

TUGAS PRAKTIKUM
ANALISIS DESAIN & BEORIENTASIOBJECT
“Disusun untuk memenuhi tugas mata kuliah analisis desain & berorientasi
objek” Dosen pengampu: Ridwan Setiawan, S.T. M.Kom



Disusun oleh

Fahreza Pratama H

2206144

INSTITUTE TEKNOLOGI GARUT
TTEKNIK INFORMATIKA

2024

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia- Nya sehingga laporan praktikum Analisis Desain Berorientasi Objek dengan judul "Pembuatan Activity Diagram Sistem Twitch Streaming" dapat diselesaikan tepat waktu.

Laporan ini disusun sebagai bagian dari tugas mata kuliah Praktikum Analisis Desain Berorientasi Objek. Tujuan utama dari praktikum ini adalah untuk mengaplikasikan konsep dan teknik analisis desain berorientasi objek dalam konteks sistem yang nyata dan relevan, dalam hal ini platform streaming Twitch.

Dalam kesempatan ini, kami ingin mengucapkan terima kasih kepada:

- 1. Ridwan Setiawan, S.T. M.Kom, selaku dosen pengampu mata kuliah Praktikum Analisis Desain Berorientasi Objek.**
- 2. Seluruh staf laboratorium yang telah memfasilitasi kegiatan praktikum.**
- 3. Rekan-rekan mahasiswa yang telah berpartisipasi dan memberikan masukan berharga selama proses pengerjaan.**

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saranyang membangun sangat kami harapkan untuk perbaikan di masa mendatang.

Akhir kata, kami berharap laporan ini dapat memberikan manfaat dan pemahaman yang lebih baik tentang penggunaan Activity Diagram dalam menganalisis sistem streaming online

DAFTAR ISI

| | |
|--|-----------|
| KATA PENGANTAR | 2 |
| DAFTAR ISI | 3 |
| DAFTAR GAMBAR | 4 |
| DAFTAR TABLE | 5 |
| 1.1. LATAR BELAKANG..... | 6 |
| 1.2. RUMUSAN MASALAH | 6 |
| 1.3. TUJUAN MASALAH..... | 6 |
| BAB II TINJAUAN PUSTAKA | 7 |
| 2.1. ANALISIS DESAIN BERORIENTASI OBJEK..... | 7 |
| 2.2. ACTIVITY DIAGRAM | 7 |
| 2.4. SISTEM TWITCH STREAMING | 8 |
| BAB III HASIL DAN PEMBAHASAN..... | 9 |
| 3.1. PEMBUATAN ACTIVITY DIAGRAM TWITCHSTREAMING | 9 |
| 3.2. HASIL ACTIVITY DIAGRAM | 11 |
| 3.3. PEMBUATAN USE CASE DIAGRAM..... | 14 |
| 3.4. HASIL USE CASE DIAGRAM | 15 |
| 3.5. SCENARIO USE CASE..... | 15 |
| 3.6. PEMBUATAN SEQUENCE DIAGRAM..... | 18 |
| 3.7. PEMBUATAN CLASS DIAGRAM | 20 |
| BAB IV HASIL DAN PEMBAHASAN | 24 |
| 4.1. KESIMPULAN | 24 |
| 4.2 SARAN..... | 24 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1 3 hasil sequenced diagrama | 19 |
| Gambar 3 2 Hasil Class diagram..... | 23 |

DAFTAR TABLE

| | |
|--|----|
| Table 3 1 Table Scenario use case..... | 15 |
|--|----|

BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG

Twitch adalah platform streaming langsung yang populer, terutama di kalangan gamers. Dengan meningkatnya minat terhadap live streaming, penting untuk memahami alur kerja dan proses yang terlibat dalam sistem Twitch streaming. Analisis dan desain berorientasi objek dapat membantu dalam memahami dan mengoptimalkan sistem ini.

1.2. RUMUSAN MASALAH

- 1. Bagaimana alur kerja sistem Twitch streaming dapat direpresentasikan dalam bentuk Activity Diagram?**
- 2. Apa saja komponen utama yang terlibat dalam proses Twitch streaming?**
- 3. Bagaimana interaksi antara streamer, viewer, dan sistem Twitch dapat digambarkan secara visual?**

1.3. TUJUAN MASALAH

- 1. Membuat Activity Diagram yang merepresentasikan alur kerja sistem Twitch streaming.**
- 2. Mengidentifikasi dan menjelaskan komponen utama dalam proses Twitch streaming.**
- 3. Menggambarkan interaksi antara streamer, viewer, dan sistem Twitch secara visual dan terstruktur.**

BAB II

TINJAUAN PUSTAKA

2.1. ANALISIS DESAIN BERORIENTASI OBJEK

Analisis Desain Berorientasi Objek (ADBO) adalah pendekatan dalam pengembangan perangkat lunak yang menggunakan konsep objek untuk merepresentasikan entitas dalam sistem. Menurut Booch et al. (2007), ADBO memungkinkan pengembang untuk memodelkan sistem kompleks dengan cara yang lebih alami dan intuitif.

