

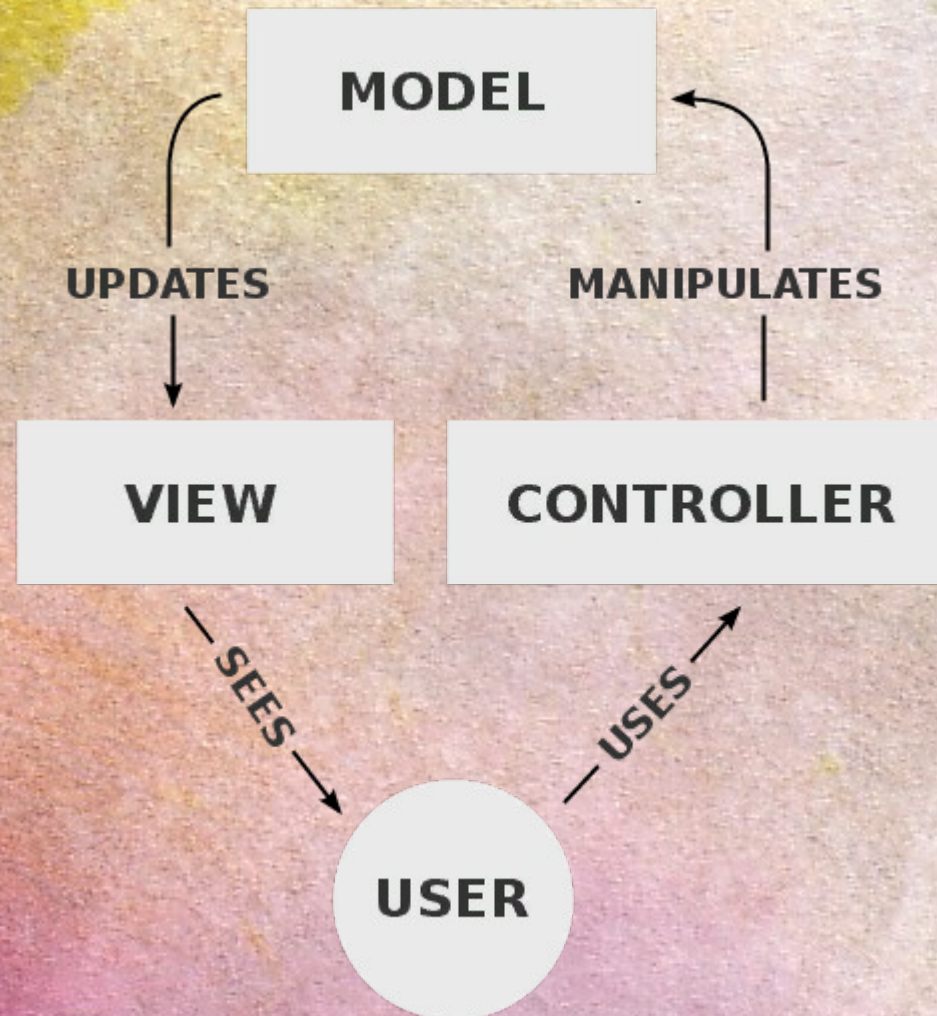
Pemrograman Framework Java

Pertemuan 5 - Aplikasi MVC

Konsep MVC

- Terdiri dari Model – View – Controller
- Sebuah teknik pengembangan antarmuka dari aplikasi yang terdiri dari 3 bagian terhubung
- Teknik ini memisahkan komponen-komponen utama yang mengizinkan penggunaan ulang kode secara efektif

Ilustrasi



Contoh MVC

Model = HTML

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished
this page.

