

2024

PBW

Modul 5 – Java Spring Boot



Modul Java Spring Boot

Java Spring Boot digunakan untuk membangun aplikasi web pada sisi server (backend). Pada modul ini kita akan mencoba membuat proyek dasar menggunakan Java Spring Boot. Pada kuliah kita, spesifikasi proyek Java Spring Boot adalah sebagai berikut:

- Spring Boot 3.3.5
- Gradle - Kotlin
- Java (minimal 17)

Untuk membuat proyek Java Spring Boot pada workspace anda, dapat dilakukan dalam beberapa alternatif.

Gradle init

Pertama-tama pastikan gradle sudah terinstall di komputer anda, dan sudah dapat diakses dari command line. (Lihat petunjuk install pada <https://gradle.org/install/>) Kemudian ikutilah instruksi berikut:

1. Pada command line lakukan gradle init seperti gambar berikut:

```
C:\Users\Ray\Downloads\pbw1\project>gradle init
Starting a Gradle Daemon, 3 busy and 3 incompatible Daemons could not be reused, use --status for details

Select type of build to generate:
 1: Application
 2: Library
 3: Gradle plugin
 4: Basic (build structure only)
Enter selection (default: Application) [1..4] 1

Select implementation language:
```

Beberapa pilihan selanjutnya adalah:

- Type of build: 1
 - Implementation language: 1
 - Java version: 17
 - Project name: (menyesuaikan kebutuhan)
 - Application structure: 1
 - Build Script: 1
 - Test Framework: 1
 - Generate build using new APIs: no
2. Tunggu sampai proyek selesai dibangun dan terbuat beberapa file pada folder tersebut. Setelah itu, masuk ke file app/build.gradle.kts. Pada bagian plugins, kita masukkan versi dari framework yang dibutuhkan:

```

8  plugins {
9      // Apply the application plugin to add support for building a CLI app
10     application
11     id("org.springframework.boot") version "3.3.4"
12     id("io.spring.dependency-management") version "1.1.6"
13 }

```

3. Lalu pada bagian dependencies tambahkan package spring boot seperti gambar berikut:

```

20 dependencies {
21     implementation("org.springframework.boot:spring-boot-starter-web")
22 }

```

4. Setelah itu pada command line di folder utama (terdapat file gradlew.bat), masukkan command `./gradlew build`. Tunggu sampai build successful.
5. Setelah itu cari file App.java di dalam folder main. Untuk menggunakannya pada proyek spring boot, kita akan hapus semua kode ini dan diubah dengan kode berikut:

```

1  package org.example;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6
7  @SpringBootApplication
8  public class App {
9      public static void main(String[] args) {
10         SpringApplication.run(App.class, args);
11     }
12 }

```

6. Lalu hapus kode dengan nama AppTest.java di folder main. Kita tidak menggunakan file tersebut pada tugas ini.
7. Di folder yang sama dengan App.java, buatlah 1 file java dengan nama HelloController.java yang berisi berikut:

```

1  package org.example;
2
3  import org.springframework.stereotype.Controller;
4  import org.springframework.web.bind.annotation.GetMapping;
5  import org.springframework.web.bind.annotation.ResponseBody;
6
7  @Controller
8  public class HelloController {
9
10     @GetMapping("/hello")
11     @ResponseBody
12     public String hello() {
13         return "Greetings from Spring Boot!";
14     }
15 }

```

8. Kemudian pada command line, lakukan perintah `“./gradlew bootRun”`. Tunggu sampai server Tomcat menyala. Lalu pada browser masukkan alamat `localhost:8080/hello`. Perhatikan hal yang terjadi.

Spring Initializr

Kita dapat melakukan hal yang sama dengan menggunakan Spring Initializr dari web (<https://start.spring.io/>). Buatlah project dengan menggunakan spring initializr dan tambahkan dependencies Spring Web.

Lalu tekan tombol Generate dan letakkan proyek ini di sebelah folder yang dibuat pada bagian pertama.

- a. Perhatikan kedua proyek yang dibuat dengan gradle init dan dari spring initializr. Bandingkan dan tuliskan apa perbedaan dari kedua struktur folder tersebut.
- b. Lengkapilah agar di proyek ini dapat membuka halaman jika kita memanggil Alamat `localhost:8080/hello` seperti pada bagian pertama.

Build --continuous

Pada bagian ini dan seterusnya kita akan menggunakan proyek kedua yang dibuat dari spring initializr. Salah satu fungsi yang berguna dalam proses development adalah melakukan compile proyek secara otomatis, sehingga kita tidak perlu memberhentikan server dan build ulang setiap kali kita mengubah kode.

Untuk menjalankan fungsi ini, kita membutuhkan 2 terminal berbeda. Terminal pertama digunakan untuk membuat web server (bootRun) dan terminal kedua digunakan untuk build otomatis. Untuk menjalankannya masukan perintah `“./gradlew build –continuous”`.

- c. Untuk menguji fitur ini, ubahlah string pada HelloController lalu save. Seharusnya isi halaman dapat berubah tanpa kita perlu menutup server dan compile ulang.

Spring Bean

Pelajari penjelasan singkat dari spring bean dari alamat berikut <https://www.baeldung.com/spring-bean>

- d. Jelaskan dengan menggunakan pengertian sendiri, apa kegunaan dari Spring Bean ini!

Spring Bean merupakan faktor kunci yang menggerakkan framework Spring. Untuk melihat bean apa saja yang digunakan di proyek kita, kita dapat melihat dari `context.getBeanDefinitionNames()`. Untuk memudahkan dan tidak mengganggu code lain, kita coba untuk tunjukkan pada bean `CommandLineRunner` sebagai berikut (tambahkan di bawah method `public static void main()`):

```

@Bean
public CommandLineRunner commandLineRunner(ApplicationContext context) {
    return args -> {
        System.out.println("Let's inspect the beans provided by Spring Boot:");
        String[] beanNames = context.getBeanDefinitionNames();
        Arrays.sort(beanNames);
        for (String beanName : beanNames) {
            System.out.println(beanName);
        }
    };
}

```

Lengkapi dengan import package yang dibutuhkan.

- e. Dimana output dari kode ini ditampilkan, dan kapan kode ini dijalankan? Jelaskan.

Tugas

Beberapa hal yang harus dilakukan untuk penyelesaian modul ini adalah sebagai berikut:

1. Ubah port server menjadi port 3000 pada file application.properties.
2. Ubah konteks aplikasi menjadi "web" pada file application.properties.
 - f. Apa efek dari perubahan konteks aplikasi ini?
3. Buatlah dan tambahkan file index.html pada folder resource static yang berisi text hello world. Lalu coba akses dari sisi server. Bukan double klik file html.
4. Tambahkan routing hello2 yang menampilkan text dengan parameter tertentu. (Lihat implementasi pada slide)
 - g. Pelajari penggunaan annotation @ResponseBody. Jelaskan apa kegunaan anotasi tersebut, dan jelaskan pula apa yang terjadi jika kita tidak menggunakan anotasi tersebut.
5. Tambahkan routing untuk redirect ke halaman lain. (Lihat implementasi pada slide)

Kumpulkan dengan nama M05xxyyy.zip yang berisi folder project (dibuat dari gradle init) dan folder project2 (dibuat dari spring initializr) dan 1 buah txt yang berisi jawaban dari pertanyaan yang ditandai text biru. Silahkan diunggah di IDE kesayangan anda.

= SELAMAT MENGERJAKAN =