

Nama = Tiara putri Elisa

Nim = 121140049

Kelas = Praktikum Pemweb RA

### \* Desain Pattern

LD merupakan pendekatan untuk memecahkan masalah desain perangkat lunak yang umum. Mereka menyediakan solusi terstruktur, efisien, dan dapat dipahami untuk tantangan yang sering muncul selama pengembangan perangkat lunak.

Beberapa desain pattern yang sering di gunakan :

#### 1. Singleton Pattern (solusi unik)

LD memastikan bahwa sebuah kelas memiliki satu instance dan menyediakan titik akses global ke instance tersebut. contoh = Pengelolaan koneksi database

#### 2. Factory Method Pattern (Pembuat Objek yang Dinamis)

LD Menyediakan antarmuka untuk membuat objek, tetapi membiarkan subclass memodifikasi jenis objek yang di buat. Contoh = Framework pengembangan game dengan kelas abstrak "character Factory" untuk menciptakan karakter

#### 3. Adapter pattern (menghubungkan yang berbeda)

LD mengubah antarmuka suatu kelas ke antarmuka lain yang diharapkan oleh klien. Contoh : adapter untuk mengintegrasikan sistem pembayaran lama dengan sistem baru dalam aplikasi e-commerce.

#### 4. Decorator Pattern (mempercantik dengan fleksibilitas)



↳ Menambahkan fungsionalitas pada objek tanpa mengubah struktur dasarnya. Contoh: penambahan fitur logging atau enkripsi pada objek stream data.

#### 5. Observer Pattern (Komunikasi Dinamis)

↳ Memungkinkan objek untuk menginformasikan objek lain tentang perubahan keadaannya. Contoh: Sistem notifikasi dalam aplikasi Perpesanan Instan.

#### 6. Strategy Pattern (Algoritma yang dipilih dengan bijak)

↳ Mendefinisikan serangkaian algoritma dan memungkinkan klien memilih algoritma yang sesuai. Contoh: pengaturan strategi pembayaran dalam sistem e-commerce

#### \* Model, View, Controller (MVC)

↳ paradigma arsitektur perangkat lunak yang memisahkan logika aplikasi menjadi 3 komponen utama.

##### 1. Model = fondasi logika dan data

↳ model mewakili data dan aturan bisnis

aplikasi. Bertanggung jawab untuk mengelola, memanipulasi, dan menyimpan data aplikasi.

Perannya = Menangani logika bisnis, berkomunikasi dengan database, dan memberikan antarmuka untuk interaksi dengan data

Contoh = Kelas yang mempresentasikan entitas seperti pengguna, produk dan faktur dalam aplikasi e-commerce



## 2. View = Representasi Visual dan Antarmuka Pengguna

↳ Tanggung jawab untuk menampilkan data dari model kepada pengguna dan menerima input pengguna

Perannya = Menyajikan informasi kepada pengguna secara grafis atau tekstual.

Menerima input dan meneruskannya ke controller.

Contoh = Halaman web, jendela aplikasi, atau elemen antarmuka seperti tombol formulir.

## 3. Controller = pengelut interaksi dan logika aplikasi

↳ controller bertindak sebagai perantara antara model dan view. Menerima input dari pengguna, memprosesnya, dan memperkarui model atau view sesuai kebutuhan.

peran = Menginterpretasikan input pengguna, mengkoordinasikan interaksi antara model dan view, mengatur alur kontrol aplikasi.

Contoh = pengontrol navigasi pada aplikasi mobile, atau skrip pengontrol yang mengelola logika pemrosesan formulir pada situs web.