Prinsip-prinsip utama ADBO meliputi:

- 1. Enkapsulasi:** Menyembunyikan detail implementasi internal objek.
- 2. Pewarisan:** Memungkinkan objek untuk mewarisi sifat dan perilaku dari objek lain.
- 3. Polimorfisme:** Kemampuan objek untuk memiliki banyak bentuk.

2.2. ACTIVITY DIAGRAM

Activity Diagram adalah salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan aspek dinamis dari sistem. Menurut Rumbaugh et al. (2004), Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis.

Komponen utama Activity Diagram meliputi:

- 1. Action:** Representasi dari langkah tunggal dalam aktivitas.
- 2. Control Flow:** Menunjukkan urutan eksekusi actions.
- 3. Decision Node:** Titik di mana aliran kontrol dapat mengambil satu dari beberapa jalur alternatif.
- 4. Fork dan Join:** Menunjukkan aliran paralel dan sinkronisasi.

2.3. USE CASE DIAGRAM

Use Case Diagram adalah salah satu diagram dalam UML yang digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna eksternal. Menurut Rumbaugh et al. (2004), Use Case Diagram menunjukkan interaksi antara aktor (pengguna eksternal) dengan sistem dan menggambarkan fitur-fitur yang disediakan oleh sistem.

2.4. SISTEM TWITCH STREAMING

Twitch adalah platform streaming langsung yang fokus pada konten gaming, meskipun juga mencakup berbagai kategori lainnya. Menurut Hamilton et al. (2014), Twitch telah menjadi platform dominan untuk live streaming game, dengan jutaan pengguna aktif setiap harinya.

Komponen utama dalam sistem Twitch streaming meliputi:

- 1. Streamer: Pengguna yang melakukan siaran langsung.**
- 2. Viewer: Pengguna yang menonton siaran langsung.**
- 3. Chat: Fitur yang memungkinkan interaksi real-time antara streamer dan viewer.**
- 4. Monetisasi: Sistem yang memungkinkan streamer mendapatkan pendapatan melalui donasi, langganan,**

