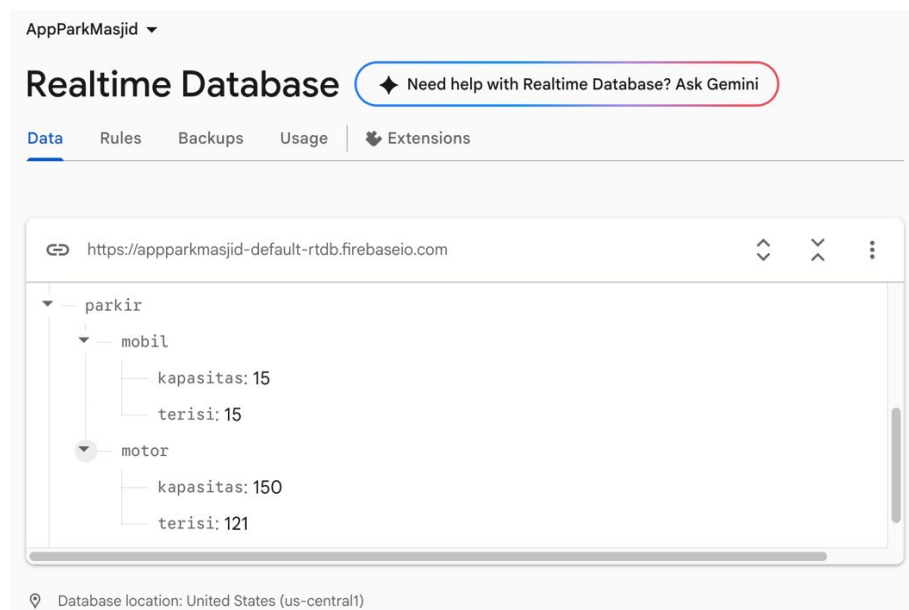
*Firebase Authentication* digunakan sebagai tempat penyimpanan akun takmir masjid yang dikelola oleh aplikasi. Pada layanan ini, akun disimpan dalam bentuk email dan kata sandi yang telah dicatat oleh Firebase. Gambar berikut menampilkan struktur penyimpanan akun yang digunakan dalam aplikasi.



Gambar 4.2 Struktur Data Akun

## B. Struktur Data Parkir

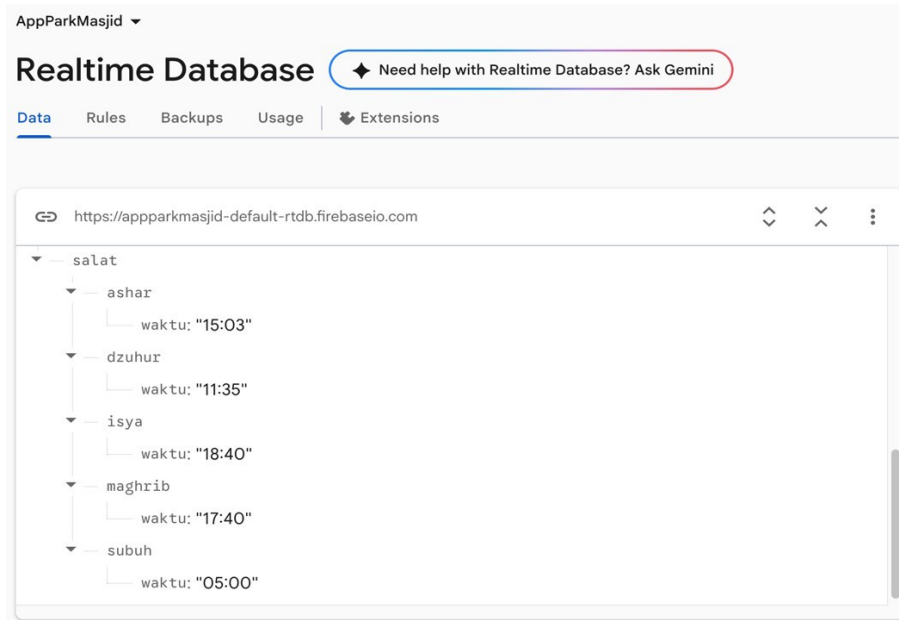
Data parkir direalisasikan pada *node* parkir dalam *Firebase Realtime Database*. *Node* ini memuat informasi terisi dan kapasitas area parkir untuk masing-masing jenis kendaraan. Struktur penyimpanan data parkir dapat dilihat pada Gambar 4.3.



Gambar 4.3 Struktur Data Parkir

### C. Struktur Data Jadwal Salat

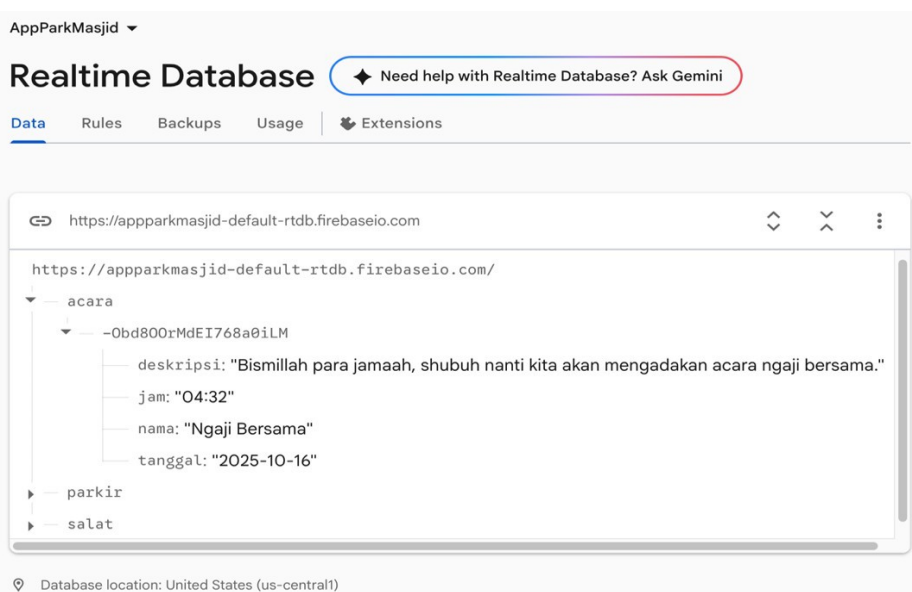
*Firestore Realtime Database* juga digunakan untuk menyimpan informasi jadwal salat. Data disusun dalam *node* jadwal, yang berisi waktu salat harian. Tampilan struktur penyimpanan jadwal salat ditunjukkan pada gambar berikut.



Gambar 4.4 Struktur Data Salat

### D. Struktur Data Acara Masjid

*Node* acara digunakan untuk menyimpan daftar acara masjid, termasuk nama, deskripsi, dan tanggal pelaksanaan. Gambar berikut memperlihatkan bentuk penyimpanan data acara yang digunakan oleh aplikasi.



Gambar 4.5 Struktur Data Acara

### 4.2.2 Mekanisme Notifikasi

Subbab ini membahas mekanisme notifikasi pada aplikasi Sistem Parkir Masjid. Notifikasi pada aplikasi dibagi menjadi dua, yaitu notifikasi acara masjid dan notifikasi jadwal salat. Penjelasan mekanisme untuk masing-masing notifikasi disajikan pada bagian berikut.

#### A. Mekanisme Notifikasi Acara Masjid

Bagian ini menjelaskan mekanisme notifikasi acara pada aplikasi. Pada aplikasi ini, notifikasi acara digunakan untuk dua kebutuhan, yaitu memberikan informasi ketika terdapat acara baru, dan memberikan pengingat (*reminder*) menjelang waktu acara dimulai. Untuk mendukung kebutuhan tersebut, implementasi notifikasi memanfaatkan dua komponen utama, yaitu WorkManager dan AlarmManager.

WorkManager digunakan untuk mengecek data acara secara berkala yang tersimpan di Firebase. Interval pengecekan ditetapkan 15 menit karena merupakan batas minimum pekerjaan periodik pada WorkManager, sehingga aplikasi mengikuti ketentuan tersebut agar proses sinkronisasi tetap berjalan konsisten. Potongan kode berikut menunjukkan pengaturan pengecekan berkala tersebut.

```
val periodic = PeriodicWorkRequestBuilder<EventSyncWorker>(15,
    TimeUnit.MINUTES)
    .setConstraints(
        Constraints.Builder()
            .setRequiredNetworkType(NetworkType.CONNECTED)
            .build()
    )
    .build()

WorkManager.getInstance(context).enqueueUniquePeriodicWork(
    "acara_sync_periodic",
    ExistingPeriodicWorkPolicy.UPDATE,
    periodic
)
```

Gambar 4.6 Kode program penjadwalan WorkManager pengecekan acara

Waktu kemunculan notifikasi diatur melalui dua parameter, yaitu jeda notifikasi acara baru dan selisih waktu *reminder* sebelum acara dimulai. Kedua parameter tersebut ditetapkan sebagai konstanta agar mudah dikelola.

```
private val first_notif = 0
private val reminder_notif = 60
```

Gambar 4.7 Kode program konfigurasi waktu notifikasi acara baru dan *reminder*

Nilai *first\_notif* digunakan untuk menentukan jeda kemunculan notifikasi “acara baru” setelah acara terdeteksi oleh proses pengecekan. Sementara itu, *reminder\_notif* digunakan untuk menentukan kapan *reminder* ditampilkan, yaitu beberapa menit sebelum waktu acara dimulai.

