

LATIHAN MINGGU KE 2
PRAKTIKUM PEMOOGRAMAN BERBASIS OBJEK (PBO)

Disusun oleh:

Alief Muhammad Sugata (223443026)

Muhammad Defval (223443040)

Muhammad Hamzah(223443042)



JURUSAN TEKNIK OTOMASI MANUFAKTUR DAN MEKATRONIKA
POLITEKNIK MANUFAKTUR NEGERI BANDUNG
BANDUNG
SEPTEMBER 2024

Apa Itu SDLC ?

SDLC (Software Development Life Cycle) adalah proses terstruktur yang digunakan untuk mengembangkan perangkat lunak secara sistematis dari awal hingga akhir. Proses ini mencakup berbagai tahap, mulai dari perencanaan, analisis kebutuhan, desain, implementasi, pengujian, hingga pemeliharaan perangkat lunak. SDLC memastikan bahwa perangkat lunak yang dikembangkan sesuai dengan kebutuhan pengguna, berkualitas tinggi, dan teruji dengan baik sebelum diluncurkan ke lingkungan produksi. Setiap tahap dalam SDLC membantu mengurangi risiko kesalahan dan meningkatkan efisiensi pengembangan, serta memastikan perangkat lunak dapat beradaptasi dengan perubahan kebutuhan atau teknologi. Dengan metode ini, tim pengembang dapat menghasilkan produk yang lebih terorganisir dan siap digunakan secara optimal oleh pengguna akhir.

SDLC juga berfungsi sebagai pedoman untuk pengelolaan proyek perangkat lunak, memastikan bahwa setiap langkah dalam proses pengembangan dapat dipantau, diukur, dan diperbaiki jika diperlukan. Berbagai model SDLC, seperti **Waterfall**, Agile, Iterative, dan DevOps, dapat digunakan tergantung pada jenis proyek dan kebutuhan bisnis. Masing-masing model memiliki pendekatan unik, tetapi tujuan akhirnya tetap sama: menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna.

Dengan menggunakan SDLC, risiko dalam pengembangan perangkat lunak, seperti ketidakpastian teknis, perubahan persyaratan, atau pengembangan yang terlalu lambat, dapat dikurangi. Selain itu, tim dapat lebih mudah berkoordinasi dan berkomunikasi dalam hal pelacakan kemajuan, pengelolaan risiko, dan evaluasi performa setiap fase pengembangan. SDLC juga memastikan bahwa proses pemeliharaan perangkat lunak setelah implementasi berjalan dengan baik, memungkinkan perangkat lunak terus berkembang dan diperbaiki sesuai dengan perkembangan teknologi atau perubahan kebutuhan bisnis.

2) Agile Scrum

Agile Scrum adalah salah satu kerangka kerja yang paling populer dalam metodologi Agile, yang dirancang untuk mengelola proyek perangkat lunak secara iteratif dan inkremental. Agile sendiri merupakan pendekatan pengembangan yang berfokus pada fleksibilitas, kolaborasi, dan pengiriman yang cepat dalam siklus yang singkat. Scrum menambahkan struktur pada Agile dengan menetapkan peran, acara, dan artefak yang jelas untuk memastikan bahwa tim pengembang dapat beroperasi secara terorganisir

dan efisien. Dalam Scrum, pekerjaan dibagi ke dalam siklus pendek yang disebut Sprint, biasanya berdurasi 2 hingga 4 minggu. Setiap Sprint bertujuan untuk menghasilkan produk atau fitur yang dapat diuji dan dievaluasi oleh pengguna atau stakeholder. Proses ini memungkinkan tim untuk merespons perubahan persyaratan dengan cepat, serta memungkinkan pengiriman nilai secara terus-menerus melalui pengembangan inkremental.

Di dalam kerangka Scrum, ada tiga peran utama: Product Owner, Scrum Master, dan Development Team. Product Owner bertanggung jawab untuk memprioritaskan backlog produk, yang merupakan daftar semua fitur, perbaikan, dan peningkatan yang dibutuhkan. Scrum Master bertindak sebagai fasilitator yang memastikan tim memahami dan mengikuti proses Scrum dengan benar, serta membantu mengatasi hambatan yang dapat mengganggu kinerja tim. Development Team adalah sekelompok individu dengan keahlian lintas fungsi yang bertanggung jawab atas pengembangan produk. Tim ini berkolaborasi erat untuk menyelesaikan tugas yang telah diprioritaskan oleh Product Owner dan berkomitmen menyelesaikan pekerjaan yang diambil dalam Sprint.