BAB III HASIL DAN PEMBAHASAN

3.1. PEMBUATAN ACTIVITY DIAGRAM TWITCHSTREAMING

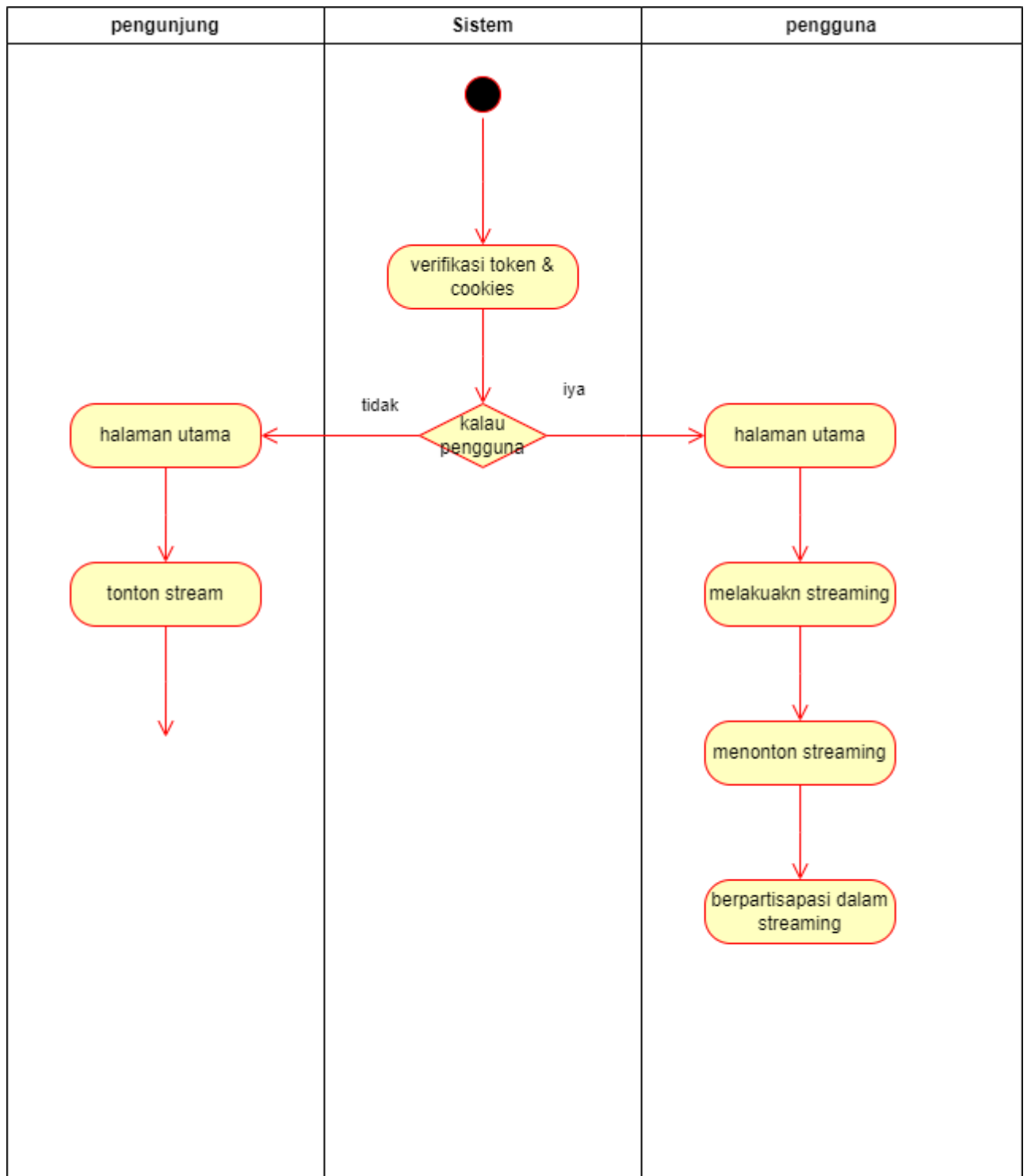
- 1. Identifikasi Aktor dan Aktivitas Utama:**
 - a. Saya mulai dengan mengidentifikasi aktor utama: Streamer, Viewer, dan Sistem Twitch.
 - b. Kemudian, saya membuat daftar aktivitas utama yang terjadi dalam proses streaming.
- 2. Penentuan Titik Awal dan Akhir:**
 - a. Saya menentukan "Streamer memulai stream" sebagai titik awal.
 - b. "Streamer mengakhiri stream" dan "Viewer meninggalkan stream" ditetapkan sebagai titik akhir.
- 3. Pemetaan Alur Aktivitas**
 - a. Saya mulai menggambar alur dari titik awal, menghubungkan setiap aktivitas dengan panah.
 - b. Aktivitas seperti "Sistem memverifikasi akun", "Viewer bergabung stream", dan "Interaksi dalam chat" ditambahkan ke diagram.
- 4. Penambahan Decision Points**
 - a. Saya menambahkan decision point "Verifikasi akun berhasil?" setelah aktivitas verifikasi.
 - b. Decision point lain seperti "Stream masih berlangsung?" dan "Viewer ingin berdonasi?" juga ditambahkan.
- 5. Penggambaran Alur Alternatif**
 - a. Untuk setiap decision point, saya menggambar alur alternatif. Misalnya, jika verifikasi gagal, alur kembali ke "Streamer memulai stream".
- 6. Penambahan Parallel Activities**
 - a. Saya menggunakan fork dan join untuk menunjukkan aktivitas paralel, seperti interaksi simultan antara streamer dan viewer dalam chat.
- 7. Penambahan Swimlanes**
 - a. Untuk memperjelas tanggung jawab setiap aktor, saya membagi diagram menjadi tiga swimlane: Streamer, Sistem Twitch, dan Viewer.
- 8. Peninjauan dan Perbaikan**

- a. **Setelah menyelesaikan draft pertama, saya meninjau diagram untuk memastikan semua alur logis dan konsisten.**
- b. **Saya melakukan beberapa perbaikan, seperti menambahkan detail pada proses donasi dan memperjelas alur saat stream berakhir.**

9. Finalisasi Diagram:

- a. **Terakhir, saya memastikan semua elemen terhubung dengan benar dan menambahkan keterangan jika diperlukan.**

3.2. HASIL ACTIVITY DIAGRAM



Gambar 1 akitivy diagram twich stream berdasarkan akses pengguna maupun pengunjung

Diagram di atas menggambarkan alur kerja sistem Twitch streaming, mulai dari streamer memulai stream hingga stream berakhir. Beberapa elemen kunci dalam diagram ini meliputi:

1. **Initial node:** Ditandai dengan lingkaran hitam solid di bagian atas diagram.

