**PROPOSAL PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK GAME SYSTEM**

PROPOSAL

*Sebagai Tugas Untuk Memenuhi Persyaratan Ujian Tengah Semester*

****

**Dosen Pengampu:**

**Agus Sulaiman S.Kom. M.T**

**DISUSUN OLEH:**

**Abdul Hayyi El Naseer**

**241552010010**

**Fakultas Sains dan Teknologi Jurusan Teknik Informatika**

**Sekolah Tinggi Manajemen Informatika dan Komputer TAZKIA. Raya Dramaga Blok Radar Baru No.8, RT.03/RW.03, Margajaya, Kec. Bogor Bar., Kota Bogor, Jawa Barat 16116, Indonesia**

**SEKOLAH TINGGI MANEJEMAN INFORMATIKA KOMPUTER (STMIK) TAZKIA** **TEKNIK INFORMATIKA**

**2024/2025**

# ABSTRAK

Industri game hiburan berbasis mobile dan PC terus berkembang pesat. Namun, seiring dengan meningkatnya popularitas, muncul kebutuhan untuk menciptakan lingkungan bermain yang sehat dan interaksi sosial yang positif. Banyak game fokus pada kompetisi tanpa memperhatikan aspek kesehatan pemain, seperti risiko kecanduan atau *overplay*. Untuk mengatasi masalah ini, diusulkan pengembangan sebuah "Game System". Aplikasi ini bertujuan untuk menjadi platform hiburan yang seimbang, menghubungkan pemain (Pemain) dalam berbagai mode permainan seperti Story Mode (single player) dan PvP (multiplayer). Sistem ini akan dikelola oleh Sistem itu sendiri, yang secara proaktif memonitor kesehatan bermain. Fitur utama mencakup mode permainan, sistem komunitas (Guild/Chat), serta fitur umpan balik (Feedback). Sebagai mekanisme *responsible gaming*, sistem akan dilengkapi fitur unik Overplay Warning yang memberikan peringatan otomatis jika pemain bermain terlalu lama. Aplikasi ini dirancang sebagai sistem *client-server* berbasis mobile (Android/iOS) dan PC. Diharapkan, aplikasi ini dapat memberikan hiburan yang berkualitas sekaligus mendorong interaksi sosial yang positif dan kebiasaan bermain yang sehat.

Kata Kunci: *Game System, PvP, Story Mode, Interaksi Sosial, Kesehatan Gaming, Anti-Overplay, Sistem Peringatan, Platform Mobile, Komunitas Guild.*

# Daftar isi

Contents

[ABSTRAK 2](#_Toc211802029)

[Daftar isi 3](#_Toc211802030)

[BAB 1 PENDAHULUAN 4](#_Toc211802031)

[1.1 Latar Belakang 4](#_Toc211802032)

[1.2 Rumusan Masalah 4](#_Toc211802033)

[1.3 Batasan Masalah 5](#_Toc211802034)

[1.4 Tujuan 5](#_Toc211802035)

[1.5 Manfaat 5](#_Toc211802036)

[BAB 2 TINJAUAN PUSTAKA 5](#_Toc211802037)

[2.1 Hasil Penelitian Terdahulu 5](#_Toc211802038)

[2.2 Dasar Teori 6](#_Toc211802039)

[BAB 3 METODOLOGI 6](#_Toc211802040)

[3.2 Bahan dan Peralatan yang Digunakan 6](#_Toc211802041)

[3.3 Urutan Pelaksanaan Penelitian 7](#_Toc211802042)

[BAB 4 PERANCANGAN SISTEM 7](#_Toc211802043)

[4.1 Gambaran Umum Sistem 7](#_Toc211802044)

[4.2 Diagram Alur Sistem / Flowchart 8](#_Toc211802045)

[4.2.1. Alur Pemain (Memulai Permainan) 8](#_Toc211802046)

[4.2.2. Alur Sistem (Overplay Warning) 8](#_Toc211802047)

[4.3 Diagram Use Case 8](#_Toc211802048)

