

# **PROPOSAL TUGAS BESAR PEMROGRAMAN BERBASIS OBJEK**



## **PSRent Max – Sistem Manajemen Rental PlayStation**

**Mata Kuliah:** Pemrograman Berbasis Objek (PBO)

**Tanggal:** 9 November 2025

**Kelompok:** [6]

---

### **ANGGOTA KELOMPOK**

1. [Muhammad Rafadi Kurniawan /103062300089]
  2. [Aditya Attabby Hidayat/103062300078]
  3. [Naufal Saifullah Yusuf /10302300091]
-

# BAB I PENDAHULUAN



## A. Latar Belakang

Di era digital saat ini, teknologi informasi telah menjadi bagian integral dari kehidupan sehari-hari manusia. Pemanfaatan teknologi tidak hanya terbatas pada komunikasi, tetapi juga merambah ke berbagai sektor bisnis, termasuk industri hiburan dan rekreasi.

Salah satu bentuk bisnis hiburan yang populer adalah rental PlayStation (PS). Usaha rental PlayStation memerlukan sistem yang terorganisir untuk mengelola berbagai aspek operasional seperti penjadwalan ruangan, pengelolaan perangkat konsol (PS4/PS5), perhitungan tarif rental yang dinamis, reservasi pelanggan, serta pencatatan transaksi pembayaran.

Selama ini, banyak usaha rental PlayStation masih mengandalkan sistem manual atau menggunakan aplikasi sederhana seperti Microsoft Excel untuk mencatat data pelanggan dan transaksi. Hal ini menimbulkan beberapa permasalahan seperti: - Sulitnya melacak ketersediaan ruangan dan console secara real-time - Risiko terjadinya double booking atau konflik jadwal - Perhitungan tarif yang tidak konsisten (tarif weekday vs weekend, member vs non-member) - Kesulitan dalam membuat laporan pendapatan harian maupun bulanan - Proses pembayaran yang lambat dan rentan kesalahan pencatatan

Oleh karena itu, diperlukan suatu sistem terkomputerisasi (aplikasi desktop) untuk mengelola seluruh proses bisnis rental PlayStation secara efisien, akurat,

dan terintegrasi. Aplikasi **PSRent Max** dirancang dengan arsitektur berbasis objek (OOP) menggunakan bahasa pemrograman Java, menerapkan design pattern modern seperti Singleton, Strategy, Factory, dan Facade untuk memastikan sistem mudah dipelihara dan dikembangkan di masa depan.

## B. Sasaran & Tujuan Proyek Sistem Informasi

Tujuan dari pembuatan sistem informasi PSRent Max adalah:

1. Untuk memahami proses bisnis rental PlayStation yang saat ini berjalan dan mengidentifikasi kebutuhan otomasi.
2. Untuk merancang dan mengimplementasikan sistem informasi berbasis desktop yang mampu mengelola:
  - Autentikasi pengguna (Admin, Operator)
  - Manajemen inventori (Console PS4/PS5, Room)
  - Manajemen sesi rental (start, pause, resume, extend, end)
  - Perhitungan tarif dinamis (Standard, Weekend, Member)
  - Pembayaran dan cetak struk
  - Reservasi dengan deteksi konflik jadwal
  - Laporan pendapatan harian dan bulanan
3. -konsep **Pemrograman Berbasis Objek (PBO)** :
  - **Abstraction** (abstract class: Person, Laporan)
  - **Encapsulation** (private attributes dengan getter/setter)
  - **Inheritance** (Admin, Operator, Member extends Person)
  - **Polymorphism** (Strategy pattern untuk tarif)
  - **Interface** (RentalTarifStrategy)
  - **Design Pattern**
4. Untuk menyediakan antarmuka pengguna (GUI) yang user-friendly menggunakan Java Swing.
5. Untuk menghasilkan sistem yang scalable dan maintainable, sehingga mudah dikembangkan di masa depan (misalnya integrasi database MySQL, ekspor laporan PDF, atau fitur loyalty program).

