

# RENCANA PEMBELAJARAN SEMESTER (RPS)

Berbasis Outcome-Based Education (OBE)

## PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS LOREM IPSUM

### 1. Identitas Mata Kuliah

Nama Program Studi : Teknik Informatika  
Nama Mata Kuliah : Pemrograman Berorientasi Objek  
Kode Mata Kuliah : TIF-301  
Semester : 3 (Tiga)  
SKS / Bobot Kredit : 3 SKS (2 Teori, 1 Praktikum)  
Dosen Pengampu : Dr. Lorem Ipsum, M.Kom.  
Tanggal Penyusunan : 31 Januari 2026

### 2. Capaian Pembelajaran Lulusan (CPL)

CPL yang dibebankan pada mata kuliah ini mencakup kompetensi lulusan dalam aspek pengetahuan, keterampilan, dan sikap:

- **CPL-1 (Pengetahuan):** Menguasai konsep teoretis bidang pengetahuan tertentu secara umum dan konsep teoretis bagian khusus dalam bidang pengetahuan tersebut secara mendalam, serta mampu memformulasikan penyelesaian masalah prosedural.
- **CPL-2 (Keterampilan Umum):** Mampu menerapkan pemikiran logis, kritis, sistematis, dan inovatif dalam konteks pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora.
- **CPL-3 (Keterampilan Khusus):** Mampu merancang, mengimplementasikan, dan mengevaluasi sistem perangkat lunak menggunakan paradigma berorientasi objek dengan mempertimbangkan standar kualitas perangkat lunak.
- **CPL-4 (Sikap):** Menunjukkan sikap bertanggung jawab atas pekerjaan di bidang keahliannya secara mandiri dan mampu bekerja sama dalam tim.

### 3. Capaian Pembelajaran Mata Kuliah (CPMK)

Kemampuan atau kompetensi spesifik yang diharapkan mahasiswa kuasai setelah menyelesaikan mata kuliah:

- **CPMK-1:** Mahasiswa mampu memahami dan menjelaskan konsep dasar pemrograman berorientasi objek (class, object, encapsulation, inheritance, polymorphism).
- **CPMK-2:** Mahasiswa mampu merancang solusi permasalahan menggunakan diagram UML (Use Case, Class Diagram, Sequence Diagram).
- **CPMK-3:** Mahasiswa mampu mengimplementasikan konsep OOP dalam bahasa pemrograman Java dengan menerapkan prinsip SOLID.
- **CPMK-4:** Mahasiswa mampu menganalisis dan mengevaluasi kualitas kode program berorientasi objek berdasarkan best practices dan design patterns.

## 4. Sub-CPMK / Indikator Pencapaian

Penjabaran CPMK menjadi indikator yang lebih terukur dan dapat diuji:

- **Sub-CPMK 1.1:** Menjelaskan perbedaan antara pemrograman prosedural dan berorientasi objek
- **Sub-CPMK 1.2:** Mengidentifikasi class, object, dan attribute dalam studi kasus nyata
- **Sub-CPMK 1.3:** Mendemonstrasikan konsep encapsulation dengan access modifier
- **Sub-CPMK 2.1:** Membuat use case diagram untuk sistem sederhana
- **Sub-CPMK 2.2:** Merancang class diagram dengan relasi yang tepat (association, aggregation, composition)
- **Sub-CPMK 3.1:** Mengimplementasikan inheritance dan method overriding
- **Sub-CPMK 3.2:** Menerapkan polymorphism dalam desain aplikasi
- **Sub-CPMK 4.1:** Menganalisis code smell dan melakukan refactoring
- **Sub-CPMK 4.2:** Mengimplementasikan minimal 3 design patterns (Singleton, Factory, Observer)

## 5. Materi Pembelajaran (Bahan Kajian)

Daftar topik materi yang relevan dengan Sub-CPMK dan CPMK:

1. Pengenalan Paradigma Berorientasi Objek
2. Konsep Class dan Object
3. Encapsulation dan Information Hiding
4. Inheritance dan Hierarchical Relationships

5. Polymorphism dan Dynamic Binding
6. Abstract Class dan Interface
7. UML Diagrams (Use Case, Class, Sequence)
8. Exception Handling dalam OOP
9. Prinsip SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion)
10. Design Patterns (Creational, Structural, Behavioral)
11. Collections dan Generics
12. File I/O dan Serialization
13. Unit Testing dan Test-Driven Development
14. Code Quality dan Refactoring

