

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bisnis kebugaran merupakan salah satu sektor yang menunjukkan pertumbuhan signifikan seiring dengan meningkatnya kesadaran masyarakat akan pentingnya gaya hidup sehat. Gym dan pusat kebugaran menjadi pilihan utama masyarakat untuk menjaga kebugaran tubuh, menurunkan berat badan, serta membentuk massa otot. Penelitian Koloay (2020) mengungkapkan bahwa 80% responden menyatakan kebugaran sebagai hal yang sangat penting, dan 85% dari mereka berkeinginan untuk rutin berolahraga. Kondisi ini mencerminkan tingginya permintaan terhadap fasilitas kebugaran yang mampu memberikan layanan berkualitas serta mendukung gaya hidup aktif masyarakat.

Namun, tingginya minat masyarakat terhadap layanan gym belum selalu diimbangi dengan sistem manajemen yang efisien dan terintegrasi. Berdasarkan hasil penelitian Setiawan et al. (2023), sebagian besar gym di Indonesia masih mengelola data keanggotaan secara manual tanpa memanfaatkan teknologi berbasis internet. Metode manual tersebut menimbulkan kendala berupa ketidaktepatan pencatatan data, keterlambatan akses informasi, kesulitan dalam evaluasi data anggota, serta risiko hilangnya data akibat kerusakan perangkat atau bencana. Selain itu, metode manual rentan terhadap human error, kurang mendukung integrasi dengan sistem pembayaran digital, serta tidak memberikan kemudahan analisis data untuk pengambilan keputusan strategis.

Kendala tersebut menyebabkan penurunan kualitas pelayanan kepada anggota. Contohnya, keterlambatan verifikasi pembayaran menghambat akses anggota ke fasilitas, sementara manajemen menghadapi kesulitan dalam memantau masa aktif keanggotaan dan tren penggunaan fasilitas. Oleh karena itu, diperlukan penerapan teknologi yang memudahkan pelaksanaan aktivitas pengelolaan gym secara efektif dan efisien (Hamdani & Priatna, 2021).

Pemanfaatan teknologi informasi, khususnya aplikasi berbasis Android yang terintegrasi dengan cloud database, dapat menjadi solusi strategis. Aplikasi ini memungkinkan proses pendaftaran, perpanjangan, dan pembayaran keanggotaan dilakukan secara cepat dan aman, sekaligus menyediakan fitur pemantauan data anggota secara real-time. Penelitian Hevner et al. (2004) menjelaskan bahwa penerapan design science pada pengembangan sistem informasi dapat menciptakan solusi inovatif yang meningkatkan efisiensi operasional dan mutu layanan organisasi.

Penelitian yang dilakukan oleh Setiawan et al. (2023) di Ni Baroe Gym menunjukkan bahwa aplikasi manajemen *member gym* berbasis Android dapat berjalan dengan baik dan efektif. Hasil penelitian tersebut menjelaskan bahwa pengelola *gym* lebih mudah melakukan proses pendaftaran dan transaksi dibandingkan dengan pencatatan manual. Sistem yang dibangun memungkinkan pengelolaan data *member* dan transaksi secara lebih efisien, termasuk pencatatan aktivitas serta pendaftaran anggota. Aplikasi ini menerima data yang diinput oleh admin, kemudian melalui tahap pembuatan prototipe dan demonstrasi sebelum digunakan secara penuh. Selain itu, aplikasi dinilai memudahkan pengguna dalam pengoperasian serta memiliki desain yang lebih *user-friendly*.

Penelitian serupa dilakukan oleh Hidayat et al. (2023) yang menemukan bahwa aplikasi serupa dapat mempermudah proses pengelolaan *member* maupun pengunjung. Aplikasi ini memungkinkan pengunjung untuk melakukan pendaftaran secara online, sehingga data tersimpan secara otomatis dan dapat diakses dengan mudah oleh pengelola. Dengan adanya sistem ini, permasalahan terkait manajemen data yang kurang efisien serta potensi kesalahan pencatatan dapat diminimalisasi. Selain itu, aplikasi juga memberikan kemudahan bagi pengelola dalam mengatur data *member* dan transaksi, sekaligus meningkatkan efisiensi pengelolaan secara keseluruhan.

Berdasarkan uraian tersebut, pengembangan aplikasi manajemen *member gym* berbasis Android menjadi relevan dan penting dilakukan. Selain menjawab kebutuhan digitalisasi pengelolaan keanggotaan, aplikasi ini juga dapat meningkatkan produktivitas pengelola, meminimalisir kesalahan pencatatan, serta memberikan pengalaman layanan yang lebih baik bagi anggota. Dengan dukungan penelitian terkait dan tren digitalisasi di

sektor kebugaran, pengembangan aplikasi ini diharapkan mampu menjadi langkah nyata dalam meningkatkan kualitas manajemen dan daya saing fasilitas *gym* di era modern.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, dapat dirumuskan beberapa masalah yang dihadapi dalam pembuatan tugas akhir yaitu sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi manajemen *member gym* berbasis *mobile* guna mengatasi permasalahan yang ada di SunuGym?
2. Bagaimana menguji dan mengevaluasi aplikasi pengelolaan *member gym* berbasis *mobile* yang telah dibangun?

1.3 Batasan Masalah

Dalam penelitian ini, ada beberapa batasan yang ditetapkan untuk menjaga agar fokus penelitian tetap jelas dan untuk memastikan hasil yang sesuai dengan tujuan yang diinginkan, yaitu:

1. Penelitian ini hanya berfokus pada lokasi SunuGym.
2. Aplikasi yang dikembangkan tidak terintegrasi dengan *database* yang saat ini digunakan oleh SunuGym.
3. Aplikasi dibangun menggunakan bahasa pemrograman *Kotlin* dan memanfaatkan *Firestore* sebagai *database*.
4. Penelitian ini hanya berfokus pada platform Android dan tidak mencakup pengembangan untuk platform *mobile* lainnya, seperti iOS atau sistem operasi lainnya.

1.4 Tujuan Penelitian

Berdasarkan masalah yang telah dirumuskan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk merancang dan membangun aplikasi pengelolaan *member gym* berbasis *mobile* yang dapat mengatasi permasalahan yang ada di SunuGym.
2. Untuk menguji dan mengevaluasi aplikasi *member gym* berbasis *mobile* yang telah dibangun.

1.5 Landasan Teori

1.5.1 SunuGym

SunuGym salah satu pusat kebugaran di Makassar yang menyediakan sejumlah fasilitas untuk mempromosikan keterlibatan anggotanya dalam olahraga dan kebugaran. Karena semakin banyak orang yang menyadari pentingnya olahraga dan kesehatan yang baik, keanggotaan SunuGym terus bertambah setiap tahun. Tetapi seperti banyak *gym* lainnya, SunuGym kesulitan melacak informasi anggotanya, terutama selama saat-saat seperti pemadaman listrik ketika akses ke sistem pencatatan berbasis komputer dibatasi. Di SunuGym, administrasi dan pengelolaan data anggota masih dilakukan dengan menggunakan sistem berbasis komputer yang tidak memiliki cadangan daya jika terjadi pemadaman daya.

Anggota dan manajemen mungkin mengalami ketidaknyamanan sebagai akibat dari kesulitan dalam memasukkan informasi anggota baru, memperpanjang keanggotaan, dan mencatat pembayaran. Solusi berbasis aplikasi seluler yang memungkinkan manajer dan anggota mengakses informasi keanggotaan secara fleksibel tanpa bergantung pada sistem berbasis PC diperlukan untuk mengatasi masalah ini. Program ini memudahkan administrator untuk menggunakan perangkat seluler untuk menambahkan, mengubah, dan menghapus data anggota secara *real time*.

Melalui aplikasi yang dikembangkan, pengguna juga dapat dengan mudah memperbarui keanggotaan mereka, memeriksa status keanggotaan mereka, dan mendapatkan informasi tentang layanan yang ditawarkan di SunuGym. Implementasi aplikasi manajemen anggota *gym* berbasis Android dengan *Firestore* sebagai *backend* diharapkan dapat meningkatkan efisiensi dalam pengelolaan data, mengurangi risiko kehilangan data akibat keterbatasan sistem yang ada, serta memberikan pengalaman yang lebih baik bagi anggota SunuGym.

1.5.2 Aplikasi Berbasis Android

Android adalah sistem operasi yang dikembangkan oleh Google dan banyak digunakan di perangkat seluler. Sejak dirilis pada tahun 2008, Android telah menjadi sistem operasi dominan yang digunakan oleh berbagai merek smartphone dan tablet. Karena permintaan pasar dan jumlah pengguna sistem operasi Android yang sangat banyak, aplikasi Android termasuk yang tercepat (Trivaika & Senubekti, 2022). Manfaat utama Android adalah bersifat open-source, memungkinkan pengembang untuk memodifikasi dan menyempurnakan aplikasi agar sesuai dengan kebutuhan pengguna.

Sistem operasi Android terdiri dari beberapa lapisan utama, yaitu:

1. *Linux Kernel*: Bertindak sebagai inti sistem operasi yang mengelola sumber daya perangkat keras seperti memori, prosesor, dan perangkat *input output*.
2. *Libraries*: Kumpulan pustaka yang mendukung pengembangan aplikasi, termasuk grafik 2D/3D, pengolahan media, dan lain-lain.
3. *Android Runtime (ART)*: Mesin eksekusi aplikasi yang memungkinkan aplikasi berjalan dengan efisien dan responsif.
4. *Application Framework*: Menyediakan *API* untuk pengembang dalam membangun berbagai jenis aplikasi.
5. *Application Layer*: Lapisan tempat aplikasi berjalan dan berinteraksi langsung dengan pengguna.