## C. Batasan Masalah

Batasan masalah dari sistem informasi PSRent Max ini adalah:

1. **Fitur yang diimplementasikan** meliputi:
  - Login dan registrasi (Admin dapat mendaftarkan Operator baru)
  - Manajemen Console dan Room (status ketersediaan)
  - Manajemen Member
  - Manajemen Sesi Rental (start, pause, resume, extend, end dengan perhitungan biaya otomatis)
  - Tarif dinamis: Standard (weekday), Weekend (+50%), Member (-20%)

- Pembayaran (Cash, E-Wallet, Transfer) dan generate struk teks
  - Reservasi (cek overlap jadwal per room)
  - Laporan Harian dan Bulanan
2. Sistem bersifat multi-user dengan role berbeda (Admin dan Operator).
  3. Aplikasi berjalan offline (desktop application) dengan data disimpan in-memory menggunakan ArrayList. dapat dihubungkan ke database MySQL jika diperlukan persistensi data.
  4. Aplikasi ini hanya dapat digunakan oleh pengelola rental PlayStation (tidak ada portal customer online).
  5. Teknologi yang digunakan: Java 8+, Swing GUI, arsitektur MVC (Model-View-Controller).
  6. Tidak mencakup: Integrasi payment gateway online, notifikasi SMS/Email, mobile app, atau analitik bisnis kompleks.
- 

## BAB II RANCANGAN SISTEM

### D. Deskripsi Sistem

Aplikasi sistem informasi **PSRent Max** dibuat menggunakan bahasa pemrograman **Java** dengan antarmuka berbasis **Java Swing (GUI Desktop)**. Sistem ini menerapkan arsitektur **MVC (Model-View-Controller)** dan berbagai **Design Pattern** untuk memastikan kode terstruktur, modular, dan mudah dipelihara.

**Penyimpanan data** menggunakan struktur in-memory (ArrayList) yang memungkinkan aplikasi berjalan tanpa ketergantungan database eksternal.

Namun, sistem dirancang dengan pola Singleton (Database) sehingga mudah diintegrasikan dengan MySQL jika diperlukan persistensi data di masa depan.

**Alur kerja aplikasi** meliputi:

1. **Login** menggunakan username dan password (role: Admin atau Operator)
2. **Admin** dapat:
  - Melihat dan mengelola inventori (Console, Room)
  - Melihat daftar Operator yang terdaftar
  - Mendaftarkan Operator baru
  - Generate laporan pendapatan harian dan bulanan
3. **Operator** dapat:
  - Memulai sesi rental (pilih room, console, durasi bermain)
  - Pause, resume, atau extend sesi yang sedang berjalan
  - Mengakhiri sesi dan menghitung biaya otomatis berdasarkan tarif dinamis
  - Memproses pembayaran (cash/e-wallet/transfer) dan mencetak struk
  - Membuat reservasi untuk pelanggan (sistem cek overlap jadwal otomatis)
  - Melihat daftar sesi aktif dan reservasi

## E. Fungsionalitas Aplikasi

Berikut adalah daftar lengkap fungsionalitas yang tersedia di aplikasi PSRent Max:

### 1. Autentikasi & Otorisasi

- Login Admin (username: admin, password: admin123)
- Login Operator (username: operator, password: op123)
- Registrasi Operator baru (hanya Admin yang bisa melakukan)

### 2. Manajemen Inventori

- **Console Management:** Lihat daftar console (PS4/PS5), status (Available/In Use/Maintenance), tarif per jam, jumlah controller tersedia
- **Room Management:** Lihat daftar ruangan, kapasitas, console yang terpasang, status ketersediaan

### 3. Manajemen Member

- Lihat data member (ID, nama, level: Silver/Gold, saldo)
- Top-up saldo member
- Deduct saldo member saat pembayaran

### 4. Manajemen Sesi Rental

- **Start Session:** Operator memilih room, console, dan durasi bermain (dalam menit), sistem generate ID sesi otomatis
- **Pause Session:** Hentikan sementara (waktu pause dicatat)
- **Resume Session:** Lanjutkan sesi yang di-pause
- **Extend Session:** Tambah waktu bermain
- **End Session:** Akhiri sesi, sistem menghitung total biaya berdasarkan:
  - Durasi aktual (start time → end time)
  - Strategi tarif yang dipilih (Standard/Weekend/Member)
  - Tarif dasar console per jam

### 5. Pembayaran & Struk

- Pilih metode pembayaran: Cash, E-Wallet, atau Transfer
- Proses pembayaran otomatis
- Generate struk pembayaran (format teks) berisi:
  - ID Payment, ID Session
  - Nama Room, Tipe Console
  - Total biaya, metode pembayaran, tanggal & waktu

