



# Pemahaman

# Spring Boot

Muhammad Daffa Raihandika - 211524050

3B D4 - Teknik Informatika

# Content

- Spring Boot
- Spring Bean
- Row Mapper
- REST API
- MVC
- JPA
- Hibernate



# Spring Boot

- Spring Boot adalah salah satu jenis framework dari Spring
- Java Spring Boot adalah alat yang merupakan bagian dari Spring Framework, dirancang untuk mempercepat proses pembuatan aplikasi dengan menyederhanakan konfigurasi.
- Spring Boot menyediakan konfigurasi standar secara otomatis, menghilangkan kebutuhan untuk mengkonfigurasi setiap aspek aplikasi secara manual.
- Dengan Spring Boot, pengembang dapat membuat aplikasi berbasis Java dengan cepat dan mudah
- Kemampuan Spring Boot dalam membuat web server dengan cepat dan mudah menjadikannya pilihan yang populer di kalangan pengembang untuk pengembangan aplikasi web berbasis Java



# Spring Bean

- Spring bean adalah objek yang dikelola oleh Spring Framework. Ini berarti Spring Framework bertanggung jawab untuk membuat, mengonfigurasi, dan menghancurkan bean
- Spring beans biasanya digunakan untuk mengimplementasikan logika bisnis dan menyediakan layanan ke komponen lain dalam suatu aplikasi.
- Untuk menggunakan spring beans dapat mencantumkan annotations pada kode yang ingin diberi beans



# Spring Beans Annotation

## **@Service**

Digunakan untuk menandai kelas sebagai layanan (service) dalam arsitektur aplikasi. Ini berguna untuk menunjukkan bahwa kelas tersebut menyediakan bisnis logic atau fungsi layanan tertentu dalam aplikasi.

## **@Repository**

Digunakan untuk menandai kelas sebagai komponen yang bertanggung jawab atas akses data, seperti pengambilan dan penyimpanan data ke dalam basis data.

## **@Autowired**

Digunakan untuk melakukan injeksi ketergantungan (dependency injection) secara otomatis ke dalam bean yang sesuai

# Spring Beans Annotation

## **@RestController**

Digunakan khusus untuk membuat endpoint REST API. Setiap metode akan secara otomatis mengembalikan data dalam format yang sesuai dengan jenis konten yang diminta oleh klien (misalnya JSON atau XML).

## **@RequestMapping:**

Digunakan untuk memetakan permintaan HTTP ke metode tertentu dalam kelas yang dianotasi. Anotasi ini dapat dikonfigurasi untuk menentukan jalur URL dan metode HTTP mana yang akan ditangani oleh metode yang dianotasi.

**@GetMapping,  
@PostMapping,  
@PutMapping,  
@DeleteMapping**

Anotasi-anotasi ini digunakan sebagai singkatan untuk @RequestMapping dengan metode HTTP tertentu

# Row Mapper

- RowMapper adalah sebuah antarmuka (interface) dalam Spring Framework yang digunakan untuk memetakan hasil setiap baris dari hasil kueri database menjadi objek Java. Ini sering digunakan untuk mengambil data dari database relasional dan mengonversinya menjadi objek yang dapat digunakan dalam aplikasi Java
- Ketika menggunakan Spring JDBC, RowMapper ini akan digunakan secara otomatis oleh JdbcTemplate untuk memetakan hasil kueri menjadi objek Java saat menggunakan metode seperti query() untuk mengambil data dari database



# Row Mapper

## Kelas Menggunakan RowMapper

```
package com.blog.mapper;

import java.sql.ResultSet;
import java.sql.SQLException;
import com.blog.vo.Post;
import org.springframework.jdbc.core.RowMapper;

public class PostMapper implements RowMapper<Post>{

    @Override
    public Post mapRow(ResultSet rs, int rowNum) throws SQLException{
        Post post = new Post();
        post.setId(rs.getLong(columnLabel:"id"));
        post.setUser(rs.getString(columnLabel:"user"));
        post.setTitle(rs.getString(columnLabel:"title"));
        post.setContent(rs.getString(columnLabel:"content"));
        post.setRegDate(rs.getDate(columnLabel:"reg_date"));
        post.setUpdtDate(rs.getDate(columnLabel:"updtt_date"));

        return post;
    }
}
```

## Contoh Penggunaan Kelas RowMapper

```
public Post findOne(Long id) {
    String sql = "SELECT * FROM post WHERE id = ?";

    RowMapper<Post> rowMapper = new PostMapper();

    return this.jdbcTemplate.queryForObject(sql, rowMapper, id);
}
```

# REST API

- REST API adalah jenis arsitektur perangkat lunak yang didasarkan pada konsep Representational State Transfer (REST).
- REST API memungkinkan aplikasi untuk berkomunikasi satu sama lain melalui internet dengan menggunakan protokol HTTP dengan cara yang sederhana dan mudah dipahami.
- REST API menggunakan protokol HTTP untuk berkomunikasi antara klien dan server. Metode HTTP seperti GET, POST, PUT, DELETE digunakan untuk berinteraksi dengan resources yang disediakan oleh server.