Scrum juga menetapkan serangkaian acara teratur untuk menjaga tim tetap fokus dan transparan. Setiap Sprint dimulai dengan Sprint Planning, di mana tim memutuskan item backlog mana yang akan dikerjakan selama Sprint. Setiap hari, tim mengadakan Daily Scrum, rapat singkat di mana anggota tim berbagi apa yang telah mereka kerjakan, apa yang akan mereka kerjakan, dan hambatan apa yang dihadapi. Setelah Sprint selesai, tim mengadakan Sprint Review untuk mendemonstrasikan hasil pekerjaan mereka kepada Product Owner dan stakeholder lainnya, serta menerima umpan balik. Setelah itu, Sprint Retrospective dilakukan untuk menilai kinerja tim dan mencari cara untuk meningkatkan proses di Sprint berikutnya.

Keunggulan utama dari Scrum adalah kemampuannya untuk merespons perubahan secara cepat dan efisien. Dengan siklus kerja yang singkat dan umpan balik yang terus-menerus, Scrum meminimalkan risiko besar yang mungkin terjadi dalam pengembangan perangkat lunak tradisional. Ini juga memungkinkan tim untuk lebih produktif dan terfokus pada hasil yang konkret, menghindari pekerjaan yang tidak relevan atau terlalu jauh dari kebutuhan pengguna. Scrum memberikan lingkungan yang dinamis di mana tim dapat bekerja secara kolaboratif, otonom, dan dengan keterlibatan yang tinggi. Kerangka kerja ini cocok untuk proyek yang kompleks atau di mana kebutuhan sering berubah, serta membantu perusahaan mencapai hasil yang lebih cepat dan lebih sesuai dengan harapan pengguna atau klien.

3) Use Case

Use case adalah deskripsi rinci tentang bagaimana pengguna atau aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu. Dalam pengembangan perangkat lunak, use case membantu mendokumentasikan dan menggambarkan fungsi yang diharapkan dari sebuah sistem berdasarkan perspektif pengguna akhir. Ini biasanya digunakan dalam tahap analisis dan desain untuk mendefinisikan bagaimana sebuah sistem seharusnya bekerja, dengan fokus pada interaksi eksternal antara pengguna dan sistem. Sebuah use case menggambarkan langkah-langkah yang harus diambil oleh pengguna dan bagaimana sistem merespons setiap aksi yang dilakukan pengguna.

Use case terdiri dari beberapa elemen utama, yaitu aktor (entitas yang berinteraksi dengan sistem, yang bisa berupa pengguna manusia atau sistem eksternal lainnya), skenario (serangkaian langkah yang menggambarkan interaksi antara aktor dan sistem), dan tujuan (hasil akhir yang ingin dicapai aktor). Setiap use case mengidentifikasi serangkaian tindakan dan respons sistem yang diperlukan untuk mendukung pencapaian tujuan ini. Use case biasanya didokumentasikan dalam bentuk diagram use case (menggunakan notasi UML – Unified Modeling Language) dan deskripsi teks yang lebih rinci untuk memberikan gambaran yang jelas tentang bagaimana fitur atau proses bekerja dalam sistem.

Use case sangat berguna dalam mengkomunikasikan persyaratan fungsional dari sistem kepada semua pihak yang terlibat, termasuk pengembang, desainer, dan stakeholder. Ini memberikan panduan yang jelas tentang apa yang harus dicapai oleh perangkat lunak dan bagaimana perangkat lunak tersebut harus berperilaku dalam berbagai situasi. Selain itu, use case juga membantu mengidentifikasi skenario normal dan skenario alternatif (misalnya, jika terjadi kesalahan atau kondisi khusus), yang memberikan gambaran lengkap tentang semua interaksi yang mungkin terjadi antara pengguna dan sistem. Dengan demikian, use case berperan penting dalam memastikan bahwa perangkat lunak yang dikembangkan sesuai dengan kebutuhan dan ekspektasi pengguna, serta membantu mengurangi risiko kesalahan desain dan pengembangan yang dapat mengakibatkan ketidakpuasan pengguna.