### 6. Reservasi

- **Buat Reservasi:** Input nama pelanggan, nomor HP, pilih room, waktu mulai & selesai, deposit
- **Cek Overlap:** Sistem otomatis mendeteksi konflik jadwal (reservasi yang overlap pada room yang sama akan ditolak)

- **Cancel Reservasi:** Batalkan reservasi yang sudah dibuat

## 7. Laporan

- **Laporan Harian:** Filter sesi berdasarkan tanggal tertentu, hitung total pendapatan hari itu
- **Laporan Bulanan:** Filter sesi berdasarkan bulan & tahun, hitung total pendapatan bulanan

## 8. Logout

- Admin/Operator dapat logout dan kembali ke menu login

## F. Target User / Pengguna

Target pengguna sistem informasi PSRent Max adalah:

1. **Admin / Pemilik Usaha Rental PlayStation**
  - Mengelola inventori (console dan room)
  - Memonitor operator yang bekerja
  - Melihat laporan pendapatan untuk analisis bisnis
2. **Operator / Karyawan Rental**
  - Melayani pelanggan (start/end session)
  - Memproses pembayaran
  - Mengelola reservasi

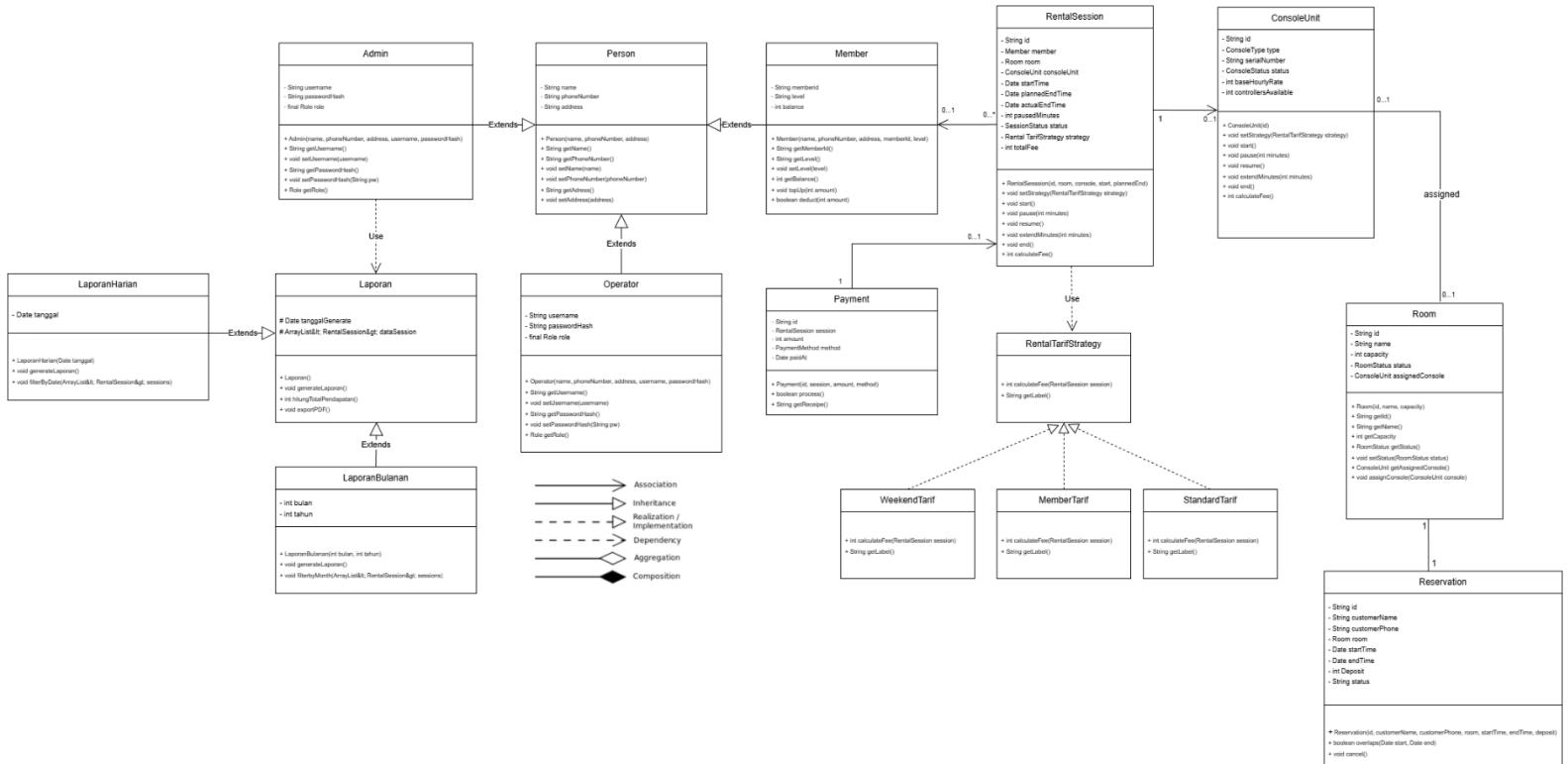
## G. Teknologi Perangkat Lunak Yang Digunakan