1.5.3 Kotlin

Kotlin adalah bahasa pemrograman modern yang dikembangkan oleh *JetBrains* dan secara resmi didukung oleh *Google* sebagai bahasa pemrograman utama untuk pengembangan aplikasi Android. Karena sebagian besar kode *Kotlin* mirip dengan *Java*, bahkan lebih ringkas, dan mudah diintegrasikan ke dalam IDE *Java* seperti *NetBeans*, *IntelliJ*, dan *Android Studio*, ini adalah bahasa pemrograman yang cukup mudah dan relatif sederhana untuk dipelajari (Dzulqarnain & Tukino, 2023). Di situs resminya, disebutkan bahwa ada tiga keunggulan utama yang dimiliki *Kotlin*, yaitu sebagai berikut.

Tabel 1. Keunggulan *Kotlin*

No	Keunggulan	Deskripsi
1	<i>Null Safety</i>	<i>Kotlin</i> memiliki fitur <i>null safety</i> yang mencegah <i>error</i> akibat <i>NullPointerException</i> (NPE), yang sering menjadi masalah dalam <i>Java</i> .
2	<i>Coroutines</i>	<i>Kotlin</i> menyediakan <i>coroutines</i> untuk menangani operasi <i>asynchronous</i> dengan lebih efisien dibandingkan dengan penggunaan <i>thread</i> tradisional.
3	Ringkas dan ekspresif	<i>Kotlin</i> mengurangi <i>boilerplate code</i> yang sering ditemukan dalam <i>Java</i> , sehingga kode menjadi lebih bersih dan mudah dibaca.

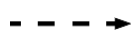
Dengan keunggulan utama tersebut, *Kotlin* menjadi pilihan utama dalam pengembangan aplikasi Android modern, termasuk dalam pengembangan aplikasi manajemen *gym* yang membutuhkan kinerja tinggi dan keamanan data yang baik.

1.5.4 Use Case Diagram

Use Case Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem yang akan dikembangkan. Diagram ini memodelkan fungsionalitas sistem dari perspektif pengguna, menampilkan hubungan antara aktor dan berbagai *use case* (kasus penggunaan) dalam sistem. *Use Case Diagram* menjelaskan aspek fungsionalitas sistem dengan menggambarkan interaksi yang terjadi antara aktor dengan sistem (A & Tri, 2018). *Use case* dideskripsikan secara tekstual dalam bentuk skenario untuk menjelaskan interaksi tersebut, kemudian diilustrasikan secara visual dalam bentuk *Use Case Diagram*

untuk menggambarkan konteks dari sistem yang dikembangkan.

Tabel 2. Komponen *use case diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berinteraksi dengan <i>use case</i>
	<i>Use case</i>	Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i>	Abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i>
	<i>Generalization</i>	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>
<<include>>	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

1.5.5 Activity Diagram

Activity Diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja atau aktivitas dalam suatu sistem atau proses bisnis. Diagram ini memodelkan urutan aktivitas dan keputusan yang terjadi, mirip dengan *flowchart*, tetapi dengan kemampuan untuk mewakili paralelisme, percabangan, dan aliran bersamaan dalam suatu sistem. *Activity Diagram* adalah pemodelan yang dilakukan pada suatu sistem dan menggambarkan aktivitas sistem berjalan. Diagram ini digunakan sebagai penjelasan aktivitas program tanpa melihat koding atau tampilan.

Tabel 3. Komponen pada *activity diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Start Point</i>	Titik awal bagaimana Objek dibentuk atau diawali
	<i>End Point</i>	Titik akhir objek
	<i>Activities</i>	Memperlihatkan bagaimana masing masing kelas Antarmuka saling berinteraksi satu sama lain
	<i>Decision</i>	Menggambarkan suatu Keputusan atau tindakan yang harus diambil pada kondisi tertentu

1.5.6 Firebase

Firebase merupakan platform pengembangan aplikasi berbasis *cloud* yang dikembangkan oleh *Google*. *Firebase* menyediakan berbagai layanan, termasuk *Firebase Authentication*, *Firebase Realtime Database*, *Firestore*, *Firebase Cloud Messaging* (FCM), dan *Firebase Hosting*, yang dapat membantu pengembang dalam membangun aplikasi yang lebih *scalable* dan responsif. *Firebase Realtime Database* adalah sistem penyimpanan berbasis *cloud* yang memungkinkan data diperbarui secara *real-time* ke semua perangkat yang terhubung. Teknologi ini menggunakan format JSON dan bekerja dengan mekanisme sinkronisasi otomatis, sehingga perubahan yang dilakukan pada satu perangkat akan langsung terlihat di perangkat lainnya tanpa perlu melakukan refresh manual.

Firestore, sebagai pengembangan lebih lanjut dari *Firebase Realtime Database*, menawarkan fitur yang lebih canggih, termasuk kemampuan kueri yang lebih kompleks, struktur data yang lebih fleksibel, serta integrasi yang lebih baik dengan berbagai layanan *Google Cloud*. Struktur *database* dari *firebase* bersifat fleksibel sehingga pas untuk digunakan pada aplikasi berbasis *mobile* (Purnomo & Onno W. Purbo, 2020). Keunggulan utama *Firebase* sebagai *database cloud* dalam pengembangan aplikasi *mobile* adalah:

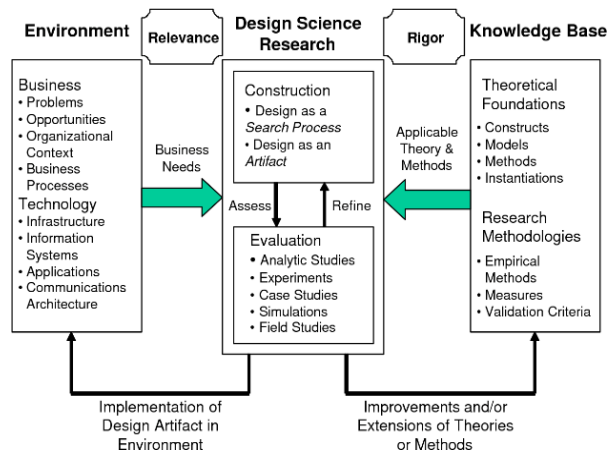
Tabel 4. Keunggulan Database *Firebase*

No	Keunggulan	Deskripsi
1	<i>Realtime Synchronization</i>	<i>Firebase</i> memungkinkan data diperbarui secara langsung antar perangkat tanpa perlu melakukan <i>request</i> manual.
2	<i>Offline Support</i>	<i>Firebase</i> menyediakan fitur <i>offline</i> yang memungkinkan aplikasi tetap berfungsi meskipun tidak terhubung ke internet. Ketika koneksi kembali aktif, data akan otomatis tersinkronisasi.
3	Keamanan yang Kuat	<i>Firebase</i> dilengkapi dengan <i>Firebase Authentication</i> yang memungkinkan pengelolaan otorisasi pengguna dengan berbagai metode, termasuk email/password, <i>Google Sign-In</i> , dan metode autentikasi lainnya.
4	Integrasi Mudah dengan Android	<i>Firebase</i> dirancang untuk bekerja secara optimal dengan aplikasi berbasis Android, dengan dukungan SDK yang memudahkan pengembang dalam mengimplementasikan berbagai fitur.
5	Dukungan Skala Besar	<i>Firebase</i> mampu menangani jumlah pengguna yang besar tanpa memerlukan pengaturan server tambahan, sehingga sangat cocok untuk aplikasi dengan jumlah anggota yang terus bertambah.

1.5.7 Kerangka Penelitian Sistem Informasi

Sistem informasi adalah pengelompokan komponen atau subsistem yang terhubung dan bekerja sama untuk mengelola data sehingga memiliki makna bagi penerima, serta dapat digunakan untuk mengambil keputusan pada masa sekarang maupun masa depan (Nugroho, 2015). Menurut Sutabri (2005), sistem informasi adalah sistem di dalam suatu organisasi yang memenuhi persyaratan pemrosesan transaksi harian, mendukung fungsi operasional yang berorientasi manajerial, dan sejalan dengan kegiatan strategis organisasi untuk menyediakan laporan yang diperlukan kepada pihak eksternal tertentu.

Hevner et al. (2004) menjelaskan bahwa kerangka penelitian sistem informasi berfungsi sebagai kerangka kerja untuk memahami, menerapkan, dan menilai penelitian *design science* dalam domain sistem informasi. *Knowledge base*, *IS research*, dan *environment* membentuk kerangka kerja penelitian sistem informasi yang didasarkan pada *design science* penelitian sistem informasi.

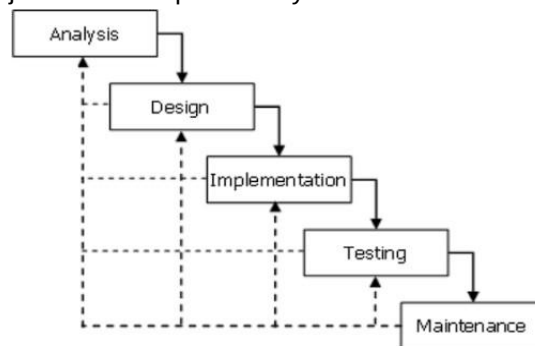


Gambar 1. Information systems research framework

People, organizations, dan technology adalah bagian dari lingkungan, yang menentukan sejauh mana masalah di mana kejadian terjadi. IS *research* adalah penelitian yang menerapkan *behavioral science* dengan menggunakan teori yang berbeda untuk menjelaskan atau membenarkan masalah bisnis, serta *design science* untuk membuat dan menilai artefak (perangkat) yang merupakan perangkat lunak, perangkat keras, atau proses bisnis yang ditujukan untuk memenuhi kebutuhan bisnis yang teridentifikasi. *Knowledge base* mencakup teori dasar dan prosedur penelitian, di antara topik teoretis lainnya.

1.5.8 Metode Pengembangan (Waterfall Metode)

Saat mengembangkan perangkat lunak atau sistem informasi, salah satu model SDLC yang sering digunakan adalah model *Waterfall* (Wahid, 2020). Metode *Waterfall* adalah model pengembangan perangkat lunak yang mengikuti pendekatan linier dan berurutan, di mana setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya. Model ini terdiri dari lima tahap utama.



