

Bab 11

Design Patterns Dasar

11.1 Design Patterns: Creational Patterns

Sub-CPMK yang Dicakup dalam Bab Ini:

- Sub-CPMK 4.2: Mengimplementasikan minimal 3 design patterns (Singleton, Factory, Observer)

Konsep Penting

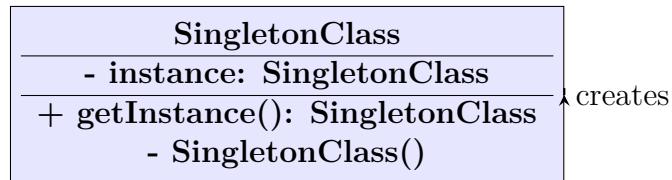
Design Pattern adalah solusi umum untuk masalah yang sering muncul dalam konteks desain perangkat lunak [2]. Pattern bukanlah desain akhir yang dapat langsung diubah menjadi kode, melainkan template tentang cara menyelesaikan masalah.

11.1.1 Singleton Pattern

Menjamin hanya ada satu instance dari class.

```
1 public class DatabaseConnection {  
2     private static DatabaseConnection instance;  
3  
4     private DatabaseConnection() {}  
5  
6     public static DatabaseConnection getInstance() {  
7         if (instance == null) {  
8             instance = new DatabaseConnection();  
9         }  
10        return instance;  
11    }  
12}
```

Kode Program 11.1: Singleton Pattern



Gambar 11.1: Struktur Singleton Pattern

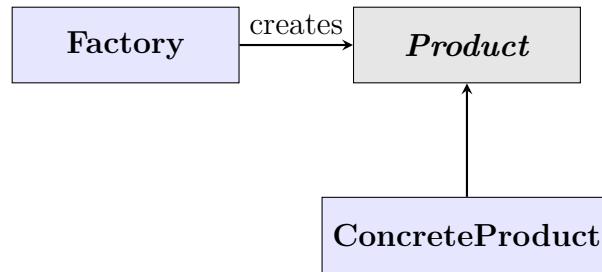
11.1.2 Factory Pattern

Menyediakan cara membuat objek tanpa mengekspos logika instansiasi.

```

1 interface Kendaraan {
2     void jalan();
3 }
4
5 class Mobil implements Kendaraan {
6     public void jalan() {
7         System.out.println("Mobil berjalan");
8     }
9 }
10
11 class Motor implements Kendaraan {
12     public void jalan() {
13         System.out.println("Motor berjalan");
14     }
15 }
16
17 class KendaraanFactory {
18     public static Kendaraan buat(String jenis) {
19         if ("mobil".equalsIgnoreCase(jenis)) return new Mobil()
20             ;
21         if ("motor".equalsIgnoreCase(jenis)) return new Motor()
22             ;
23         return null;
24     }
25 }
```

Kode Program 11.2: Factory Pattern



Gambar 11.2: Struktur Factory Pattern

11.2 Behavioral Pattern: Observer

11.2.1 Observer Pattern

Observer digunakan ketika satu objek (subject) perlu memberi tahu banyak objek lain (observer) saat ada perubahan.

```

1 interface Observer {
2     void update(String pesan);
3 }
4
5 class Subscriber implements Observer {
6     private String nama;
7
8     public Subscriber(String nama) {
9         this.nama = nama;
10    }
11
12    public void update(String pesan) {
13        System.out.println(nama + " menerima: " + pesan);
14    }
15 }
  
```

Kode Program 11.3: Observer Pattern Sederhana

11.2.2 Kapan Menggunakan Design Pattern

Gunakan pattern saat:

- Solusi berulang muncul di banyak bagian aplikasi
- Dibutuhkan struktur yang konsisten
- Tim perlu pola komunikasi yang jelas

Aktivitas Pembelajaran

1. **Implementasi Singleton:** Buat class **Logger** sebagai singleton.
2. **Factory:** Buat factory untuk membuat objek pembayaran.
3. **Observer:** Buat sistem notifikasi berita dengan subject dan observer.
4. **Diskusi:** Identifikasi pattern pada framework Java (misalnya Spring).

Latihan dan Refleksi

1. Jelaskan perbedaan Factory dan Singleton.
2. Buat contoh kasus yang cocok menggunakan Observer.
3. Apa resiko jika Singleton digunakan berlebihan?
4. Tambahkan pattern Strategy pada studi kasus diskon.
5. **Refleksi:** Pattern mana yang paling membantu desain Anda?

Asesmen (Evaluasi Kinerja) Instrumen Penilaian untuk Sub-CPMK 4.2

A. Pilihan Ganda

1. Pattern yang menjamin hanya satu instance adalah:
 - (a) Factory
 - (b) Singleton
 - (c) Observer
 - (d) Strategy
2. Observer pattern cocok digunakan untuk:
 - (a) Membuat banyak objek
 - (b) Mengelola notifikasi perubahan
 - (c) Menghitung data
 - (d) Menyimpan konfigurasi

B. Tugas Praktik

- Implementasikan Singleton, Factory, dan Observer pada satu studi kasus sederhana.

Rubrik Penilaian: Lihat Lampiran A

Checklist Pencapaian Kompetensi

Centang item berikut setelah Anda yakin telah menguasainya:

- Saya memahami konsep design patterns
- Saya dapat menerapkan Singleton
- Saya dapat menerapkan Factory
- Saya dapat menerapkan Observer
- Saya dapat memilih pattern sesuai kebutuhan

Rangkuman

Design patterns menyediakan solusi teruji untuk masalah desain umum [2, 1]. Singleton, Factory, dan Observer adalah pola dasar yang sering digunakan di aplikasi nyata.

Kata Kunci: *Design Pattern, Singleton, Factory, Observer, Strategy*

Daftar Pustaka

- [1] Eric Freeman and Elisabeth Robson. *Head First Design Patterns*. O'Reilly Media, 2nd edition, 2020.
- [2] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.