Teknologi perangkat lunak yang digunakan untuk membuat sistem informasi PSRent Max adalah:

1. **Java Development Kit (JDK) 8+** – Platform dan compiler untuk Java
2. **Java Swing** – Library GUI untuk membuat antarmuka desktop
3. **NetBeans / IntelliJ IDEA / Eclipse** – IDE untuk pengembangan (opsional, dapat dikompilasi via command line)
4. **Git & GitHub** – Version control dan kolaborasi tim
5. **Trello / Notion** – Manajemen task per anggota tim (Scrum board)
6. **Mermaid / Visual Paradigm** – Tools untuk membuat Class Diagram dan Use Case Diagram

## H. Class Diagram

Class Diagram menggambarkan struktur kelas, atribut, method, access modifier, serta relasi antar kelas dalam sistem PSRent Max.



### Penjelasan Relasi:

**Inheritance:** Admin, Operator, Member mewarisi Person. LaporanHarian dan LaporanBulanan mewarisi Laporan. –

**Implementasi Interface:** StandardTarif, WeekendTarif, MemberTarif mengimplementasikan RentalTarifStrategy. –

**Asosiasi:** RentalSession berasosiasi dengan Room, ConsoleUnit, dan Member. Payment berasosiasi dengan RentalSession. Reservation berasosiasi dengan Room.

**Dependency:** RentalSession menggunakan RentalTarifStrategy untuk kalkulasi biaya. Application menggunakan Database sebagai Facade dan Admin menggunakan Laporan untuk.

### 1. Inheritance (Pewarisan)

Inheritance adalah hubungan "is-a" (adalah seorang/sebuah). Sebuah kelas anak (subclass) mewarisi atribut dan method dari kelas induk (superclass), yang memungkinkan reusabilitas kode dan pembentukan hierarki yang logis.

Dalam diagram ini, kita melihat dua contoh utama Inheritance:

- **Hierarki Person:**
  - **Person** adalah kelas induk abstrak.
  - **Admin, Operator, dan Member** semuanya adalah Person (Admin is a Person). Mereka mewarisi atribut dasar seperti name, phoneNumber, dan address.
  - **Keuntungan:** Kita tidak perlu menulis ulang atribut name, address, dll. di setiap kelas. Setiap kelas anak kemudian bisa menambahkan atribut dan method spesifiknya sendiri (misal, username untuk Admin, dan balance untuk Member).
- **Hierarki Laporan:**
  - **Laporan** adalah kelas induk abstrak yang.
  - **LaporanHarian** dan **LaporanBulanan** adalah jenis-jenis spesifik dari Laporan. Keduanya mewarisi properti dasar dari Laporan tetapi memiliki implementasi yang berbeda untuk menghasilkan laporan.
  - **Keuntungan:** Kita bisa memperlakukan LaporanHarian dan LaporanBulanan sebagai objek Laporan (Polimorfisme), yang menyederhanakan kode saat memproses berbagai jenis laporan.

## 2. Asosiasi

Asosiasi adalah hubungan "has-a" (memiliki sebuah) atau hubungan struktural antar kelas. Ini menunjukkan bahwa objek dari satu kelas terhubung dengan objek dari kelas lain dan mereka bisa saling berinteraksi.