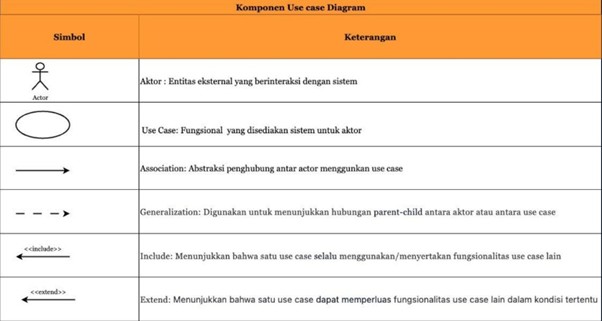
[4.4 Struktur Modul / Arsitektur Sistem 9](#_Toc211802049)

[4.5 Desain Antarmuka / Mockup 9](#_Toc211802050)

[4.6 Spesifikasi Fungsional dan Non-Fungsional 9](#_Toc211802051)

[4.6.2 Kebutuhan Non-Fungsional 10](#_Toc211802052)

[DAFTAR PUSTAKA 10](#_Toc211802053)

****

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Game mobile dan PC telah menjadi salah satu bentuk hiburan utama di seluruh dunia. Selain menyediakan hiburan, game juga berfungsi sebagai platform interaksi sosial. Namun, fokus yang berlebihan pada kompetisi seringkali mengabaikan pentingnya membangun komunitas yang positif dan kebiasaan bermain yang sehat. Masalah seperti perilaku *toxic* dan kecanduan bermain (*overplay*) menjadi tantangan signifikan.

Perlunya sistem yang tidak hanya menghibur tetapi juga proaktif dalam menjaga komunitasnya. Fitur seperti *feedback* dan pelaporan *toxic* sangat penting, namun seringkali reaktif. Diperlukan sebuah mekanisme preventif untuk kesehatan pemain. Oleh karena itu, diperlukan sebuah "Game System" yang mengintegrasikan mode permainan (Story dan PvP) dengan fitur komunitas (Guild/Chat) serta pengingat kesehatan (Anti-Overplay) untuk mendukung hiburan yang sehat dan interaksi sosial yang positif.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sebuah sistem game yang dapat memfasilitasi mode permainan *single player* (Story Mode) dan *multiplayer* (PvP, Kompetisi Rank) secara seimbang?
2. Bagaimana mengimplementasikan sistem komunitas (Guild, Chat) dan sistem umpan balik (Feedback, Report) yang efektif untuk mendorong interaksi sosial positif?
3. Bagaimana membangun sebuah fitur peringatan otomatis (*Overplay Warning*) yang dapat mendeteksi durasi bermain berlebih dan memberikan notifikasi kesehatan kepada pemain?

## 1.3 Batasan Masalah

Ruang lingkup dan batasan untuk pengembangan perangkat lunak ini adalah sebagai berikut:

* Fungsionalitas Utama: Sistem mencakup registrasi/login pemain, mode permainan (Story, PvP, Rank), fitur komunitas (Guild, Chat), sistem feedback, pelaporan (*cheating/toxic*), dan sistem peringatan *overplay*.
* Aktor: Aktor utama yang berinteraksi adalah Pemain dan Sistem (untuk fitur otomatis).
* Platform: Game akan berjalan di platform mobile (Android min. 8.0 dan iOS min. 13).
* Teknis: Sistem merupakan aplikasi *client-server*. Mode PvP membutuhkan koneksi internet stabil, sedangkan Story Mode dapat dimainkan *offline* (opsional).

## 1.4 Tujuan

Tujuan dari pengembangan aplikasi ini adalah:

1. Menyediakan platform digital untuk hiburan yang mencakup mode permainan *solo* (Story Mode) dan *multiplayer* (PvP Match, Kompetisi Rank).
2. Membangun fitur komunitas yang terintegrasi (Guild/Chat) untuk mendorong interaksi sosial yang positif antar pemain.
3. Mengimplementasikan sistem pengingat kesehatan (*Overplay Warning*) secara otomatis untuk mendukung kebiasaan bermain yang sehat.