# MVC (Model, View, Controller)

## Model

Merupakan representasi dari data atau informasi yang digunakan dalam aplikasi. Ini dapat berupa objek yang disesuaikan dengan entitas pada database. Model bertanggung jawab untuk mengelola data dan berinteraksi dengan database.

## View

Merupakan bagian yang bertanggung jawab untuk menampilkan informasi kepada pengguna. Biasanya, view berupa halaman web, antarmuka pengguna, atau tampilan lainnya yang berinteraksi langsung dengan pengguna.

## Controller

Bertindak sebagai perantara antara model dan view yang mengendalikan alur logika aplikasi. Controller menerima input melalui view, memprosesnya menggunakan model, dan kemudian mengirimkan hasilnya kembali ke view untuk ditampilkan kepada pengguna.



# Konsep MVC Pada Spring Boot

## Model

Model dalam Spring MVC biasanya merujuk pada kelas Java yang mewakili data yang akan ditampilkan atau diproses dalam aplikasi. Ini dapat berupa Value Object (VO), yang digunakan untuk mengemas dan mengelola data aplikasi

## View

View dalam Spring MVC adalah template atau halaman web yang digunakan untuk menampilkan informasi kepada pengguna. Dalam Spring Boot, view sering kali didefinisikan menggunakan teknologi template seperti FreeMarker

## Controller

Controller dalam Spring MVC adalah kelas Java yang menangani permintaan HTTP dari klien. Controller juga bertanggung jawab untuk menyiapkan data yang diperlukan untuk ditampilkan di view. Dianotasi dengan @Controller atau @RestController



# JPA

- JPA adalah singkatan dari Java Persistence API. Ini adalah spesifikasi Java yang menyediakan kerangka kerja untuk (ORM) dalam aplikasi Java.
- Tujuannya adalah untuk menyederhanakan dan memudahkan pengembangan aplikasi yang berinteraksi dengan database relasional menggunakan objek-objek Java.
- Dengan menggunakan JPA dapat menterjemahkan nama method menjadi query SQL
- Contohnya:

<b>Method Name</b>	List<Post> findAllOrderByUpdtDateAsc();
<b>Query SQL</b>	SELECT * FROM post ORDER BY upd_date ASC



# JPA Syntax for SQL

And	findByUserAndTitle	... where user = ?1 and title = ?2
Or	findByUserOrTitle	... where user = ?1 or title = ?2
Between	findByRegDateBetween	... where regDate between ?1 and ?2
Like	findByTitleLike	... where title like ?1
Containing	findByTitleContaining	... where Title like ?1
OrderBy	findByUserOrderByRegDateDesc	... where user = ?1 order by regDate desc