Gambar 2. Tahapan dari metode waterfall

1. *Requirement Analysis* (Analisis Kebutuhan) yaitu pengumpulan dan analisis persyaratan untuk mengetahui dan memahami komponen perangkat lunak yang memenuhi kebutuhan pengguna.
2. *Design* (Desain) Proses perancangan pembuatan perangkat lunak, termasuk arsitektur perangkat lunak dan representasi antarmuka perangkat lunak.
3. *Implementation* (Implementasi) Proses implementasi suatu perancangan pada perangkat lunak berupa program komputer yang saling terintegrasi, sesuai dengan perancangan yang dibuat pada tahap sebelumnya.
4. *Testing* (Pengujian) Proses pengujian bertujuan untuk memverifikasi apakah sistem yang dihasilkan memenuhi kebutuhan pengguna dan meminimalkan kesalahan yang ada.

5. *Maintenance* (Pemeliharaan) Proses pemeliharaan dapat berupa pembaruan perangkat lunak ketika beberapa komponen baru dirasa perlu, atau proses perbaikan *bug* yang tidak terdeteksi pada proses pengujian sebelumnya.

1.5.9 Blackbox Testing

Blackbox testing merupakan metode pengujian yang berfokus pada spesifikasi fungsional perangkat lunak, penguji dapat menentukan serangkaian kondisi *input* dan melakukan pengujian terhadap spesifikasi fungsional program. Proses pengujian *blackbox* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap fiturnya. Pengujian ini diperlukan untuk mengetahui program tersebut sesuai dengan kebutuhan atau tidak.

1.5.10 User Acceptance Test (UAT)

User Acceptance Testing (UAT) adalah proses dimana klien atau pengguna akhir akan menilai apakah solusi yang dihasilkan sistem telah memenuhi kebutuhan mereka (Suprpto, 2021). Memastikan bahwa aplikasi memenuhi kebutuhan pengguna dan praktis serta efisien adalah tujuan utama dari UAT. Sebelum program tersedia untuk umum, UAT juga dapat digunakan untuk menemukan *bug* atau masalah lain dengan aplikasi. UAT adalah proses penting yang mencakup penilaian kegunaan, fungsionalitas, desain sistem, dan memastikan bahwa aplikasi mudah digunakan dan mampu menangani tugas dalam skenario dunia nyata (Juliandiny, 2023). Indikator keberhasilan yang diukur dalam penelitian *User Acceptance Testing* (UAT) dapat bervariasi tergantung pada konteks dan tujuan pengujian. Salah satu indikator utama adalah kepuasan pengguna, yang mencerminkan seberapa puas pengguna terhadap sistem yang diuji. Pada penelitian ini, penulis akan menggunakan skala *likert* yang terdiri dari empat penilaian yaitu Tidak Setuju (TS), Kurang Setuju (KS), Setuju (S), dan sangat setuju (SS) kepada dua *role* yaitu admin dan *user*.

BAB II METODE PENELITIAN

2.1 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan mulai dari minggu pertama bulan Desember 2024 hingga minggu terakhir bulan Maret 2025 di SunuGym.

Tabel 5. Jadwal Kegiatan Penelitian

No	Tahapan Penelitian	Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Observasi																
2	Studi Literatur																
3	Requirement																
4	Design																
5	Implementation																
6	Testing																
7	Maintenance																

2.2 Analisis Kebutuhan Sistem

Dalam Penelitian ini, dibutuhkan perangkat lunak dan perangkat keras sebagai *tools* atau alat yang dapat mendukung penelitian, serta pengguna sistem (*user*) sebagai bahan analisis kebutuhan dalam perancangan sistem. Adapun kebutuhan sistem dalam penelitian ini, yaitu:

1. Perangkat Lunak

- Figma
- Android Studio
- Chrome
- Android Emulator

2. Perangkat Keras

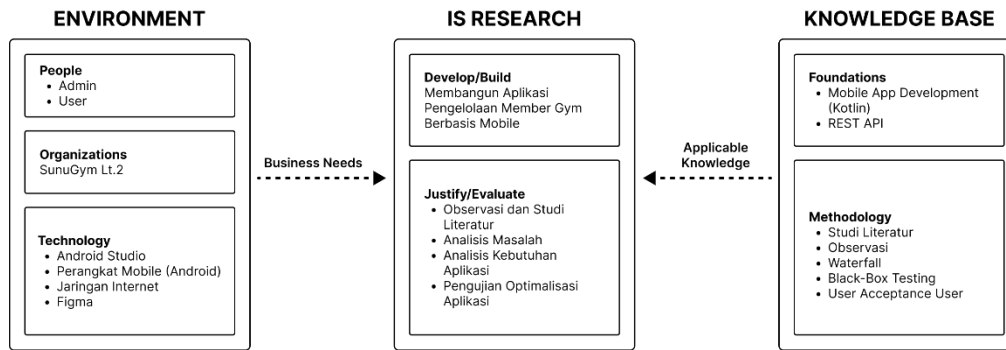
Selama penelitian, peneliti menggunakan Laptop Asus TUF Gaming F15 dengan spesifikasi *processor* NVIDIA Geforce GTX 1650, *RAM* 16 GB, dan *SSD* 512 GB.

3. Pengguna Sistem

- Admin
Admin merupakan pengguna yang berperan untuk mengelola data yang ada didalam aplikasi, contohnya menghapus, mengedit, dan menambahkan data baru di Aplikasi SunuGym.
- User
User hanya memiliki fitur untuk melihat data keanggotaan mereka, menghitung BMI, melihat dan memperbarui profil yang ada di dalam Aplikasi SunuGym.

2.3 Design Science Research

Design Science Research melibatkan pengembangan dan evaluasi pengetahuan preskriptif untuk menciptakan solusi yang meningkatkan kinerja fungsional (Hevner A. R., 2007). Gambar berikut menggambarkan bagaimana *Design Science Research* diterapkan dalam penelitian ini.



Gambar 3. Penerapan *Design Science Research*

Berikut adalah uraian dari masing-masing elemen *design science* yang telah diterapkan dalam penelitian ini.

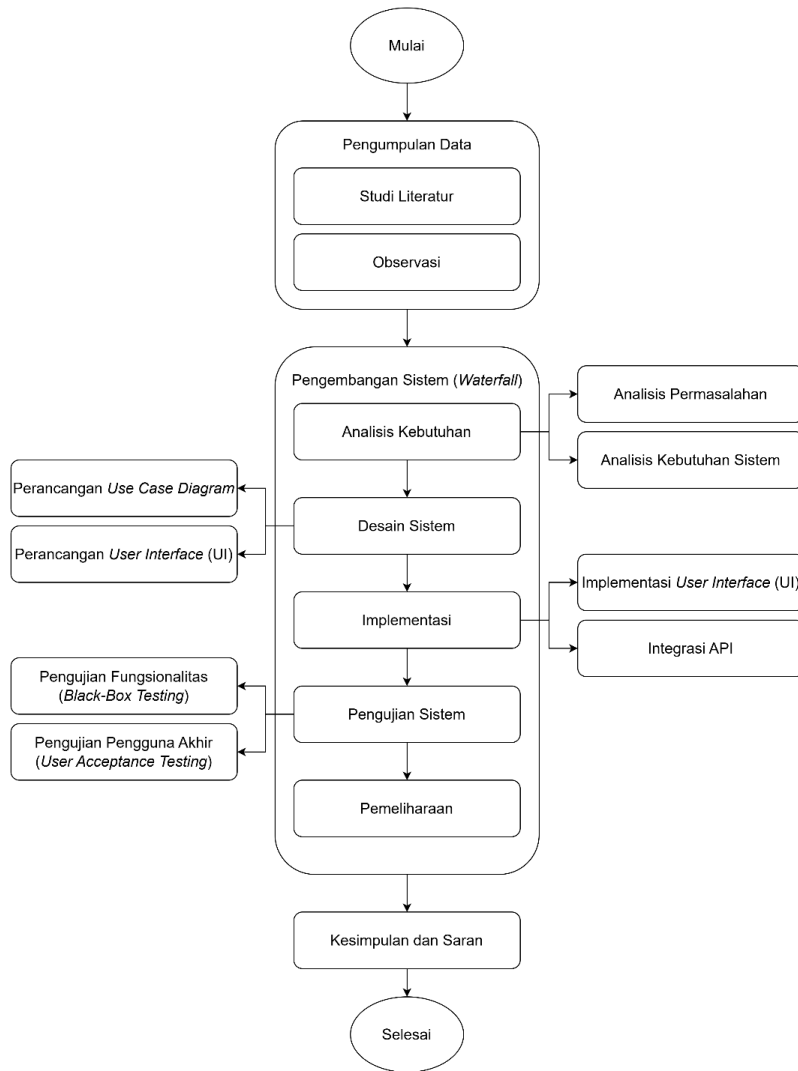
Tabel 6. Elemen-elemen *design science*

No	Elemen	Unsur	Deskripsi	Penerapan pada Penelitian
1	Lingkungan (<i>Environment</i>)	Orang (<i>People</i>)	Individu atau kelompok yang terlibat dalam penelitian.	Penelitiannya melibatkan Admin (Pengelola Gym) dan User (<i>Member Gym</i>) yang terlibat langsung dalam Aplikasi SunuGym.
		Organisasi (<i>Organizations</i>)	Konteks dimana penelitian dilakukan termasuk struktur, budaya, proses, Maupun sumber daya yang tersedia.	SunuGym adalah organisasi yang terlibat dalam penelitian ini sebagai studi kasus.
		Teknologi (<i>Technology</i>)	Infrastruktur, perangkat lunak, ataupun perangkat keras yang digunakan dalam penelitian.	Perangkat <i>mobile</i> berupa <i>smartphone</i> atau tablet yang terhubung dengan jaringan internet.
2	Penelitian Sistem Informasi (<i>IS Research</i>)	Membangun (<i>Build</i>)	Pembuatan artefak, yaitu solusi yang dirancang untuk mengatasi masalah yang telah diidentifikasi.	Aplikasi manajemen <i>member gym</i> berbasis <i>mobile</i> .
		Mengevaluasi (<i>Evaluate</i>)	Evaluasi artefak untuk mengukur efektivitas dan efisiensinya dalam mencapai tujuan penelitian.	Memberikan kuesioner kepada masing-masing peran pengguna untuk mengukur tingkat kegunaan,

No	Elemen	Unsur	Deskripsi	Penerapan pada Penelitian
3	Basis Pengetahuan (<i>Knowledge Base</i>)	Fondasi (<i>Foundations</i>)	Teori, konsep, maupun kerangka kerja yang mendasari penelitian.	efektivitas, efisiensi, dan kepuasan. Pengembangan aplikasi <i>mobile</i> Menggunakan <i>Android Studio</i> Dengan Bahasa Pemrograman <i>Kotlin</i> .
		Metodologi (<i>Methodologies</i>)	Metode yang digunakan untuk pengembangan penelitian.	Metode yang digunakan dalam pengumpulan data dengan studi literatur dan observasi, yaitu Metode pengembangan sistem dengan <i>waterfall</i> ,serta pengujian atau evaluasi sistem dengan menggunakan metode <i>black-box testing</i> dan <i>User Acceptance Testing (UAT)</i> .

