

Nama : Elisabet Nandra Hagaina Sembiring  
NIM : 121140101  
Kelas : RA

### Design Pattern

merupakan sebuah metode yang digunakan dalam rekayasa perangkat lunak untuk menyelesaikan masalah - masalah yang sering muncul. Pola design ini merupakan solusi umum yang dapat diterapkan kembali dalam berbagai konteks pengembangan perangkat lunak.

Ada 3 pola utama dalam design Pattern :

- Creational Pattern
- Behavioral Pattern
- Structural Pattern

#### 1. Creational Pattern

Creational Pattern berkaitan dengan berkaitan dengan proses pembuatan objek atau instance dari suatu kelas atau tipe data. Tujuan utama dari creational pattern adalah untuk menyederhanakan kompleksitas dan pembuatan objek dan membuatnya lebih fleksibel.

Contoh Creational Pattern : Singleton pattern, Factory Method pattern, Abstract Factory Pattern, dll.

#### 2. Behavioral Pattern

Behavioral Pattern adalah pola perilaku, satu jenis design patterns dalam paradigma pemrograman berorientasi objek atau kelas berinteraksi dan bekerjasama. Tujuan behavioral Pattern adalah memfasilitasi komunikasi yang baik antar objek, mengelola tanggungjawab dan meningkatkan fleksibilitas dalam perubahan perilaku sistem.

Contoh behavioral Pattern : Observer Pattern, Strategy Pattern, Command Pattern.

#### 3. Structural Pattern

Structural Pattern adalah salah satu kategori design pattern dalam pemrograman berorientasi objek yang berkaitan dengan cara kelas dan objek dibentuk agar membentuk struktur yang lebih besar.

Tujuannya adalah untuk mempermudah komposisi objek dan kelas kedalam struktur yang lebih besar serta memastikan bahwa perubahan dalam suatu bagian dari sistem tidak mempengaruhi seluruh sistem.

Contoh Structural Pattern : Adapter Pattern, Composite Pattern, Bridge Pattern.

Design Pattern memiliki beberapa kelebihan, antara lain:

- memberi solusi atas masalah yang timbul saat pengembangan perangkat lunak.
- membuat penulisan kode menjadi lebih rapi, terstruktur dan mudah dibaca.
- mempercepat proses pengembangan perangkat lunak dengan memungkinkan penggunaan kembali pola design.

Kekurangan design Pattern adalah kompleksitas yang tinggi dan membutuhkan usaha yang lebih dalam pengembangan perangkat lunak.



## Konsep MVC (Model View Controller) Design Pattern.

Konsep MVC adalah sebuah design pattern yang digunakan dalam pengembangan perangkat lunak untuk memisahkan komponen aplikasi menjadi 3 bagian utama yaitu Model, View dan Controller

Tujuan utama dari MVC untuk mengorganisir dan memisahkan tugas tugas berbeda dalam aplikasi agar lebih mudah dimengerti, dikelola dan diubah.

Komponen MVC :

### 1. Model.

- merupakan bagian dari aplikasi yang bertanggungjawab untuk mengelola data dan logika bisnis
- menyediakan antar muka untuk mengakses dan memanipulasi data.
- Tidak mengetahui tentang tampilan, Controller sehingga dapat digunakan kembali tanpa memodifikasi ketika tampilan atau logika pengendali berubah.

### 2. View.

- menangani tampilan atau presentasi data kepada pengguna.
- mengambil informasi dari model dan menampilkan data sesuai dengan aturan tertentu.
- tidak berurusan dengan logika bisnis atau perubahan data, hanya bertanggungjawab untuk tampilan.

### 3. Controller

- Bertindak sebagai perantara model dan view
- menangani input dari pengguna, memprosesnya dan memperbaharui model sesuai dengan tindakan yang diambil.
- memisahkan logika bisnis dari tampilan, sehingga perubahan dalam tampilan atau model tidak mempengaruhi logika pengendali

Contoh Kode Model :

```
class User
{
    protected $table = 'users';

    protected $fillable = [
        'name',
        'email',
        'password',
        'created-at',
        'update-at',
    ];
}
```

Penjelasan Kode :

- Kode model user memiliki tiga properti : id, name dan email. Properti-properti ini digunakan untuk menyimpan informasi tentang pengguna.
- Properti id digunakan untuk menyimpan ID pengguna. ID ini biasanya merupakan nilai numerik yang unik untuk setiap pengguna.
- Properti name digunakan untuk menyimpan nama pengguna. Nama pengguna ini dapat berupa nama lengkap atau panggilan.
- Properti email digunakan untuk menyimpan alamat email pengguna. alamat digunakan untuk mengirimkan informasi kepada pengguna.

Contoh Kode View :

```
<? php
$user = User::all();
echo '<h1> Daftar user </h1>';
:
?>
```

Penjelasan Kode :

- \$user = User::all(); digunakan untuk mendapatkan data dari model, dalam hal ini akan didapatkan adalah daftar semua user.
- echo '<h1> daftar user </h1>'; Baris kode ini digunakan untuk menampilkan judul halaman.



Contoh Kode Controller:

```
<? php
```

```
class UserController
```

```
{
```

```
    protected $userModel;
```

```
    public function __construct()
```

```
    {
```

```
        $this->userModel = new UserModel();
```

```
    }
```

```
    public function index()
```

```
    {
```

```
        $user = $this->userModel->getAllUser();
```

MVC digunakan secara luas dalam pengembangan perangkat lunak berbasis GUI (Graphical User Interface) seperti aplikasi web dan desktop. Ada variasi dan implementasi khusus dari MVC seperti Model - View - Presenter (MVP) dan Model - View - View Model (MVVM).

Penjelasan Kode:

- Mendefinisikan kelas `UserController`.
- Membuat properti `$userModel` yang akan menyimpan objek dari kelas `User Model`. digunakan untuk berinteraksi dengan model pengguna.