2. **Action nodes:** Direpresentasikan oleh kotak dengan sudut melengkung, misalnya "Streamer memulai stream".
3. **Decision nodes:** Ditunjukkan dengan diamond, seperti "Verifikasi akun berhasil?".
4. **Fork dan join:** Digunakan untuk menunjukkan aktivitas paralel dalam interaksi chat.
5. **Final nodes:** Ditandai dengan lingkaran hitam dengan lingkaran putih di luarnya.

3.3. PEMBUATAN USE CASE DIAGRAM

1. Identifikasi Aktor:

- a. Saya mengidentifikasi aktor utama dalam sistem Twitch streaming:
User (pengguna umum yang bisa menjadi viewer atau streamer).

2. Identifikasi Use Case:

Saya membuat daftar fungsionalitas utama yang disediakan oleh sistem Twitch, seperti:

- a. Halaman utama
- b. Register
- c. Login
- d. Melihat daftar streaming
- e. Nonton streaming
- f. Melakukan streaming
- g. Melihat profile
- h. Dan lain-lain

3. Penggambaran Aktor dan Use Case

- a. Saya menggambar aktor menggunakan simbol stick figure.
- b. Use case digambarkan dengan oval dan diberi label sesuai fungsionalitasnya.

4. Penentuan Relasi

- a. Saya menghubungkan aktor dengan use case yang relevan menggunakan garis.
- b. Untuk relasi antar use case, saya menggunakan relasi <<include>> dan <<extend>> sesuai kebutuhan.

5. Penambahan Relasi Khusus

- a. Saya menambahkan relasi <<include>> untuk menunjukkan fungsionalitas yang selalu dijalankan sebagai bagian dari use case lain
- b. Relasi <<extend>> ditambahkan untuk menunjukkan fungsionalitas opsional atau kondisional.

