

<i>Session Date</i>	:	13 Desember 2019
<i>Semester</i>	:	Gasal
<i>Subject</i>	:	1132102 – Pemrograman Berorientasi Objek
<i>Week/Session</i>	:	3/3
<i>Key Topics</i>	:	<i>Creating Web Application using Spring Framework</i>
<i>Activity</i>	:	<i>Exercise</i>
<i>Duration</i>	:	100 minutes
<i>Objective</i>	:	Mahasiswa mampu membuat web aplikasi dengan memanfaatkan fitur: Spring Boot
<i>Lecturer</i>	:	TMP
<i>Instructor</i>	:	SEP

Prerequisites:

- JDK 8
- STS IDE
- Koneksi internet yang stabil

Spring Framework

Framework Spring merupakan *open source framework* berbasis bahasa pemrograman Java yang dapat digunakan untuk membangun aplikasi Java CUI (*Command User Interface*), GUI (Graphical User Interface) dan berbasis Web yang berjalan pada Browser. Spring framework termasuk *portable* karena aplikasi yang dikembangkan dapat berjalan pada JVM manapun. Untuk menggunakan Spring, *developer* dapat menggunakan Eclipse atau Netbeans. Spring Tool Suites merupakan *tool* pengembangan berbasis Eclipse yang disesuaikan untuk mengembangkan aplikasi Spring. Spring Tool Suite juga termasuk *tools* yang siap digunakan, *debug*, *run*, dan *deploy* aplikasi termasuk integrasi untuk Pivotal TC Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, dan lainnya. Berikut ini merupakan beberapa fitur yang terdapat pada Spring Framework, yaitu:

- *Core technologies*: dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP.
- *Testing*: mock objects, TestContext framework, Spring MVC Test, WebTestClient.
- *Data Access*: transactions, DAO support, JDBC, ORM, Marshalling XML.
- Spring MVC and Spring WebFlux web frameworks.
- *Integration*: remoting, JMS, JCA, JMX, email, tasks, scheduling, cache.

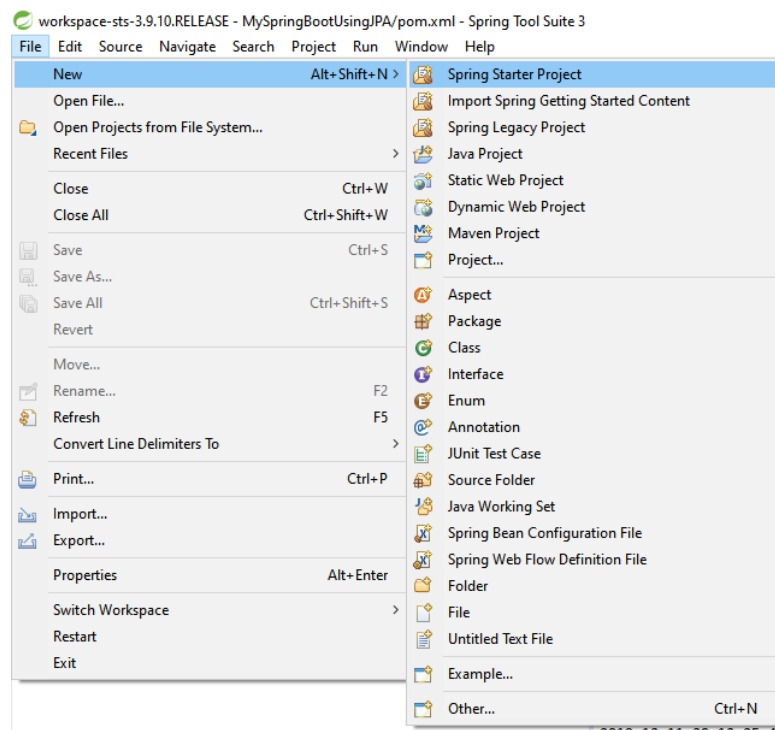
- *Languages*: Kotlin, Groovy, dynamic languages.