4) Activity Diagram

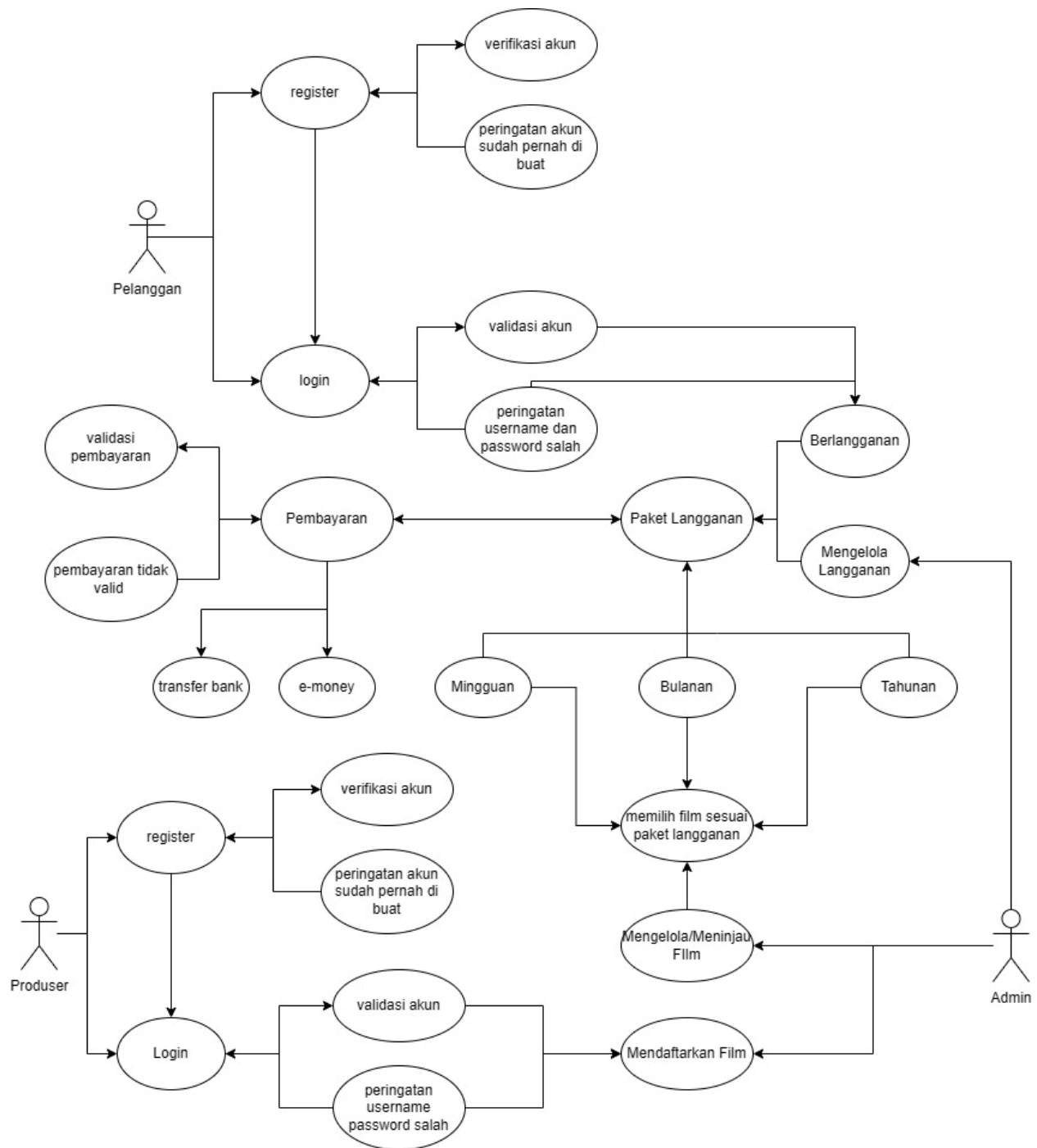
Activity diagram adalah jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan alur kerja atau aktivitas dalam suatu sistem. Diagram ini menggambarkan aliran langkah-langkah logis dari satu aktivitas ke aktivitas lainnya dalam sebuah proses, dan memberikan gambaran visual tentang bagaimana proses tersebut berjalan, termasuk keputusan yang dibuat, aksi yang diambil, dan hasil yang dicapai. Activity diagram biasanya digunakan dalam pengembangan perangkat lunak untuk memodelkan proses bisnis atau alur eksekusi logis dari sebuah sistem, mulai dari awal hingga akhir.