2.4 Tahapan Penelitian

Agar penelitian dapat berjalan secara terstruktur dan terencana, maka perlu dibuat suatu tahapan atau alur yang harus diikuti sebagai panduan pelaksanaan penelitian. Penelitian ini sendiri memiliki beberapa tahapan yang dimulai dari proses pengumpulan data dengan metode studi literatur dan observasi, kemudian dilanjutkan dengan proses pengembangan sistem menggunakan metode *waterfall*. Berdasarkan metode *waterfall*, tahapan-tahapan yang harus dilakukan dalam mengembangkan sebuah sistem, yaitu dimulai dari analisis kebutuhan, desain sistem, implementasi sistem, pengujian sistem, dan diakhiri dengan pemeliharaan sistem. Untuk lebih jelasnya, tahapan dalam penelitian ini dapat dilihat pada Gambar di bawah.



Gambar 4. Tahapan Penelitian
2.5 Metode Pengumpulan Data

Dalam penelitian ini, dilakukan beberapa tahap pengumpulan data untuk mendukung pengembangan sistem yang lebih baik. Pemilihan metode pengumpulan data sangat krusial karena hasil yang diperoleh akan menjadi dasar dalam proses pengembangan. Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur adalah metode pengumpulan data dengan menelaah berbagai sumber tertulis yang relevan dengan topik penelitian. Sumber-sumber ini meliputi buku, jurnal, artikel ilmiah, laporan penelitian, serta dokumen dokumen lain yang terkait. Dalam penelitian ini, penulis melakukan studi literatur dengan mencari berbagai informasi dari jurnal, buku, dan website yang relevan mengenai pengembangan aplikasi pengelolaan *member gym*.

2. Observasi

Observasi merupakan metode pengumpulan data dengan cara mengamati langsung objek penelitian di lapangan. Dalam penelitian ini, observasi dilakukan untuk memahami secara mendalam kondisi yang ada di objek penelitian. Proses observasi ini dilakukan di kawasan SunuGym.

2.6 Metode Pengembangan Sistem

Dalam penelitian ini, metode pengembangan sistem yang digunakan adalah metode *waterfall*. Metode ini terdiri dari beberapa tahapan yang harus dilalui secara berurutan, sebagai berikut:

1. Analisis Kebutuhan (*Requirements*)

Tahap ini merupakan awal dari pengembangan sistem yang bertujuan untuk memahami kebutuhan pengguna dan sistem yang akan dikembangkan. Pada tahap ini, peneliti mengidentifikasi permasalahan berdasarkan hasil observasi untuk memperoleh informasi terkait sistem yang sedang berjalan sehingga peneliti mampu memahami dan menentukan spesifikasi dari sistem yang ingin dikembangkan. Dalam penelitian ini, teridentifikasi beberapa permasalahan utama, yaitu ketidaktepatan pencatatan data, keterlambatan akses informasi, kesulitan dalam evaluasi data anggota, serta tingginya risiko kehilangan data akibat faktor teknis seperti kerusakan perangkat atau bencana, ditambah dengan pencatatan manual yang rentan terhadap human error, sulit diintegrasikan dengan sistem pembayaran digital, dan tidak mendukung analisis data untuk pengambilan keputusan strategis. Sebagai solusi, analisis kebutuhan sistem mencakup fitur pencatatan data otomatis, akses informasi cepat, evaluasi data anggota melalui *dashboard*, pencegahan kehilangan data dengan *backup cloud*, integrasi pembayaran digital, serta analisis strategis untuk pengambilan keputusan.

2. Desain (*Design*)

Setelah kebutuhan teridentifikasi, tahap selanjutnya adalah merancang arsitektur, komponen, dan juga antarmuka pengguna (*User Interface*) dari sistem yang akan dibangun. Tahap ini bertujuan untuk memberikan panduan yang jelas mengenai cara mengkonstruksi setiap bagian dari sistem sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

3. Implementasi (*Implementation*)

Tahap Implementasi merupakan pelaksanaan dari tahap desain. Pada tahap ini, desain dari sistem yang telah dibuat diimplementasikan ke dalam kode program dengan menggunakan *Kotlin*.

4. Pengujian (*Verification*)

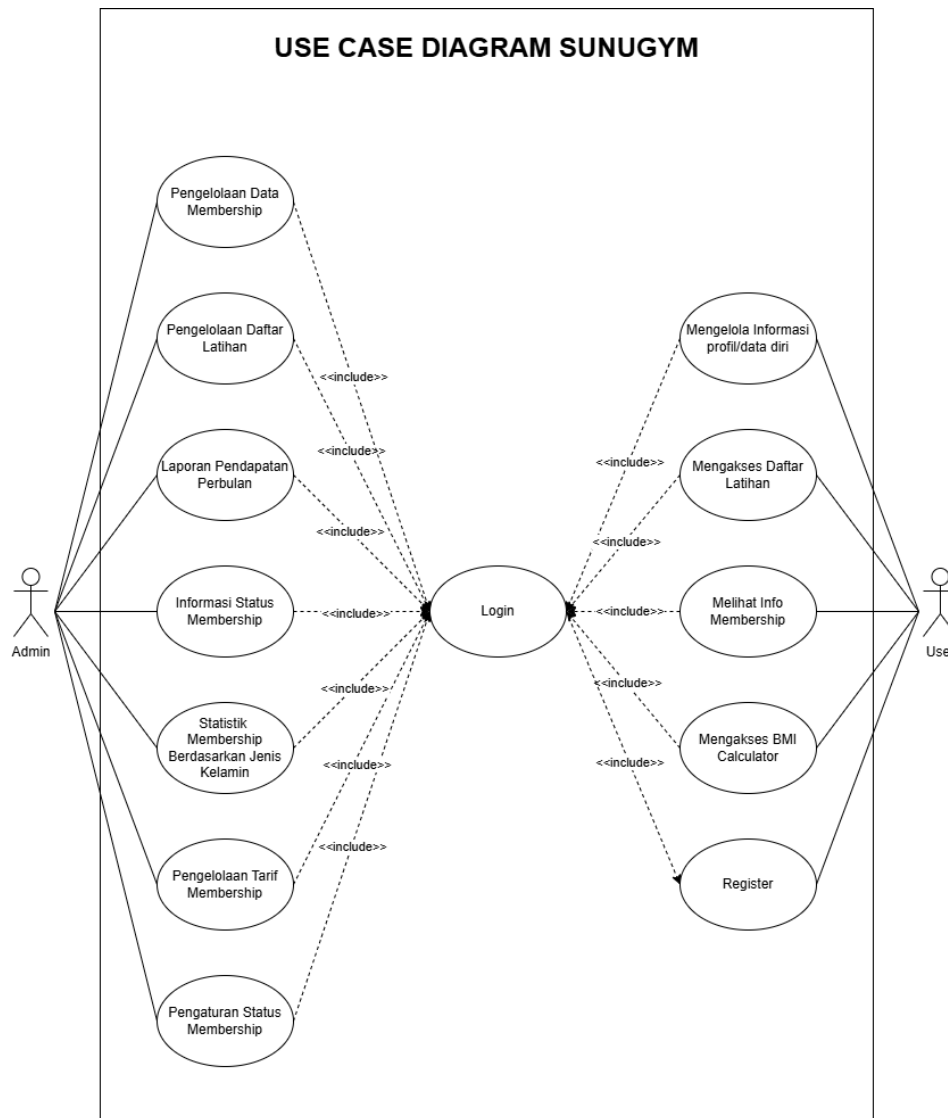
Setelah sistem selesai diimplementasikan, tahap pengujian dilakukan untuk memastikan bahwa sistem yang dibangun telah memenuhi kebutuhan pengguna dan berfungsi dengan baik sesuai dengan spesifikasi yang telah ditetapkan sebelumnya. Pengujian sistem pada penelitian ini dilakukan dengan menggunakan metode *Black-box Testing* dan *User Acceptance Testing (UAT)* untuk menguji fungsionalitas sistem. Pengujian sistem juga dilakukan untuk mengidentifikasi dan memperbaiki *bug* atau kesalahan yang mungkin muncul pada aplikasi sebelum dirilis kepada pengguna.

5. Pemeliharaan (*Maintenance*)

Tahap akhir dalam pengembangan sistem adalah Pemeliharaan. Tahap ini bertujuan untuk memperbaiki kesalahan dan kekurangan yang ditemukan pada sistem atau aplikasi setelah digunakan oleh pengguna. Tahap pemeliharaan juga dapat mencakup penambahan fitur baru, peningkatan kinerja sistem, serta pembaruan sistem jika diperlukan.

