

Memulai Maven

April 2015

Oleh : Feri Djuandi

Tingkat:



Pemula



Menengah



Mahir

Secara umum **Maven** disebut sebagai sebuah *build automation tool*. Maven sangat berguna bagi programmer Java di dalam proses pengembangan aplikasi mulai dari pembuatan project, mereferensi *external library* (file-file JAR), pembuatan dokumentasi hingga *compiling* dan pembuatan *application package* (misalnya JAR atau WAR). Dengan demikian Maven adalah tool yang menemani programmer dari awal hingga akhir proses pengembangan.

Maven dikembangkan oleh *Apache Software Foundation* dan tipe lisensinya adalah *free software license*. Maven ditulis dalam bahasa pemrograman Java dan pada awal masa pembuatannya dikenal sebagai *Jakarta Project*.

Informasi selengkapnya mengenai Maven dapat dijumpai pada alamat situ:

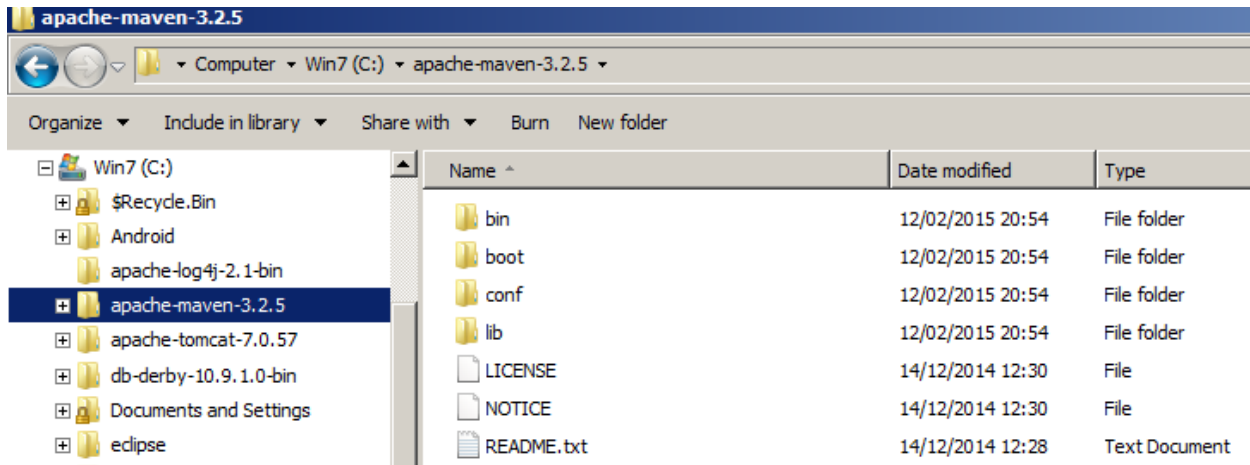
<http://maven.apache.org>

Pembaca juga dapat men-download Maven secara gratis pada situs yang sama. Pembahasan pada dokumentasi ini merujuk pada Maven versi 3.2.5, namun demikian pembaca bebas menggunakan versi yang berbeda. Pembahasan ini juga mengasumsikan sistem operasi yang digunakan adalah Microsoft Windows.

