

# Praktikum 7

## Projek Web ala Spring Roo

### 1. Pengantar

- Spring Framework adalah framework yang digunakan untuk membangun aplikasi berbasis web dan Java. Dengan menggunakan Spring Framework, mahasiswa dapat membangun aplikasi dengan mudah dan cepat
- Dalam proses pembuatan projek, semua dilakukan menggunakan Netbeans IDE yang terkenal akan keramahan tampilan dan penggunaannya. Nantinya mahasiswa dapat memulai mengerjakan kode-kode aplikasi tersebut
- Praktikum ini akan membantu mahasiswa untuk mempelajari dan menggunakan Spring Framework dan Java

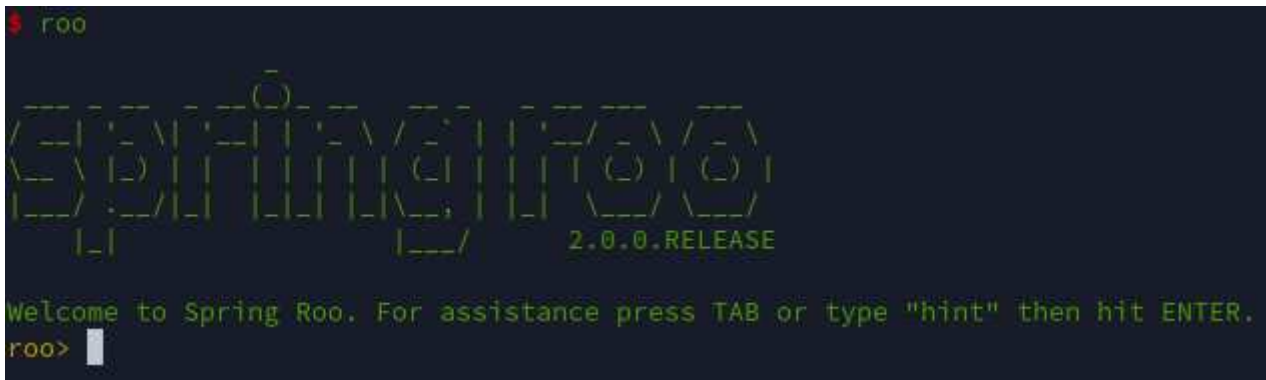
### 2. Pembahasan

Praktikum ini dimulai dengan pembuatan projek pertama dengan menggunakan Spring Roo. Namun sebelum projek dimulai, kebutuhan Spring Roo serta Maven harus terpenuhi seperti: Tomcat Server/GlassFish, Java EE plugin, Java Web plugin. Jika sudah terpenuhi mahasiswa dapat memulai projeknya.

- 1) Pastikan Spring Roo dan Maven sudah bisa diakses melalui CMD menggunakan tutorialnya.
- 2) Praktikum ini akan membuat Diari berbasis Web tanpa login
- 3) Bukalah CMD untuk mengetikkan perintah-perintah berikut
- 4) Buatlah folder dan masuk ke dalamnya dengan perintah

```
mkdir web  
cd web
```

- 5) Lalu kita harus membuka Spring Roo dengan mengetik perintah roo di CMD



6) Kemudian kita harus membuat projek awalnya dengan mengetik perintah berikut, lalu Enter:

```
project setup --topLevelPackage com.projek
```

```

root> project setup --topLevelPackage com.projek
Created ROOT/pom.xml
Created SRC_MAIN_RESOURCES
Created SRC_MAIN_RESOURCES/application-dev.properties
Created SRC_MAIN_JAVA/com/projek
Created SRC_MAIN_JAVA/com/projek/ProjekApplication.java
Created SRC_MAIN_RESOURCES/banner.txt

```

7) Setelah itu kita perlu menambahkan database, kita dapat menggunakan DB internal hanya dengan mengetikkan perintah berikut, lalu Enter:

```
jpa setup --provider HIBERNATE --database HYPERSONIC IN_MEMORY
```

```
root> jpa setup --provider HIBERNATE --database HYPERSONIC_IN_MEMORY
Created SRC_MAIN_RESOURCES/application.properties
Updated SRC_MAIN_RESOURCES/application.properties
Updated SRC_MAIN_RESOURCES/application-dev.properties
Updated ROOT/pom.xml [added dependencies org.springframework.boot:spring-boot-starter-data-jpa:, org.springframework.boot:spring-boot-starter-jdbc:, org.hsqldb:hsqldb::; added property 'springlets.version' = '1.2.0.RELEASE'; added dependencies io.springlets:springlets-data-jpa:${springlets.version}, io.springlets:springlets-data-jpa:${springlets.version}; added dependencies io.springlets:springlets-data-commons:${springlets.version}, io.springlets:springlets-data-commons:${springlets.version}]
```