- **RentalSession sebagai Pusat Asosiasi:**
  - RentalSession **berasosiasi dengan** Room, ConsoleUnit, dan Member.
    - **Multiplicity 0..1 (antara RentalSession dan Member):** Menunjukkan bahwa sebuah sesi bisa jadi tidak memiliki member (0) atau memiliki satu member (1). Ini masuk akal karena tidak semua pelanggan adalah member.
    - Sebuah RentalSession **harus memiliki** sebuah Room dan sebuah ConsoleUnit.
- **Payment dan RentalSession:**
  - Sebuah Payment **berhubungan dengan** satu RentalSession. Hubungan ini penting untuk melacak transaksi mana yang membayar sesi rental mana.
- **Reservation dan Room:**
  - Sebuah Reservation **berhubungan dengan** satu Room. Ini mengikat sebuah pemesanan di masa depan ke asset fisik tertentu.

### 3. Implementation (Implementasi Interface)

Ini adalah hubungan di mana sebuah kelas "mengimplementasikan" atau "memenuhi kontrak" dari sebuah interface. Interface mendefinisikan serangkaian method yang harus disediakan oleh kelas yang mengimplementasikannya.

- **RentalTarifStrategy** dan Turunannya:

- **RentalTarifStrategy** adalah sebuah interface.
- **StandardTarif**, **WeekendTarif**, dan **MemberTarif** semuanya **mengimplementasikan** **RentalTarifStrategy**. Ini berarti ketiganya *wajib* menyediakan implementasi konkret untuk method `calculateFee()`.
- **Keuntungan:** Ini adalah inti dari **Strategy Pattern**. Kelas **RentalSession** tidak perlu tahu *bagaimana* cara menghitung biaya, ia hanya perlu tahu bahwa ia memiliki objek yang bisa melakukan `calculateFee()`.

### 4. Dependency (Ketergantungan)

Dependency adalah hubungan yang lebih lemah daripada asosiasi. Ini terjadi ketika satu kelas *menggunakan* kelas lain (biasanya sebagai parameter dalam method atau sebagai variabel lokal), tetapi tidak menyimpannya sebagai atribut permanen. Hubungan ini sering digambarkan dengan panah putus-putus.

- **RentalSession** dan **RentalTarifStrategy**:

- **RentalSession** **bergantung pada** **RentalTarifStrategy** untuk menghitung biaya. Ia tidak menyimpan strategi ini selamanya, melainkan **menggunakannya** saat metode `calculateFee()` dipanggil.
- Alur Interaksi Admin dan Laporan

Seorang Admin tidak secara permanen "memiliki" (has-a) sebuah Laporan. Sebaliknya, Admin adalah aktor yang memicu pembuatan dan penggunaan sebuah Laporan.

Jenis Hubungan	Dari Kelas	Ke Kelas	Deskripsi Fungsional
Inheritance	Admin, Operator, Member	Person	Admin, Operator, dan Member adalah jenis dari Person, mewarisi sifat umum.
Inheritance	LaporanHarian, LaporanBulanan	Laporan	Laporan harian & bulanan adalah jenis dari Laporan, memiliki cara kerja spesifik.
Association	RentalSession	Room, ConsoleUnit	Sebuah sesi rental "memiliki" atau berlangsung di satu ruangan dan menggunakan satu konsol.
Association	RentalSession	Member	Sebuah sesi "mungkin memiliki" satu member yang terkait (untuk diskon, dll.).
Implementation	StandardTarif, WeekendTarif	RentalTarifStrategy	Kelas-kelas tarif ini "adalah sebuah" strategi perhitungan, memenuhi kontrak dari interface.

<b>Dependency</b>	RentalSession Laporan	RentalTarifStrategy Admin	Sesi "menggunakan" sebuah strategi untuk menghitung biayanya pada waktu tertentu.
<b>Dependency</b>	Application	Database	Aplikasi "menggunakan" database sebagai backend untuk semua operasinya.

## I. Skenario User

### 1. Login

**Aktor:** Admin / Operator

**Deskripsi:** Pengguna memasukkan username dan password, memilih role (Admin/Operator), sistem memvalidasi kredensial.

### 2. Registrasi Operator Baru

**Aktor:** Admin

**Deskripsi:** Admin mendaftarkan operator baru ke sistem.

### 3. Kelola Inventori Console & Room

**Aktor:** Admin

**Deskripsi:** Admin melihat daftar console (PS4/PS5) dan room yang tersedia.

### 4. Mulai Sesi Rental

**Aktor:** Operator

**Deskripsi:** Operator memulai sesi rental untuk pelanggan.

### 5. Pause, Resume, Extend Sesi

**Aktor:** Operator

**Deskripsi:** Operator pause, resume, atau extend sesi yang sedang berjalan. sistem update plannedEndTime

### 6. Akhiri Sesi dan Pembayaran

**Aktor:** Operator

**Deskripsi:** Operator mengakhiri sesi, sistem hitung biaya, proses pembayaran.

## J. Mock-up Design

Berikut adalah deskripsi antarmuka (mock-up) untuk setiap layar utama aplikasi PSRent Max:

### 1. MenuLogin

- **Layout:** GridBagLayout dengan form di tengah
- **Komponen:**
  - Label: "PSRent Max - Login"
  - Label & TextField: "Username"
  - Label & PasswordField: "Password"
  - Label & ComboBox: "Login as" (pilihan: Admin, Operator)
  - Button: "Login" (primary)
  - Button: "Register Operator" (secondary)
- **Fungsi:** Validasi kredensial, buka MenuAdmin/MenuKaryawan sesuai role



### 2. MenuRegister

- **Layout:** GridBagLayout dengan form
- **Komponen:**
  - Label: "Register Operator Baru"
  - TextField: Nama, Nomor HP, Alamat, Username, Password
  - Button: "Register", "Cancel"
- **Fungsi:** Simpan operator baru, validasi input tidak boleh kosong

**Register Operator**

Name:

Phone:

Address:

Username:

Password:

**Register** **Cancel**

### 3. Menu Admin

- Layout: JTabbedPane dengan 3 tab
- Tab 1 - Inventori:
  - JTable menampilkan daftar Console (ID, Type, Status, Tarif, Controllers)
  - JTable menampilkan daftar Room (ID, Name, Capacity, Console, Status)
- Tab 2 - Operator:
  - JTable menampilkan daftar Operator (Nama, Username, HP, Alamat)
- Tab 3 - Laporan:
  - ComboBox: pilih "Harian" atau "Bulanan"
  - DatePicker atau TextField untuk input tanggal/bulan/tahun
  - Button: "Generate Laporan"
  - TextArea: tampilkan hasil laporan (jumlah sesi, total pendapatan)
- Button: "Logout" di pojok kanan atas

**CONSOLE INVENTORY**

Kelola stok dan status konsol secara terpusat.

ID	Type	Serial	Status	Rate/Hour
C001	PS4	SN-PS4-001	AVAILABLE	Rp 15000
C002	PS5	SN-PS5-001	AVAILABLE	Rp 25000
C003	PS4	SN-PS4-002	AVAILABLE	Rp 15000

#### 4. Menu Karyawan (Operator)

- Layout: JTabbedPane dengan 3 tab
- Tab 1 - Sesi Aktif:
  - JTable menampilkan daftar sesi yang sedang berjalan (Session ID, Room, Console, Start Time, Status)
  - Button: "Pause", "Resume", "Extend", "End Session"
- Tab 2 - Mulai Sesi:
  - ComboBox: pilih Room (hanya yang Available)
  - ComboBox: pilih Console (hanya yang Available)
  - TextField: Durasi bermain (menit)
  - TextField (opsional): Member ID
  - Button: "Start Session"
- Tab 3 - Reservasi:
  - Form input: Nama pelanggan, HP, pilih Room, Tanggal/Jam mulai, Tanggal/Jam selesai, Deposit
  - Button: "Buat Reservasi", "Cancel Reservasi"
  - JTable: daftar reservasi yang ada
- Button: "Logout" di pojok kanan atas

The screenshot shows a software interface titled "ROOM MANAGEMENT". At the top, there is a navigation bar with tabs: DASHBOARD, CONSOLES, ROOMS, SESSIONS (which is currently selected), and REPORTS. Below the title, a subtitle reads "Atur kapasitas dan status ruangan untuk distribusi pelanggan optimal." A table displays three room entries:

ID	Name	Capacity	Status
R001	Room A	2	AVAILABLE
R002	Room B	2	AVAILABLE
R003	VIP Room	4	AVAILABLE

## BAB III METODOLOGI KERJA

### K. Dasar Pengelompokan Fitur

#### **Modul Administrasi & Pelaporan (Muhammad Rafadi Kurniawan)**

- **Fokus:** Pengelolaan sistem dari sisi Admin, mencakup sumber daya (aset, operator) dan analisis bisnis.
- **Fitur:** Fitur 1 (Manajemen Operator & Inventori) dan Fitur 2 (Sistem Pelaporan).
- **Konsep OOP Kunci:** Inheritance (untuk struktur pengguna) dan Abstraction (untuk skalabilitas laporan).