Spring Boot merupakan salah satu jenis *framework* Spring yang mempermudah pengguna dalam membuat program, karena:

1. Sudah disediakan Tomcat dan beberapa server lain, sehingga pengguna hanya perlu *run*
2. Menggunakan Maven sebagai *build manager* yang dapat di-setting melalui *Project Object Model* (POM)
3. Memiliki beberapa anotasi yang dapat digunakan untuk mempermudah kita dalam menentukan komponen kelas, dsb.

Langkah-langkah membuat proyek menggunakan Spring Boot:

1. Klik File > New > Spring Starter Project.



Gambar 1. Create new project

2. Kemudian Akan muncul *form* untuk konfigurasi proyek yang akan dikembangkan:

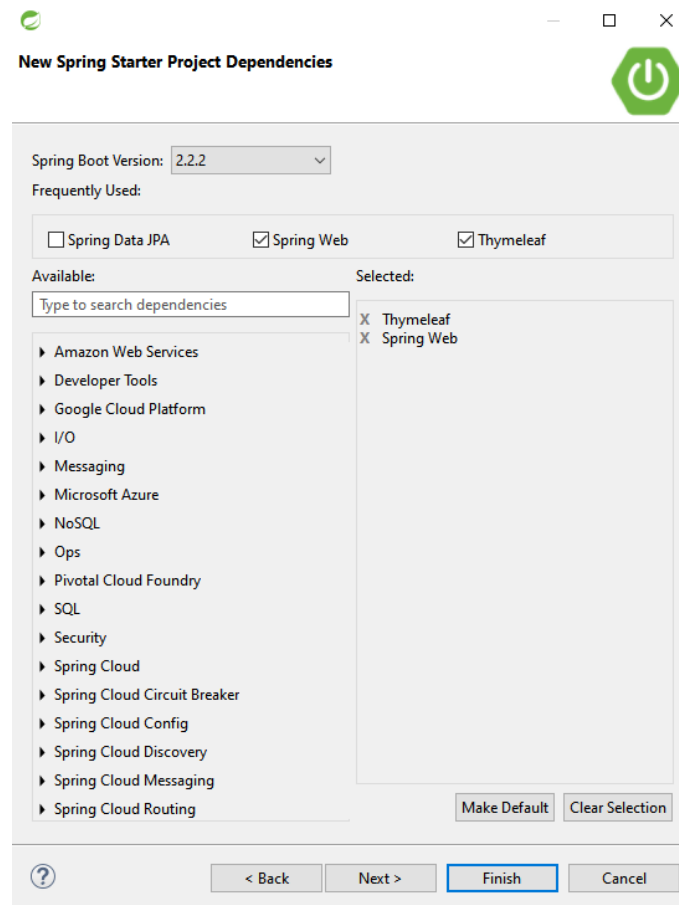
The image shows a 'New Spring Starter Project' dialog box. It has a title bar with a green power icon. The main area contains several input fields and checkboxes. The 'Service URL' is set to 'https://start.spring.io'. The 'Name' field contains 'MySpringBootExercise'. The 'Location' field shows a file path and a 'Browse' button. The 'Type' is 'Maven', 'Packaging' is 'Jar', 'Java Version' is '8', and 'Language' is 'Java'. The 'Group' is 'com.example', 'Artifact' is 'MySpringBootExercise', 'Version' is '0.0.1-SNAPSHOT', 'Description' is 'Demo project for Spring Boot', and 'Package' is 'com.example.demo'. There is a 'Working sets' section with a checkbox 'Add project to working sets' and buttons 'New...' and 'Select...'. At the bottom, there are buttons '< Back', 'Next >', 'Finish', and 'Cancel'.

Gambar 2. Form Konfigurasi Proyek

Konfigurasi proyek dengan rincian sebagai berikut:

- Group: nama package default Anda
- Artifact: nama proyek Anda
- Version: versi proyek Anda
- Packaging: hasil build proyek Anda
- Name: nama proyek Anda (default: artefact)
- Description: deskripsi proyek Anda
- Package name: nama package default Anda, bias disamakan dengan group id
- Language: Bahasa yang digunakan
- Java version: versi JDK yang Anda punya

3. Selanjutnya akan dilakukan konfigurasi terhadap *dependencies* yang akan digunakan pada proyek. Pada praktikum pengenalan Spring Boot ini akan digunakan: **Thymeleaf** sebagai *template* dari engine yang akan digunakan dan **Web** yang akan dikembangkan dalam proyek ini.



