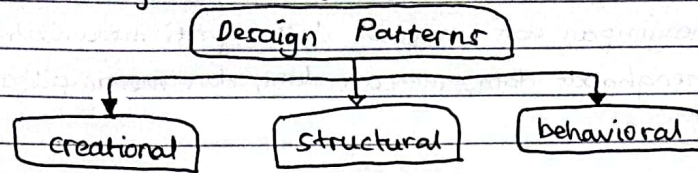


Nama : RA Siti Zakiyah  
NIM : 121140103  
Kelas : RA

5 Desember 2023

## Design Pattern

- ↳ Design Pattern adalah solusi umum untuk masalah yang biasa terjadi dalam desain perangkat lunak. Design pattern bukan merupakan metode yang langsung dapat diubah menjadi kode program melainkan sebuah pola dasar atau template dari cara untuk menyelesaikan masalah



### ↳ Jenis - jenis Design pattern

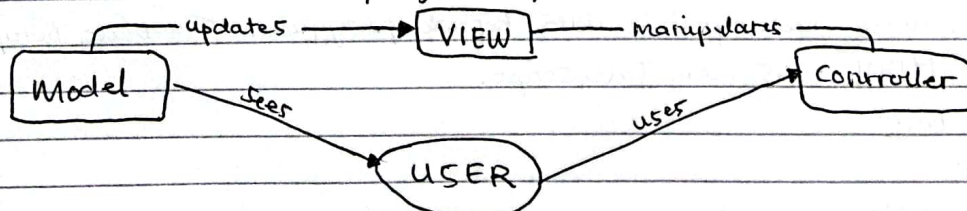
- ↳ Creational pattern adalah design pattern yang digunakan untuk membuat objek
- ↳ structural pattern adalah design pattern yang digunakan untuk mengatur hubungan antar objek.
- ↳ behavioral pattern adalah desain pattern yang digunakan untuk mengatur hubungan antar objek.

### ↳ Keuntungan Menggunakan Design Pattern

- ↳ aplikasi menjadi lebih mudah dipelihara
- ↳ aplikasi menjadi lebih mudah dikembangkan
- ↳ aplikasi menjadi lebih fleksibel
- ↳ aplikasi menjadi lebih terstruktur
- ↳ aplikasi menjadi lebih mudah di uji

### ↳ konsep MVC ( Model - View - Controller ) Design Pattern

- ↳ MVC adalah suatu pola desain yang digunakan untuk memisahkan logika aplikasi menjadi tiga komponen utama, yaitu Model, view dan Controller. Konsep ini bertujuan untuk meningkatkan modularitas, fleksibilitas, dan pemeliharaan kode dalam pengembangan perangkat lunak.



### PHP MVC framework

- |               |                  |           |           |
|---------------|------------------|-----------|-----------|
| ↳ laravel     | ↳ zero framework | ↳ Yii     | ↳ PHPIXIE |
| ↳ Codeigniter | ↳ Cake PHP       | ↳ Slim    |           |
| ↳ Symfony     | ↳ Fuel PHP       | ↳ Phalcon |           |



## ↳ Java Script MVC Framework

- ↳ Angular JS
- ↳ Backbone.js
- ↳ Ember.js
- ↳ Meteor
- ↳ React
- ↳ Vue.js

## ↳ Komponen MVC

1. Model : untuk menyimpan dan mengolah data seperti menambah data, mengubah data, menghapus data, mencari data, dan menampilkan data.

### contoh kode

```
class User {  
    protected $table = 'users';  
  
    protected $fillable = [  
        'name',  
        'email',  
        'password',  
        'created_at',  
        'updated_at'  
    ];  
}
```

### penjelasan :

- ↳ kode model User memiliki tiga properti: id, name dan email yang digunakan untuk menyimpan informasi tentang pengguna.
- ↳ properti id digunakan untuk menyimpan id pengguna yang merupakan nilai numerik yang unik untuk setiap pengguna.
- ↳ properti name digunakan untuk menyimpan nama pengguna, dapat berupa nama lengkap atau nama panggilan.
- ↳ properti email digunakan untuk menyimpan alamat email pengguna yang digunakan untuk mengirimkan informasi kepada pengguna.

2. View : untuk menampilkan data kepada pengguna. View biasa berupa HTML, CSS, dan JavaScript.

### contoh kode

```
<? php  
$users = User::all();  
echo '<h1> Daftar User</h1>';  
foreach ($users as $user) {  
    echo '<p> Nama: ' . $user->name . '</p>';  
    echo '<p> Email: ' . $user->email . '</p>';  
}
```





```
}  
?>
```

Penjelasan :

- ↳ `$users = User::all();` Baris kode ini digunakan untuk mendapatkan data dari model. Dalam hal ini, data yang akan didapatkan adalah daftar semua user
- ↳ `echo '<h1>Daftar User</h1>';` Baris kode ini digunakan untuk menampilkan judul halaman.
- ↳ Baris code `foreach - echo` digunakan untuk menampilkan daftar semua user. setiap user akan ditampilkan satu baris

3. Controller : untuk menerima input dari pengguna dan memprosesnya dan untuk mengirimkan data dari model ke view.

contoh code

```
<?php
```

```
class UserController
```

```
{
```

```
    protected $userModel;
```

```
    public function __construct()
```

```
    {
```

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

```
    }
```

```
    public function index () {
```

```
        $users = $this->userModel->getAllUsers();
```

```
    }
```

```
?>
```

penjelasan :

↳ mendefinisikan kelas UserController

↳ membuat properti (variabel anggota) `$userModel` yang akan menyimpan objek dari kelas `UserModel`. properti ini akan digunakan untuk berinteraksi dengan model pengguna.