Varell Anthonio (121140122)

Laporan Praktikum 8
Pemrograman web RA

A. Design Pattern

↳ Solusi utk masalah desain perangkat lunak merupakan panduan utk memecahkan masalah desain dlm pengembangan suatu perangkat lunak

Jenis" design pattern
1. Creational Patterns, mengatur proses penciptaan objek
2. Structural Patterns, fokus pd komposisi objek dan kelas
3. Behavioral Patterns, mengatur alur eksekusi dan komunikasi antar objek

Design Patterns
- → Creational → factory method, Builder, abstract factory, singleton
- → Structural → Adapter, Decorator, Facade, Proxy
- → Behavioral → observer, strategy, command, state, chain of responsibility, Visitor

Keuntungan penggunaan design pattern
1. reusability, dapat digunakan kembali karena menyediakan solusi untuk masalah umum
2. flexibility, memungkinkan perubahan tanpa merusak struktur keseluruhan
3. Maintainability, lebih mudah dipahami sehingga mempermudah pemeliharaan
4. Scalability, mengatur kompleksitas seiring waktu
5. encapsulation of best practices, mengikuti praktik terbaik dan sudah teruji tepat dan handal

B. Konsep MVC (Model - View - Controller)

MVC adalah pola design pattern yg membagi aplikasi menjadi 3 komponen utama setiap komponen memiliki tanggung jawab tertentu dlm sistem

Komponen MVC
1. Model, mewakili struktur data dan logika bisnis, model mengelola data dan mengolahnya untuk keperluan dari data tsb
2. View, bertanggung jawab utk menampilkan informasi kpd pengguna, menjadi tampilan grafis atau lainnya
3. Controller, mengatur input pengguna menginterpretasikan perintah dan menginstruksikan model dan View utk melakukan tindakan

Framework MVC:

1. Ruby on Rails (RoR) → Ruby
2. Django → Python
3. Laravel → PHP
4. Spring framework → Java
5. express.js. → Javascript (Node.js)
6. ASP.net MVC → ASP.NET (C#)

Contoh Kode program:

Model PHP

```
Class User {
    Protected $table = "users";
    Protected $Fillable = [
            'name',
            'email',
            'Password',
            'Created_at',
            'Updated_at', ];
}
```

View Php

```
<?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>';
   }
?>
```

Controller Php

```
<?Php
 Class UserController {
      Protected $userModel;
      Public function __construct() {
              $this -> userModel = new UserModel();
      }
      Public function index() {
          $users = $this -> userModel -> getAllUser();
?>
```