8) Berikutnya kita membuat Diari namun masih kosong tanpa ada tempat input nama dan lain-lain. Ketik perintah berikut, lalu Enter:

```
entity jpa --class ~.praktikum.Diari
```

```
roo> entity jpa --class ~.praktikum.Diari
Created SRC_MAIN_JAVA/com/projek/praktikum
Created SRC_MAIN_JAVA/com/projek/praktikum/Diari.java
Updated SRC_MAIN_JAVA/com/projek/praktikum/Diari.java
Updated ROOT/pom.xml [added dependencies io.springlets:springlets-context:${springlets.version}, io.springlets:springlets-context:${springlets.version}]
Created SRC_MAIN_JAVA/com/projek/praktikum/Diari_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA/com/projek/praktikum/Diari_Roo_JavaBean.aj
```

9) Bagian ini adalah bagian yang paling penting, dan jangan sampai melakukan kesalahan urutan pengetikan!. Di dalam Diari yang kita butuhkan adalah **Nomor**, **Tanggal**, **Subjek**, dan **Isi** Diari. Perlu diketahui bahwa Nomor dan Tanggal memerlukan input tersendiri. Maka ketiklah perintah berikut secara berurutan:

Untuk **Nomor** → field **number** --fieldName **Nomor** --type **int**

Untuk **Tanggal** → field **date** --fieldName **Tanggal** --type **java.util.Date**

Untuk **Subjek** → field **string** --fieldName **Subjek** --notNull

Untuk **Isi** → field **string** --fieldName **Isi** --notNull

10) Berikut adalah daftar perintah yang bisa digunakan:

```
~.praktikum.Diari roo> field

field boolean      field date          field embedded      field enum
field file         field list          field number        field other
field reference    field set           field string
```

11) Field input sudah siap, berikutnya adalah membuat repository jpa dengan perintah berikut:

```
repository jpa --all
```

```
~.praktikum.Diari roo> repository jpa --all
Created SRC_MAIN_JAVA/com/projek/config
Created SRC_MAIN_JAVA/com/projek/config/SpringDataJpaDetachableRepositoryConfiguration.java
Created SRC_MAIN_JAVA/com/projek/repository
Created SRC_MAIN_JAVA/com/projek/repository/DiariRepository.java
Created SRC_MAIN_JAVA/com/projek/repository/DiariRepositoryCustom.java
Created SRC_MAIN_JAVA/com/projek/repository/DiariRepositoryImpl.java
Updated ROOT/pom.xml [added dependency com.querydsl:querydsl-jpa;; added depende
```

12) Setelah itu ketik perintah berikut:

```
service --all
```

13) Berikutnya adalah membuat tampilan web beserta controllernya dengan kode berikut ini:

```
web mvc setup  
web mvc view setup --type THYMELEAF  
web mvc controller --all --responseType THYMELEAF  
web mvc controller --all --pathPrefix /api
```