2.7 Use Case Diagram

Perancangan sistem pada penelitian ini divisualisasikan melalui *use case diagram*. *Use case diagram* memberikan gambaran umum mengenai fungsionalitas sistem dari perspektif pengguna (aktor) dan bagaimana aktor berinteraksi dengan *use case* dalam sistem tersebut. *Use case diagram* juga memungkinkan kita untuk melihat aktivitas atau tindakan apa saja yang dapat dilakukan oleh masing-masing peran pengguna pada aplikasi yang dibuat. Gambar di bawah menunjukkan *use case diagram* dari sistem manajemen *member gym* yang dirancang.



Gambar 5. Penerapan *Use Case Diagram*

Dapat dilihat pada Gambar di atas, terdapat dua pengguna atau aktor dengan peran yang berbeda, yaitu Admin dan User. Kedua aktor tersebut harus melakukan *login* terlebih dahulu agar dapat mengakses fitur-fitur lain, sesuai dengan peran mereka masing-masing. Berikut ini adalah uraian dari kedua aktor yang terdapat pada sistem.

1. Admin

Admin atau administrator adalah pengguna yang memiliki hak untuk mengelola seluruh data *Membership* yang ada di SunuGym. Dalam konteks penelitian ini, pengelola *gym* adalah satu satunya orang yang berperan sebagai admin. Fokus utama seorang admin, yaitu:

- **Pengelolaan Data Membership**

Fitur ini memungkinkan admin untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap data *membership*. Admin dapat menambahkan data *member* baru, memperbarui informasi *member*, melihat detail *membership*, serta menghapus data *member* jika diperlukan.

- **Pengelolaan Daftar Latihan**

Fitur ini digunakan oleh admin untuk mengelola berbagai jenis latihan yang tersedia di *gym*. Admin dapat menambahkan latihan baru, mengedit latihan yang sudah ada, menghapus latihan yang tidak digunakan lagi, serta melihat seluruh daftar latihan.

- **Laporan Pendapatan Perbulan**

Fitur ini memungkinkan admin untuk melihat total pendapatan dari membership selama satu bulan. Selain itu, admin juga bisa melihat perbandingan antara pendapatan bulan ini dan bulan sebelumnya untuk mengetahui apakah terjadi peningkatan atau penurunan pendapatan (plus/minus).

- **Informasi Status Membership**

Fitur ini menyajikan informasi jumlah *member* yang masih aktif (masa *membership* masih berlaku) dan yang sudah tidak aktif (masa *membership* telah habis). Informasi ini berguna untuk monitoring dan perencanaan strategi layanan *gym*.

- **Statistik Membership Berdasarkan Jenis Kelamin**

Fitur ini menampilkan statistik jumlah member berdasarkan jenis kelamin. Admin dapat mengetahui berapa banyak *member* laki-laki dan perempuan yang terdaftar, sehingga bisa membantu dalam analisis demografis pengguna *gym*.

- **Pengelolaan Tarif Membership**

Fitur ini memberikan akses kepada admin untuk mengatur harga dari masing-masing tipe *membership*. Tipe *membership* yang tersedia yaitu:

- 1 Bulan
- 3 Bulan
- 6 Bulan
- 1 Tahun
- 3 Tahun

Admin dapat menaikkan atau menurunkan harga sesuai kebijakan atau promosi tertentu.

- **Pengaturan Status Membership**

Fitur ini digunakan untuk mengatur status aktif atau tidak aktif suatu membership. Admin dapat menonaktifkan (*inactivate*) *membership* secara manual atau mengaktifkan kembali (*reactivate*) *membership* sesuai permintaan atau kebutuhan.

2. **User**

User atau pengguna dalam sistem SunuGym adalah *member* yang telah terdaftar dan memiliki akses untuk memanfaatkan fitur-fitur yang disediakan sesuai kebutuhan mereka. Fokus utama *user* adalah memanfaatkan layanan *gym* untuk mendukung aktivitas olahraga mereka, dengan fitur-fitur sebagai berikut:

- **Register**

Fitur ini memungkinkan calon member untuk mendaftar sebagai *user* baru di sistem SunuGym. *User* dapat mengisi data diri dan memilih tipe *membership* yang diinginkan untuk memulai keanggotaan mereka.

- **Mengelola Informasi Profil dan Data Diri**

Fitur ini memungkinkan *user* untuk melihat dan memperbarui informasi pribadi mereka, seperti nama, alamat, nomor telepon, atau data lain yang relevan. *User* juga dapat memastikan bahwa data mereka selalu akurat dan sesuai dengan kondisi terkini.

- **Mengakses Daftar Latihan**

Fitur ini memberikan akses kepada *user* untuk melihat daftar latihan yang tersedia di *gym* dan rekomendasi latihan. *User* dapat memilih latihan yang sesuai dengan kebutuhan atau tujuan olahraga mereka, sehingga mereka dapat merencanakan aktivitas *gym* dengan lebih baik.

- **Melihat Informasi Status *Membership***

Fitur ini memungkinkan *user* untuk memeriksa status *membership* mereka, apakah masih aktif atau telah habis. Informasi ini membantu *user* untuk mengetahui masa berlaku *membership* mereka dan merencanakan perpanjangan jika diperlukan.

- **Mengakses BMI Calculator**

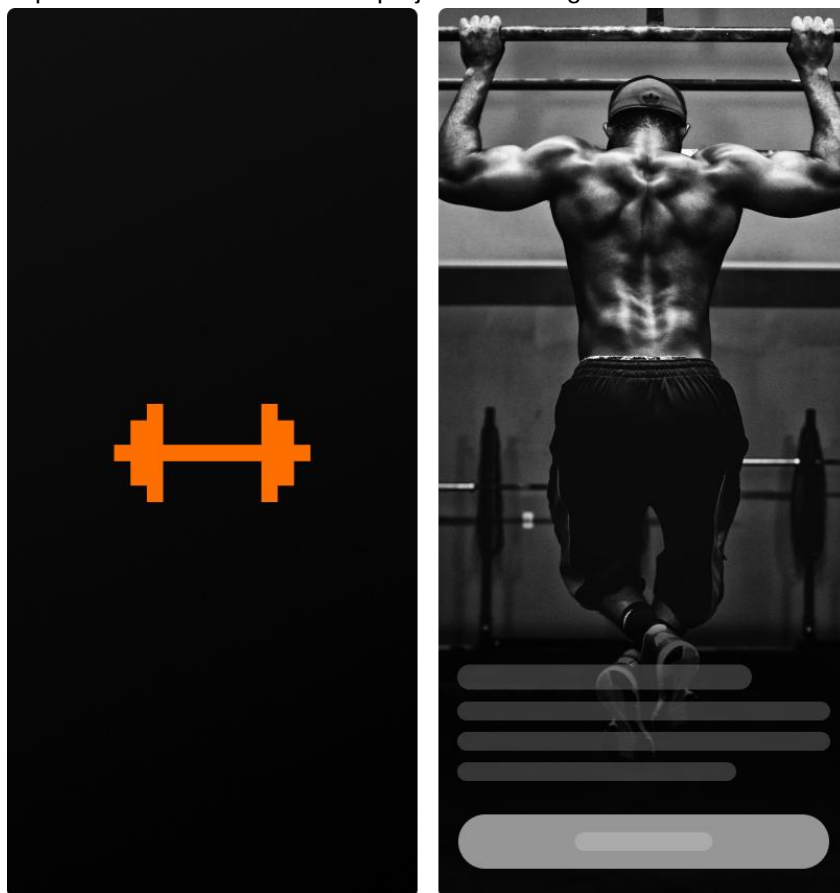
Fitur ini menyediakan alat perhitungan *Body Mass Index* (BMI) yang dapat digunakan *user* untuk mengetahui indeks massa tubuh mereka. Dengan fitur ini, *user* dapat memantau kondisi tubuh mereka dan menyesuaikan program latihan untuk mencapai tujuan kesehatan yang diinginkan.

2.8 Perancangan *User Interface*

Desain *User Interface* (UI) merupakan representasi visual yang berinteraksi langsung dengan pengguna, bertujuan untuk memberikan gambaran mengenai fitur-fitur yang tersedia dalam sistem Rancangan *User Interface* pada penelitian ini adalah sebagai berikut:

1. Halaman *Splash* dan *Welcome*

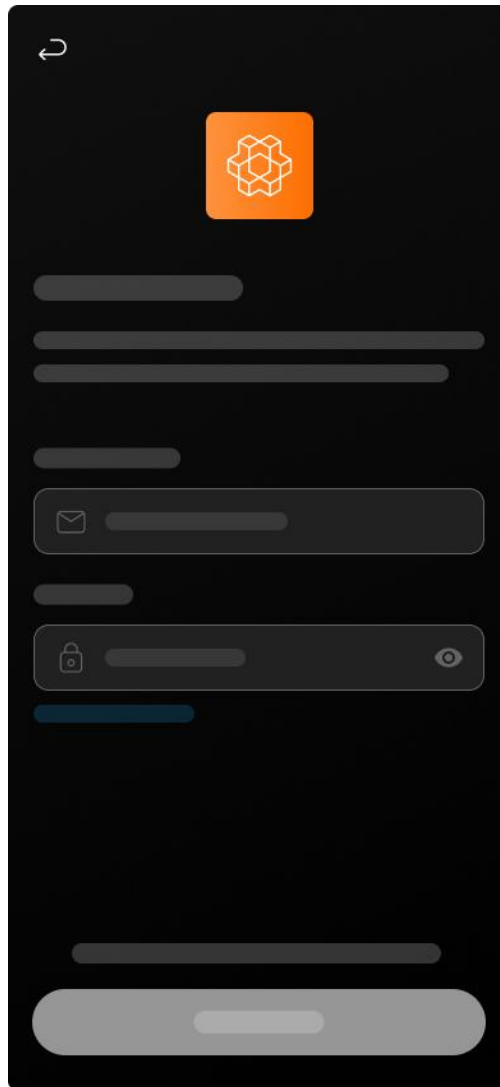
Halaman *Splash* menampilkan logo aplikasi SunuGym yang berupa ikon dumbel berwarna oranye dengan latar belakang hitam. Desain ini sederhana namun langsung mencerminkan tema kebugaran dan olahraga. Halaman ini berfungsi sebagai layar pembuka yang muncul saat aplikasi pertama kali dibuka. Setelah ditampilkan, pengguna akan langsung di arahkan ke Halaman *Welcome*. Pada Gambar 6, Halaman ini bertujuan untuk memberikan sambutan dan motivasi kepada *user* sebelum memulai perjalanan kebugaran mereka.