## 1.5 Manfaat

Manfaat yang diharapkan dari implementasi sistem ini adalah:

* Bagi Pemain: Mendapatkan pengalaman bermain yang seimbang antara kompetisi, cerita, dan interaksi sosial, serta mendapatkan pengingat untuk menjaga kesehatan bermain.
* Bagi Developer: Memiliki data *feedback* dan laporan pemain yang terstruktur untuk perbaikan game, serta membangun citra game yang bertanggung jawab secara sosial.
* Bagi Komunitas: Menciptakan lingkungan bermain yang lebih sehat dan positif, mengurangi perilaku *toxic* melalui fitur pelaporan dan moderasi.

# BAB 2 TINJAUAN PUSTAKA

# 2.1 Hasil Penelitian Terdahulu

Pada bagian ini, umumnya akan dibahas penelitian-penelitian sebelumnya yang relevan dengan pengembangan sistem game, psikologi pemain, desain komunitas online, atau efektivitas sistem peringatan kesehatan (*warning systems*) dalam aplikasi. Analisis terhadap kelebihan dan kekurangan dari sistem yang sudah ada akan menjadi dasar untuk merancang sistem yang lebih baik.

Namun, berdasarkan dokumen yang diberikan, tidak terdapat informasi mengenai penelitian terdahulu yang spesifik.

## 2.2 Dasar Teori

Pengembangan perangkat lunak ini didasarkan pada beberapa konsep dan metodologi rekayasa perangkat lunak.

Software Requirements Specification (SRS)

SRS adalah dokumen yang secara komprehensif mendeskripsikan tujuan, fungsi, kebutuhan, dan batasan dari sebuah perangkat lunak yang akan dikembangkan28. Dokumen ini menjadi landasan komunikasi antara tim pengembang dengan pemangku kepentingan. Dokumen SRS untuk game ini merinci kebutuhan fungsional dan non-fungsional 29, arsitektur client-server 30, dan batasan platform31.

Unified Modeling Language (UML)

UML adalah bahasa pemodelan visual standar yang digunakan untuk merancang, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak. Dalam perancangan ini, digunakan dua jenis diagram UML utama:

1. Use Case Diagram: Diagram ini menggambarkan interaksi antara aktor (pengguna) dengan fungsionalitas utama sistem. Ini membantu untuk memahami apa saja yang dapat dilakukan sistem dari perspektif pengguna. Aktor dalam sistem ini adalah Pemain dan Sistem.
2. Class Diagram: Diagram ini mendeskripsikan struktur statis dari sebuah sistem dengan menunjukkan kelas-kelas sistem, atribut, metode, dan hubungan antar kelas. Ini menjadi cetak biru untuk implementasi kode.

# BAB 3 METODOLOGI

**3.1 Metode yang Digunakan**

Berdasarkan dokumen SRS dan diagram UML yang telah disiapkan, metodologi pengembangan perangkat lunak yang cocok digunakan adalah model Waterfall. Model ini memiliki alur kerja sekuensial yang terstruktur, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Tahapan yang terdokumentasi dengan baik seperti SRS dan diagram UML sangat sesuai dengan pendekatan ini.

## 3.2 Bahan dan Peralatan yang Digunakan

**Perangkat Keras:**

* Infrastruktur server (Cloud) untuk *hosting backend*, database, dan *matchmaking*.
* Perangkat mobile (ponsel pintar/tablet) dengan sistem operasi Android dan iOS untuk pengujian aplikasi klien.
* Komputer/Laptop untuk pengembangan.

**Perangkat Lunak:**

* *Game Engine* (Unity/Unreal Engine).
* Database relasional (MySQL atau PostgreSQL) untuk menyimpan data akun, progres, dan guild.
* REST API untuk komunikasi server-client dan WebSocket untuk fitur *real-time* (battle/chat).
* Alat perancangan UML untuk membuat diagram sistem.