Pada saat WorkManager melakukan pengecekan, aplikasi juga menyimpan daftar id acara yang sudah diproses. Penyimpanan ini dilakukan untuk memastikan satu acara tidak memunculkan notifikasi secara berulang. Jika ditemukan acara yang belum pernah diproses, maka acara tersebut dijadwalkan notifikasinya dan ID-nya dicatat.

```
if (!seen.contains(id)) {
    scheduleAlarm(applicationContext,      data,      initialTrigger,
"initial")
    seen.add(id)
}
```

Gambar 4.8 Kode program pencegahan notifikasi ganda berdasarkan ID acara

Setelah acara terdeteksi dan dijadwalkan, AlarmManager digunakan sebagai pemicu notifikasi sesuai waktu yang sudah dihitung. Penggunaan AlarmManager terutama diperlukan untuk notifikasi *reminder* agar dapat tampil mendekati waktu acara. Berikut adalah potongan kode pemakaian *alarm* dalam penjadwalan notifikasi.

```
alarmManager.setExactAndAllowWhileIdle(
    AlarmManager.RTC_WAKEUP,
    triggerAt,
    pendingIntent
)
```

Gambar 4.9 Kode program penjadwalan notifikasi dengan AlarmManager

Ketika *alarm* dijalankan. *Receiver* menentukan jenis notifikasi yang perlu ditampilkan, yaitu notifikasi acara baru atau notifikasi *reminder*.

```
when (type) {
    "initial" ->
NotificationHelper.showNewEventNotification(context, acara)
    "reminder" ->
NotificationHelper.showReminderNotification(context, acara)
}
```

Gambar 4.10 Kode program pemilihan notifikasi (acara baru/*reminder*)

Pesan notifikasi pada aplikasi ini ditampilkan sesuai teks yang sudah ditetapkan pada masing-masing jenis notifikasi. Potongan kode berikut menunjukkan teks yang digunakan pada notifikasi acara baru dan notifikasi *reminder*.

```
// NoTIFIKASI: Acara Baru Terdeteksi

@SuppressLint("MissingPermission")

fun showNewEventNotification(
    context: Context,
    acara: AcaraModel
) {
    if (!canNotify(context)) return

    val waktu = formatWaktu(acara.tanggal, acara.jam)
    val title = "🔔 Acara Baru: ${acara.nama ?: "Acara"}"
    val shortText = "Jadwal: $waktu"
    val bigText = """
        Assalamu'alaikum Jemaah,

        Ada acara baru: ${acara.nama}

        Waktu: $waktu

        Yuk agendakan dan ajak keluarga atau teman untuk hadir ya
    """.trimIndent()
}

// NoTIFIKASI: Reminder Sebelum Acara Dimulai

@SuppressLint("MissingPermission")

fun showReminderNotification(
    context: Context,
    acara: AcaraModel
) {
```

```

if (!canNotify(context)) return

val waktu = formatWaktu(acara.tanggal, acara.jam)

val title = "🕒 Reminder Acara: ${acara.nama ?: "Acara"}"

val shortText = "Dimulai pada $waktu"

val bigText = """

    Bismillah,

    Acara "${acara.nama}" akan segera dimulai pada:

    $waktu ✨

    """.trimIndent()
}

```

Gambar 4.11 Kode program teks pesan notifikasi acara baru dan *reminder*

## B. Mekanisme Notifikasi Jadwal Salat

Subbab ini menjelaskan mekanisme notifikasi jadwal salat pada aplikasi. Notifikasi jadwal salat dirancang sebagai pengingat harian waktu adzan yang dapat diaktifkan atau dinonaktifkan oleh jemaah melalui halaman jadwal salat. Dalam implementasinya, mekanisme ini memanfaatkan `AlarmManager` agar pengingat dapat dipicu pada waktu yang mendekati jadwal salat.

Pada saat fitur notifikasi adzan diaktifkan, sistem menyimpan status aktif pada penyimpanan lokal dan memulai penjadwalan alarm berdasarkan jadwal salat yang tersedia. Proses ini juga memperhatikan kebutuhan izin notifikasi dan izin alarm pada perangkat, sehingga notifikasi dapat ditampilkan sesuai ketentuan sistem operasi `Android`.

```

private fun activateNotificationsAndSetAlarms() {
    preferenceHelper.setAzanNotificationEnabled(true)
    if (currentJadwalSalat.isNotEmpty()) {
        maybeRequestExactAlarmPermission()
        AlarmUtils.setAlarmForAllSalat(requireContext(),
currentJadwalSalat)
    } else {
        loadJadwalSalat(true)
    }
}
}

```

Gambar 4.12 Kode program aktivasi notifikasi adzan

Jadwal salat dibaca dari Firebase kemudian ditampilkan pada antarmuka pengguna. Setelah jadwal berhasil dimuat, data waktu salat disimpan ke SharedPreferences sebagai data cadangan. Penyimpanan ini diperlukan agar jadwal dapat digunakan kembali untuk penjadwalan alarm dan proses penjadwalan ulang tanpa harus bergantung pada pembacaan ulang data dari Firebase setiap waktu.

```
currentJadwalSalat = mapof(
    "Subuh" to subuhWaktu,
    "Dzuhur" to dzuhurWaktu,
    "Ashar" to asharWaktu,
    "Maghrib" to maghribWaktu,
    "Isya" to isyaWaktu
)
preferenceHelper.saveJadwalSalat(currentJadwalSalat)
```

Gambar 4.13 Kode program penyimpanan jadwal salat ke SharedPreferences

Setelah jadwal tersedia dan notifikasi dalam keadaan aktif, aplikasi menjadwalkan *alarm* untuk masing-masing waktu salat. Penjadwalan dilakukan menggunakan AlarmManager dengan mode *alarm* agar notifikasi dapat ditampilkan mendekati waktu yang ditentukan. Apabila waktu salat pada hari yang sama sudah terlewati, sistem akan menggeser penjadwalan ke hari berikutnya sehingga notifikasi tetap berjalan pada waktu berikutnya.

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    alarmManager.setExactAndAllowWhileIdle(
        AlarmManager.RTC_WAKEUP,
        calendar.timeInMillis,
        pendingIntent
    )
} else {
    alarmManager.setExact(AlarmManager.RTC_WAKEUP,
calendar.timeInMillis, pendingIntent)
}
```

Gambar 4.14 Kode program penjadwalan AlarmManager

Ketika waktu salat tiba, AlarmManager akan memanggil BroadcastReceiver (AlarmReceiver). Receiver ini bertugas menampilkan notifikasi adzan sesuai nama salat yang sedang berlangsung. Pada tahap ini, aplikasi menampilkan informasi adzan

melalui judul dan isi notifikasi, sehingga pengguna memperoleh pengingat untuk melaksanakan ibadah.

```

val n = NotificationCompat.Builder(context, CHANNEL_ID)
    .setSmallIcon(R.drawable.baseline_mosque)
    .setContentTitle("Adzan $salatName Berkumandang")
    .setContentText("Yuk tunaikan Salat $salatName. Cek juga info
parkir masjid di aplikasi.")
    .setPriority(NotificationCompat.PRIORITY_HIGH)
    .setAutoCancel(true)
    .setSound(soundUri)
    .build()

nm.notify(notificationId, n)

```

Gambar 4.15 Kode program tampilan notifikasi adzan

Setelah notifikasi adzan berhasil ditampilkan, aplikasi melakukan penjadwalan ulang secara otomatis untuk hari berikutnya. Penjadwalan ulang ini bertujuan agar pengingat tetap berjalan setiap hari tanpa memerlukan pengaturan ulang dari pengguna, selama fitur notifikasi adzan masih diaktifkan.