Diagram ini mirip dengan flowchart dalam konsepnya, tetapi memiliki fitur tambahan untuk mendukung pemodelan sistem yang lebih kompleks, seperti paralelisme, kondisi percabangan, dan tindakan konkuren. Elemen utama dalam activity diagram termasuk activity nodes (yang mewakili tindakan atau aktivitas yang dilakukan), decision nodes (untuk menangani percabangan atau kondisi), start nodes dan end nodes (untuk menandai awal dan akhir dari aliran proses), serta swimlanes (yang digunakan untuk menunjukkan siapa atau komponen mana yang bertanggung jawab atas setiap aktivitas). Diagram ini juga bisa menunjukkan aliran data, input, dan output yang relevan dengan aktivitas yang sedang berlangsung.

Activity diagram sangat berguna untuk menganalisis proses bisnis dan alur sistem karena memberikan pandangan menyeluruh tentang bagaimana tugas dilakukan dan bagaimana keputusan dibuat sepanjang aliran proses. Ini membantu pengembang dan pemangku kepentingan lainnya memahami secara lebih mendalam bagaimana sistem akan berfungsi dalam skenario nyata dan bagaimana berbagai komponen dalam sistem berinteraksi. Selain itu, activity diagram sering digunakan dalam tahap desain untuk mengidentifikasi potensi masalah seperti kemacetan (bottlenecks) atau ketidakefisienan dalam aliran kerja yang diusulkan.

Dengan activity diagram, tim pengembang dapat memastikan bahwa proses yang mereka rancang sesuai dengan tujuan dan kebutuhan bisnis. Diagram ini juga memfasilitasi komunikasi yang lebih baik antara analis bisnis, desainer sistem, dan pengembang perangkat lunak, karena mereka dapat memahami aliran proses dengan cara yang visual dan mudah dipahami. Di berbagai industri, activity diagram sering digunakan untuk menggambarkan operasi internal sistem, dari otomatisasi proses di perangkat lunak hingga pemodelan prosedur operasional dalam organisasi.

Use Case Aplikasi Subscription Commerce



Penjelasan :

Aktor

1. Pelanggan: Pengguna yang menggunakan sistem untuk berlangganan paket dan mengakses film.
2. Produser: Pihak yang mendaftarkan dan mengelola film di dalam sistem.
3. Admin: Pihak yang berperan dalam pengelolaan film dan sistem langganan.

Use Case untuk Pelanggan

1. Register (Daftar)

- Pelanggan melakukan registrasi akun.
- Setelah register, ada verifikasi akun untuk memastikan keaslian data.
- Jika akun sudah ada, pelanggan menerima peringatan akun sudah pernah dibuat.

2. Login:

- Setelah register, pelanggan bisa login.
- Jika username atau password salah, sistem menampilkan peringatan username dan password salah
- Jika berhasil, akun divalidasi dan pelanggan dapat melanjutkan ke langkah berikutnya.

3. Pembayaran

- Pelanggan memilih metode pembayaran untuk berlangganan paket.
- Sistem melakukan validasi pembayaran
- Jika pembayaran tidak valid, muncul peringatan pembayaran tidak valid.
- Metode pembayaran yang ditawarkan meliputi transfer bank dan e-money.

4. Paket Langganan:

- Setelah pembayaran, pelanggan memilih jenis paket yang diinginkan: mingguan, bulanan, atau tahunan.
- Berdasarkan paket yang dipilih, pelanggan dapat memilih film sesuai paket langganan.

5. Berlangganan dan Mengelola Langganan:

- Setelah memilih paket, pelanggan dapat berlangganan dan mengelola paket langganannya, termasuk memperbarui atau mengubah paket sesuai kebutuhan.

Use Case untuk Produser:

1. Register dan Login:

- Sama seperti pelanggan, produser juga harus melakukan register dan login ke sistem.
- Setelah login, mereka juga melalui proses verifikasi akun.
- Jika ada kesalahan saat login, mereka mendapatkan peringatan username dan password salah.

2. Mendaftarkan Film:

- Setelah login dan verifikasi, produser dapat mendaftar film yang akan ditambahkan ke sistem.

