

Nama : Putri Naphtali Manurung
NIM : 121140027
Kelas : RA

Design Pattern

Merupakan sebuah metode yang digunakan dalam rekayasa perangkat lunak untuk menyelesaikan masalah-masalah yang sering muncul. Pola design merupakan solusi umum yang dapat diterapkan kembali dalam berbagai konteks pengembangan perangkat lunak.

Ada 3 pola utama dalam design pattern :

- Creational Pattern
- Behavioral Pattern
- Structural Pattern

1. Creational Pattern

Creational Pattern berkaitan dengan proses pembuatan objek atau instance dari suatu kelas atau tipe data. Tujuan utama dari creational pattern adalah untuk menyembunyikan kompleksitas dari pembuatan objek dan membuatnya lebih fleksibel.

Contoh creational Pattern : Singleton Pattern, Factory Method Pattern, Abstrak Factory Pattern, dll.

2. Behavioral Pattern

Behavioral pattern adalah pola perilaku, satu jenis design patterns dalam paradigma pemrograman berorientasi objek atau kelas berinteraksi dan bekerjasama.

Tujuan behavioral pattern adalah memfasilitasi komunikasi yang baik antar objek, mengelola tanggungjawab dan meningkatkan fleksibilitas dalam perubahan perilaku sistem.

Contoh behavioral pattern : observer pattern, strategy pattern, command pattern.

3. Structural Pattern

Structural Pattern adalah salah satu kategori design pattern dalam pemrograman berorientasi objek yang berkaitan dengan cara kelas dan objek dibentuk agar membentuk struktur yang lebih besar. Tujuannya adalah untuk mempermudah komposisi objek dan kelas ke dalam struktur yang lebih besar serta memastikan bahwa perubahan dalam suatu bagian dari sistem tidak mempengaruhi seluruh sistem.

Contoh Structural Pattern : Adapter Pattern, Composite Pattern, Bridge Pattern.

Design Pattern memiliki beberapa kelebihan, antara lain:

- memberi solusi atas masalah yang timbul saat pengembangan perangkat lunak
- membuat penulisan kode menjadi lebih rapih, terstruktur dan mudah dibaca.
- mempercepat proses pengembangan perangkat lunak dengan memungkinkan penggunaan kembali pada design.

Kekurangan design pattern adalah kompleksitas yang tinggi dan membutuhkan usaha yang lebih dalam pengembangan perangkat lunak.

Konsep MVC (Model - View - Controller) Design Pattern

Suatu pola desain yang digunakan untuk memisahkan logika aplikasi menjadi tiga komponen utama: Model, View dan controller yang bertujuan untuk meningkatkan modularitas, fleksibel, dan pemeliharaan kode dalam perangkat lunak.

Model → bagian yang mengelola dan berhubungan langsung dengan database.

View → bagian yang akan menyajikan tampilan informasi pengguna.

Controller → bagian yang menghubungkan model dan view dalam setiap proses request dari user.

Contoh kode Model:

```
class User
```

```
{
```

```
    protected $table = 'user';
```

```
    protected $fillable = [
```

```
        'name'
```

```
        'email'
```

```
        'password'
```

```
        'created_at'
```

```
        'update_at'
```

```
    ];
```

```
}
```

Penjelasan kode

- Terdapat tiga properti: id, name, dan email
- Properti id: untuk menyimpan id pengguna
- Properti name: untuk menyimpan nama pengguna.
- Properti email: untuk menyimpan alamat email pengguna.

Contoh 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 kode :

- digunakan untuk mendapatkan data dari model.
- echo '<h1>Daftar User </h1>';
- digunakan untuk menampilkan judul halaman

Contoh Controller

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

Penjelasan kode :

- Mendefinisikan properti \$userModel
- Menyimpan objek dari kelas UserModel
- digunakan untuk berinteraksi dengan model pengguna.