## 3.3 Urutan Pelaksanaan Penelitian

Proses pengembangan akan mengikuti tahapan model Waterfall berikut:

1. Analisis Kebutuhan: Mengumpulkan dan mendefinisikan semua kebutuhan fungsional dan non-fungsional, yang telah didokumentasikan dalam dokumen SRS.
2. Perancangan Sistem: Membuat arsitektur perangkat lunak, termasuk perancangan diagram UML (Use Case dan Class Diagram), skema database, dan desain antarmuka (mockup).
3. Implementasi: Menerjemahkan hasil desain ke dalam kode program yang fungsional menggunakan *game engine* dan teknologi *backend* yang dipilih.
4. Pengujian: Melakukan serangkaian pengujian untuk memastikan semua fungsi berjalan sesuai spesifikasi (mis. *loading* maks 5 detik, *uptime* 99%), tidak ada *bug*, dan memenuhi kebutuhan keamanan (enkripsi data login).
5. Deployment: Merilis aplikasi ke server produksi dan toko aplikasi (Google Play Store & Apple App Store).
6. Pemeliharaan: Melakukan pemeliharaan rutin, perbaikan *bug*, dan pengembangan fitur di masa depan.

# BAB 4 PERANCANGAN SISTEM

## 4.1 Gambaran Umum Sistem

"Game System" adalah sebuah sistem mandiri berbasis *client-server*.

* Client: Terdiri dari aplikasi game *mobile* (Android/iOS) yang digunakan oleh Pemain.
* Server: Bertindak sebagai *backend* yang terhubung ke database pusat, mengelola semua logika bisnis, otentikasi, *matchmaking*, data progres pemain, dan logika *Overplay Warning*.

**Sistem ini dirancang untuk dua jenis aktor utama:**

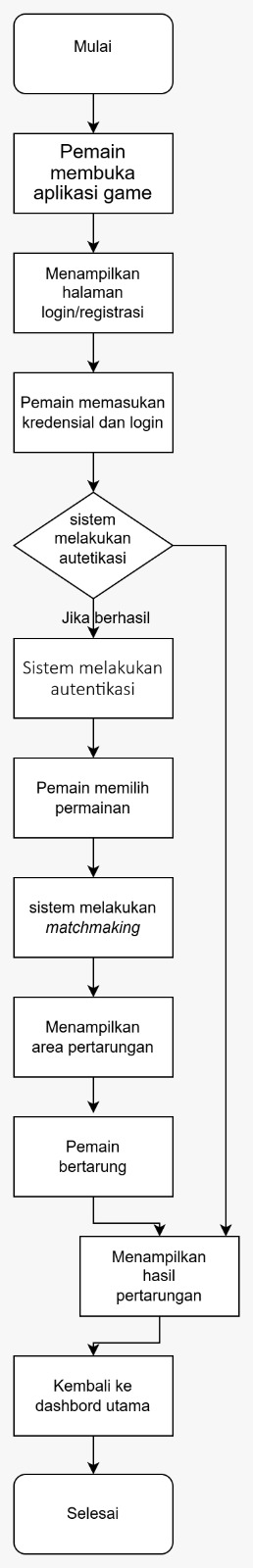
* Pemain: Dapat melakukan registrasi/login, memilih mode permainan (Story, PvP, Rank), bergabung dengan Guild/Chat, mengirim *feedback*, dan melaporkan pemain lain.
* Sistem: Aktor non-manusia yang secara otomatis mengirimkan notifikasi *Overplay Warning* kepada Pemain jika durasi bermain melebihi batas yang ditentukan (mis. > 2 jam)5555.

## 4.2 Diagram Alur Sistem / Flowchart

Diagram alur sistem secara umum (dideskripsikan secara tekstual) adalah sebagai berikut:

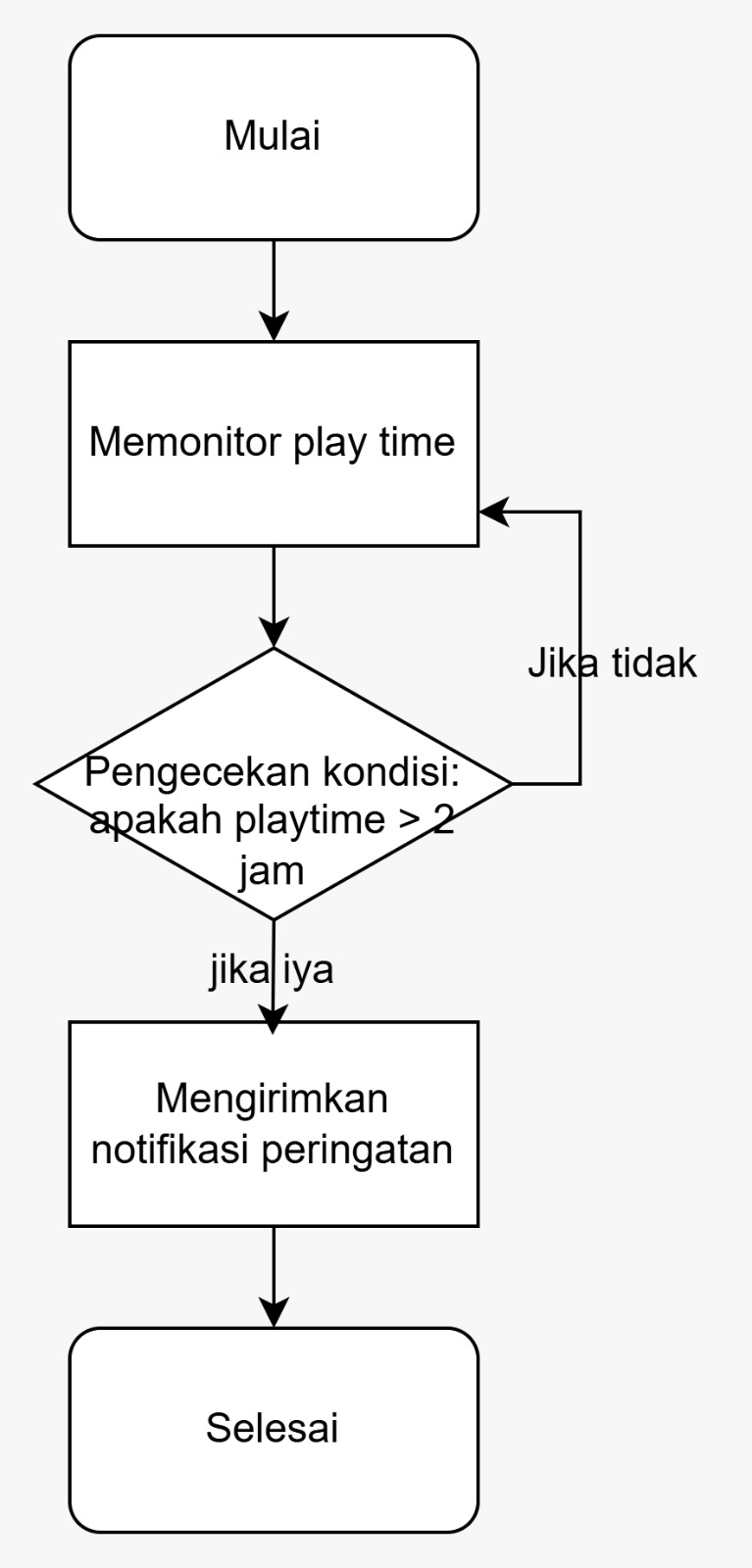
## 4.2.1. Alur Pemain (Memulai Permainan)

1. Mulai
2. Pemain membuka aplikasi game.
3. Menampilkan Halaman Login/Registrasi.
4. Pemain memasukkan kredensial dan *login*.
5. Sistem melakukan autentikasi.
6. (Jika berhasil) Menampilkan Dasbor Utama.
7. Pemain memilih mode permainan (Misal: PvP Match).
8. Sistem melakukan proses *matchmaking*.
9. Menampilkan Arena Pertarungan.
10. Pemain bertarung.
11. Menampilkan Hasil Pertarungan (dan *update rank* jika mode kompetisi).
12. Kembali ke Dasbor Utama.
13. Selesai.



