

## Laporan Praktikum Pemrograman web Modul 9

Nama: Umy Afifah

NIM : 121140087

Kelas : RA

### A. Design Pattern

adalah solusi umum untuk masalah yang biasa terjadi dalam desain Perangkat lunak. Design Pattern bukan merupakan kode berupa Program melainkan pada dasar, atau template dan cara untuk menyelesaikan masalah.

Barangknya, design Pattern Seperti resep makanan.

Saat kita ingin membuat sebuah makanan, kita bisa mengikuti resep yang ada, resep tersebut akan memberikan panduan tentang bahan serta langkah-langkahnya. Dengan mengikuti resep tersebut kita dapat membuat masakan yang lezat sesuai yang diinginkan.

• Terdapat 3 Jenis Design Pattern

1.) Creational Pattern : Untuk membuat Objek

2.) Structural Pattern : Untuk mengatur hubungan antar objek

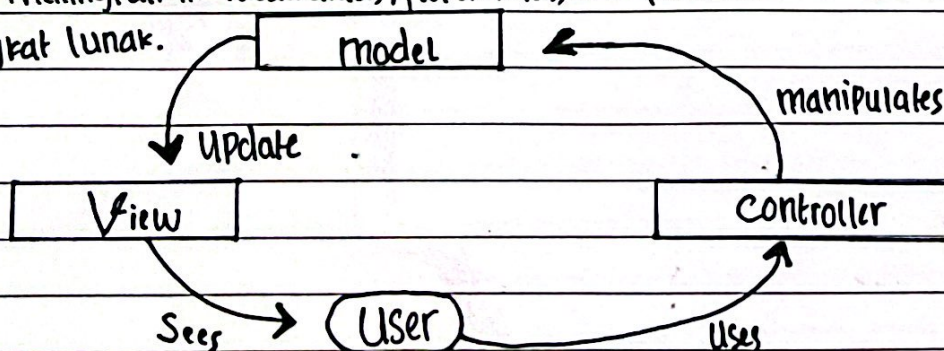
3.) Behavioral Pattern : Untuk mengatur perilaku objek

• Keuntungan menggunakan Design Pattern

Aplikasi menjadi lebih mudah dipelihara, lebih mudah dikembangkan, lebih fleksibel, lebih terstruktur, lebih mudah diuji.

### B. Konsep MVC

Model - View - Controller adalah suatu pola desain yang digunakan untuk memisahkan logika aplikasi menjadi 3 komponen utama : Model, View, dan Controller. Konsep ini bertujuan untuk meningkatkan modularitas, fleksibilitas, dan pemeliharaan kode dalam pengembangan Perangkat lunak.





## • PHP MVC Framework

Diantaranya: Laravel, Condelgiter, Sympony, Zen Framework, Cake PHP, Fuel PHP, Yii, Slim, Phalcon.

## • Java Script MVC Framework

Diantaranya: Angular JS, Backbone JS, Ember JS, Meteor, React dan Vue JS.

## • Komponen MVC

### 1.) Model

Model bertanggung jawab untuk menyimpan dan mengelola data. Model tidak boleh bergantung pada tampilan atau kontrol. Model dapat digunakan untuk mengakses dan manipulasi data yang disimpan dalam database atau table. Model dapat digunakan untuk melakukan berbagai operasi seperti: Menambah data, Mengubah data, Menghapus data, Mencari data, Menampilkan data.

### 2.) View

View bertanggung jawab untuk menampilkan data pengguna. View tidak boleh bergantung pada model atau kontrol. View adalah bagian dari aplikasi web yang bertanggung jawab untuk menampilkan data ke pengguna. View biasanya berupa file HTML, CSS dan Java Script. View dapat menampilkan data dari berbagai sumber termasuk: model, database, API.

### 3.) Controller

Controller bertanggung jawab untuk menerima input dari pengguna dan memprosesnya. Controller bertanggung jawab untuk mengirimkan data dari model ke View.

## • Contoh KODE MODEL

```
class User
```

```
{
```

```
protected $table = 'users';
```

```
protected $fillable = [
```

```
    'name'
```

```
    'email'
```

```
    'password'
```

```
    'created_at'
```

```
    'updated_at'
```

```
];
```

```
}
```

## Penjelasan KODE MODEL

- Kode model user memiliki tiga properti: id, name dan

- Email. Property ini digunakan untuk menyimpan

- Informasi tentang pengguna.

- Property id: Untuk menyimpan id pengguna.

- Property name: Untuk menyimpan nama pengguna.

- Property Email: Untuk menyimpan alamat email

- Pengguna.



### • Contoh Kode View

### Penjelasan Kode View

```
<? 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>';
```

```
}
```

```
?>
```

• \$user = User::all(); Baris ini digunakan untuk mendapatkan data dari model.

Data yang didapatkan adalah daftar semua user.

• echo '<h1> Daftar User </h1>';

Baris kode ini digunakan untuk menampilkan judul halaman.

### • Contoh Output

Daftar User

Nama : John Doe

Email : johndoe@example.com

Nama : John Doe

Email : janedoe@example.com

### • Contoh Kode Controller

### Penjelasan Kode Controller

```
<? PHP
```

```
class UserController
```

```
{
```

```
protected $user Model;
```

```
public function _Construct()
```

```
{
```

```
$this->User Model = new User model();
```

```
}
```

```
public function index()
```

```
{
```

```
$users = $this->User Model->getAllUsers();
```

```
}
```

```
?>
```

• Mengidentifikasi kelas User Controller.

• Membuat Properti (Variabel anggota)

\$user model yang akan menyimpan

objek dari kelas User model. Properti ini akan digunakan untuk berinteraksi dengan model pengguna.