Download the sample [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past re

Today, we must clear the mind of past practice
W3C, WaSP and the major browser creators

The css Zen Garden invites you to relax and re

View = CSS

```
body {  
    font: 12px/16px arial, helveti  
    color: #555;  
    background: url(bg_left.gif) r  
    margin: 0;  
    padding: 0;  
}  
  
a {  
    text-decoration: none;  
    font-weight: bold;  
    color: #655;  
}  
  
a:hover {  
    text-decoration: none;  
    font-weight: bold;  
    color: #e60;  
}
```

Controller = Browser



Komponen MVC

- **Model:** Komponen pusat dari teknik ini. Mengandung struktur data dinamis dari aplikasi, mandiri dari UI
- **View:** Representasi dari informasi yang berupa chart, diagram, atau tabel
- **Controller:** Menerima input dan mengubahnya menjadi perintah untuk model atau view

Lanjutan

- **Teknik ini biasanya dipakai oleh pengembang untuk membuat aplikasi desktop**
- **Namun akhirnya populer juga ke pembuatan aplikasi web**
- **Contoh Java C# Python Ruby PHP**

MVC Aplikasi Web

- **MVC sekarang sudah diadopsi secara luas untuk aplikasi World Wide Web**
- **Beberapa framework web juga menggunakan cara ini**
- **Interpretasi MVC bisa berbeda, utamanya tugas MVC dibagi di antara client dan server**

Tujuan MVC

- **Pengembangan secara bersamaan**
 - Di karena ada pemecahan komponen aplikasi, pengembang dapat mengerjakan secara terpisah
- **Penggunaan ulang kode**
 - Dengan membuat komponen mandiri, pengembang dapat menggunakan ulang komponen secara cepat

Kelebihan

- **Pengembangan Simultan**
- **Kohesi Tinggi**
- **Low Coupling**
- **Kemudahan Modifikasi**
- **Banyak View untuk Model**