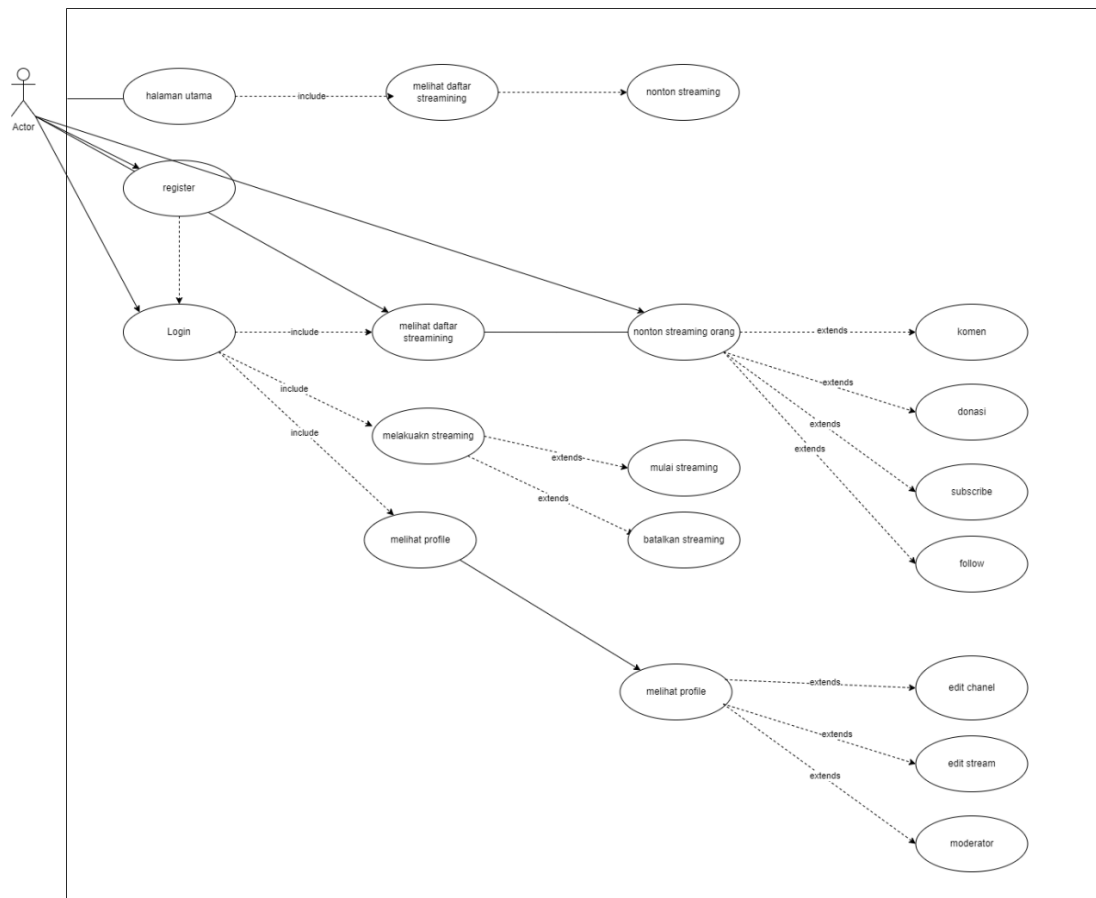
6. Peninjauan dan Perbaikan

- a. Setelah menyelesaikan draft pertama, saya meninjau diagram untuk memastikan semua interaksi telah tercakup.
- b. Saya melakukan perbaikan yang diperlukan, seperti menambahkan

7. Finalisasi Diagram use case

- a. Terakhir, saya memastikan semua elemen terhubung dengan benar dan tepat

3.4. HASIL USE CASE DIAGRAM



3.5. SCENARIO USE CASE

| No | | Aktor | Sistem |
|----|------------------------------|---|---|
| 1 | Register | Memilih opsi "register". Mengisi informasi yang diperlukan | Menampilkan form registrasi. Memvalidasi informasi. Membuat akun user baru. Mengkonfirmasi registrasi berhasil |
| 2 | Login | Memilih opsi "login". Memasukkan kredensial | Menampilkan form login. Memverifikasi kredensial. Memberikan akses ke akun user |
| 3 | Melihat Halaman Utama | Mengakses halaman utama | Menampilkan konten utama dan opsi |

| | | | |
|----------|---------------------------------|--|--|
| | | | navigasi |
| 4 | Melihat Daftar Streaming | Memilih opsi "lihat daftar streaming" | Mengambil konten streaming yang |