Gambar 6. Desain *Wireframe* Halaman *Splash* dan *Welcome*

2. Halaman *Login*

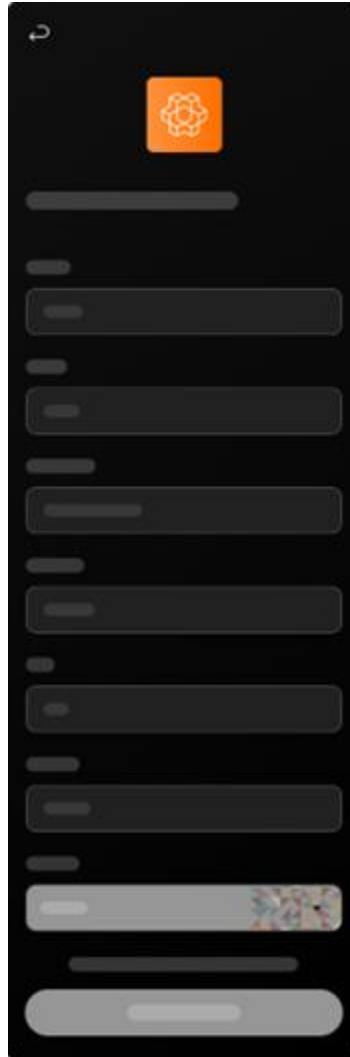
Halaman *login* merupakan halaman yang meminta pengguna untuk memasukkan email dan kata sandi sebagai langkah autentikasi sebelum dapat mengakses fitur-fitur dalam aplikasi. Tampak desain antarmuka yang sederhana dengan kolom input untuk email dan kata sandi, serta tombol login dan opsi "Lupa Kata Sandi" untuk keamanan pengguna, sebagaimana ditunjukkan pada Gambar 7.



Gambar 7. Desain Wireframe Halaman Login

3. Halaman *Register*

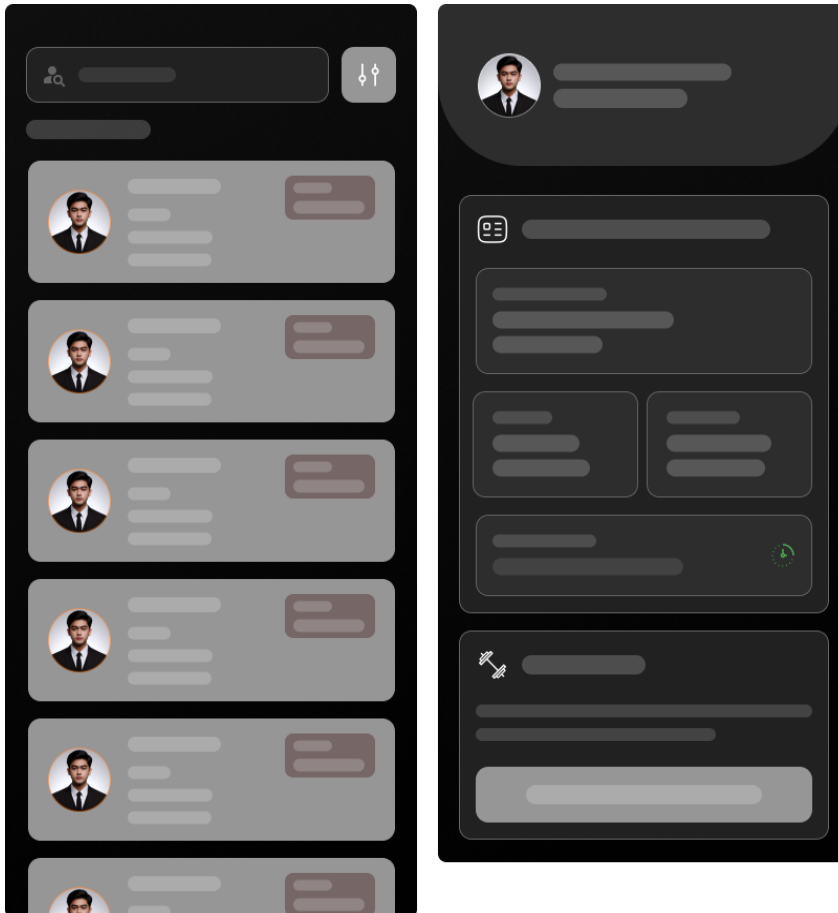
Halaman *Register* merupakan halaman yang memungkinkan pengguna membuat akun baru dengan memasukkan informasi, seperti nama, email, kata sandi, alamat, umur, berat badan, dan jenis kelamin untuk dapat mengakses fitur-fitur dalam aplikasi. Gambar 8 menunjukkan desain antarmuka yang rapi dengan kolom isian untuk setiap data, tombol “Daftar” untuk menyimpan, serta opsi untuk kembali ke halaman Login.



Gambar 8. Desain Wireframe Halaman Register

4. Halaman Menu Utama

Halaman menu utama merupakan halaman yang pertama kali muncul setelah pengguna berhasil melakukan *login*. Halaman ini memiliki tampilan yang berbeda antara admin dan *user*. Halaman Menu Utama Admin menampilkan daftar lengkap pengguna yang memiliki *membership*, memungkinkan admin untuk melihat dan mengelola informasi pengguna dengan mudah. Sementara itu, Halaman Menu Utama *User* menampilkan informasi *membership* pengguna, termasuk jenis *membership*, tanggal bergabung, tanggal kadaluarsa, dan sisa hari. Terdapat juga *Card Exercise List* dengan deskripsi singkat dan tombol yang mengarahkan ke Halaman *Exercise* saat diklik. Pada Gambar 9, halaman di sebelah kiri merupakan menu utama admin, sedangkan halaman di sebelah kanan merupakan menu utama *user*.



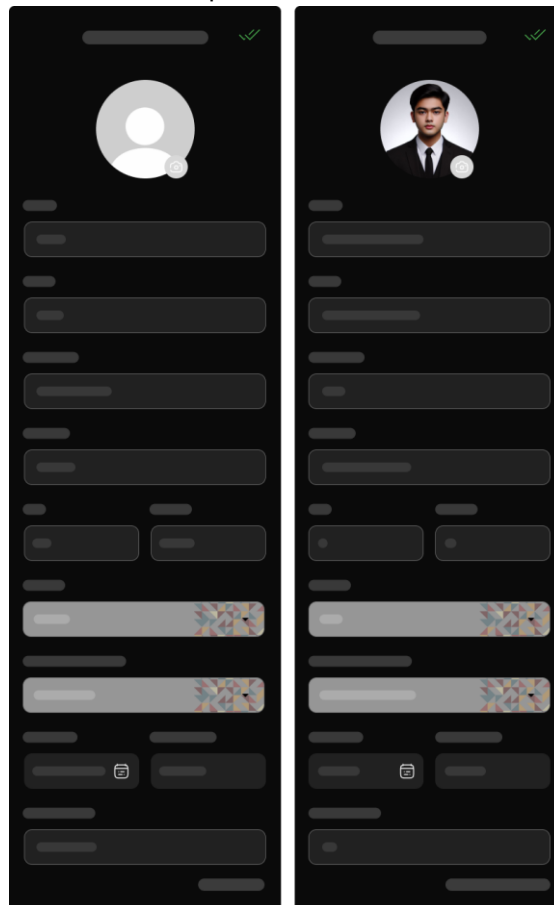
Gambar 9. Desain *Wireframe* Halaman Menu Utama

5. Halaman Admin

Berikut ini adalah rancangan *user interface* dari beberapa halaman yang hanya dapat diakses oleh peran admin.

- Halaman Tambahkan dan Edit *Membership*

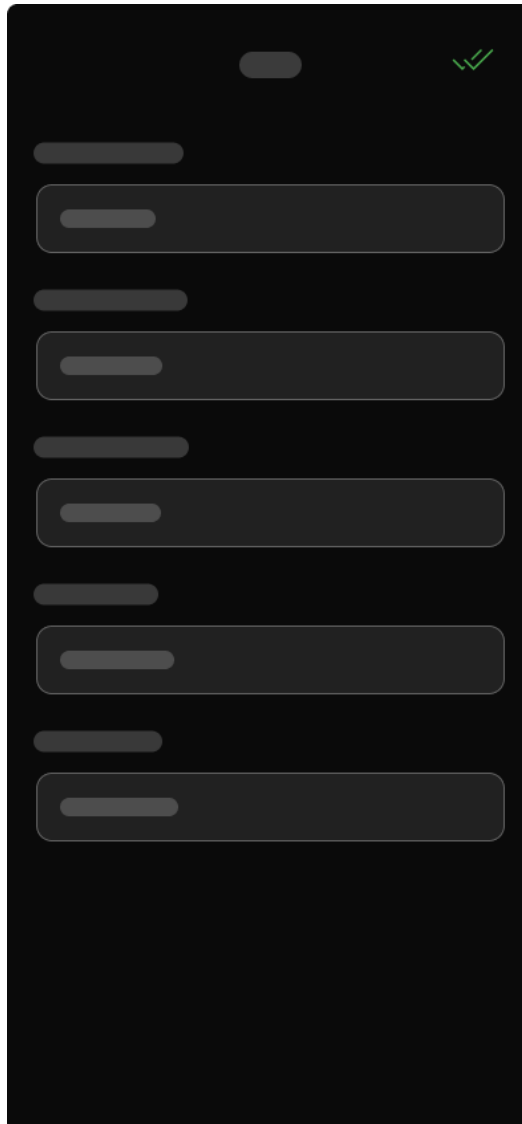
Halaman Tambahkan dan Edit *Membership* memungkinkan admin untuk menambah atau memperbarui data *Membership* yang ingin mendaftar atau sudah terdaftar sebagai *membership*. Admin dapat mengisi atau memperbarui informasi seperti nama, email, kata sandi, alamat, usia, berat badan, jenis kelamin, tipe *membership*, tanggal bergabung, dan diskon. Tanggal kadaluarsa dan total biaya akan dihitung otomatis berdasarkan tipe *membership* dan tanggal bergabung yang dipilih atau diperbarui. Pada Gambar 10, halaman di sebelah kiri merupakan Halaman Tambahkan member, sedangkan halaman di sebelah kanan merupakan Halaman Edit *Membership*.