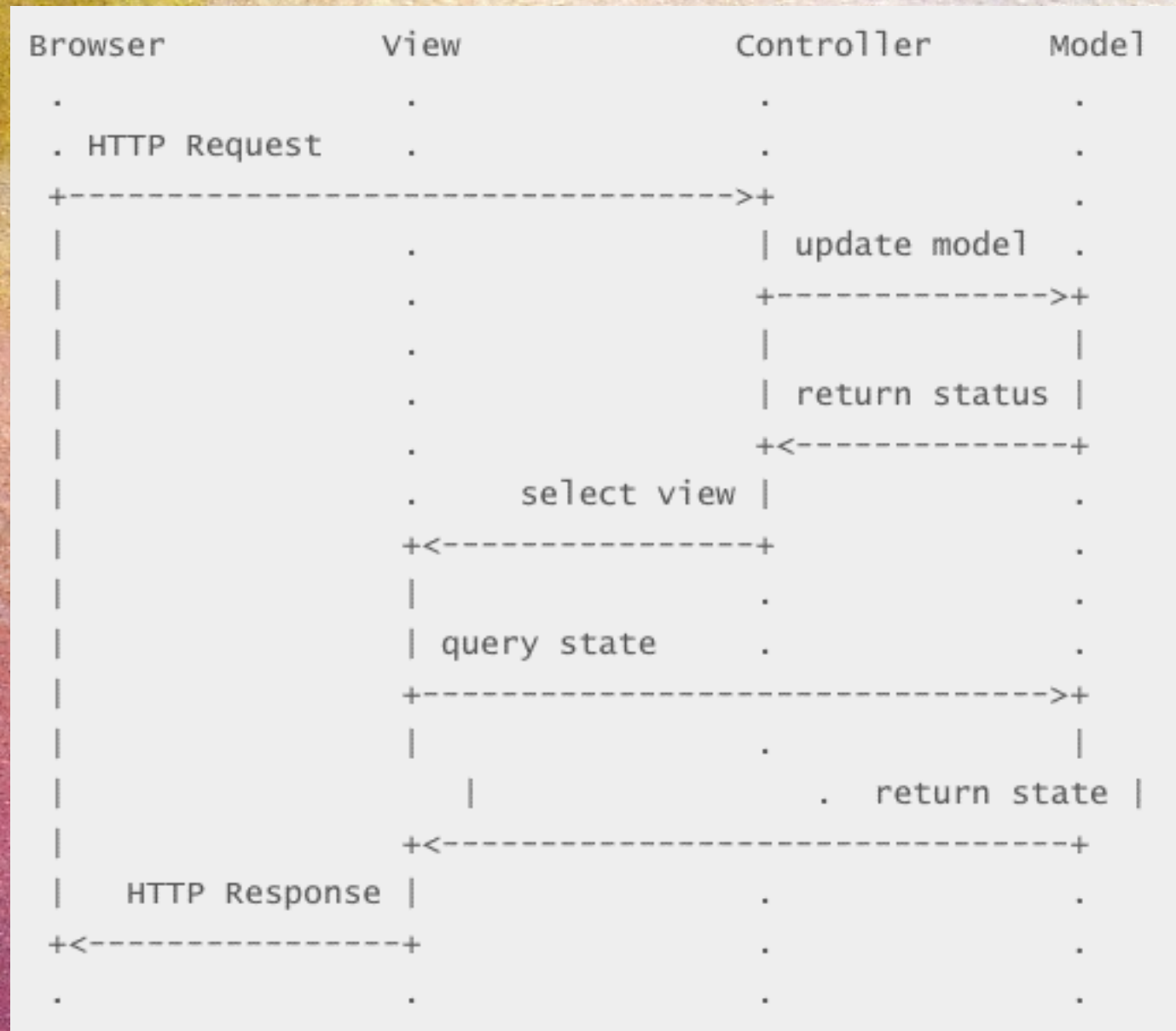
Kekurangan

- **Navigasi kode yang kompleks**
- **Konsistensi artifak yang tidak teratur**
- **Kesulitan dalam pembelajaran**