## 6. Metode Pembelajaran

Strategi atau pendekatan pembelajaran yang dipilih sesuai OBE yang menekankan aktivitas mahasiswa:

- **Ceramah Interaktif:** Penjelasan konsep dengan diskusi tanya jawab
- **Problem-Based Learning (PBL):** Mahasiswa menyelesaikan permasalahan nyata menggunakan OOP
- **Project-Based Learning:** Pengembangan aplikasi mini sebagai proyek kelompok
- **Praktikum Terbimbing:** Latihan coding dengan bimbingan asisten
- **Peer Review:** Mahasiswa melakukan code review terhadap pekerjaan rekan
- **Flipped Classroom:** Mahasiswa mempelajari materi sebelum kelas, diskusi mendalam di kelas
- **Studi Kasus:** Analisis implementasi OOP pada aplikasi open source

## 7. Pengalaman Belajar Mahasiswa

Deskripsi tugas, aktivitas, atau pengalaman belajar yang mendukung pencapaian Sub-CPMK:

- Menganalisis studi kasus untuk mengidentifikasi class dan object
- Merancang class diagram untuk sistem perpustakaan, e-commerce, atau sistem akademik
- Mengimplementasikan aplikasi console berbasis OOP dengan minimal 5 class
- Melakukan refactoring pada kode yang diberikan untuk meningkatkan kualitas
- Berkolaborasi dalam tim untuk mengembangkan proyek aplikasi desktop sederhana
- Mempresentasikan hasil analisis design pattern pada framework populer
- Menulis unit test untuk class yang telah dibuat
- Melakukan peer code review dan memberikan feedback konstruktif
- Membuat dokumentasi teknis menggunakan Javadoc

## 8. Kriteria, Indikator, dan Bobot Penilaian

Teknik/alat asesmen dipetakan ke Sub-CPMK/CPMK dengan bobot yang jelas:

Komponen	Teknik Asesmen	Indikator/CPMK	Bobot (%)
Tugas Individu	Coding Assignment	Sub-CPMK 1.1, 1.2, 1.3, 3.1	15
Kuis	Multiple Choice & Essay	Sub-CPMK 1.1, 2.1, 2.2	10
Praktikum	Lab Exercise	Sub-CPMK 3.1, 3.2, 4.1	15
UTS	Written Exam & Coding	CPMK-1, CPMK-2	20
Proyek Kelompok	Application Development	CPMK-2, CPMK-3, CPMK-4	20
UAS	Comprehensive Exam	CPMK-1, CPMK-2, CPMK-3, CPMK-4	20
<b>Total</b>			<b>100</b>

### Kriteria Penilaian:

- A (85-100): Menguasai semua CPMK dengan sangat baik, mampu menerapkan dalam kasus kompleks
- B (70-84): Menguasai sebagian besar CPMK dengan baik
- C (60-69): Menguasai CPMK dasar dengan cukup

- D (50-59): Menguasai sebagian kecil CPMK
- E (<50): Belum menguasai CPMK yang ditetapkan

## 9. Evaluasi dan Refleksi Pembelajaran

Penilaian sumatif/formatif untuk memantau ketercapaian outcome secara menyeluruh:

- **Evaluasi Formatif:** Kuis mingguan, latihan coding, peer review untuk memberikan feedback berkelanjutan
- **Evaluasi Sumatif:** UTS dan UAS untuk mengukur pencapaian CPMK secara komprehensif
- **Refleksi Mahasiswa:** Jurnal belajar mingguan untuk refleksi diri terhadap pemahaman materi
- **Evaluasi Dosen:** Survey kepuasan mahasiswa di tengah dan akhir semester
- **Continuous Improvement:** Analisis hasil penilaian untuk perbaikan RPS di semester berikutnya

## 10. Daftar Referensi

Sumber belajar utama yang digunakan dalam penyusunan materi dan asesmen:

1. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
2. Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley Professional.
3. Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
4. Horstmann, C. S. (2019). *Core Java Volume I—Fundamentals* (11th ed.). Prentice Hall.
5. Freeman, E., & Robson, E. (2020). *Head First Design Patterns* (2nd ed.). O'Reilly Media.
6. Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code* (2nd ed.). Addison-Wesley Professional.
7. Oracle. (2024). *The Java Tutorials*. Retrieved from <https://docs.oracle.com/javase/tutorial/>
8. Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* (3rd ed.). Prentice Hall.

Disusun oleh,

**Dr. Lorem Ipsum, M.Kom.**

NIP. 123456789012345678