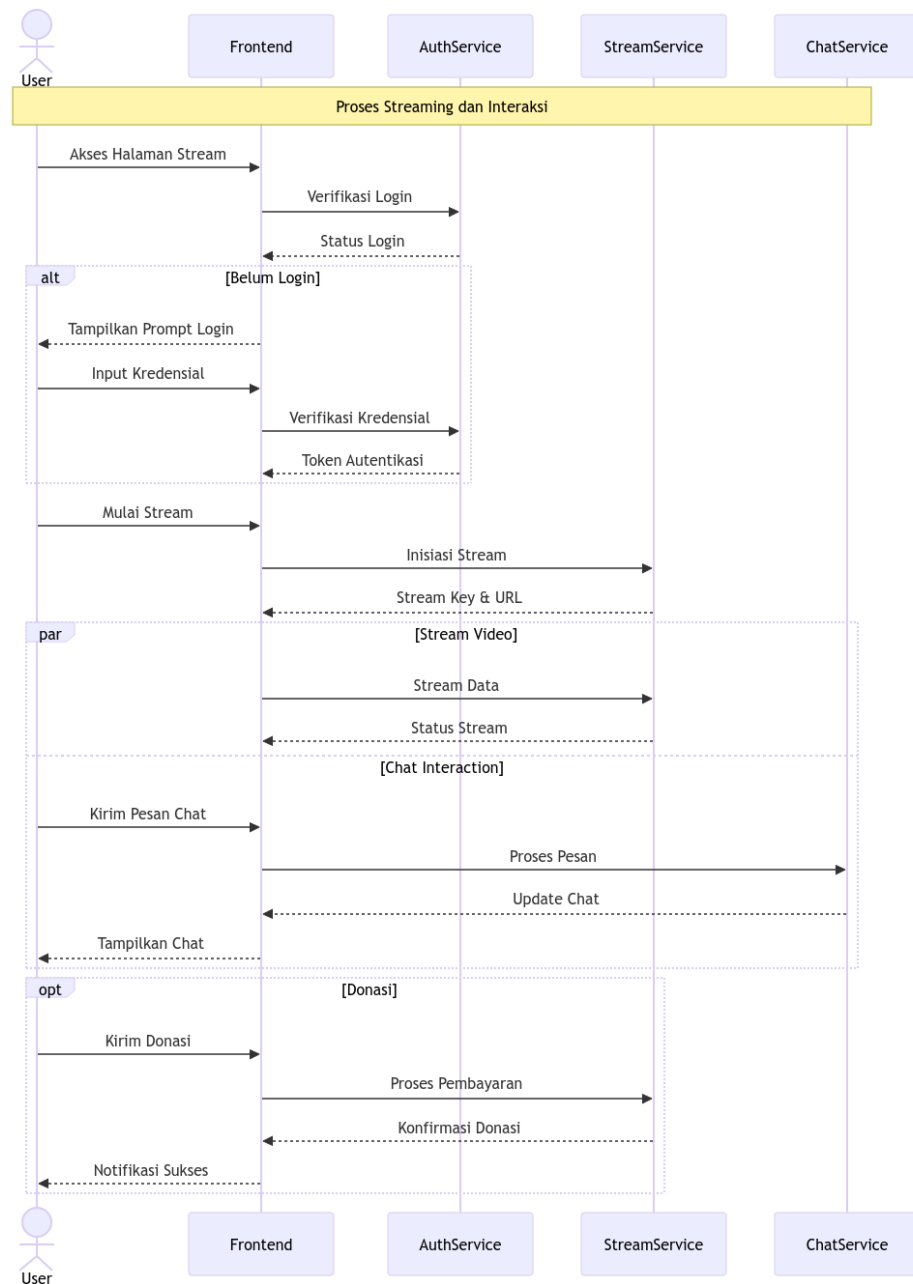
| | | | |
|---|----------------------------|--|--|
| | | | tersedia. Menampilkan daftar opsi streaming |
| 5 | Menonton Streaming | Memilih konten streaming tertentu. Menonton konten | Memulai pemutaran streaming |
| 6 | Melihat Profil | Memilih opsi "lihat profil" | Mengambil informasi profil user 2. Menampilkan profil user |
| 7 | Edit Profil | Memilih opsi "edit profil". Memodifikasi informasi. Mengirim perubahan | Menampilkan field profil yang dapat diedit. Memperbarui informasi profil |
| 8 | Mengelola Streaming | Mengakses opsi manajemen streaming. Melakukan tindakan pengelolaan (edit channel, edit stream, moderasi) | Menampilkan antarmuka manajemen streaming. Memproses tindakan pengelolaan |
| 9 | Berinteraksi dengan Konten | Memilih opsi interaksi (komentar, donasi, subscribe, follow) | Memproses interaksi user2. Memperbarui statistik konten/creator |
| | | | |

Table 3 1 Table Scenario use case

3.6. PEMBUATAN SEQUENCE DIAGRAM

Sequence diagram dibuat untuk menggambarkan interaksi antar objek dalam sistem Twitch streaming secara berurutan. Langkah-langkah pembuatan sequence diagram meliputi:

- 1. Identifikasi Objek dan Aktor**
 - a. User sebagai aktor utama**
 - b. Frontend sebagai antarmuka sistem**
 - c. AuthService untuk manajemen autentikasi**
 - d. StreamService untuk manajemen streaming**
 - e. ChatService untuk manajemen chat**
- 2. Penentuan Alur Interaksi**
 - a. Alur autentikasi**
 - b. Proses streaming**
 - c. Interaksi chat**
 - d. Sistem donasi**
- 3. Penggambaran Messages**
 - a. Synchronous messages**
 - b. Asynchronous messages**
 - c. Return messages**
 - d. Alternative flows**



Gambar 3 1 hasil sequenced diagrama

3.7. PEMBUATAN CLASS DIAGRAM

LANGKAH-LANGKAH PEMBUATAN CLASS DIAGRAM

1. Identifikasi Kelas

- 1. Menganalisis entitas utama dalam sistem**
- 2. Menentukan kelas-kelas yang dibutuhkan**
- 3. Memisahkan tanggung jawab antar kelas**

2. Penentuan Atribut

- 1. Mengidentifikasi properti setiap kelas**
- 2. Menentukan tipe data yang sesuai**
- 3. Menentukan visibility (public/private)**

3. Penentuan Method

- 1. Mengidentifikasi operasi yang dapat dilakukan**
- 2. Menentukan parameter dan return type**
- 3. Menentukan visibility method**