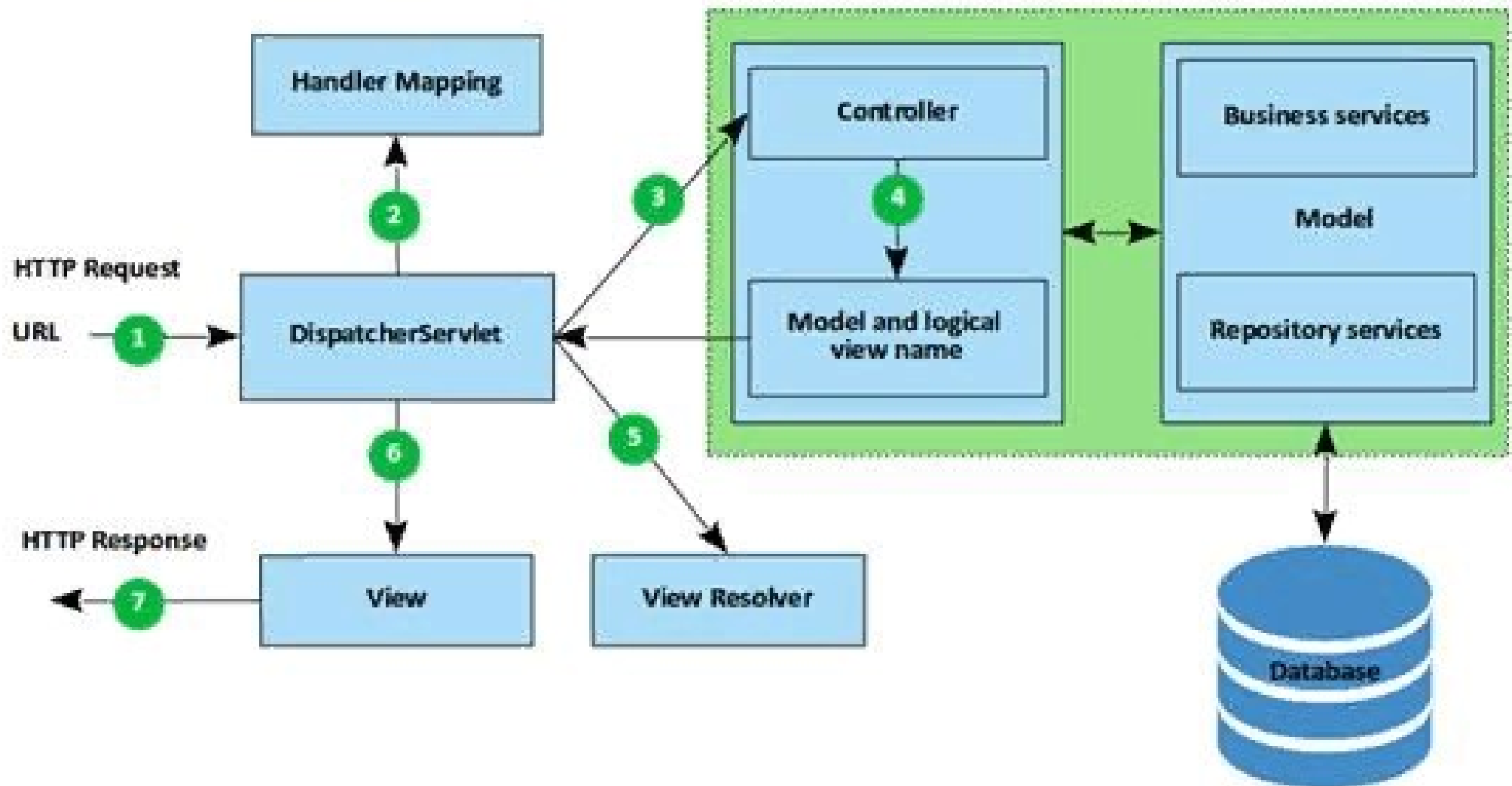
Aplikasi Web dan MVC

- **Dikarenakan MVC diadopsi untuk pemrograman web, maka ada adaptasi yang harus dilakukan oleh pengembang untuk menerapkannya**

