

Nama : Novia Eka Putri

NIM : 121140030

Resume Pratik Pemweb RA

Design Pattern

Design pattern adalah solusi untuk masalah desain perangkat lunak. Merupakan panduan/template untuk memecahkan masalah desain dalam pengembangan suatu perangkat lunak.

⇒ Jenis-jenis Design Pattern

1. Creational Pattern, adalah design pattern yang digunakan untuk membuat objek.
2. Structural Pattern, adalah design pattern yang digunakan untuk mengatur hubungan antar objek.
3. Behavioral Pattern, adalah design pattern yang digunakan untuk mengatur perilaku objek.

⇒ Keuntungan Menggunakan Design Pattern

1. Aplikasi menjadi lebih mudah dipelihara.
2. Aplikasi menjadi lebih mudah dikembangkan.
3. Aplikasi menjadi lebih fleksibel.
4. Aplikasi menjadi lebih terstruktur.
5. Aplikasi menjadi lebih mudah diuji.

Design Patterns

- Creational → Factory method → Builder → Abstract factory → Singleton
- Structural → Adapter → decorator → facade → Proxy
- Behavioral → Observer → Strategy → Command → State → Chain of responsibility → Visitor

Konsep MVC (Model-View-Controller) Design Pattern

MVC adalah pola design pattern yang membagi aplikasi menjadi tiga komponen utama. Setiap komponen memiliki tanggung jawab tertentu dalam sistem.

⇒ Komponen MVC

1. Model, mewakili struktur data dan logika bisnis. Model mengelola data dan menyediakan antarmuka untuk berinteraksi dengan data tersebut.
2. View, bertanggung jawab untuk menampilkan informasi kepada pengguna menjadi tampilan antarmuka/grafis.
3. Controller, menangani input pengguna, menginterpretasikan perintah, dan menginstruksikan model dan view untuk 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 Code Program:

1. Model PHP
2. View

```

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

```

<?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>';
}
?>
    
```

3. Controller

```

<?php
class UserController {
    protected $userModel;
    public function __construct() {
        $this->userModel = new UserModel();
    }
    public function index() {
        $users = $this->userModel->getAllUsers();
    }
}
    
```

⇒