#### **Fitur 1: Manajemen Operator & Inventori (Inheritance)**

Fitur ini mengelola data master dalam sistem. Konsep utamanya adalah menggunakan kelas induk abstrak Person sebagai cetak biru untuk kelas anak Admin dan Operator. Ini menerapkan Inheritance untuk menciptakan hierarki pengguna yang logis. Admin dapat menggunakan metode registrasiOperator(data) dan lihatDaftarInventori() untuk mengelola data master di dalam sistem, dengan atribut yang terjaga melalui Encapsulation.

#### **Fitur 2: Sistem Pelaporan (Abstraction)**

Untuk fitur ini, dibuat sebuah kelas induk abstrak Laporan yang memiliki *abstract method* generate(filter). Kelas ini menjadi kerangka kerja untuk kelas anak seperti LaporanHarian dan LaporanBulanan. Setiap kelas anak akan mengimplementasikan metode generate() dengan logika filtering yang spesifik.

Penggunaan Abstraction dan Polymorphism ini memudahkan penambahan jenis laporan baru di masa depan tanpa mengubah kode yang sudah ada.

---

#### **Modul Arsitektur & Transaksi (Aditya Attabby Hidayat)**

- **Fokus:** Fondasi teknis aplikasi, keamanan, dan logika inti transaksi yang dinamis.
- **Fitur:** Fitur 3 (Autentikasi & Database) dan Fitur 4 (Perhitungan Tarif & Pembayaran).
- **Konsep OOP Kunci:** Singleton Pattern (untuk database), Strategy Pattern, dan Factory Pattern (untuk tarif).

#### **Fitur 3: Autentikasi & Database (Singleton Pattern)**

Fondasi sistem ini bergantung pada kelas Database yang diimplementasikan menggunakan Singleton Pattern. Dengan *private constructor* dan metode statis getInstance(), dipastikan hanya ada satu objek database yang diakses di seluruh aplikasi, menjaga konsistensi data. Metode validasiUser() akan menggunakan instance tunggal ini untuk memverifikasi kredensial pengguna saat login.

#### **Fitur 4: Perhitungan Tarif & Pembayaran (Strategy & Factory Pattern)**

Untuk perhitungan biaya yang fleksibel, digunakan **Strategy Pattern** melalui interface RentalTarifStrategy. Kelas-kelas konkret seperti StandardTarif, WeekendTarif, dan MemberTarif mengimplementasikan logika perhitungan yang berbeda. Untuk menyederhanakan pemilihan strategi ini, digunakan **Factory Pattern** melalui kelas TariffFactory. Saat sesi berakhir, metode endSession() akan otomatis memanggil factory untuk mendapatkan strategi tarif yang tepat dan menghitung total biaya.

---

#### **Modul Operasional & Alur Kerja Sesi (Naufal Saifullah Yusuf)**

- Fokus: Alur kerja utama yang dihadapi Operator saat melayani pelanggan, dari sesi hingga reservasi.
- Fitur: Fitur 5 (Manajemen Sesi Rental) dan Fitur 6 (Sistem Reservasi).
- Konsep OOP Kunci: Association (hubungan antar objek) dan Dependency (ketergantungan logika).

#### **Fitur 5: Manajemen Sesi Rental (Association & State Management)**

Fitur ini berpusat pada kelas RentalSession. Menggunakan konsep Association, setiap objek RentalSession terhubung dengan objek Room, ConsoleUnit, dan Member (opsional), yang mencerminkan hubungan logis di dunia nyata. Metode seperti start(), pause(), resume(), dan end() akan memodifikasi status internal RentalSession dan objek terkait. Perubahan status yang terkontrol ini merupakan implementasi dari Encapsulation.

#### **Fitur 6: Sistem Reservasi (Dependency & Algorithm)**

Fitur ini mengelola pemesanan di masa depan melalui kelas Reservation. Logika pembuatannya memiliki Dependency pada data reservasi lain yang sudah ada untuk proses validasi. Metode buatReservasi(data) berisi algoritma penting untuk mendeteksi tumpang tindih jadwal (overlap) dengan memeriksa waktu mulai dan selesai reservasi baru terhadap semua reservasi yang ada untuk ruangan yang sama, sehingga mencegah *double booking*.