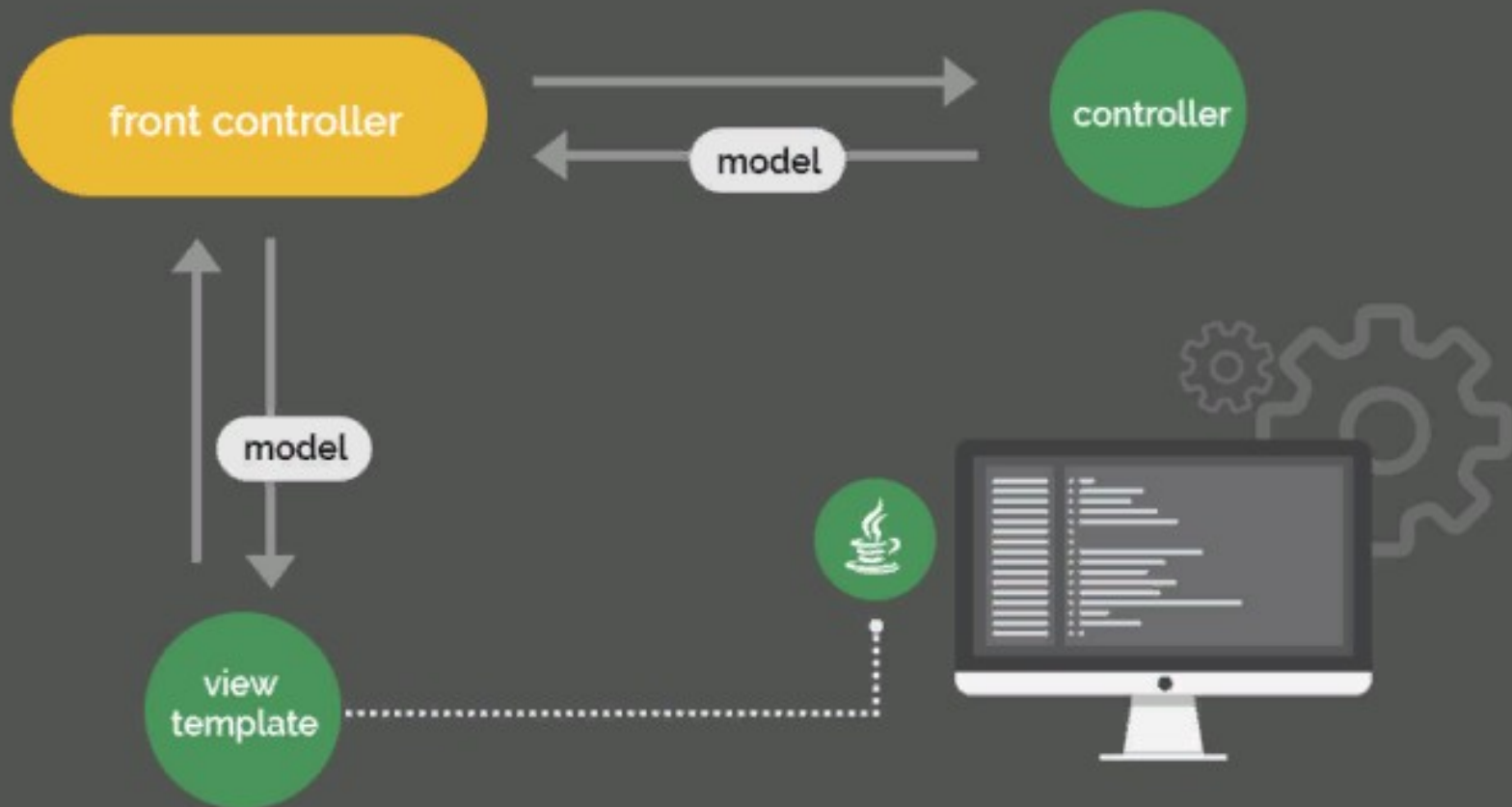
Contoh – Request HTTP



Ilustrasi



Ilustrasi



Apa Yang Berguna?

- **Pemisahan Permintaan (Request) dan Halaman (Pages)**
 - **Dikarenakan Controller memiliki tugas untuk handle permintaan dan memilih View yang tepat, maka tidak ada coupling diantara request dari user dengan hasil halaman**

Lanjutan

- **View itu simpel**
 - Sejak semua kode melakukan sesuatu hal kecuali membuat halaman bagus untuk user di luar dari objek View
 - Mengganti layout tidak perlu menyentuh logika aplikasi

Lanjutan

- **Melindungi Implementasi Model**
 - **Dikarenakan semua aksi dari status aplikasi itu dihandle oleh Model, sangat memungkinkan untuk mengganti implementasi model tanpa menyentuh UI.**
 - **Selama API Public Model tidak berubah**

Masalah dan Pembatasan

- **Apa Pergi Ke Mana**
 - **Terkadang sangat sulit untuk mencari tahu dimana satu potong dari aplikasi seharusnya pergi**
 - **Khususnya membagi Model dari Controller bisa menjadi sangat sulit, sesuai dengan aturan Ibu Jari Controller seharusnya seminimal mungkin**

Lanjutan

- **Coupling Antara View dan model**
 - **Satu masalah ketika View dan Controller sedang meminta Model mengubah API Public Model adalah View dan Controller juga harus adaptasi**
 - **Mengubah Controller tidak sesulit mengubah View**