4. Penentuan Relasi

- 1. Menganalisis hubungan antar kelas**
- 2. Menentukan jenis relasi (asosiasi, agregasi, komposisi)**
- 3. Menentukan multiplicity**

DESKRIPSI KELAS

1. User

- 1. Merepresentasikan pengguna sistem**
- 2. Atribut: userID, username, email, password, profilePicture, followers**
- 3. Method: register, login, updateProfile, followUser, getFollowers**

2. Stream

- 1. Merepresentasikan sesi streaming**
- 2. Atribut: streamID, title, description, startTime, endTime, viewer, category**
- 3. Method: startStream, endStream, updateTitle, updateCategory, getViewerCount**

3. Channel

- 1. Merepresentasikan channel streamer**
- 2. Atribut: channelID, ownerID, description, followers, subscribers**
- 3. Method: updateDescription, getStats, addModerator, removeModerator**

4. ChatMessage

- 1. Merepresentasikan pesan dalam chat**
- 2. Atribut: messageID, userID, content, timestamp, isModerated**
- 3. Method: sendMessage, deleteMessage, moderateMessage**

5. Subscription

- 1. Merepresentasikan langganan channel**
- 2. Atribut: subID, userID, channelID, tier, startDate, endDate**
- 3. Method: subscribe, unsubscribe, upgradeTier**

6. Donation

- 1. Merepresentasikan donasi ke streamer**
- 2. Atribut: donationID, amount, message, timestamp**
- 3. Method: processDonation, refundDonation, getDonationHistory**

RELASI ANTAR KELAS

1. User - Stream (1 to Many)

- 1. Satu user dapat memiliki banyak stream**
- 2. Relasi ini menunjukkan aktivitas streaming user**

2. User - Channel (1 to 1)

- 1. Setiap user dapat memiliki satu channel**
- 2. Merepresentasikan kepemilikan channel**

3. User - ChatMessage (1 to Many)

- 1. Satu user dapat mengirim banyak chat message**
- 2. Menunjukkan aktivitas chat user**

4. Channel - Stream (1 to Many)

- 1. Satu channel dapat memiliki banyak stream**
- 2. Merepresentasikan konten channel**

5. Channel - Subscription (1 to Many)

- 1. Satu channel dapat memiliki banyak subscriber**
- 2. Menunjukkan sistem berlangganan**

6. Channel - Donation (1 to Many)

- 1. Satu channel dapat menerima banyak donasi**
- 2. Merepresentasikan sistem donasi**

1. Struktural

- 1. Memberikan gambaran jelas struktur sistem**
- 2. Memudahkan pemahaman hubungan antar komponen**
- 3. Membantu dalam perencanaan database**

2. Pengembangan

- 1. Panduan implementasi kode**
- 2. Memudahkan pembagian tugas development**
- 3. Referensi untuk testing**

3. Maintenance

- 1. Memudahkan identifikasi impact perubahan**
- 2. Membantu dalam refactoring kode**
- 3. Dokumentasi sistem yang terstruktur**

IMPLEMENTASI DATABASE

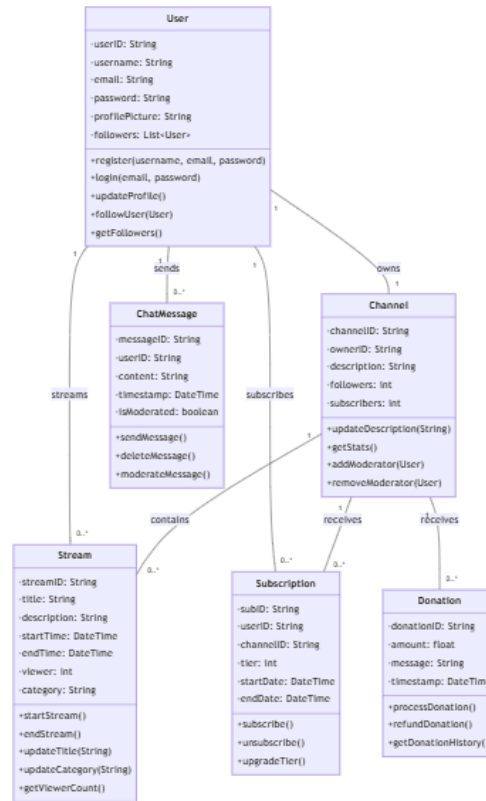
Class diagram ini dapat digunakan sebagai dasar pembuatan skema database:

1. Tabel Utama

- 1. Users**
- 2. Streams**
- 3. Channels**
- 4. ChatMessages**
- 5. Subscriptions**
- 6. Donations**

2. Relasi Database

- 1. Foreign keys sesuai relasi class diagram**
- 2. Indeks untuk optimasi query**
- 3. Constraints untuk integritas data**



Gambar 3 2 Hasil Class diagram

BAB IV HASIL DAN PEMBAHASAN

4.1. KESIMPULAN

Berdasarkan hasil analisis dan perancangan sistem Twitch streaming menggunakan pendekatan berorientasi objek, dapat ditarik beberapa kesimpulan:

1. Analisis Kebutuhan Sistem

- a. Use Case Diagram berhasil mengidentifikasi kebutuhan fungsional sistem dari perspektif pengguna dengan jelas
- b. Interaksi antara user dan sistem telah terdefinisi dengan baik melalui berbagai use case yang saling terhubung
- c. Pemisahan peran antara viewer dan streamer telah terstruktur dengan tepat

2. Perancangan Sequence Diagram

- a. Sequence diagram menunjukkan alur interaksi yang kompleks namun terorganisir dengan baik
- b. Proses autentikasi, streaming, dan interaksi real-time telah tergambar secara sequential
- c. Penanganan proses parallel seperti chat dan streaming telah diimplementasikan dengan efektif

3. Implementasi Class Diagram

- a. Struktur class menunjukkan modularitas sistem yang baik
- b. Relasi antar class telah terdefinisi dengan jelas dan sesuai kebutuhan
- c. Encapsulation dan inheritance telah diterapkan sesuai prinsip OOP

4. Keseluruhan Sistem

- a. Ketiga diagram (Use Case, Sequence, dan Class) saling melengkapi dalam menggambarkan sistem
- b. Arsitektur sistem mendukung skalabilitas dan maintainability
- c. Sistem telah dirancang dengan mempertimbangkan aspek keamanan dan performa

4.2 SARAN

Berdasarkan hasil analisis dan perancangan, berikut beberapa saran untuk pengembangan sistem lebih lanjut:

1. Pengembangan Teknis

- a. Implementasikan sistem caching untuk mengoptimalkan performa streaming

- b. Tambahkan mekanisme load balancing untuk menangani traffic yang tinggi
 - c. Kembangkan sistem backup dan recovery untuk menjaga keamanan data
 - d. Implementasikan mikroservis untuk meningkatkan skalabilitas sistem
- 2. Peningkatan Fitur
 - a. Tambahkan fitur clip creation untuk highlight moment penting
 - b. Implementasikan sistem reward yang lebih kompleks
 - c. Kembangkan integrasi dengan platform sosial media
 - d. Tambahkan fitur analitik yang lebih detail untuk streamer
- 3. Aspek Keamanan
 - a. Tingkatkan sistem autentikasi dengan implementasi 2FA
 - b. Tambahkan enkripsi end-to-end untuk chat privat
 - c. Implementasikan sistem deteksi konten berbahaya secara otomatis
 - d. Tingkatkan perlindungan terhadap serangan DDoS
- 4. User Experience
 - a. Kembangkan antarmuka yang lebih responsif
 - b. Tambahkan customization options untuk channel
 - c. Implementasikan sistem rekomendasi yang lebih akurat
 - d. Tingkatkan aksesibilitas untuk pengguna disabilitas
- 5. Dokumentasi dan Testing
 - a. Buat dokumentasi API yang lebih komprehensif
 - b. Implementasikan automated testing untuk semua komponen
 - c. Tingkatkan coverage unit testing
 - d. Dokumentasikan best practices untuk pengembangan
- 6. Monitoring dan Maintenance
 - a. Implementasikan sistem monitoring yang lebih robust
 - b. Buat prosedur maintenance yang lebih terstruktur
 - c. Tingkatkan sistem logging untuk troubleshooting
 - d. Kembangkan dashboard monitoring real-time

Semua saran di atas perlu dipertimbangkan dan diprioritaskan sesuai dengan kebutuhan dan resources yang tersedia. Implementasi dapat dilakukan secara bertahap dengan memperhatikan dampak terhadap sistem yang sudah berjalan.

Pengembangan sistem Twitch streaming ini merupakan proses yang berkelanjutan dan membutuhkan evaluasi serta penyesuaian secara regular untuk memastikan sistem tetap relevan dengan kebutuhan pengguna dan perkembangan teknologi.