```

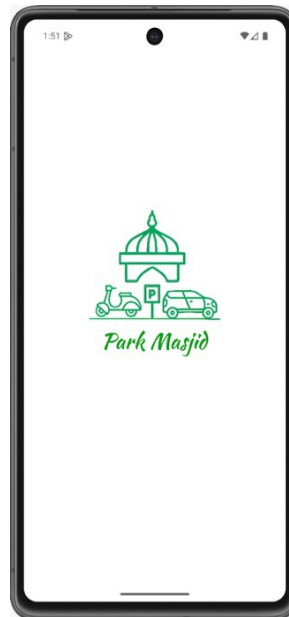
val waktu = pref.getSavedJadwalSalat()[salatName]
if (!waktu.isNullorBlank()) {
    AlarmUtils.scheduleNextDayForoneSalat(context, waktu,
    waktu, salatName,
    waktu)
}

```

Gambar 4.16 Kode program penjadwalan ulang notifikasi

Melalui mekanisme tersebut, notifikasi jadwal salat dapat ditampilkan secara terjadwal sesuai waktu salat, tetap berjalan harian melalui proses penjadwalan ulang selama pengguna masih mengaktifkan fitur notifikasi adzan.

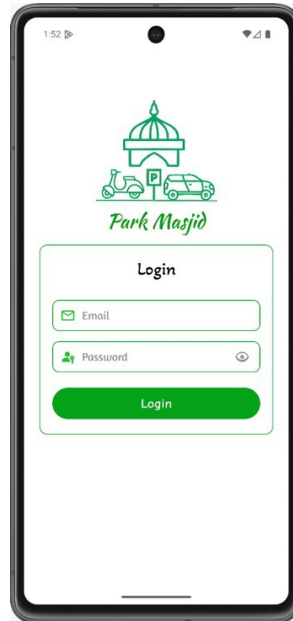
### 4.2.3 Halaman *Splash Screen*



Gambar 4.17 Tampilan *Splash Screen* Aplikasi

Halaman ini berfungsi sebagai tampilan pembuka sebelum pengguna diarahkan ke halaman utama. Pada halaman *splash screen* ditampilkan logo aplikasi yang merupakan kombinasi antara ilustrasi masjid dan simbol parkir. Selain sebagai identitas visual, halaman ini juga memuat mekanisme akses khusus bagi takmir masjid. Dengan mengetuk logo sebanyak tiga kali berturut-turut, takmir akan diarahkan ke halaman login untuk masuk ke fitur pengelolaan. Sementara itu, jemaah akan secara otomatis diarahkan ke halaman beranda tanpa perlu melakukan proses autentikasi.

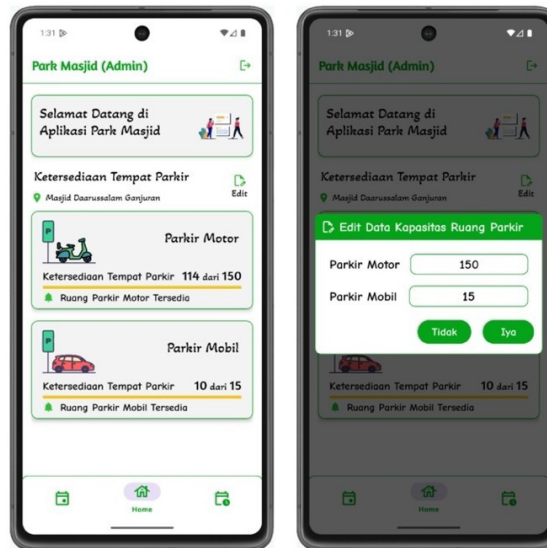
#### 4.2.4 Halaman Login Takmir Masjid



Gambar 4.18 Tampilan Halaman Login Takmir Masjid

Pada halaman ini tersedia dua kolom isian, yaitu email dan kata sandi, yang harus diisi dengan data akun yang valid. Proses autentikasi dilakukan menggunakan *Firestore Authentication* untuk memastikan hanya admin terdaftar yang dapat mengakses fitur pengelolaan. Setelah data login dikirim, sistem akan memverifikasi informasi tersebut di Firestore. Jika sesuai, admin akan diarahkan ke halaman beranda, namun jika salah, aplikasi akan menampilkan pesan kesalahan. Fitur ini memastikan keamanan akses terhadap data dan fungsi di dalam sistem.

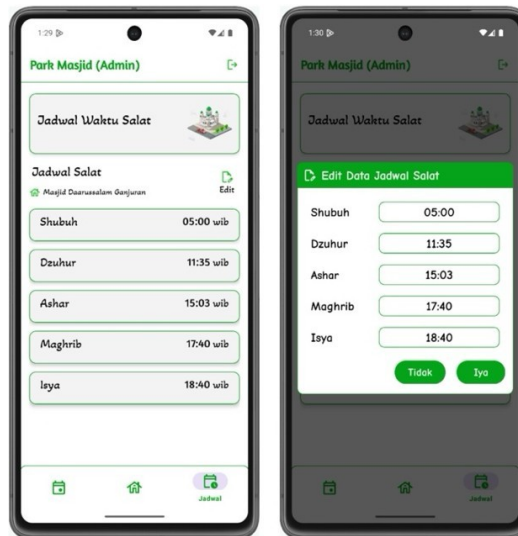
#### 4.2.5 Halaman Beranda Takmir Masjid



Gambar 4.19 Tampilan Halaman Beranda Takmir Masjid

Pada halaman ini, takmir masjid dapat memantau jumlah slot parkir yang tersedia serta jumlah slot yang sedang terpakai untuk area mobil dan motor. Seluruh data terhubung langsung ke *Firestore Realtime Database* dan akan diperbarui secara otomatis setiap terjadi perubahan. Perubahan dapat berupa bertambah atau berkurangnya jumlah kendaraan yang terparkir. Ketika terdapat kendaraan masuk, nilai terisi akan bertambah, sedangkan ketika kendaraan keluar nilai terisi akan berkurang. Pembaruan nilai tersebut menjadi dasar penyesuaian informasi ketersediaan parkir, yaitu slot tersedia dan slot penuh pada aplikasi. Takmir dapat menekan ikon ubah kapasitas untuk menampilkan dialog yang digunakan mengubah jumlah total kapasitas parkir sesuai kondisi lapangan. Setelah data diperbarui, nilai kapasitas pada Firestore otomatis tersinkronisasi dan langsung terlihat juga pada sisi jemaah.

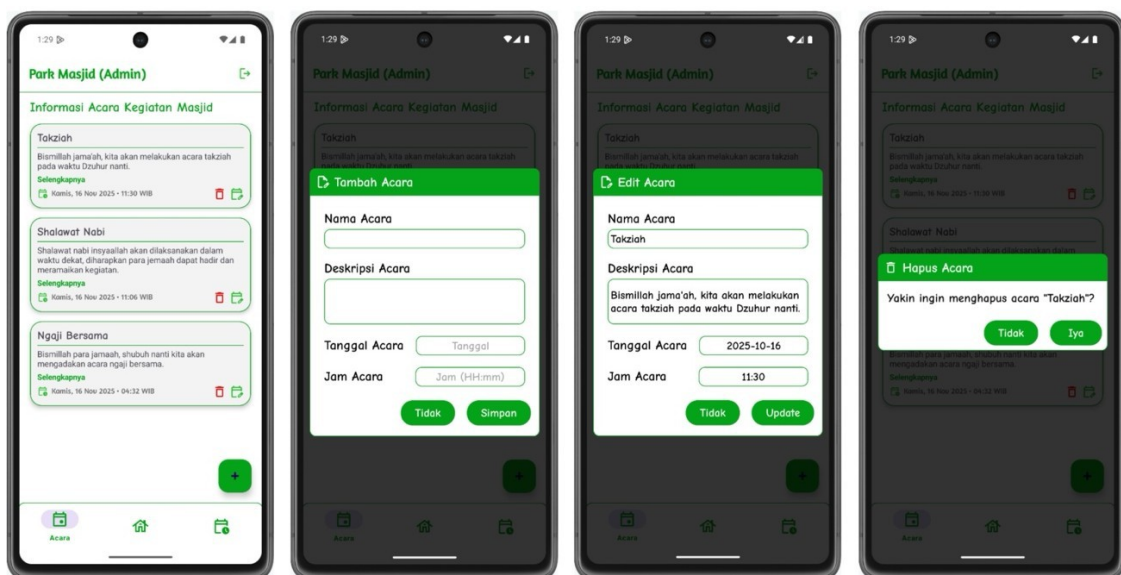
#### 4.2.6 Halaman Jadwal Salat Takmir Masjid



Gambar 4.20 Tampilan Halaman Jadwal Salat Takmir Masjid

Pada halaman ini, terdapat informasi lima waktu salat yang ditampilkan, yaitu Subuh, Dzuhur, Asar, Maghrib, dan Isya. Takmir masjid dapat mengubah jam salat sesuai kondisi terkini melalui tombol ubah jadwal yang akan menampilkan form pengeditan waktu. Setelah perubahan dilakukan, data baru akan otomatis tersimpan di Firebase dan diperbarui pada sisi jemaah secara *real-time*.

#### 4.2.7 Halaman Acara Masjid Takmir Masjid

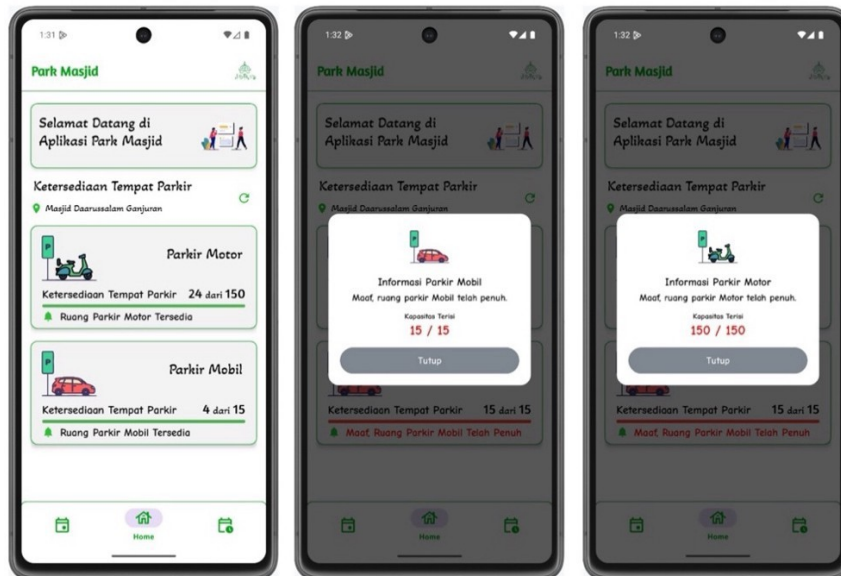


Gambar 4.21 Tampilan Halaman Acara Masjid Takmir Masjid

Halaman ini digunakan untuk mengelola daftar acara masjid, termasuk menambah, mengubah, dan menghapus informasi acara. Setiap acara memuat nama kegiatan, tanggal

pelaksanaan, waktu mulai, dan deskripsi singkat. Selain itu, halaman ini menampilkan daftar seluruh acara agar takmir dapat memantau aktivitas masjid dengan lebih mudah. Seluruh pembaruan yang dilakukan melalui dialog *form* akan tersinkronisasi secara otomatis ke Firebase dan langsung tsmphil pada aplikasi jemaah.

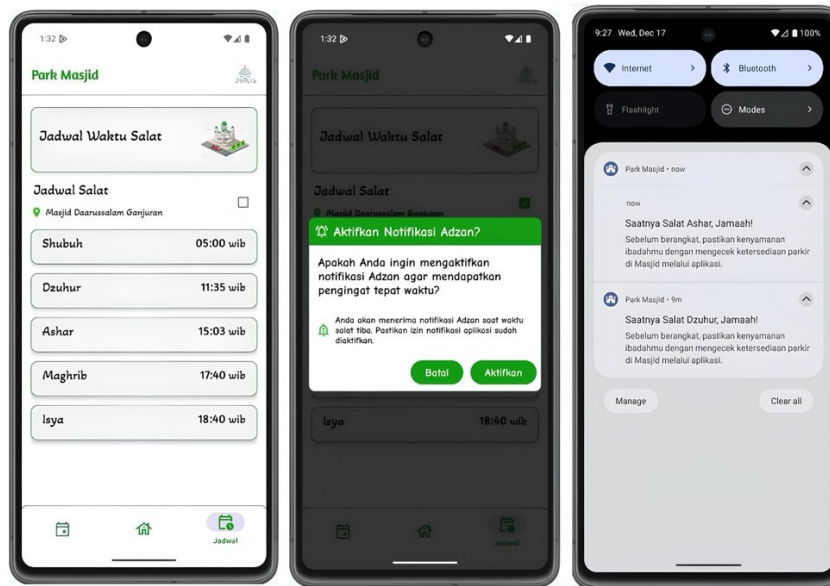
#### 4.2.8 Halaman Beranda Jemaah



Gambar 4.22 Tampilan Halaman Beranda Jemaah

Pada halaman ini, jemaah dapat melihat informasi terkini mengenai ketersediaan lahan parkir kendaraan mobil dan motor. Data yang ditampilkan diperoleh dari integrasi *Firestore Realtime Database*. Informasi ketersediaan akan menyesuaikan perubahan nilai terisi, termasuk ketika kendaraan masuk ataupun keluar. Jika jumlah kendaraan terisi telah mencapai batas kapasitas, secara otomatis akan menampilkan dialog pemberitahuan parkir telah penuh. Selain pembaruan otomatis, jemaah juga dapat memuat tombol *refresh* untuk memperbarui data secara manual.

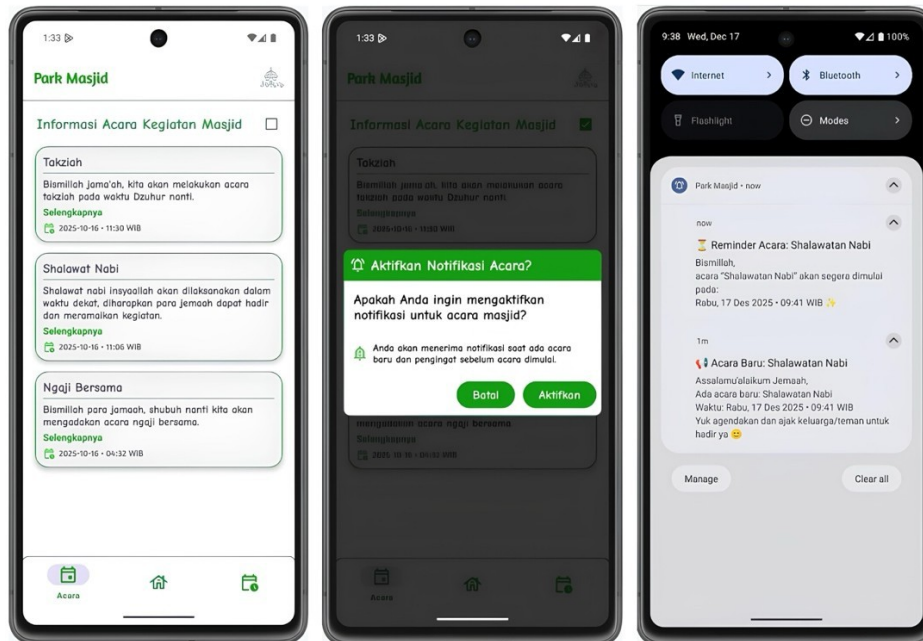
#### 4.2.9 Halaman Jadwal Salat Jemaah



Gambar 4.23 Tampilan Halaman Jadwal Salat Jemaah

Gambar 4.23 menampilkan halaman jadwal salat yang berfungsi untuk memberikan informasi waktu salat lima waktu bagi jemaah. Jadwal ini diperoleh dari Firebase dan akan langsung menyesuaikan setiap kali takmir masjid memperbarui data di sistem. Selain menampilkan jadwal, halaman ini juga dilengkapi dengan fitur pengingat waktu salat. Pengguna dapat mengaktifkan notifikasi azan melalui dialog konfirmasi izin notifikasi yang tampil di dalam aplikasi. Setelah izin diberikan, sistem akan menampilkan pemberitahuan tepat saat waktu salat tiba.

#### 4.2.10 Halaman Acara Masjid Jemaah



Gambar 4.24 Tampilan Halaman Acara Masjid Jemaah

Gambar 4.24 menampilkan daftar acara masjid yang telah dibuat oleh takmir masjid. Setiap kegiatan ditampilkan dengan informasi nama acara, tanggal pelaksanaan, waktu mulai, dan deskripsi singkat untuk memberikan gambaran mengenai isi acara tersebut. Selain melihat daftar kegiatan, jemaah juga dapat mengaktifkan fitur notifikasi acara melalui dialog izin yang disediakan. Setelah diaktifkan, sistem akan mengirimkan dua jenis pemberitahuan, yaitu ketika takmir membuat acara baru dan pengingat satu jam sebelum acara dimulai.

### 4.3 Pengujian Sistem

Bagian ini menjelaskan hasil pengujian yang dilakukan untuk memastikan bahwa aplikasi Sistem Parkir Masjid berfungsi sesuai dengan kebutuhan dan rancangan. Pengujian dilakukan dalam dua tahap, yaitu pengujian fungsional menggunakan metode *black-box testing* serta pengujian tingkat kegunaan (*usability*) menggunakan metode *System Usability Scale* (SUS). Kedua pengujian ini digunakan untuk menilai stabilitas fungsi sistem dan persepsi kenyamanan pengguna saat menggunakan aplikasi.

