

LAPORAN PRAKTIKUM PEMWEB

KELAS RA

Nama : Fadillah Rizky Junglunan

NIM : 121140194

Pertemuan : 9 (Design Pattern)

A. Design Pattern

adalah solusi umum untuk masalah yg biasa terjadi dalam desain perangkat lunak.

Design Pattern bukan merupakan kode berupa program melainkan pola dasar atau template dari cara untuk menyelesaikan masalah.

ibaratnya, design pattern seperti resep makanan.

Saat kita ingin membuat sebuah makanan, kita bisa mengikuti resep yg ada, resep tersebut akan memberikan panduan tentang bahan serta langkah-langkahnya. dengan mengikuti resep tersebut kita dapat membuat masakan yg lezat sesuai yg di inginkan.

• Jenis-jenis Design Pattern

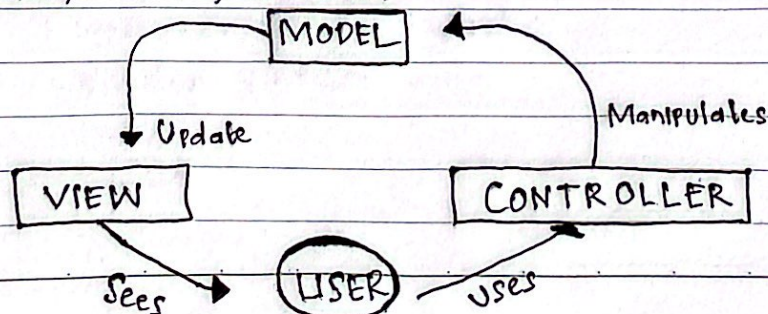
- terdapat 3 Jenis :
- 1). Creational pattern : untuk membuat objek
 - 2). Structural Pattern : untuk mengatur hubungan antar objek
 - 3). Behavioral Pattern : untuk mengatur perilaku objek

• Keuntungan menggunakan Design Pattern

diantaranya : Aplikasi menjadi lebih mudah dipelihara, lebih mudah dikembangkan, lebih fleksibel, lebih terstruktur, lebih mudah di uji

B. Konsep MVC

MVC atau Model - View-Controller adalah suatu pola desain yg digunakan untuk memisahkan logika aplikasi menjadi tiga 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 , CodeIgniter, Symfony , Zen Framework , Cake PHP, Fuel PHP, Yii, Slim , Phalcon dan PHPixie

• JavaScript 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 memanipulasi data yg disimpan dalam database atau table . Model dapat digunakan untuk melakukan berbagai operasi , seperti :

1). menambah data	4). Mencari data
2). mengubah data	5). Menampilkan data
3). menghapus data	

2). View

View bertanggung jawab untuk menampilkan data kepada pengguna . View tidak boleh bergantung pada model atau kontrol . View adalah bagian dari aplikasi web yg bertanggung jawab untuk menampilkan data ke pengguna . View biasanya berupa file HTML , CSS dan JavaScript .

• View dapat menampilkan data dari berbagai sumber , termasuk :

- 1). model
- 2). Database
- 3). 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

1). MODEL : class User

```
{
    protected $table = 'users' ;
    protected $fillable = {
        'name',
        'email',
        'password',
        'created',
        'updated' ;
    } ;
}
```


• Penjelasan :

- Kode model User memiliki tiga properti : id, name dan email. Properti - properti ini digunakan untuk menyimpan informasi tentang pengguna.
- Properti id : untuk menyimpan id pengguna
- Properti name : untuk menyimpan nama pengguna
- Properti email : untuk menyimpan alamat email pengguna

• CONTOH 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> ' ;
}
```

```
?>
```

• Penjelasan :

- \$users = User :: all () ;

Baris kode ini digunakan untuk mendapatkan data dari model. dalam hal ini, data yg akan didapatkan adalah daftar semua user.

- echo ' <h1> Daftar User </h1>' ;

Baris kode ini digunakan untuk menampilkan judul halaman.

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

Baris kode ini digunakan untuk menampilkan daftar semua user. Setiap user akan ditampilkan dalam satu baris

• Contoh OUTPUT

Daftar User

Nama : John Doe

Email : johndoe@example.com

Nama : Jane Doe

Email : janedoe@example.com

• CONTOH KODE CONTROLLER

< ?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 yg akan menyimpan objek dari kelas UserModel. properti ini akan digunakan untuk berinteraksi dengan model pengguna.