## 4.2.2. Alur Sistem (Overplay Warning)

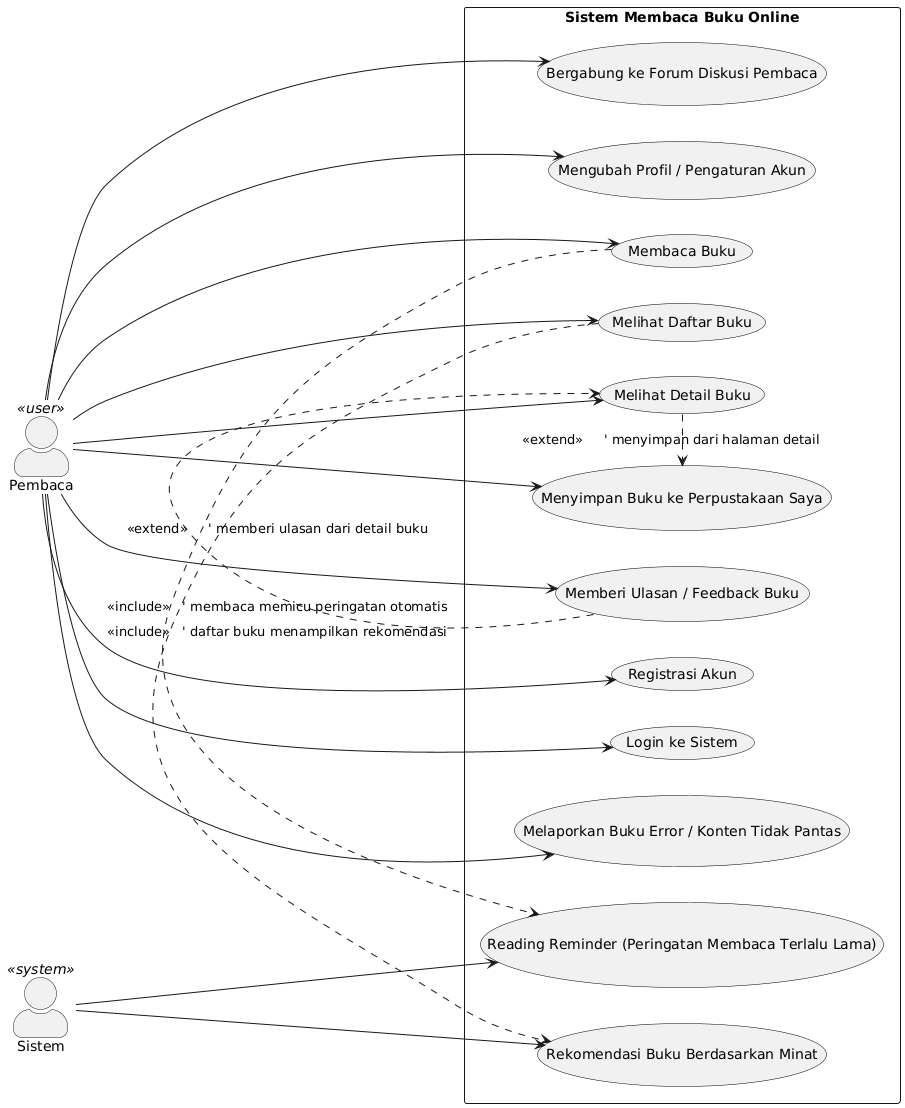
1. Mulai (Proses dipicu oleh Aktor Sistem).
2. Sistem memonitor playTime (durasi bermain) milik Pemain.
3. Pengecekan kondisi: Apakah playTime > 2 jam?.
4. (Jika Ya) -> Sistem mengirimkan notifikasi peringatan (*showWarning*) ke antarmuka Pemain.
5. (Jika Tidak) -> Proses kembali ke langkah 2 (melanjutkan monitoring).
6. Selesai.



## 4.3 Diagram Use Case

Deskripsi Aktor dan Use Case:

* Aktor: Pemain
  + Use Case: Login/Registrasi: Pemain mendaftar atau masuk ke dalam sistem.
  + Use Case: Main Story Mode: Pemain memainkan mode cerita *single player*.
  + Use Case: Main PvP Match: Pemain melawan pemain lain melalui *matchmaking*.
  + Use Case: Ikut Kompetisi Rank: Fungsionalitas tambahan (*extends*) dari PvP Match untuk mode kompetitif.
  + Use Case: Bergabung ke Guild/Chat: Pemain berinteraksi dengan komunitas.
  + Use Case: Kirim Feedback: Pemain mengirimkan saran atau keluhan ke *developer*.
  + Use Case: Lapor Cheating/Toxic: Pemain melaporkan perilaku negatif pemain lain.