Gambar 3. *Project Dependencies*

4. Kemudian Anda dapat membuat *package* baru yang dapat digunakan untuk meletakkan *controller file* yang akan digunakan. Buatlah sebuah Java Class dengan nama HomeController.java pada package *controller* Anda.

```
package com.example.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
public class GreetingController {
    @GetMapping("/greeting")
    public String greeting
        (@RequestParam
         (name="name", required=false, defaultValue="World")
         String name,
         Model model)
    {
        model.addAttribute("name", name);
        return "greeting";
    }
}
```

Kode Program 1

5. Pada bagian src/main/resource/templates Anda dapat menambahkan *file* HTML dengan nama index.html (Kode Program 2) dan greeting.html (Kode Program 3) yang akan digunakan untuk menampilkan *output* sesuai dengan *mapping* yang sudah di *setting* pada *file controller* Anda.

```
<!DOCTYPE HTML>
<html>
<head>
    <title>Getting Started: Serving Web Content</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <p>Get your greeting <a href="/greeting">here</a></p>
</body>
</html>
```

Kode Program 2

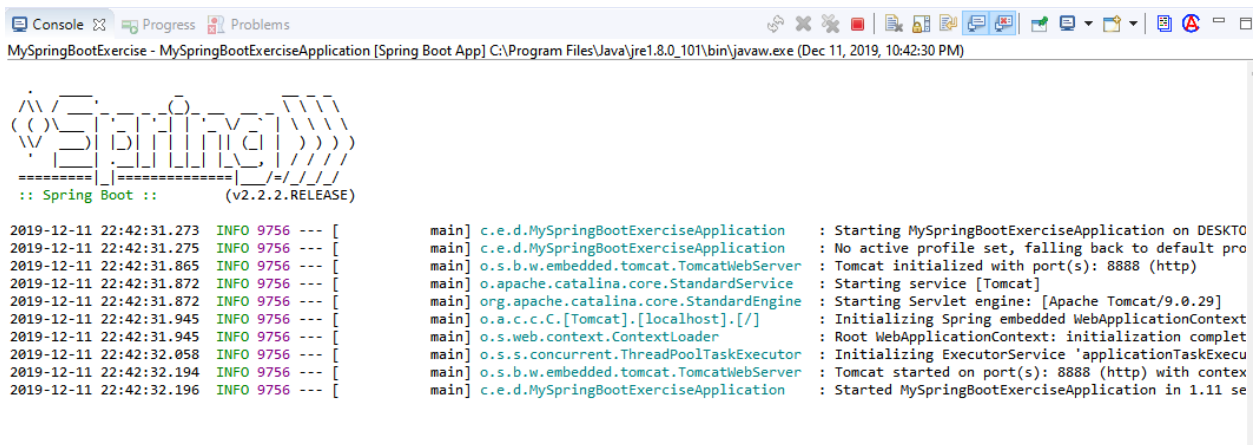
```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Getting Started: Serving Web Content</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<p th:text="'Hello, ' + ${name} + '!'" />
</body>
</html>

```

Kode Program 3

- Run proyek yang telah Anda buat. Jika proyek Anda berhasil, maka akan tampil gambar berikut pada layar *console* Anda.



Gambar 4. Output in console

- Jalankan pada *browser* Anda dengan URL <https://localhost:port>, maka akan diperoleh *output*:

Get your greeting [here](#)

- Jalankan pada *browser* Anda dengan URL <http://localhost:port/greeting>, maka akan diperoleh *output*:

Hello, World!

- Jalankan kembali program Anda pada *browser* dengan URL <http://localhost:port/greeting?name=D3TI>, maka akan diperoleh *output*:

Hello, D3TI!

Selamat Mengerjakan ☺