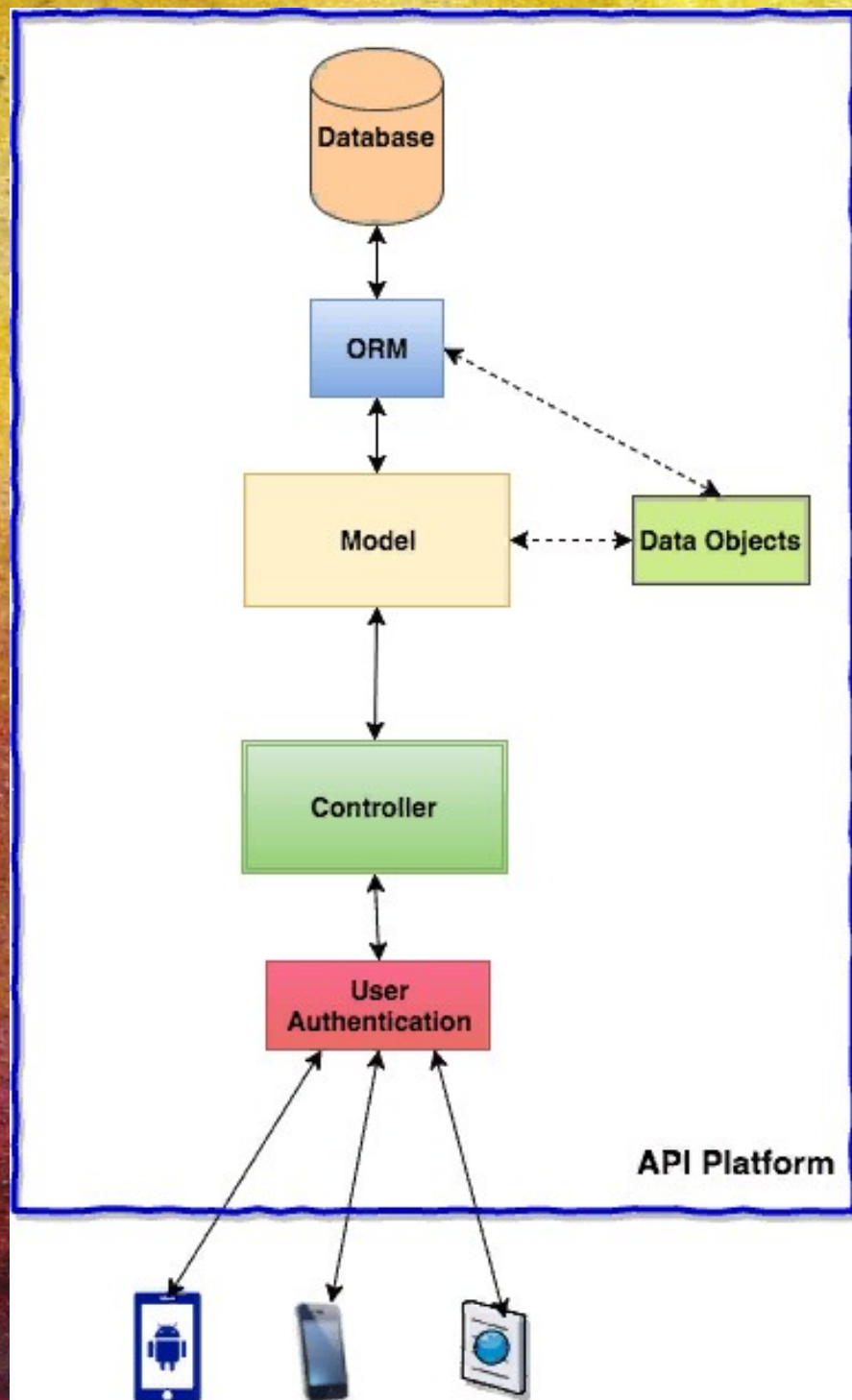
Lanjutan

- **Banyak Objek**
 - **Ketika membuat aplikasi MVC bisa menghasilkan banyak sekali Class dan Objek daripada cara tradisional**

Integrasi MVC dengan API

- **Spring MVC mendukung banyak sekali jenis API**
- **Seperti: RESTful, Backend atau API platform**