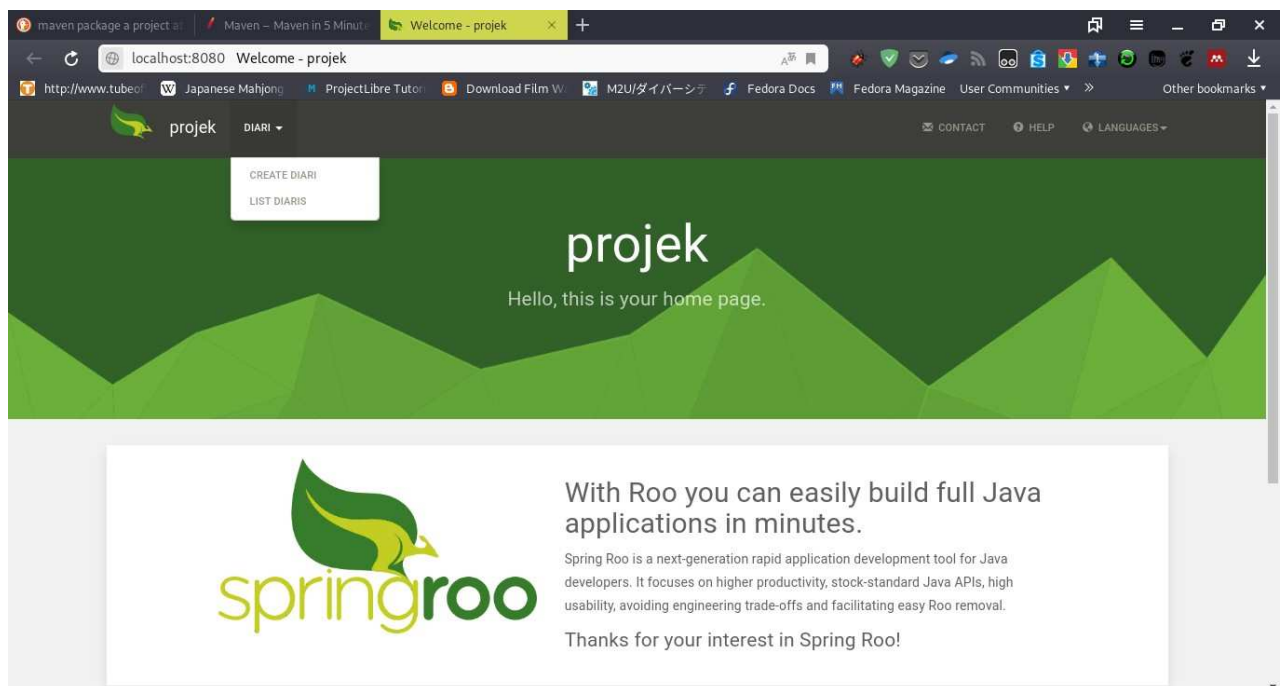
14) Setelah itu kita dapat keluar dari Spring Roo dengan perintah quit.

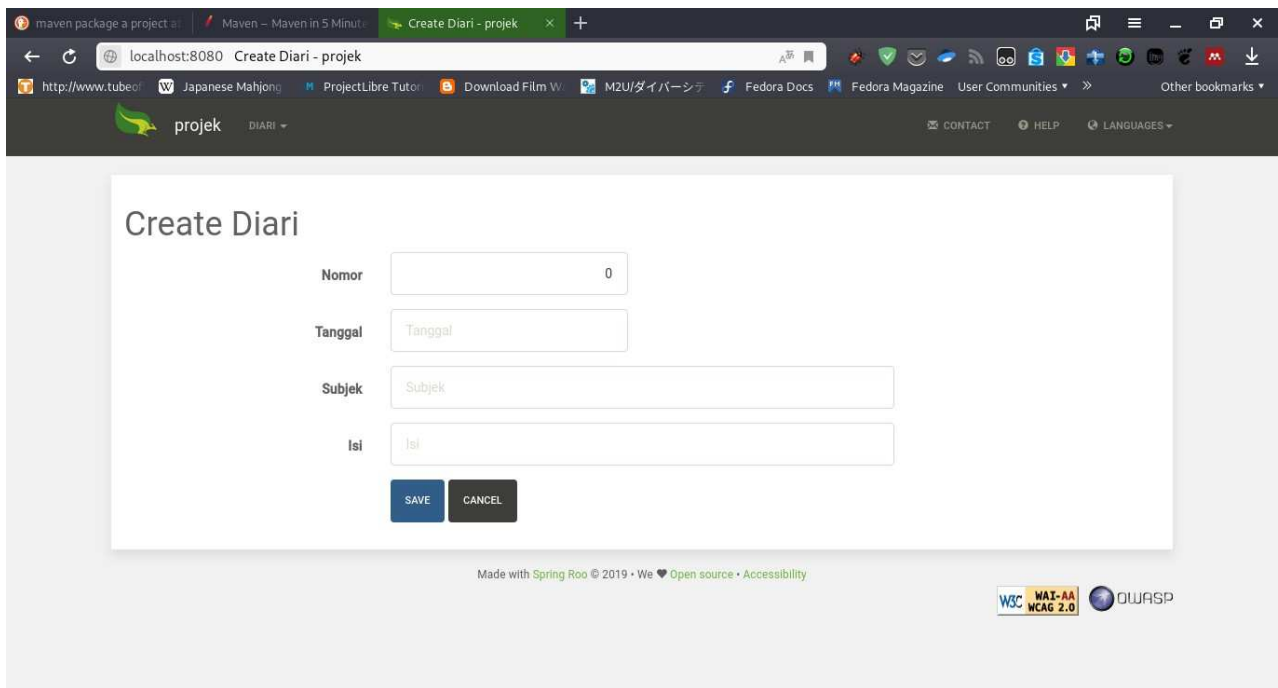
15) Hal terakhir yang dilakukan adalah Build, Run dan Package dengan perintah berikut:

Untuk **Build dan Run** → `mvn spring-boot:run`

Tunggu hingga muncul:

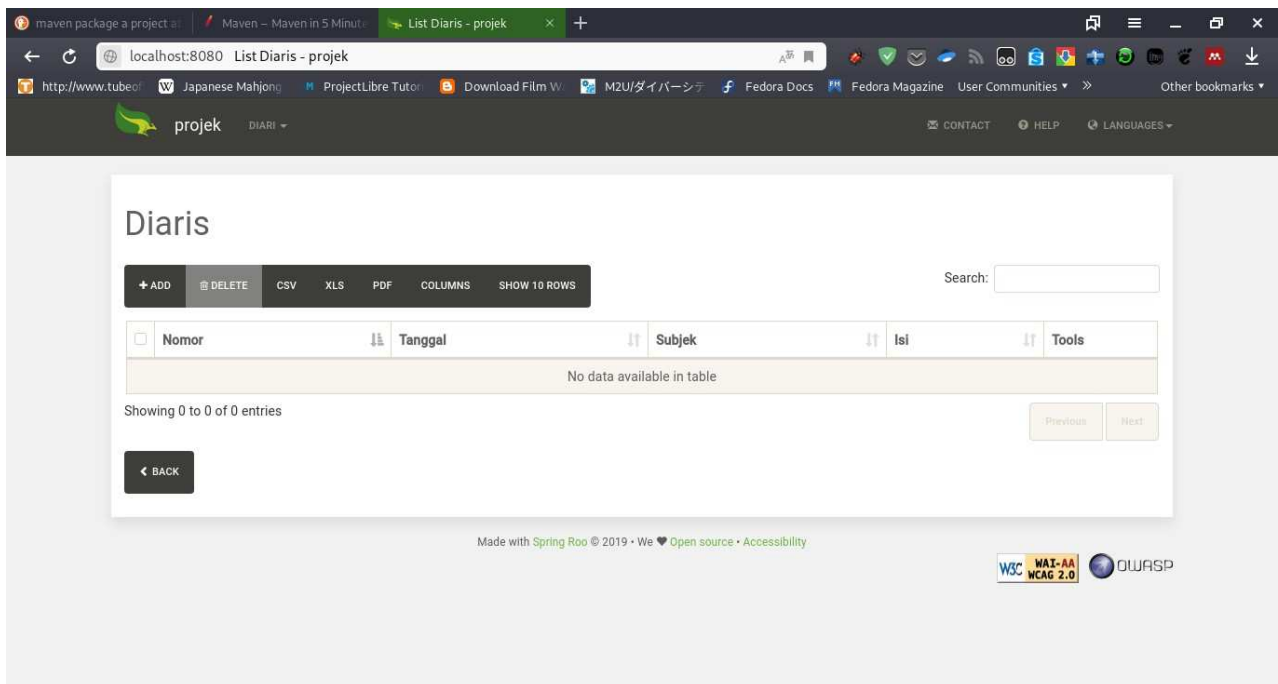
TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (<http://localhost:8080>)  
com.projek.ProjekApplication : Started ProjekApplication in 20.967 seconds





Make with [Spring Roo](#) © 2019 • We ♥ [Open source](#) • [Accessibility](#)

WCAG 2.0 OWASP



Make with [Spring Roo](#) © 2019 • We ♥ [Open source](#) • [Accessibility](#)

WCAG 2.0 OWASP

- Untuk membungkus projek menjadi satu file, eksekusi perintah berikut
- mvn package
- Lalu Maven akan membuat file jar di dalam folder target

```
[INFO] Building jar: /home/maulana/web/target/projek-1.0.0.BUILD-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:1.4.1.RELEASE:repackage (default) @ projek ---
[INFO] Attaching archive: /home/maulana/web/target/projek-1.0.0.BUILD-SNAPSHOT-exec.jar, with classifier: exec
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 41.736 s
[INFO] Finished at: 2019-05-16T18:33:49+07:00
[INFO] -----
```