Gambar 10. Desain Wireframe Halaman Tambahkan dan Edit *Membership*

- Halaman Tarif *Membership*

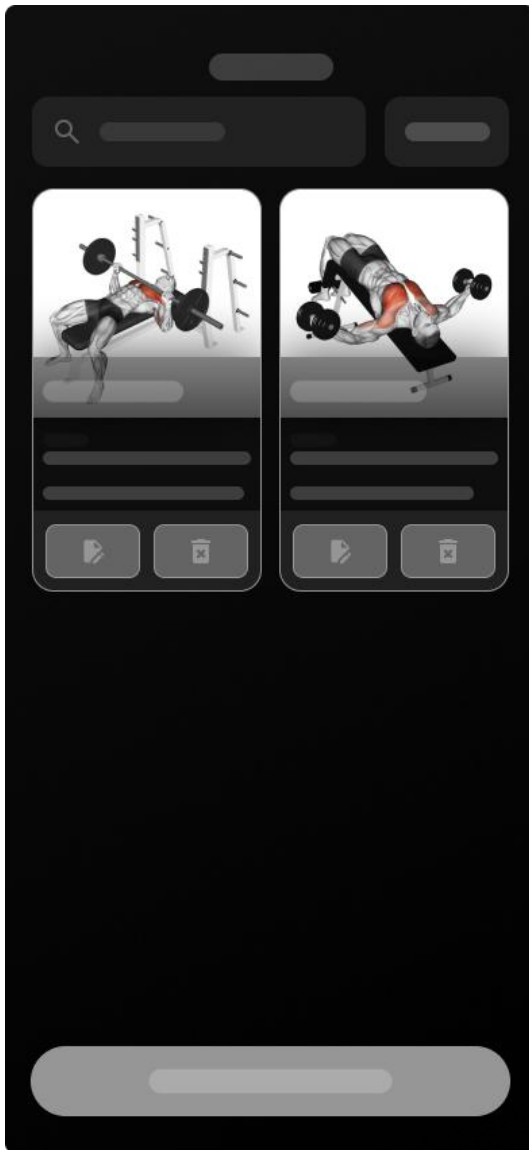
Halaman Tarif *Membership* menampilkan daftar biaya *membership* berdasarkan waktunya, seperti 1 Bulan (Rp.50.000), 3 Bulan (Rp.450.000), 6 Bulan (Rp.950.000), 1 Tahun (Rp.1.800.000), 3 Tahun (Rp.5.400.000), untuk referensi admin dalam mengelola tarif *Membership*. Gambar 11, ditampilkan antarmuka halaman Tarif *Membership* yang berisi daftar pilihan biaya sesuai dengan jangka waktu yang telah ditentukan, sehingga admin dapat dengan mudah melihat, dan melakukan perubahan tarif *membership* sesuai kebutuhan.



Gambar 11. Desain Wireframe Halaman Tarif Membership

- Halaman Daftar Latihan

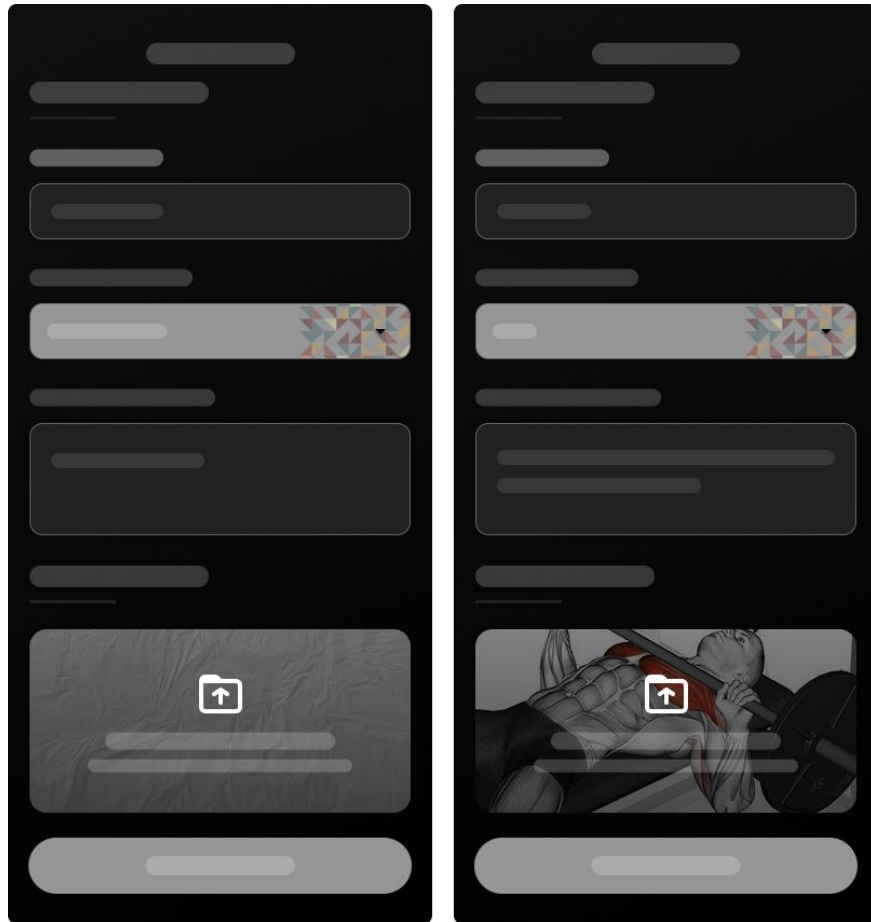
Halaman Daftar latihan menyajikan gerakan latihan *gym*, seperti *Bench Press*, *Dumbell Fly*, dan lainnya, untuk membantu *member* baru memahami penggunaan alat dengan mudah. Setiap latihan dilengkapi dengan kategori dan tingkat intensitas latihan, serta tombol edit dan hapus, serta opsi untuk menambahkan latihan baru melalui tombol "*Add New Exercises*". Gambar 12 menampilkan antarmuka halaman Daftar Latihan yang memuat informasi gerakan latihan beserta kategori, tingkat intensitas latihan, serta fitur pengelolaan seperti edit, hapus, dan penambahan latihan.



Gambar 12. Desain *Wireframe* Halaman Daftar Latihan

- Halaman Tambah dan Edit Latihan

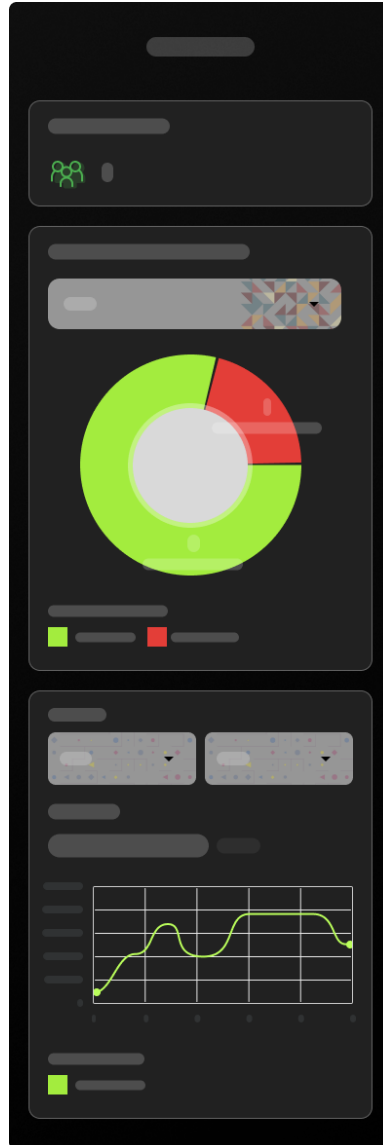
Halaman Tambah dan Edit latihan memungkinkan admin untuk menambahkan atau memperbarui data latihan dengan mengisi informasi seperti nama latihan, kategori dan deskripsi, serta mengunggah GIF gerakan latihan untuk memandu *user* dalam melakukan latihan. Perubahan atau data baru dapat disimpan menggunakan tombol “*Save Exercises*”. Pada Gambar 13, halaman di sebelah kiri merupakan Halaman Tambah Latihan, sedangkan halaman di sebelah kanan merupakan Halaman Edit Latihan.



Gambar 13. Desain Wireframe Halaman Tambah dan Edit Latihan

- Halaman *Dashboard*

Berdasarkan Gambar 14, ditampilkan antarmuka halaman *Dashboard* yang menyajikan informasi ringkas berupa total anggota, status *membership* (aktif dan tidak aktif), pendapatan, serta grafik pendapatan harian yang dapat difilter berdasarkan bulan dan tahun, sehingga memudahkan admin dalam memantau kondisi *gym* secara keseluruhan. Halaman *Dashboard* memberikan gambaran umum kepada admin, termasuk total anggota, rincian status *membership* aktif dan tidak aktif, pendapatan, dan grafik pendapatan harian, dengan filter berdasarkan bulan dan tahun.



Gambar 14. Desain Wireframe Halaman *Dashboard*

6. Halaman *User*

Berikut ini adalah rancangan *user interface* dari beberapa halaman yang hanya dapat diakses oleh peran *user*.