Ilustrasi



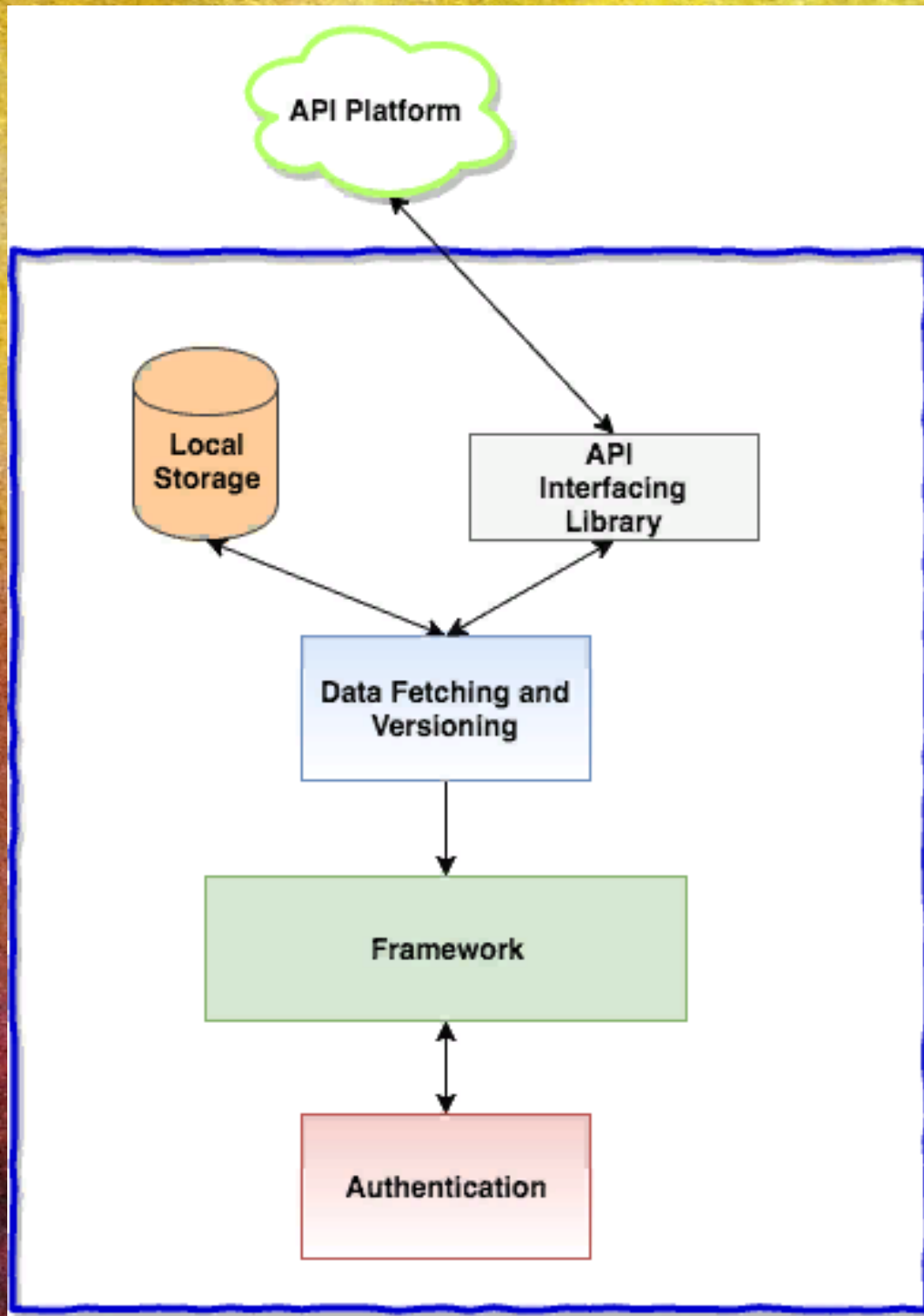
Aplikasi Web

- **Terdapat beberapa opsi:**
 - **API + DB + End to End Framework + ORM + Front End Technologies**
 - **API + DB + Teknologi Sisi Server + ORM + Front End Technologies**
 - **DB + ORM + End to End Framework + Front End Technologies**

Aplikasi Mobile

- **Terdapat opsi:**
 - **API + DB + End to End Framework + Hybrid/Native Mobile Technology**
 - **API + DB + Teknologi Sisi Server + Hybrid/Native Mobile Technology**

Ilustrasi



Spring API