# Hibernate

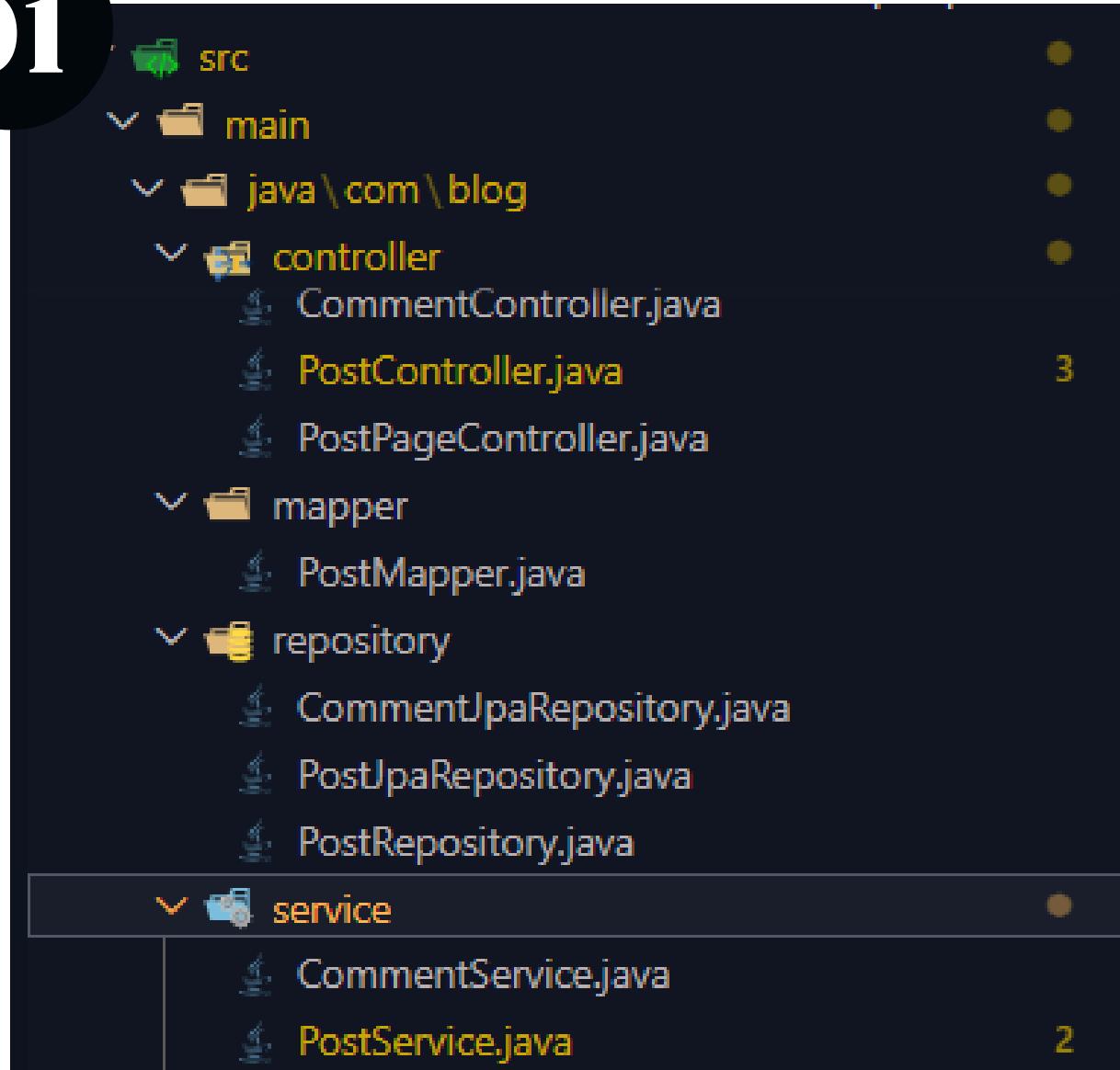
- Salah satu framework ORM (Object-Relational Mapping) yang paling populer untuk bahasa pemrograman Java.
- Hibernate memungkinkan pengembang untuk melakukan pemetaan objek-ke-relasional (ORM) dengan mudah antara struktur data dalam aplikasi Java dengan tabel-tabel dalam database relasional. Menggunakan annotation @Entity pada model

```
@Entity  
@Table(name = "post")  
public class Post {
```