* Aktor: Sistem
  + Use Case: Overplay Warning: Sistem secara otomatis memicu dan menampilkan peringatan kepada Pemain jika durasi bermain terlalu lama.

## 4.4 Struktur Modul / Arsitektur Sistem

Struktur statis sistem dan hubungan antar modul dijelaskan dalam Diagram Kelas berikut.

Deskripsi Komponen Diagram Kelas:

* Terdapat kelas utama Player yang memiliki atribut dasar seperti username, password, level, rank, dan playTime.
* Player memiliki fungsi utama seperti login(), selectMode(), sendFeedback(), dan reportPlayer().
* Seorang Player dapat bergabung dengan satu Guild dan mengirimkan banyak (0..\*) Feedback.
* Player berinteraksi dengan GameSystem. GameSystem bertanggung jawab untuk authenticate(), matchmaking(), sendNotification(), dan saveProgress().
* GameSystem mengelola berbagai mode permainan seperti StoryMode dan PvPMode.
* GameSystem juga bertanggung jawab untuk memicu OverplayWarning, yang menggunakan fungsi checkPlayTime() dan showWarning().

## 4.5 Desain Antarmuka / Mockup

Desain antarmuka akan dirancang dengan prinsip yang bersih, intuitif, dan responsif untuk platform mobile. Fokus utama adalah kemudahan navigasi antar mode permainan dan fitur komunitas.

Berdasarkan dokumen SRS, mockup UI yang detail untuk menu utama, battle, chat, dan form feedback akan dilampirkan secara terpisah69. (Catatan: Mockup tidak disertakan dalam file yang diunggah, hanya dirujuk).

## 4.6 Spesifikasi Fungsional dan Non-Fungsional

Spesifikasi kebutuhan sistem diambil dari dokumen SRS.

**4.6.1 Kebutuhan Fungsional**

**Tabel 4: Kebutuhan Fungsional**

|  |  |
| --- | --- |
| ID | Deskripsi |
| FR1 | Sistem menyediakan akun/login pemain. |
| FR2 | Pemain dapat memilih mode permainan (Story, PvP, Rank). |
| FR3 | Sistem kompetisi rank dengan leaderboard. |
| FR4 | Fitur komunitas (chat global, guild/clan). |
| FR5 | Sistem feedback untuk request update/balance. |
| FR6 | Sistem peringatan otomatis setelah main > 2 jam. |
| FR7 | Fitur report (melaporkan toxic/cheating). |

## 4.6.2 Kebutuhan Non-Fungsional

**Tabel 5: Kebutuhan Non-Fungsional 72**

|  |  |
| --- | --- |
| ID | Deskripsi |
| NFR1 | Kinerja: Loading screen maksimal 5 detik. |
| NFR2 | Skalabilitas: Mendukung minimal 10.000 pemain aktif bersamaan. |
| NFR3 | Performa: Game responsif di smartphone RAM 3 GB. |
| NFR4 | Keamanan: Semua data login dienkripsi (AES-256). |
| NFR5 | Ketersediaan: Uptime server minimal 99%. |

# DAFTAR PUSTAKA

**(Catatan: Daftar pustaka ini adalah contoh referensi standar dalam rekayasa perangkat lunak, serupa dengan gaya pada dokumen proposal "Bank Sampah", karena tidak ada daftar pustaka yang spesifik dalam dokumen SRS yang diberikan.)**

**Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley Professional.**

**Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.**

**Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modeling Language Reference Manual* (2nd ed.). Addison-Wesley Professional.**

**Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.**

**LAMPIRAN**

**Berikut dokumen-dokumen pendukung tambahan, antara lain:**

* **Dokumen Lengkap *Software Requirements Specification (SRS)*73.**
* **Rancangan *Mockup* Antarmuka Pengguna (UI)74.**
* **Skema Desain Database (Tabel users, characters, matches, feedback, guilds)75.**