## L. Jadwal Proyek

Jadwal proyek PSRent Max disusun berdasarkan metodologi Scrum dengan 5 sprint, dimulai dari **10 November 2025** hingga **11 Januari 2026** (pekan UAS).

No	Tanggal	Kegiatan Utama	Deliverable
<b>1</b>	10-16 Nov 2025	- Kick-off meeting- Finalisasi requirement & use case- Pembuatan Class Diagram detail- Setup repository Git- Pembagian tugas per	<b>- Class Diagram lengkap- Product</b>
<b>2</b>	17-30 Nov 2025	- Implementasi Model: Person, Admin, Operator, Member, ConsoleUnit, Room, Enum- Implementasi Database (Singleton) + dummy data- Unit test untuk model	<b>- Model classes (100%)- Database dengan data dummy- Unit test pass</b>
<b>3</b>	1-14 Des 2025	- Implementasi View: MenuLogin, MenuRegister- Implementasi Controller: ControllerLogin, ControllerRegister- Implementasi RentalSession	<b>- Login &amp; Register UI functional- RentalSession logic complete- Demo: login sebagai Admin/Operator</b>
<b>4</b>	15-28 Des 2025	- Implementasi Strategy Pattern (StandardTarif, WeekendTarif, MemberTarif)- Implementasi TariffFactory- Implementasi Payment & generate struk- Implementasi Reservation + <b>overlap detection</b>	<b>- Tarif dinamis functional- Pembayaran &amp; struk selesai- Reservasi dengan validasi overlap</b>

No	Sprint	Tanggal	Kegiatan Utama	Deliverable
<b>5</b>	<b>Sprint 4</b>	29 Des 2025	- Implementasi	<b>- MenuAdmin</b>
		- 4 Jan 2026	MenuAdmin (tab inventori, operator, laporan)- Implementasi MenuKaryawan (tab sesi, reservasi)-	<b>functional-MenuKaryawan functional-Laporan dapat di-generate-Demo: alur lengkap start sesi</b>
			Laporan Harian & Bulanan- Integrasi	<b>→ end sesi → payment</b>
<b>6</b>	<b>Sprint (Final)</b>	5-11 Jan 2026	- Testing seluruh fitur (unit test + integration test)- Bug fixing- Dokumentasi (README, Javadoc, Persiapan demo & presentasi- Ekspor	<b>- Aplikasi final (bug-free)- Dokumentasi lengkap- Video demo- Proposal</b>

## **BAB IV KESIMPULAN**

Aplikasi ini dirancang dengan menerapkan konsep-konsep Pemrograman Berbasis Objek (PBO) secara menyeluruh, meliputi:

- Abstraction: Menggunakan abstract class (Person, Laporan) dan interface (RentalTarifStrategy)
- Encapsulation: Semua atribut bersifat private dengan akses melalui getter/setter
- Inheritance: Hierarki class Person → Admin/Operator/Member, Laporan → LaporanHarian/Bulanan
- Polymorphism: Strategy pattern untuk tarif dinamis (Standard/Weekend/Member)

Selain itu, aplikasi PSRent Max juga mengimplementasikan Design Pattern modern seperti:

- Singleton (Database): Memastikan satu-satunya instance untuk manajemen data
- Strategy (RentalTarifStrategy): Memudahkan penambahan jenis tarif baru tanpa mengubah kode existing
- Factory (TariffFactory): Otomatis memilih strategi tarif berdasarkan konteks
- Facade (Application): Menyederhanakan operasi kompleks menjadi API yang mudah digunakan

Dengan dibuatnya aplikasi ini, kami berharap kendala-kendala yang ada pada pengelolaan rental PlayStation secara manual dapat teratasi, meliputi:

- Efisiensi operasional: Proses start/end session, pembayaran, dan reservasi menjadi lebih cepat dan akurat
- Pencegahan konflik jadwal: Sistem otomatis mendeteksi overlap reservasi
- Perhitungan tarif konsisten: Tarif dinamis dihitung otomatis berdasarkan hari (weekday/weekend) dan status pelanggan (member/non-member)
- Pelaporan real-time: Admin dapat dengan mudah generate laporan pendapatan harian dan bulanan untuk analisis bisnis