

Laporan Praktikum 8 Pemrograman Web

Disusun oleh:

Sindikia Al Ulya Nurussyifa | 121140095

Kelas RA



**Program Studi Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sumatera
Lampung Selatan
2023**

1. Design Pattern

Desain pattern, dikenal sebagai pendekatan yang digunakan untuk menangani permasalahan yang sering muncul dalam pengembangan aplikasi, memberikan solusi terhadap masalah-masalah yang berulang. Tantangan yang muncul dalam pembuatan aplikasi seringkali memiliki pola yang serupa, dan dari situlah muncul desain pattern sebagai kerangka dasar atau template yang dapat diterapkan untuk menyelesaikan masalah tersebut. Meskipun masalahnya bervariasi dan muncul dalam konteks yang berbeda, desain pola menyediakan metode dasar atau templat yang dapat digunakan sebagai landasan untuk menyelesaikan permasalahan tersebut.

a. Manfaat menggunakan design pattern

- Menyediakan solusi untuk permasalahan yang muncul selama proses pengembangan perangkat lunak
- Penulisan kode menjadi lebih teratur, terstruktur, dan lebih mudah dipahami.
- Meningkatkan efisiensi komunikasi di antara anggota tim pengembang aplikasi
- Aplikasi menjadi lebih mudah diuji
-

b. Jenis-jenis design pattern

Masalah yang muncul sangatlah beragam, keberagaman tersebut dapat dipelajari dan diorganisir ke dalam pola yang serupa. Dengan demikian, penyelesaian masalah dalam kategori tertentu dapat dilakukan dengan cepat dan efisien. Berikut tiga pola utama yang dapat digunakan :

- Creational Pattern (Pola Pembuatan)

Desain pola ini merupakan solusi untuk menangani permasalahan yang muncul ketika menciptakan suatu instance atau objek. Sebagai contoh, penggunaan pola kreatif ini terlihat ketika seorang pengembang ingin membuat sebuah kelas untuk database yang dapat diakses dari berbagai lokasi. Pola ini terbagi menjadi 5 macam :

- Abstract Factory
- Builder
- Factory Method

- Prototype
- Singleton
- Behavioral Pattern (Pola Perilaku)

Desain pola ini digunakan untuk menangani permasalahan yang muncul dalam komunikasi antar objek. Pola ini terbagi menjadi 11 macam :

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor
- Structural Pattern (Pola Struktural)

Pola ini digunakan untuk menyelesaikan permasalahan yang ditemukan dalam pengaturan komposisi dari class dan object. Pola ini terbagi menjadi 7 macam :

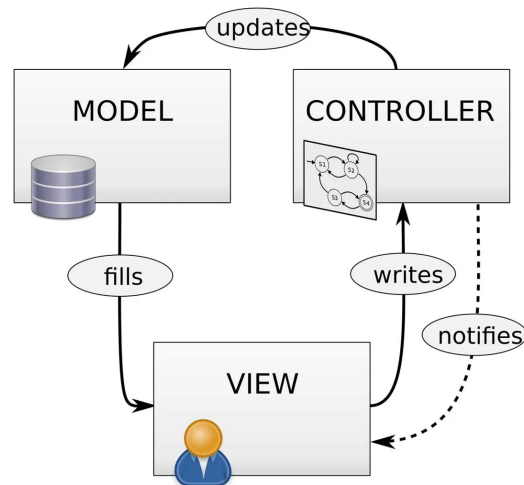
- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

2. Konsep MVC (Model-View-Controller) Design Pattern

Model View Controller (MVC) adalah sebuah pola desain perangkat lunak yang saat ini umum digunakan dalam pengembangan aplikasi dan situs web. Pola

desain ini secara luas diterapkan pada kerangka kerja PHP terkenal di dunia seperti Laravel, Symfony, Code Igniter, dan Yii. MVC memisahkan pengembangan program menjadi tiga komponen terpisah yang saling terhubung, sesuai dengan namanya: Model, View, dan Controller. Berikut penjelasan mengenai 3 komponen tersebut :

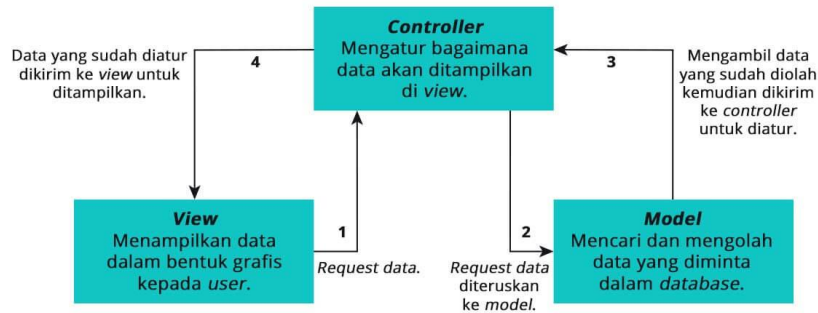
- Model, bagian yang bertanggung jawab untuk menyiapkan, mengelola, memanipulasi, dan mengorganisir data yang tersimpan dalam basis data.
- View, Bagian yang bertanggung jawab untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI)
- Controller, bagian yang bertanggung jawab untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung satu sama lain.



Gambar 1.1 Diagram Interaksi antar Komponen MVC

a. Alur Model View Controller

Langkah awal melibatkan tindakan view yang mengajukan permintaan data grafis kepada pengguna. Kontrol menerima permintaan ini dan mengalirkannya ke model untuk diproses. Model berperan dalam mencari dan memproses data yang diminta di dalam basis data. Setelah data berhasil diambil dan diolah, model mengirimkannya kembali kepada kontrol untuk ditampilkan di dalam view. Kontrol mengambil hasil pengolahan data dari model dan menyusunnya di dalam view untuk kemudian ditampilkan kepada pengguna.



Gambar 1.2 Alur Model View Controller

b. Manfaat menerapkan MVC dalam pembuatan aplikasi

- Penggunaan MVC dapat meningkatkan efisiensi dalam pengembangan aplikasi karena memungkinkan beberapa pengembang untuk bekerja pada kode tersebut. Sebagai contoh, pada pengembangan aplikasi web, tugas-tugas terkait model dan controller dapat diambil oleh pengembang back-end, sementara pengembang front-end dapat fokus pada bagian view. Hal ini dapat mempercepat proses pengembangan secara keseluruhan.
- Dengan pembagian menjadi tiga bagian, penulisan kode akan menjadi lebih teratur, menciptakan kerapian, dan memudahkan pengembang lain dalam mengembangkan kode tersebut.
- Untuk memastikan bahwa aplikasi berfungsi sesuai dengan rencana, pengujian atau testing menjadi langkah yang tidak dapat dihindari. Dengan menerapkan pola model view controller ini, proses pengujian dapat dilakukan pada setiap komponen aplikasi.
- Konsep MVC memudahkan tugas pemeliharaan aplikasi karena skrip atau kode yang terorganisir dan terstruktur dengan baik, menyediakan kemudahan bagi pengembang dalam menjalankan proses pemeliharaan aplikasi.