#### 4.3.1 Pengujian Fungsional (*Black-Box Testing*)

Pengujian fungsional pada aplikasi dilakukan dengan metode *black-box testing* untuk memastikan bahwa setiap fitur mampu menghasilkan keluaran yang sesuai tanpa memeriksa logika internal program. Proses pengujian ini mencakup dua jenis pengguna, yaitu takmir

masjid sebagai admin dan jemaah sebagai pengguna aplikasi, dengan skenario yang disesuaikan berdasarkan fungsi yang dijalankan oleh masing-masing peran. Pada sisi admin, pengujian meliputi proses autentikasi, pembaruan kapasitas parkir, pengeditan jadwal salat, serta pengelolaan data kegiatan masjid. Sementara pada sisi jemaah, pengujian difokuskan pada akses informasi parkir, jadwal salat, kegiatan masjid, serta penerimaan notifikasi. Pemisahan pengujian berdasarkan peran ini dilakukan agar evaluasi fitur lebih terarah dan sesuai ruang lingkup penggunaan. Hasil pengujian untuk kedua peran disajikan pada bagian berikut.

#### A. Pengujian *black-box* untuk takmir masjid (admin)

Pengujian *black-box* pada sisi takmir masjid (admin) dilakukan untuk mengevaluasi hasil pengujian terhadap fungsi-fungsi pengelolaan sistem. Pengujian ini mencakup proses autentikasi, pengelolaan kapasitas parkir, pembaruan jadwal salat, serta pengelolaan data acara masjid. Hasil pengujian untuk sisi takmir masjid disajikan pada Tabel 4.1.

Tabel 4.1 Hasil pengujian *black-box* fitur admin

Halaman	Pengujian	Hasil yang diharapkan	Hasil
<i>Splash screen</i> Admin	Akses halaman admin (tap logo 3x)	Sistem membuka ke halaman login admin.	Sesuai dengan hasil yang diharapkan
Login admin	Login dengan kredensial valid	Admin berhasil masuk ke halaman <i>dashboard</i> admin.	Sesuai dengan hasil yang diharapkan
	Login dengan kredensial salah	Sistem menampilkan pesan <i>toast</i> gagal login.	Sesuai dengan hasil yang diharapkan
	Login dengan <i>form</i> kosong	Sistem menampilkan pesan <i>toast</i> “ <i>email</i> dan <i>password</i> harus diisi”.	Sesuai dengan hasil yang diharapkan
<i>Dashboard</i> admin	<i>Logout</i> admin	Sistem kembali ke halaman <i>splash screen</i> .	Sesuai dengan hasil yang diharapkan

Halaman	Pengujian	Hasil yang diharapkan	Hasil
Beranda admin	Menampilkan data kapasitas dan ketersediaan parkir	Sistem menampilkan data terisi dan kapasitas total parkir untuk motor dan mobil.	Sesuai dengan hasil yang diharapkan
	Mengubah kapasitas total parkir	Sistem menyimpan perubahan kapasitas total parkir, dan memperbarui data tampilan parkir.	Sesuai dengan hasil yang diharapkan
	Sinkronisasi perubahan kapasitas ke sisi jemaah	Perubahan nilai kapasitas tercermin pada tampilan parkir sisi jemaah secara sinkron	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> , dan gagal menyimpan perubahan.	Sesuai dengan hasil yang diharapkan
Jadwal salat admin	Menampilkan jadwal salat	Sistem menampilkan data jadwal salat.	Sesuai dengan hasil yang diharapkan
	Mengubah jadwal salat	Data jadwal salat tersimpan dan diperbarui.	Sesuai dengan hasil yang diharapkan
	Mengubah lebih dari satu jadwal salat	Sistem menyimpan seluruh perubahan secara bersamaan.	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> , dan gagal menyimpan perubahan.	Sesuai dengan hasil yang diharapkan
	Menampilkan daftar acara	Sistem menampilkan daftar acara masjid.	Sesuai dengan hasil yang diharapkan

Halaman	Pengujian	Hasil yang diharapkan	Hasil
Acara masjid admin	Menambah acara	Sistem menyimpan data acara baru, dan menampilkannya pada daftar acara.	Sesuai dengan hasil yang diharapkan
	Validasi salah satu <i>field</i> ada yang kosong	Sistem gagal menyimpan dan menampilkan pesan <i>toast</i> “mohon lengkapi semua <i>field</i> ”.	Sesuai dengan hasil yang diharapkan
	Mengedit acara	Data acara pada salah satu item berhasil diperbarui.	Sesuai dengan hasil yang diharapkan
	Menghapus acara	Data acara pada salah satu item berhasil dihapus.	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem gagal menyimpan data acara.	Sesuai dengan hasil yang diharapkan

## B. Pengujian *black-box* untuk jemaah

Pengujian *black-box* pada sisi jemaah dilakukan untuk mengevaluasi kesesuaian fungsi sistem dalam mendukung penggunaan aplikasi oleh pengguna. Pengujian difokuskan pada akses informasi ketersediaan parkir, jadwal salat, agenda acara masjid, serta penerimaan notifikasi. Hasil pengujian untuk sisi jemaah disajikan pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian *black-box* fitur jemaah

Halaman	Pengujian	Hasil yang diharapkan	Hasil
<i>Splash screen</i> jemaah	Jemaah membuka aplikasi	Sistem langsung menampilkan ke halaman sisi jemaah.	Sesuai dengan hasil yang diharapkan
Beranda jemaah	Menampilkan data ketersediaan parkir	Sistem menampilkan data terisi dan kapasitas ketersediaan parkir.	Sesuai dengan hasil yang diharapkan

Halaman	Pengujian	Hasil yang diharapkan	Hasil
	Melakukan <i>refresh</i> data parkir	Sistem mengambil data parkir terbaru.	Sesuai dengan hasil yang diharapkan
	Perubahan data jumlah kapasitas parkir dari admin	Data kapasitas parkir berubah sesuai update dari admin.	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai <i>default</i> untuk data parkir.	Sesuai dengan hasil yang diharapkan
Jadwal salat jemaah	Menampilkan jadwal salat	Sistem menampilkan data jadwal salat.	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem menampilkan nilai default pada jadwal salat.	Sesuai dengan hasil yang diharapkan
	Mengaktifkan notifikasi salat	Notifikasi salat diaktifkan.	Sesuai dengan hasil yang diharapkan
	Menonaktifkan notifikasi salat	Notifikasi salat dinonaktifkan.	Sesuai dengan hasil yang diharapkan
	Penerimaan notifikasi salat	Sistem menampilkan notifikasi azan sesuai jadwal salat.	Sesuai dengan hasil yang diharapkan
Acara masjid jemaah	Menampilkan daftar data acara masjid	Sistem menampilkan daftar acara masjid.	Sesuai dengan hasil yang diharapkan
	Mengaktifkan notifikasi acara masjid	Notifikasi acara masjid diaktifkan.	Sesuai dengan hasil yang diharapkan

Halaman	Pengujian	Hasil yang diharapkan	Hasil
	Menonaktifkan notifikasi acara masjid	Notifikasi acara masjid dinonaktifkan.	Sesuai dengan hasil yang diharapkan
	Notifikasi ketika acara masjid baru dibuat oleh admin	Sistem menampilkan notifikasi acara masjid.	Sesuai dengan hasil yang diharapkan
	Notifikasi pengingat satu jam sebelum waktu acara	Sistem menampilkan notifikasi acara masjid sekitar 1 jam sebelum waktu acara.	Sesuai dengan hasil yang diharapkan