Use Case untuk Admin:

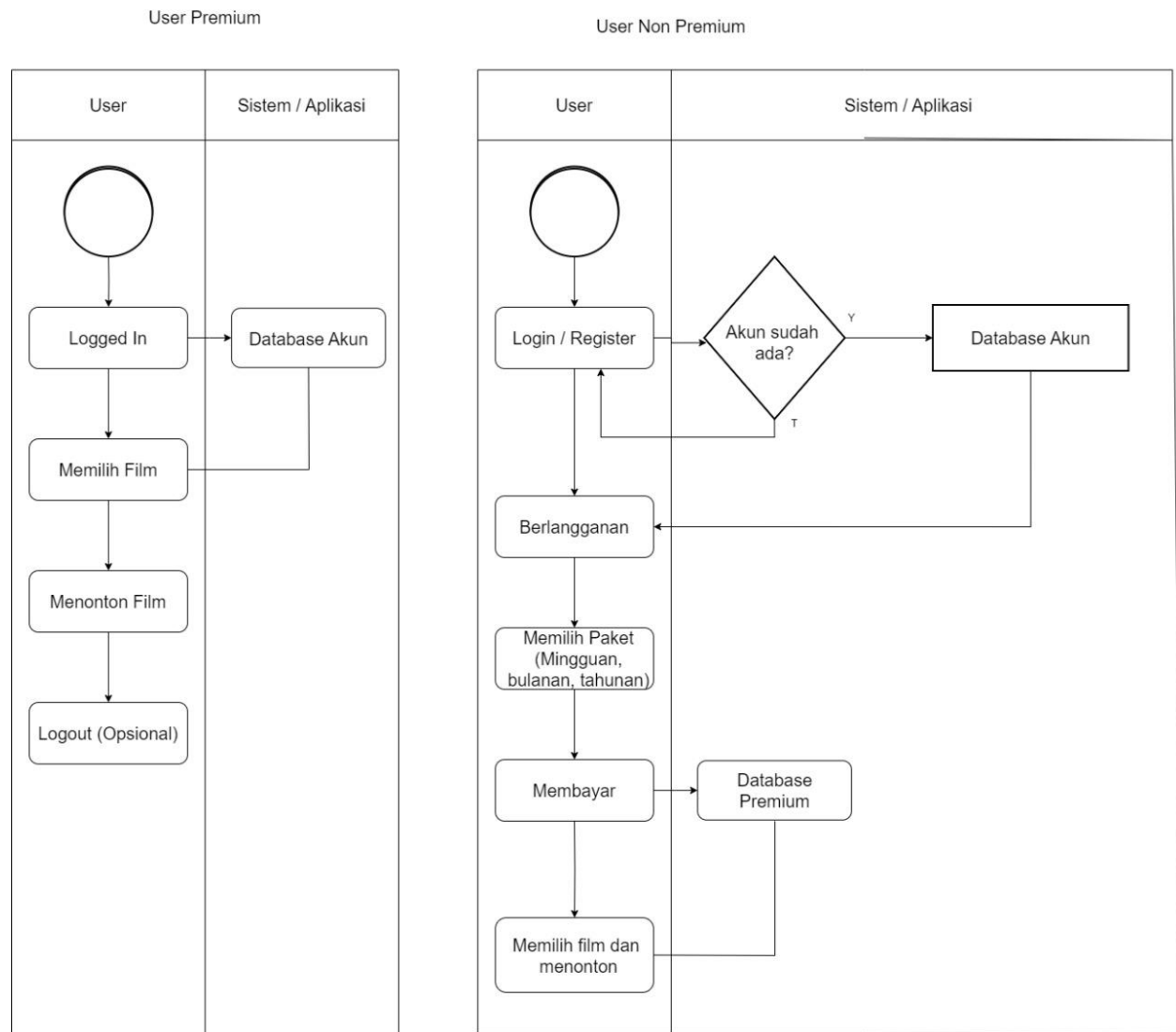
1. Mengelola/Meninjau Film:

- Admin memiliki peran dalam mengelola dan meninjau film yang telah didaftarkan oleh produser.
- Mereka juga memiliki akses untuk membantu dalam proses validasi paket langganan dan memastikan pelanggan dan produser mengikuti aturan sistem.

Interaksi Sistem.