- Halaman Body Mass Index Calculator (BMI)

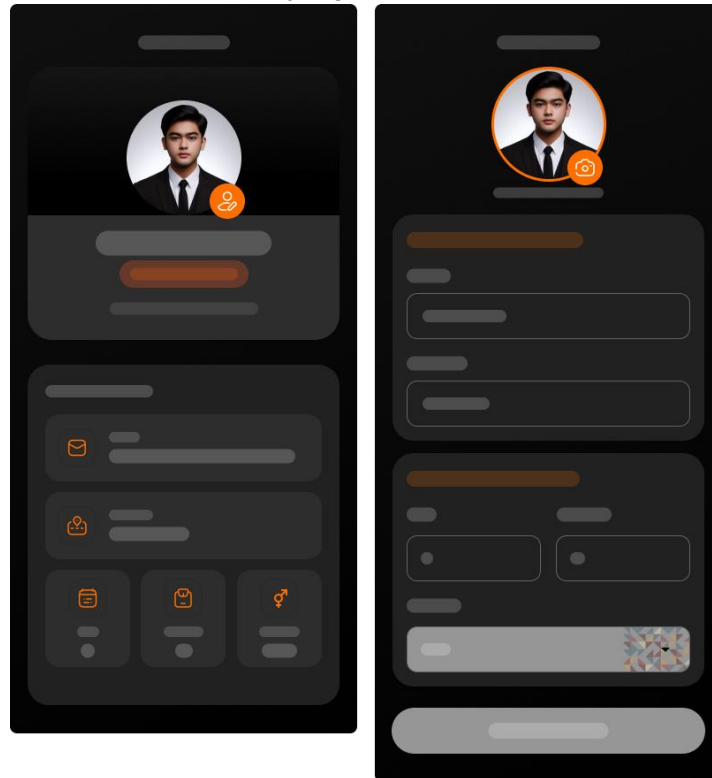
Halaman BMI Calculator adalah fitur untuk user yang memungkinkan mereka menghitung Indeks Massa Tubuh (BMI) dengan memasukkan berat badan (kg) dan tinggi badan (cm). Pada Gambar 15 halaman ini menampilkan hasil BMI, kategori (seperti obesitas, berat badan berlebih, normal, kurang), skala rentang BMI, serta catatan bahwa BMI hanya indikator umum dan saran untuk berkonsultasi dengan ahli gizi untuk analisis lebih mendalam.



Gambar 15. Desain Wireframe Halaman *Body Mass Index*

- Halaman Profil dan Edit Profil

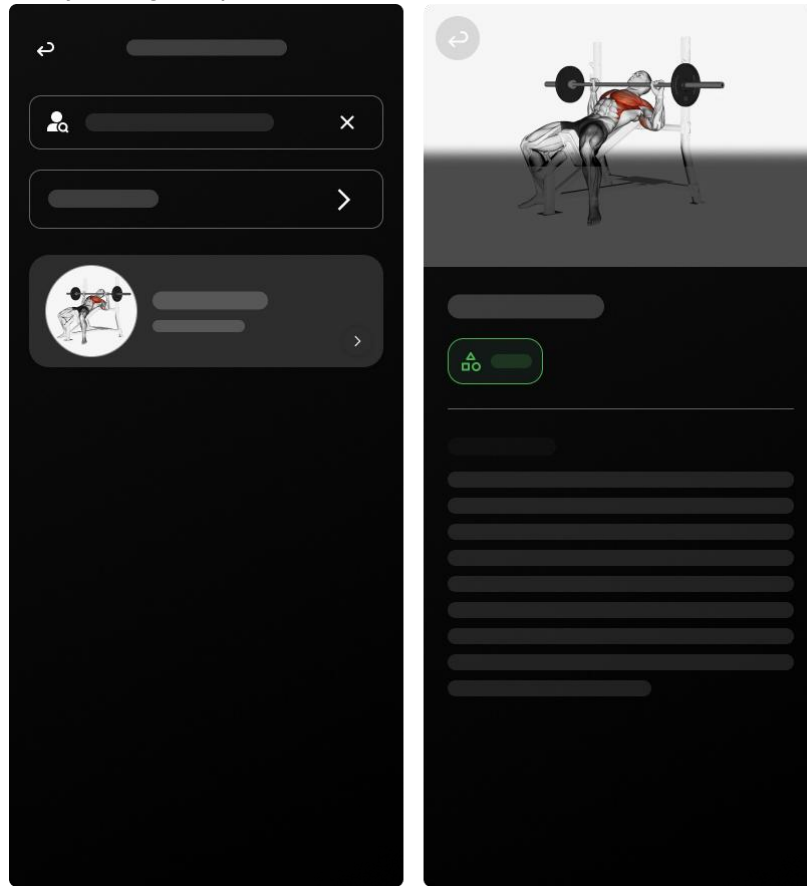
Gambar 16 menampilkan bahwa halaman di sisi kiri adalah Halaman Profil, sementara halaman di sisi kanan adalah halaman Edit Profil. Halaman Profil memperlihatkan data pribadi pengguna, seperti nama, status *membership*, tanggal bergabung, email, alamat, usia, berat badan, dan jenis kelamin, serta memberikan pilihan untuk mengubah profil. Di sisi lain, halaman Edit Profil memberikan kesempatan bagi pengguna untuk memperbarui data pribadi dan mengganti foto profil, lalu menyimpan modifikasi melalui tombol yang disediakan.



Gambar 16. Desain *Wireframe* Halaman Profil dan Edit Profil

- Halaman Lihat Latihan

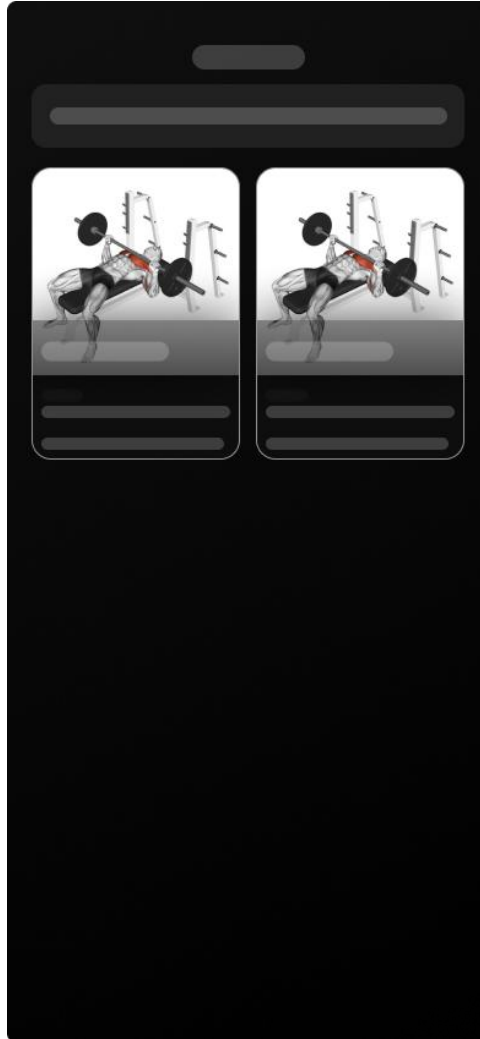
Gambar 17 menunjukkan bahwa halaman di sisi kiri adalah halaman latihan yang menampilkan berbagai macam variasi latihan, sementara halaman di sisi kanan adalah halaman untuk detail latihan yang menyajikan deskripsi latihan berdasarkan kategori beserta gerakannya yang benar. Halaman Lihat Latihan menampilkan daftar Latihan beserta kategorinya untuk referensi latihan anggota di gym. Sedangkan Halaman Detail Latihan menguraikan detail latihan yang meliputi deskripsi dan cara melakukan gerakannya dengan tepat.



Gambar 17. Desain Wireframe Halaman Latihan dan Detail Latihan

- Halaman Rekomendasi Latihan

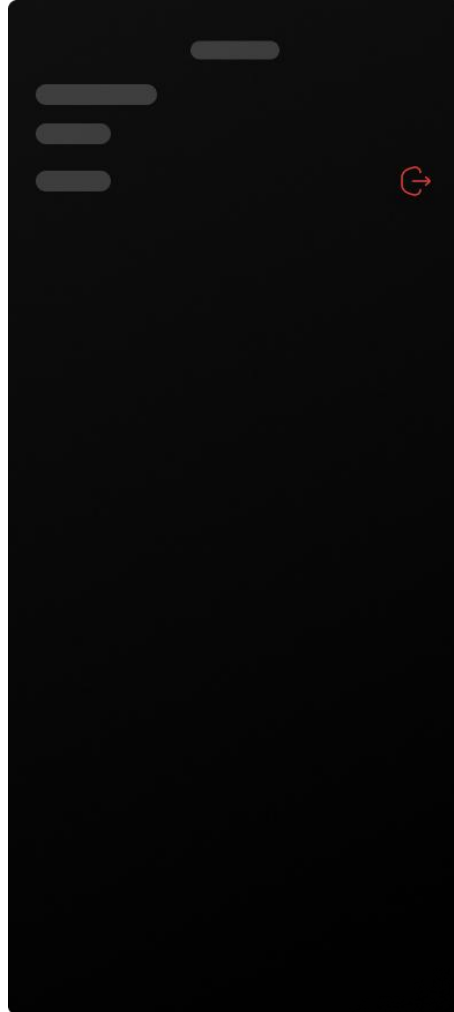
Halaman Rekomendasi latihan menampilkan daftar rekomendasi latihan beserta tingkat intensitas latihannya. Pada Gambar 18, ditampilkan antarmuka halaman Rekomendasi Latihan yang menyajikan daftar latihan beserta tingkat intensitasnya, sehingga user dapat memilih jenis latihan sesuai kemampuan dan kebutuhan.



Gambar 18. Desain Wireframe Halaman Rekomendasi Latihan

- Halaman *Options*

Pada Gambar 19, diperlihatkan *wireframe* halaman *Options* menyediakan fitur sederhana bagi *user* untuk mengelola akun, dengan tombol “*Logout*” yang memungkinkan *user* keluar dari akun dan fitur ganti Bahasa yang mendukung Bahasa Indonesia dan Bahasa Inggris.



Gambar 19. Desain *Wireframe* Halaman *Options*