	Perilaku saat <i>offline</i>	Sistem tidak menampilkan daftar acara masjid.	Sesuai dengan hasil yang diharapkan

Berdasarkan hasil pengujian *black-box* yang telah dilakukan, dapat disimpulkan bahwa seluruh fitur utama pada aplikasi telah berfungsi sesuai dengan kebutuhan sistem dan berjalan tanpa kendala yang signifikan. Pada sisi takmir masjid (admin), pengujian menunjukkan bahwa fitur autentikasi, pengaturan kapasitas parkir, pembaruan jadwal salat, serta pengelolaan data acara masjid telah terimplementasikan dengan baik sesuai perancangan. Proses pengelolaan data, yang meliputi penambahan, pengubahan, dan penghapusan acara, menghasilkan respon sistem yang konsisten, termasuk pembaruan data pada *Firestore Realtime Database*, sehingga fungsi pada sisi admin dinilai telah memenuhi persyaratan kebutuhan fungsional sistem.

Pada sisi jemaah, hasil pengujian menunjukkan bahwa fitur seperti informasi kapasitas parkir, jadwal salat, dan daftar acara masjid mampu beroperasi secara responsif sesuai skenario pengujian. Temuan spesifik teridentifikasi pada mekanisme notifikasi acara yang kinerjanya beradaptasi dengan status aktif pada aplikasi. Dalam kondisi aplikasi terbuka, notifikasi dapat ditampilkan secara *real-time* melalui *Firestore Realtime Listener*, sedangkan pada kondisi tertutup sistem beralih menggunakan kombinasi *WorkManager* dan *AlarmManager* yang menyebabkan informasi tidak tampil secara *real-time*. Akibat mekanisme pada proses latar belakang tersebut, terjadi latensi pengiriman notifikasi mulai dari 15 hingga 45 menit dengan pola keterlambatan dominan sekitar 30 sampai 40 menit pada perangkat dengan mode hemat daya. Meskipun demikian, proses pengiriman notifikasi tetap berjalan sesuai mekanisme

penjadwalan *WorkManager*, sehingga fitur ini dinilai telah memenuhi kriteria keberhasilan fungsional dan tidak mengganggu operasional aplikasi secara keseluruhan.

#### 4.3.2 Pengujian Pengguna Menggunakan *System Usability Scale (SUS)*

Pengujian terhadap aspek *usability* aplikasi dilakukan menggunakan metode *System Usability Scale (SUS)*. Metode ini dipilih karena mampu memberikan pengukuran *usability* secara ringkas melalui sepuluh pernyataan dengan skala penilaian *Likert* 1 hingga 5. Pengujian ini melibatkan 25 responden yang terdiri dari 23 jemaah dan 2 takmir masjid. Para responden diminta memberikan penilaian setelah melihat dan mencoba menggunakan aplikasi. Rekapitulasi hasil penilaian SUS ditampilkan pada Tabel 4.3 berikut.

Tabel 4.3 Rekapitulasi Jawaban *System Usability Scale (SUS)*

Responden	Pertanyaan										Total Skor Kontribusi	Skor (Total x 2,5)
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
R1	4	2	5	2	4	2	5	1	4	3	32	80,0
R2	4	2	4	2	4	2	4	2	4	2	30	75,0
R3	4	2	4	2	4	2	5	2	4	2	31	77,5
R4	3	1	5	1	4	1	5	1	5	2	36	90,0
R5	4	2	4	2	4	2	4	2	4	5	27	67,5
R6	4	2	4	2	4	2	4	2	4	2	30	75,0
R7	5	1	5	1	5	1	5	1	5	1	40	100,0
R8	4	3	4	3	4	3	4	2	4	2	27	67,5
R9	5	1	5	1	5	1	5	5	5	1	36	90,0
R10	5	2	4	2	4	1	4	2	3	3	30	75,0
R11	5	2	5	3	4	2	5	2	4	3	31	77,5
R12	4	1	5	2	4	2	5	1	5	2	35	87,5
R13	5	1	5	1	5	1	5	1	5	3	38	95,0
R14	4	3	5	3	4	1	5	1	5	3	32	80,0
R15	5	2	5	2	5	2	4	2	5	4	32	80,0
R16	4	2	4	2	4	2	4	2	4	2	30	75,0
R17	4	2	5	2	4	2	5	1	4	2	33	82,5
R18	5	1	5	2	5	2	5	1	5	2	37	92,5

Responden	Pertanyaan										Total Skor Kontribusi	Skor (Total x 2,5)
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
R19	4	2	4	2	4	2	4	4	4	2	28	70,0
R20	4	1	5	2	5	1	5	1	5	2	37	92,5
R21	3	1	5	1	5	1	1	1	5	1	34	85,0
R22	5	2	4	4	5	2	4	1	5	4	30	75,0
R23	4	1	5	1	5	1	4	1	4	1	37	92,5
R24	5	2	4	5	5	4	5	1	5	1	31	77,5
R25	4	2	4	2	5	2	4	1	4	5	29	72,5
Rata-rata												81,30

Berdasarkan hasil pengujian *usability* menggunakan metode *System Usability Scale* (SUS), aplikasi memperoleh skor akhir sebesar 81,30 yang menempatkan aplikasi dalam kategori *excellent*, sehingga menunjukkan tingkat kemudahan penggunaan yang baik. Nilai ini mencerminkan bahwa aplikasi dapat dipahami dan dioperasikan dengan mudah oleh pengguna, baik takmir masjid maupun jemaah, serta memungkinkan akses fitur-fitur utama secara efisien tanpa proses yang kompleks. Dengan capaian tersebut, aplikasi dapat dinilai telah memenuhi aspek *usability* dan siap digunakan sebagai media pendukung penyampaian informasi parkir, jadwal salat, dan acara masjid dalam lingkungan operasional masjid.

#### 4.3.3 Evaluasi dan *Feedback* Pengguna

Pada tahapan evaluasi usabilitas menggunakan metode *System Usability Scale* (SUS), responden yang terdiri dari perwakilan Takmir dan Jemaah diminta untuk mengoperasikan aplikasi secara langsung guna memberikan penilaian. Selama proses pengujian berlangsung, responden tidak hanya mengisi kuesioner, tetapi juga memberikan tanggapan secara spontan mengenai pengalaman penggunaan aplikasi. Selain itu, ditemukan pula beberapa catatan teknis selama proses pengujian fungsional berlangsung. Berikut adalah rangkuman evaluasi mengenai kelebihan dan kendala sistem yang ditemukan selama proses pengujian.

##### A. Kelebihan Sistem

Berdasarkan respon yang diberikan saat pengujian, terdapat beberapa aspek positif yang dinilai memberikan nilai tambah signifikan bagi pengguna.

##### 1. Antarmuka yang mudah dipahami

Tampilan aplikasi dinilai memiliki desain yang sederhana dan ringkas. Struktur navigasi yang disajikan cukup jelas, sehingga memudahkan pengguna untuk memahami fungsi utama aplikasi secara cepat. Kesederhanaan visual ini membuat pengguna dapat langsung mengakses informasi pada aplikasi tanpa mengalami kebingungan dalam pengoperasian awal.

## 2. Fungsionalitas Informasi Ketersediaan Parkir

Keberadaan fitur yang menampilkan status ketersediaan lahan parkir dinilai memberikan manfaat fungsional bagi pengguna. Responden memandang bahwa informasi status “Penuh” atau “Tersedia” yang tersaji pada aplikasi dapat menjadi acuan dalam mendukung pengambilan keputusan jemaah sebelum tiba di lokasi. Meskipun kondisi fisik area parkir masih memiliki tantangan terkait ketidakteraturan posisi kendaraan, penyajian informasi status pada aplikasi ini tetap dinilai mampu memberikan gambaran awal mengenai kondisi kepadatan area parkir, khususnya saat berlangsungnya kegiatan besar di masjid.

## 3. Integrasi informasi masjid

Keberadaan fitur tambahan berupa informasi kegiatan masjid dan jadwal salat mendapatkan apresiasi positif. Pihak takmir menilai fitur ini efektif sebagai alternatif media penyebaran informasi *digital* pengganti papan pengumuman fisik, sehingga informasi acara masjid dapat diterima oleh jemaah secara lebih cepat dan *personal* melalui perangkat masing-masing.