- Verifikasi Akun dilakukan setelah registrasi untuk memastikan pengguna atau produser sudah terdaftar dan datanya valid.
- Validasi Pembayaran dilakukan untuk memastikan bahwa transaksi dari pelanggan sah sebelum melanjutkan ke pemilihan paket langganan.
- Pilihan Paket Langganan membantu pelanggan memilih opsi yang paling sesuai dengan kebutuhan mereka, dan ada kemungkinan pelanggan mengelola paket langganan tersebut setelah berlangganan.

Activity Diagram User



1. User Premium

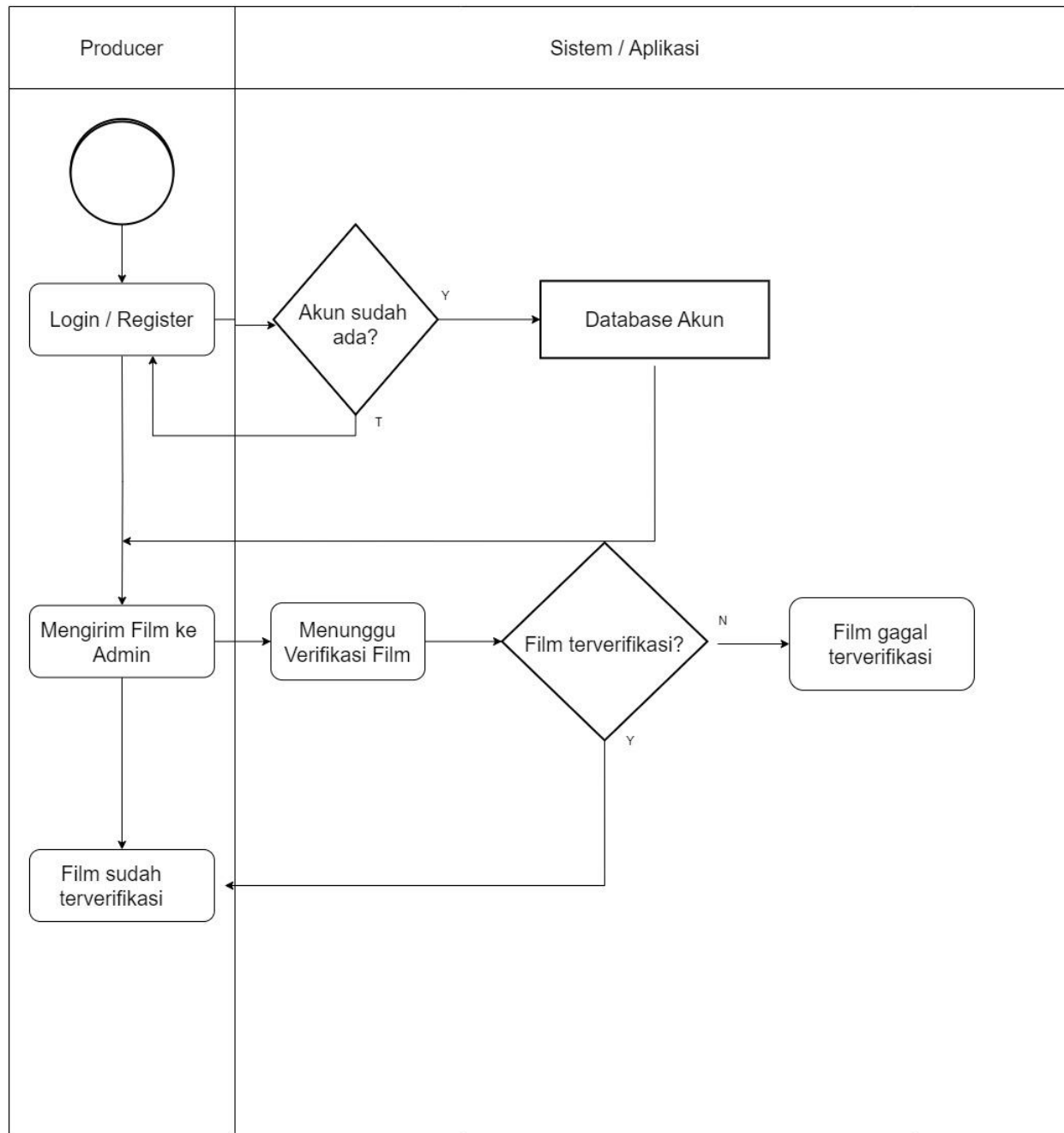
- Logged In: Pengguna premium yang sudah memiliki akun langsung melakukan login.
- Memilih Film: Setelah login, pengguna bisa langsung memilih film yang ingin ditonton.
- Menonton Film: Pengguna menonton film yang dipilih.
- Logout (Opsional): Setelah selesai menonton, pengguna memiliki opsi untuk logout dari akun mereka.