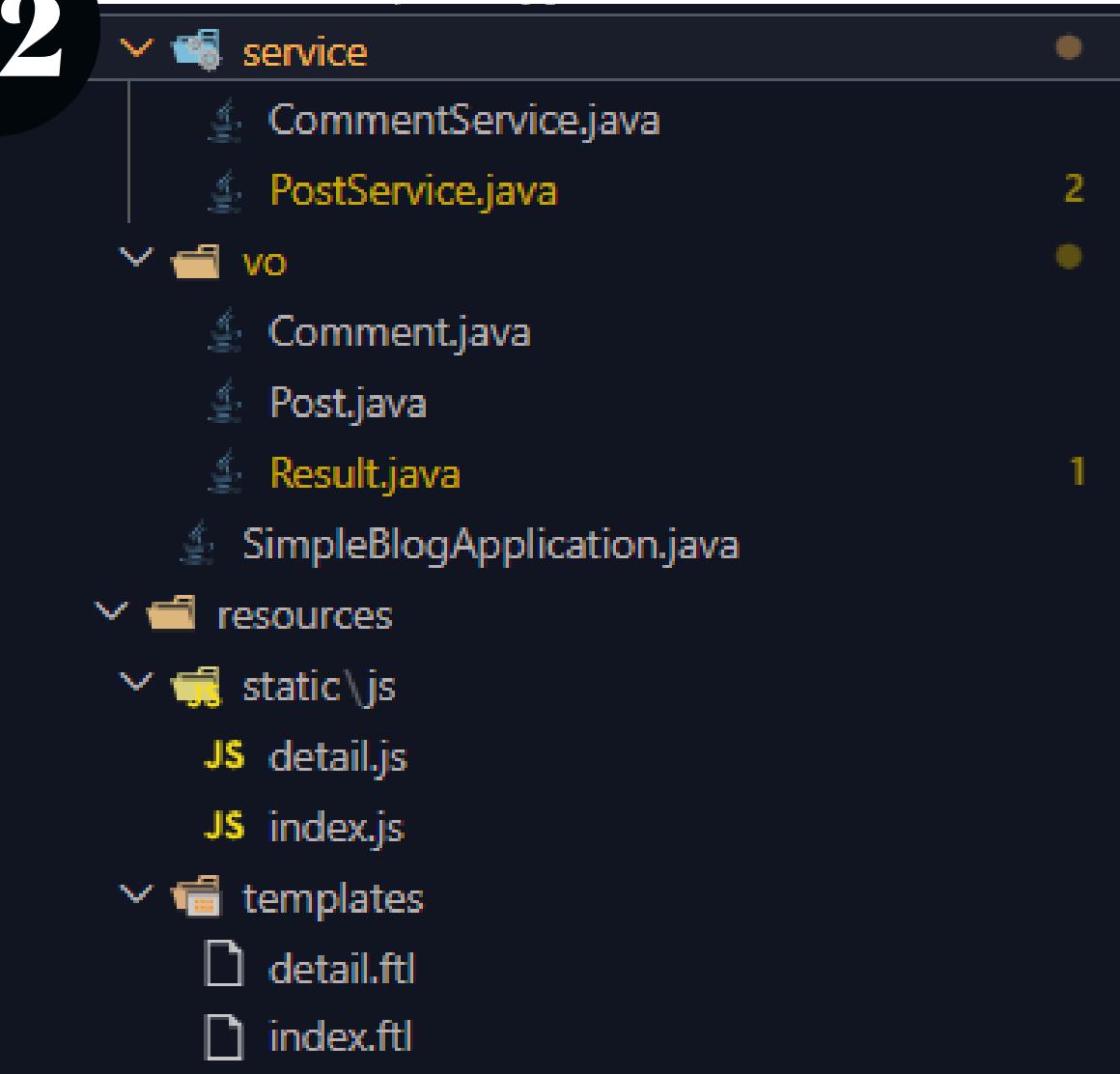


# Struktur Folder

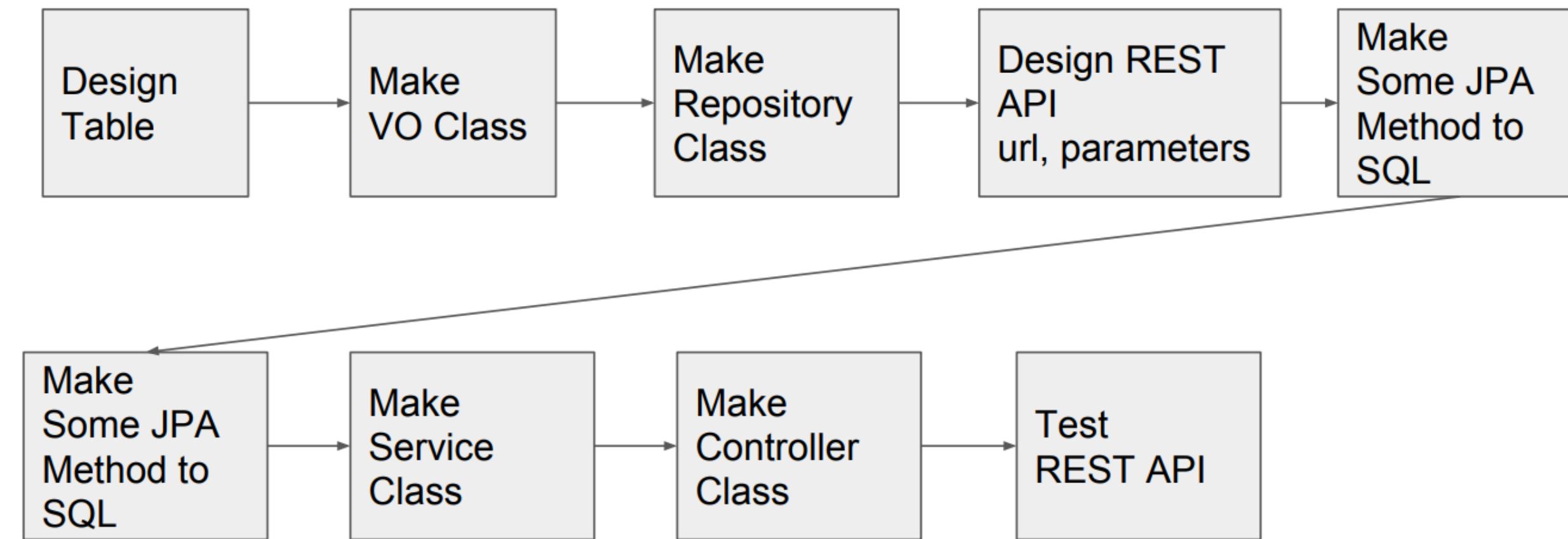
01



02



# Proses Membuat REST API Pada Spring Boot



Thank's For  
Watching