## 4. Efektifitas fitur notifikasi

Implementasi fitur notifikasi sebagai pengingat waktu salat dan penyampai informasi acara masjid mendapatkan apresiasi yang positif. Pengguna menilai fitur ini strategis dalam meningkatkan kesadaran terhadap agenda masjid. Lebih lanjut, fitur ini menjaga relevansi aplikasi agar tetap bermanfaat digunakan setiap hari sebagai pengingat ibadah, meskipun pengguna sedang tidak membutuhkan layanan parkir.

## B. Kendala dan evaluasi sistem

Selain aspek positif, ditemukan pula beberapa catatan evaluasi yang berasal dari masukan pengguna serta temuan teknis selama pengujian fungsional.

### 1. Keterbatasan lingkup implementasi

Dari perspektif jemaah, utilitas aplikasi dinilai memiliki batasan karena fungsionalitasnya masih dirancang spesifik hanya untuk lingkungan Masjid Daarussalam Ganjuran saja. Hal ini berpotensi mengurangi minat penggunaan berkelanjutan, terutama ketika jemaah sedang melaksanakan ibadah di lokasi lain yang belum terjangkau oleh sistem.

#### 2. Efisiensi penyimpanan dan aksesibilitas

Pihak Takmir menyoroti kendala teknis terkait kapasitas perangkat pengguna. Tidak semua jemaah memiliki perangkat dengan spesifikasi yang mumpuni untuk mengakomodasi instalasi aplikasi tambahan. Hambatan ini dipicu oleh variasi kondisi perangkat jemaah, mulai dari faktor usia perangkat, hingga kondisi ruang penyimpanan internal (*storage*) yang sudah penuh. Oleh karena itu, disarankan pengembangan versi berbasis *Website* sebagai solusi alternatif yang lebih ringan dan mudah diakses tanpa membebani sumber daya perangkat.

#### 3. Mekanisme pembaruan jadwal salat

Sistem saat ini masih dirancang menggunakan mekanisme pembaruan jadwal salat secara manual oleh admin. Saat proses pengujian dilakukan, pengguna memberikan masukan bahwa metode ini sangat bergantung pada kedisiplinan takmir masjid dalam memperbarui data. Apabila admin terlambat melakukan pembaruan, terdapat risiko ketidaksesuaian antara jadwal yang tercantum di aplikasi dengan waktu salat terkini.

#### 4. Latensi pada notifikasi acara

Berdasarkan hasil pengujian fungsional terhadap fitur notifikasi acara, terindikasi adanya jeda waktu (*delay*) pada penerimaan notifikasi di perangkat pengguna. Hal ini disebabkan oleh mekanisme *WorkManager* dan *AlarmManager* pada sistem operasi Android yang menyelaraskan pengiriman notifikasi dengan manajemen daya perangkat, sehingga notifikasi tidak selalu diterima secara *real-time*. Adapun pada saat aplikasi aktif, pembaruan informasi acara dapat berlangsung lebih responsif karena aplikasi memantau perubahan data secara langsung.

### 4.4 Pembahasan

Berdasarkan rangkaian pengujian yang telah dilaksanakan, pengembangan aplikasi sistem parkir berbasis *mobile* pada penelitian ini terbukti mampu merealisasikan kebutuhan fungsional sesuai dengan tujuan perancangan awal. Secara garis besar, hasil pengujian

fungsional memvalidasi bahwa seluruh fitur esensial, baik pada antarmuka takmir masjid (admin) maupun jemaah, dapat beroperasi dengan stabil. Hal ini mengindikasikan adanya keselarasan antara identifikasi kebutuhan pengguna dengan implementasi teknis sistem, di mana pembagian peran antara takmir dan jemaah berhasil menciptakan ekosistem informasi yang terpusat.

Selaras dengan keberhasilan fungsi utama tersebut, hasil pengujian juga menyoroti karakteristik pada mekanisme pengiriman notifikasi acara. Temuan menunjukkan bahwa ketepatan waktu penerimaan notifikasi dipengaruhi oleh status aktivitas aplikasi. Meskipun notifikasi dapat diterima secara *real-time* saat aplikasi aktif, teridentifikasi adanya latensi pengiriman ketika aplikasi berada dalam kondisi tertutup akibat mekanisme penjadwalan latar belakang pada sistem operasi perangkat. Walaupun demikian, keterlambatan ini dinilai masih berada dalam batas toleransi fungsional karena notifikasi tetap berhasil tersampaikan dan diterima oleh pengguna.

Karakteristik tersebut kemudian dapat dipahami melalui cara aplikasi memproses pembaruan data pada dua kondisi operasional. Pada saat aplikasi berada dalam kondisi aktif, pembaruan data acara dapat terbaca lebih cepat karena aplikasi memantau perubahan data Firebase secara langsung melalui mekanisme *listener*, sehingga informasi acara baru dapat segera diproses dan ditampilkan. Sebaliknya, ketika aplikasi berada dalam kondisi tertutup, proses pemantauan langsung tidak berjalan karena aplikasi tidak berada dalam keadaan aktif. Pada kondisi ini, sistem mengandalkan mekanisme pengecekan berkala yang dijalankan di latar belakang melalui *WorkManager*. Interval pengecekan tersebut mengikuti ketentuan periodik minimum, yaitu 15 menit, sehingga pendeteksian acara baru pada kondisi aplikasi tertutup tidak dapat berlangsung seketika. Selanjutnya, setelah acara terdeteksi pada siklus pengecekan, pemicu notifikasi dijadwalkan menggunakan *AlarmManager*. Oleh karena itu, notifikasi acara pada kondisi tertutup tidak selalu tersampaikan secara *real-time* karena mengikuti siklus pengecekan serta penyesuaian eksekusi oleh sistem operasi, terutama pada perangkat yang menerapkan penghematan daya.

Sejalan dengan pembahasan tersebut, pemilihan mekanisme notifikasi dalam penelitian ini juga mempertimbangkan aspek implementasi dan pengelolaan layanan. Penelitian ini tidak menerapkan *push notification* berbasis Firebase Cloud Messaging (FCM) sebagai mekanisme utama. Hal ini disebabkan karena mekanisme *push* yang sepenuhnya otomatis umumnya membutuhkan komponen pengirim dari sisi server, misalnya layanan *backend* atau fungsi

*cloud* yang memantau perubahan data dan memicu pengiriman notifikasi ke perangkat pengguna. Dalam konteks penelitian ini, pendekatan tersebut tidak dipilih karena menambah kebutuhan pengelolaan layanan di luar aplikasi, termasuk potensi keterkaitan dengan skema penagihan pada layanan *cloud* apabila penggunaan melampaui batas tertentu. Oleh karena itu, penelitian ini memilih pendekatan yang berjalan pada sisi perangkat pengguna, yaitu kombinasi *WorkManager* untuk pengecekan berkala dan *AlarmManager* untuk pemicu notifikasi terjadwal. Konsekuensi dari pendekatan tersebut adalah notifikasi acara tidak selalu bersifat *real-time* ketika aplikasi tertutup, namun secara fungsional notifikasi tetap dapat diterima pengguna dan tidak mengganggu operasional aplikasi secara keseluruhan.