Proses ini lebih sederhana karena pengguna premium tidak perlu melalui proses pendaftaran atau pembayaran lagi, karena mereka sudah memiliki akses penuh ke layanan.

2. User Non Premium

- Login / Register: Pengguna non-premium memulai dengan login atau mendaftar jika belum memiliki akun.
 - Jika sudah memiliki akun, data pengguna diambil dari Database Akun.
- Berlangganan: Setelah login atau registrasi, pengguna diarahkan untuk berlangganan layanan premium.
 - Memilih Paket (Mingguan, Bulanan, Tahunan): Pengguna memilih jenis paket berlangganan (mingguan, bulanan, atau tahunan).
 - Membayar: Pengguna kemudian melakukan pembayaran untuk paket yang dipilih.
- Memilih Film dan Menonton: Setelah pembayaran berhasil, pengguna dapat memilih film dan menontonnya. Akses premium diatur dalam ****Database Premium****.

Activity Diagram Producer



1.Login / Register:

Producer harus login jika sudah memiliki akun, atau mendaftar terlebih dahulu jika belum memiliki akun.

Jika akun sudah ada, data pengguna akan diambil dari Database Akun.

Jika akun belum ada, pengguna harus mendaftar, dan data tersebut akan disimpan di Database Akun.

2.Mengirim Film ke Admin:

Setelah berhasil login atau mendaftar, Producer dapat mengirim film mereka ke admin untuk proses verifikasi.

3.Menunggu Verifikasi Film:

Setelah film dikirim, film akan masuk ke tahap verifikasi oleh admin. Producer harus menunggu proses verifikasi ini.

4.Film Terverifikasi

Setelah proses verifikasi selesai, ada dua kemungkinan hasil:

Ya (Y): Jika film lolos verifikasi, statusnya berubah menjadi Film sudah terverifikasi, dan siap untuk ditayangkan atau digunakan.

Tidak (N): Jika film tidak lolos verifikasi, statusnya berubah menjadi Film gagal terverifikasi.

