

Spring MVC

Fitur Web Framework

- Struktur dan Konsistensi
- Form Binding : konversi dari request parameter menjadi object
- Validasi : JSR-303 (Declarative Validation)
- Upload File
- Multiple View

Setup Spring MVC

- Dependensi di pom.xml

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>4.0.5.RELEASE</version>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
    <scope>runtime</scope>
</dependency>
```

- Mendaftarkan/mengaktifkan DispatcherServlet : bisa di web.xml atau melalui annotation

```
<servlet>
    <servlet-name>belajar</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
```

```

</servlet>
<servlet-mapping>
    <servlet-name>belajar</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>

```

- Membuat konfigurasi DispatcherServlet dan mengaktifkan view JSP/JSTL

```

<?xml version='1.0' encoding='UTF-8'?>
<beans xmlns='http://www.springframework.org/schema/beans'
    xmlns:mvc='http://www.springframework.org/schema/mvc'
    xmlns:context='http://www.springframework.org/schema/context'
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xsi:schemaLocation='
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc.xsd'>
    <mvc:annotation-driven />
    <mvc:default-servlet-handler />

    <context:component-scan base-package='com.artivisi.belajar.controller' />

    <bean id='viewResolver' class='org.springframework.web.servlet.view.InternalRes
        <property name='viewClass' value='org.springframework.web.servlet.view.Jstl
    >
        <property name='prefix' value='/WEB-INF/templates/jsp' />
        <property name='suffix' value='.jsp' />

```

```
</bean>
</beans>
```

- Membuat handler / controller

```
@Controller
public class HaloController {

    @RequestMapping('/halo')
    public ModelAndView halo(){
        System.out.println('Menjalankan method halo');
        ModelAndView mm = new ModelAndView();

        mm.addAttribute('waktu', new Date());

        return mm;
    }
}
```

Jenis-jenis View yang didukung SpringMVC

- JSP/JSTL : output HTML
- Jasper Report : output PDF, XLS, DOC
- XML
- JSON
- [Freemarker](#) : output HTML
- [Velocity](#) : output HTML
- [Thymeleaf](#) : output HTML

Checklist implementasi request handler

- Implement class @Controller yang memiliki method dengan annotation @RequestMapping
- Buat file jsp untuk menampilkan data (ModelMap atau ModelAndView)