Dari aspek usability, evaluasi menggunakan metode *System Usability Scale* (SUS) menghasilkan skor akhir sebesar 81,30, yang menempatkan aplikasi dalam kategori *excellent*. Capaian angka ini sejalan dengan hasil evaluasi *feedback*, di mana pengguna menilai aplikasi memiliki antarmuka yang sederhana serta navigasi yang mudah dipahami. Desain antarmuka yang intuitif ini terbukti efektif mengakomodasi kebutuhan interaksi dari berbagai kalangan pengguna di lingkungan masjid tanpa mengalami kendala berarti saat pengoperasian awal. Selain itu, integrasi fitur informasi jadwal salat dan kegiatan masjid dinilai memberikan nilai tambah signifikan, menjadikan aplikasi tidak hanya berfungsi sebagai alat pemantauan parkir, tetapi juga sebagai media informasi harian jemaah.

Namun, pembahasan ini juga mencatat tantangan implementasi yang berkaitan dengan kondisi perangkat pengguna. Masukan dari pihak takmir menyoroti perlunya alternatif akses berbasis *Website* untuk masa mendatang. Hal ini ditujukan secara spesifik untuk mengakomodasi perangkat jemaah yang memiliki keterbatasan ruang penyimpanan (*storage* penuh) atau perangkat model lama yang mungkin mengalami kendala jika harus menginstal aplikasi tambahan. Selain itu, mekanisme pengelolaan jadwal salat yang masih berbasis input manual menjadi perhatian tersendiri. Ketergantungan ini berpotensi menimbulkan ketidaksesuaian antara informasi waktu ibadah yang tampil di aplikasi dengan kondisi terkini apabila admin tidak melakukan pembaruan data secara disiplin.

Secara keseluruhan, dapat disimpulkan bahwa sistem parkir yang dikembangkan telah memiliki kinerja fungsional yang andal serta tingkat penerimaan pengguna (*usability*) yang masuk dalam kategori *excellent*. Kendala teknis maupun operasional yang ditemukan selama penelitian ini tidak mengurangi esensi keberhasilan sistem, melainkan menjadi temuan yang dapat dijadikan landasan strategis untuk pengembangan di masa mendatang.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan seluruh tahapan perancangan, implementasi, dan pengujian yang telah dilakukan, penelitian ini berfokus pada pengembangan aplikasi parkir masjid berbasis *mobile* sebagai media penyajian informasi parkir dan informasi pendukung masjid. Adapun kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut.

1. Penelitian ini berhasil mengembangkan aplikasi sistem informasi parkir masjid berbasis *mobile* yang dapat digunakan sebagai sarana penyampaian informasi bagi jemaah serta sebagai alat pendukung pengelolaan informasi oleh takmir masjid.
2. Aplikasi menyediakan tampilan informasi status ketersediaan parkir untuk kendaraan mobil dan motor yang dihitung berdasarkan perbandingan antara jumlah kapasitas parkir dan jumlah kendaraan terisi, sebagai bentuk penyajian informasi parkir pada sisi aplikasi.
3. Selain fitur utama tentang ketersediaan parkir, aplikasi juga dilengkapi dengan fitur pendukung berupa informasi jadwal salat dan agenda acara masjid yang disertai dengan notifikasi azan dan notifikasi acara.
4. Hasil pengujian fungsional menggunakan metode *black-box testing* menunjukkan bahwa seluruh fitur utama aplikais telah berjalan sesuai dengan kebutuhan sistem yang dirancang, termasuk mekanisme penjadwalan notifikasi.
5. Berdasarkan hasil evaluasi *usability* menggunakan *system usability scale* (SUS), aplikasi memperoleh skor 81,30 yang termasuk dalam kategori excellent, sehingga dapat disimpulkan bahwa aplikasi memiliki tingkat kemudahan penggunaan yang baik dan dapat diterima oleh pengguna.

#### 5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ada, beberapa saran untuk pengembangan selanjutnya adalah sebagai berikut.

1. Integrasi Sistem dengan Perangkat IoT  
Penelitian selanjutnya diharapkan dapat mengintegrasikan aplikasi dengan perangkat IoT, seperti sensor parkir, agar data ketersediaan parkir dapat diperoleh secara otomatis dan merepresentasikan kondisi lapangan secara lebih akurat.

## 2. Pengembangan Sistem Multi Masjid

Aplikasi dapat dikembangkan lebih lanjut agar mendukung penggunaan data lebih dari satu masjid dalam satu sistem, sehingga sistem menjadi lebih fleksibel dan dapat diterapkan pada cakupan yang lebih luas.

## 3. Alternatif Akses Berbasis *Web*

Untuk mengakomodasi jemaah yang memiliki keterbatasan perangkat atau tidak memungkinkan melakukan instalasi aplikasi tambahan, disarankan pengembangan versi berbasis *web* sebagai alternatif akses agar jemaah tetap dapat mengakses dan menggunakan sistem informasi masjid melalui peramban tanpa memerlukan instalasi aplikasi tambahan.

## 4. Otomatisasi Jadwal Salat

Disarankan agar sistem jadwal salat dapat dikembangkan dengan mekanisme pembaruan data otomatis melalui integrasi dengan sumber data resmi, sehingga mengurangi ketergantungan pada input manual dan meningkatkan akurasi informasi.

## 5. Penyempurnaan Mekanisme Notifikasi

Sistem notifikasi pada aplikasi ini masih mengalami jeda waktu dalam pengirimannya. Oleh karena itu, pada pengembangan selanjutnya disarankan dilakukan penyempurnaan atau penerapan mekanisme notifikasi yang lebih responsif, sehingga notifikasi dapat diterima oleh pengguna secara lebih tepat waktu tanpa keterlambatan yang signifikan.

## DAFTAR PUSTAKA

- Adiswara, D. R., & Irianto, K. D. (2024). Perancangan UI/UX Aplikasi GoMasjid Menggunakan Pendekatan Human Centered Design. *Technologia : Jurnal Ilmiah*, 15(3). <https://doi.org/https://dx.doi.org/10.31602/tji.v15i3.15565>
- Alfikri, H., & Irianto, K. D. (2024). IMPLEMENTASI INTERNET OF THINGS PADA SISTEM PARKIR MASJID. *EDUSAINTEK: Jurnal Pendidikan, Sains Dan Teknologi*, 11(4), 1907–1920. <https://doi.org/https://doi.org/10.47668/edusaintek.v11i4.1323>
- Andriyani, R., & Sari, A. O. (2023). PENERAPAN METODE SYSTEM USABILITY SCALE UNTUK EVALUASI SISTEM INFORMASI E-PROCUREMENT. *Jurnal Multi Media Dan IT*, 7(2). <https://doi.org/10.46961/jommit.v7i2>
- Android Studio. (2024). *Mengenal Android Studio*. Android Studio. <https://developer.android.com/studio/intro>
- Aura Arasy, S., Shafiyatun Nuha, F., Awalia, N., Meiliyani, K., & Setiadi, T. (2024). Analisis dan Perancangan Sistem Smart Parking Berbasis Mobile. *Jurnal Teknik*, 18(2), 763–774. <https://doi.org/https://doi.org/10.5281/zenodo.13318152>
- Badan Pusat Statistik. (2023). *Jumlah Kendaraan Bermotor Menurut Provinsi dan Jenis Kendaraan (Unit)*. BPS–Statistics Indonesia. <https://www.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTIU1bVNFOTVVbmQyVURSTVFUMDkjMyMwMDAw/jumlah-kendaraan-bermotor-menurut-provinsi-dan-jenis-kendaraan-unit.html?year=2023>
- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6), 574–594. <https://doi.org/10.1080/10447310802205776>
- Bose, S., Kundu, Aditi, Mukherjee, M., & Banerjee, M. (2018). A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT. *International Journal of Advanced Research in Computer Science*, 9(3), 41–45. <https://doi.org/10.26483/ijarcs.v9i3.5978>
- Chougale, P., Yadav, V., & Gaikwad, A. (2021). FIREBASE-OVERVIEW AND USAGE. *International Research Journal of Modernization in Engineering Technology and Science*, 3(1).
- Devani, A., Nugroho, S., Jannah, atul, Hartami Santi, I., Balitar Jl Majapahit No, I., Blitar, K., & Timur, J. (2024). PERANCANGAN SISTEM INFORMASI TEMPAT PARKIR

- BERBASIS APLIKASI WEB DI UNIVERSITAS ISLAM BALITAR. *Jurnal Mahasiswa Teknik Informatika*, 8(3), 4329–4335.  
<https://doi.org/https://doi.org/10.36040/jati.v8i3.9894>
- Dharwiyanti, S., & Wahono, R. S. (2003). *Pengantar Unified Modeling Language (UML)*.  
[https://www.academia.edu/download/50995520/Modul\\_UML.pdf](https://www.academia.edu/download/50995520/Modul_UML.pdf)
- Gahara, L. A. N. G., & Irianto, K. D. (2025). Mosque Parking System: An Implementation Using Camera and IoT. *International Journal Software Engineering and Computer Science (IJSECS)*, 5(2), 645–654.  
<https://doi.org/https://doi.org/10.35870/ijsecs.v5i2.4856>
- Gerungan, S. W. J., Wahongan, A., & Lembong, R. (2022). PERTANGGUNGJAWABAN PERDATA PENGELOLA PARKIR TERHADAP KENDARAAN KONSUMEN. *Lex Administratum*, 10(5).
- Guluh, M. A., & Sulkify, M. (2023). SISTEM MONITORING KETERSEDIAN SLOT PARKIR DENGAN ESP8266 BERBASIS INTERNET OF THINGS. *Jurnal Sistem Informasi Dan Teknik Komputer*, 8(1), 130–134.  
<https://doi.org/https://doi.org/10.51876/simtek.v8i1.247>
- Hyzy, M., Bond, R., Mulvenna, M., Bai, L., Dix, A., Leigh, S., & Hunt, S. (2022). System Usability Scale Benchmarking for Digital Health Apps: Meta-analysis. *JMIR MHealth and UHealth*, 10(8). <https://doi.org/10.2196/37290>
- Ilham, N. A., & Naziro. (2019). IMPLEMENTASI KONSEP PEMROGRAMAN BERORIENTASI OBJEK PADA APLIKASI SISTEM PARKIR MENGGUNAKAN BAHASA PEMROGRAMAN JAVA. *Jurnal Edukasi Elektro*, 3(2), 63–69.
- Kusumaningtyas. (2016). EVALUASI DAN PERANCANGAN SISTEM INFORMASI LAHAN PARKIR. *Studia Informatika: Jurnal Sistem Informasi*, 9(1), 15–27.
- Marwita, F. (2023). RANCANG BANGUN PROTOTYPE REBSEKRESI PARKING GEDUNG PERKANTORAN BERBASIS IoT. *Sinusoida*, 25(1).
- Nugraha, B. F., Aditama, F., Arrofi, M., Ahmad, S. U., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Penghitungan Parkir Swalayan ADA Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 5(2), 146–151.  
<https://doi.org/10.32493/informatika.v5i2.5350>
- Oktaviani, R., Nazwirman, Djamiludin, & Windyarsari, V. S. (2020). Aplikasi Sistem Parkir Kendaraan Bermotor Menggunakan Teknologi Radio Frequency Identification (RFID) di Universitas Islam Syekh Yusuf Tangerang. *Jurnal Ilmiah Fakultas Teknik*, 1(2), 96–103.

- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. L. (2023). PENGUJIAN BLACK BOX DAN WHITE BOX SISTEM INFORMASI PARKIR BERBASIS WEB BLACK BOX AND WHITE BOX TESTING OF WEB-BASED PARKING INFORMATION SYSTEM. *Jurnal Testing Dan Implementasi Sistem Informasi*, 4(1), 1–16.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). McGraw-Hill.
- Purniawan, A., & Nanang. (2024). Pengembangan Aplikasi Parkir Berbasis Web Menggunakan Metode RFID Untuk Manajemen Parkir (Studi Kasus: PT. Prama Jaya Arja). *Jurnal Ilmu Komputer, Teknik Dan Multimedia*, 2(4), 373–395.
- Putera, W. A., Sandiyasa, I. G. E., Pradana, I. B. E. W., Dalem, I. B. G. W. A., & Dewi, N. W. S. (2022). INTERNATIONAL SEMINAR COMMEMORATING THE 100 TH ANNIVERSARY OF TAMANSISWA | Yogyakarta. *International Seminar Commemorating the 100th Anniversary of Tamansiswa*, (1), 322–326.
- Rahardjo, P., Safitri, N., & Nur Rizky, as. (2023). MONITORING SLOT PARKIR BERBASIS WEB DAN ANDROID. *Orbith: Majalah Ilmiah Pengembangan Rekayasa Dan Sosial*, 18(3), 221–231. <https://doi.org/https://doi.org/10.32497/orbith.v18i3.4366>
- Rahmatillah, A. (2022). *MOPARKING: SISTEM MONITORING PARKIRAN MOBIL BERBASIS IOT* [Skripsi, Universitas Islam Indonesia]. <https://dspace.uii.ac.id/handle/123456789/42079>
- Ramadhan, M. M. Z., & Angelia, F. (2023). MENGOPTIMALKAN PENGEMBANGAN APLIKASI MOBILE MELALUI PERBANDINGAN METODE PENGEMBANGAN PERANGKAT LUNAK (WATERFALL, PROTOTYPE, MOBILE-D, AGILE, RAD). *Jurnal Ilmiah Teknologi Informasi Dan Sains*, 3(2), 13–19. <https://doi.org/https://doi.org/10.36815/submit.v3i2.2993>
- Romdoni, M. R., & Ramadhan, D. A. (2022). Prototipe Sistem Smart Car Parking Berbasis IoT dengan Monitoring Melalui Web dan Android. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 5(1). <https://prosiding.unipma.ac.id/index.php/SENATIK/article/view/2970>
- Setlight, C. Y. Y., Cahyono, M. S. D., & Rahayu, Y. E. (2022). Evaluasi Kinerja Kapasitas Parkir Dan Kebutuhan Ruang Parkir Di Masjid Al-Ikhlas Kota Malang. *Jurnal Anggapa*, 1(2). <https://doi.org/https://doi.org/10.61293/anggapa.v1i2.301>

- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. *Prosiding 9th Industrial Research Workshop and National Seminar*, 9. <https://doi.org/10.35313/irwns.v9i0.1116>
- Suria, O. (2024). A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform. *Journal of Information Systems and Informatics*, 6(2), 992–1007. <https://doi.org/10.51519/journalisi.v6i2.750>
- Tannujaya, F., Hasanah, F., & Tio Nugroho, A. (2021). Design and Realization of Parking System Prototype Using RFID Based Microcontroller with Reservation Service via Website for Four-wheel Vehicle. *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, 207. <https://doi.org/10.2991/aer.k.211106.070>
- Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan, Pemerintah Republik Indonesia (2009).
- Yulianggara, R. (2022). *PENGEMBANGAN APLIKASI INFORMASI TEMPAT PARKIR TERDEKAT DI PROVINSI DKI JAKARTA MENGGUNAKAN LOCATION BASED SERVICE (LBS) DAN GOOGLE MAPS API (STUDI KASUS: UNIT PERPAKIRAN DINAS PERHUBUNGAN DKI JAKARTA)*. Universitas Islam Negeri Syarif Hidayatullah.

## LAMPIRAN

Lampiran A: Tempat Parkir di Masjid Daarussalam Ganjuran



Lampiran B: Wawancara dengan Ketua Takmir Masjid



Lampiran C: Wawancara dengan Jemaah Masjid



Lampiran D: Wawancara dengan Takmir Masjid