Activity Diagram Admin

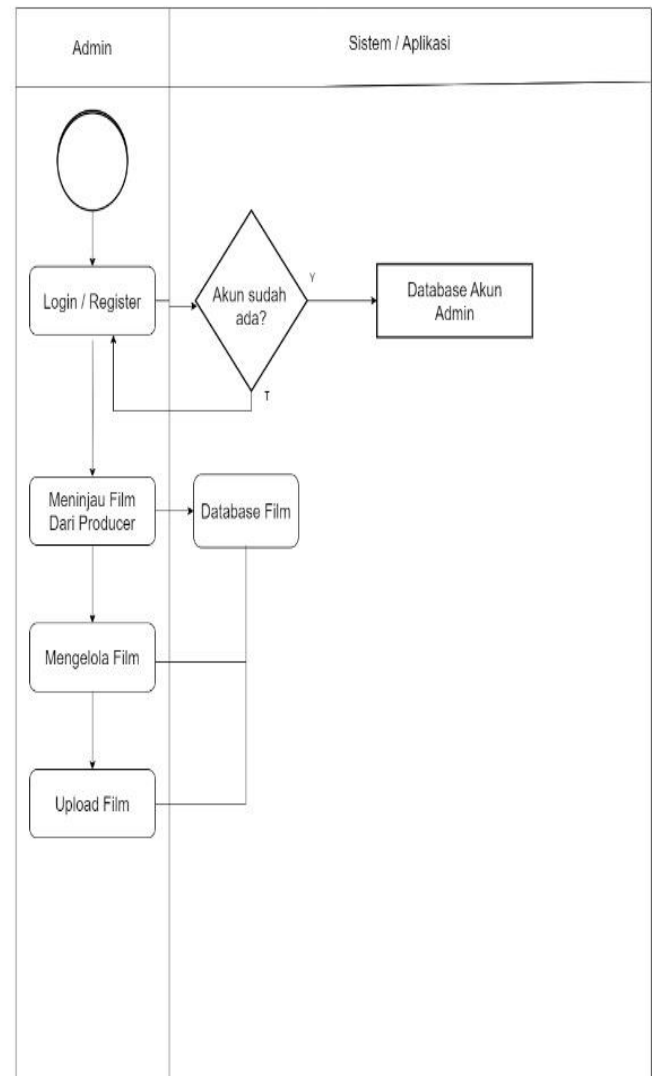
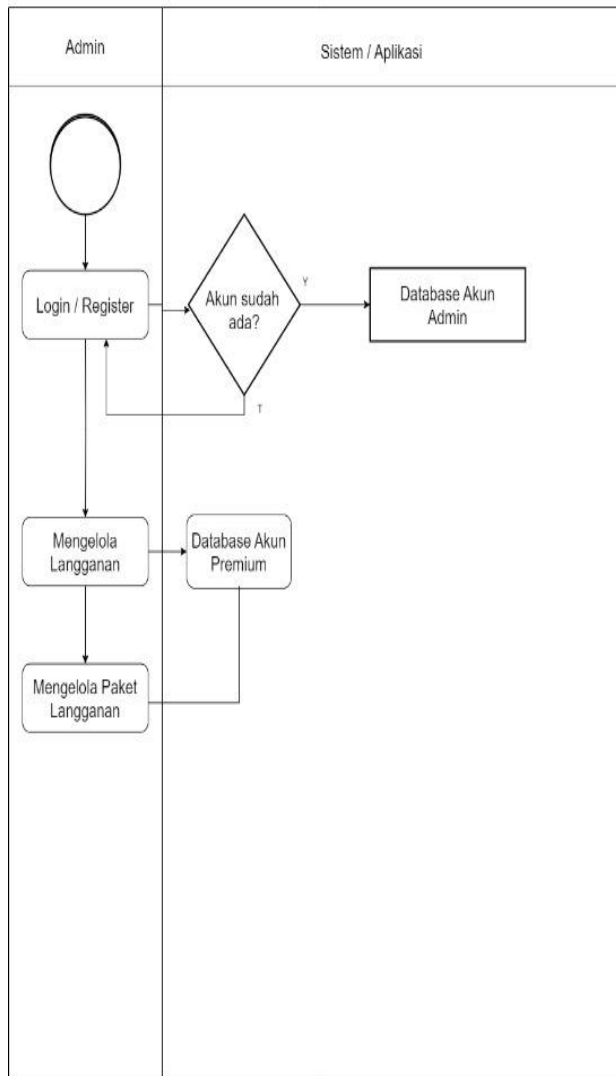


Diagram Kiri:

Ini menggambarkan aktivitas admin terkait pengelolaan akun dan langganan:

1. Login/Register: Admin memulai dengan melakukan login atau pendaftaran ke dalam sistem.
2. Apakah akun sudah ada Sistem akan memeriksa apakah akun admin sudah terdaftar.
 - Jika ya, admin diarahkan ke Database Akun Admin.
 - Jika tidak, admin melanjutkan dengan pendaftaran akun baru.
3. Mengelola Langganan: Admin memiliki kemampuan untuk mengelola informasi langganan.
4. Mengelola Paket Langganan: Admin juga dapat mengelola paket langganan, seperti membuat, mengubah, atau menghapus paket-paket langganan yang ada.
5. Semua informasi terkait langganan dan paket disimpan di Database Akun Premium.

Diagram Kanan:

Diagram ini menunjukkan proses yang terkait dengan pengelolaan film dalam sistem:

1. Login/Register: Sama dengan proses di kiri, admin mulai dengan login atau pendaftaran.
2. Apakah akun sudah ada: Sistem akan memeriksa apakah akun admin sudah ada.
 - Jika ya, admin diarahkan ke Database Akun Admin.
 - Jika tidak, admin mendaftar terlebih dahulu.
3. Meninjau Film Dari Producer: Admin meninjau film yang diunggah oleh produser.
4. Mengelola Film: Admin dapat mengelola film-film tersebut, seperti menyetujui, menolak, atau mengedit film sebelum dipublikasikan.
5. Upload Film: Setelah meninjau dan mengelola, admin mengunggah film tersebut ke dalam sistem.

6. Semua informasi terkait film yang diunggah akan disimpan di Database Film.

Secara keseluruhan, kedua diagram ini menggambarkan aktivitas admin yang berbeda: satu terkait pengelolaan akun dan langganan, dan satu lagi terkait pengelolaan konten film.