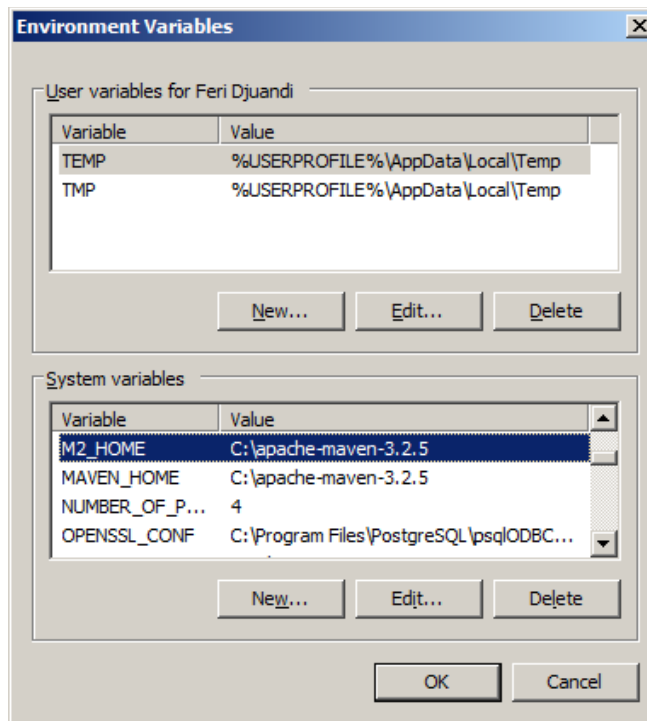
Sebelum mulai menggunakan Maven pastikan pada komputer pembaca telah terinstall *Java Development Kit (JDK)*. JDK yang digunakan pada pembahasan ini adalah 1.8.0, namun demikian pembaca dapat juga menggunakan versi yang lain.

Instalasi Maven

1. Silakan men-download Maven dari situs yang disebutkan di atas. File instalasi adalah sebuah file kompresi.
2. Ekstrak file instalasi tersebut ke komputer lokal, misalnya: C:\apache-maven-3.2.5. Perhatikan struktur folder dari hasil ekstrak tersebut pada gambar berikut ini.



3. Buka **Control Panel** dan lakukan setting secara manual pada **Environment Variables**.

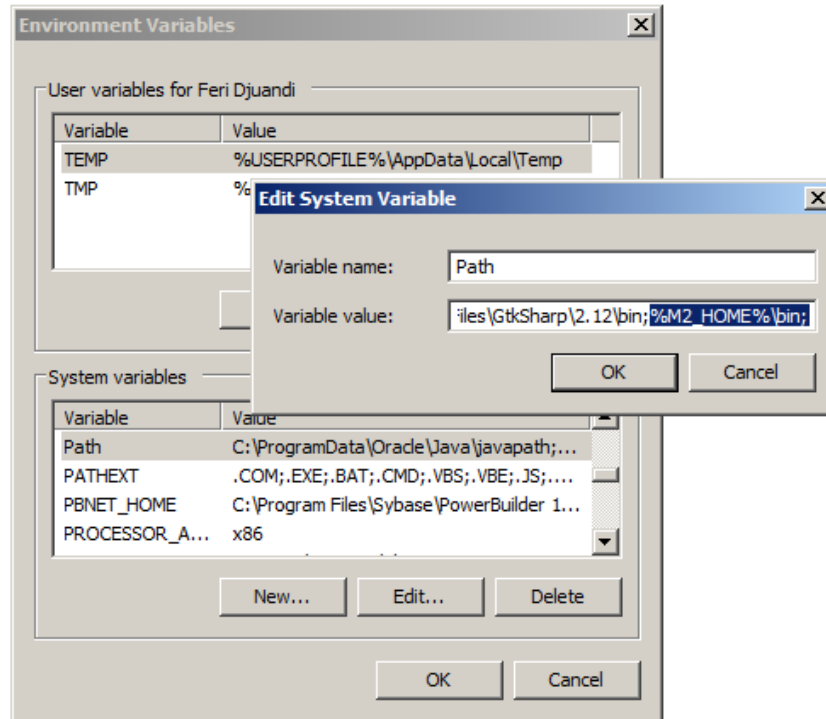


4. Pada **System Variables** buat dua buah *variable* dengan *value* sebagai berikut:

| Nama Variabel | Value |
|---------------|-----------------------|
| MS_HOME | C:\apache-maven-3.2.5 |
| MAVEN_HOME | C:\apache-maven-3.2.5 |

Silakan menyesuaikan value tersebut dengan hasil ekstrak pada langkah sebelumnya.

5. Buka variable **PATH** yang ada. Di akhir path yang sudah ada tambahkan teks **%M2_HOME%\bin;** yang merujuk ke folder Maven.

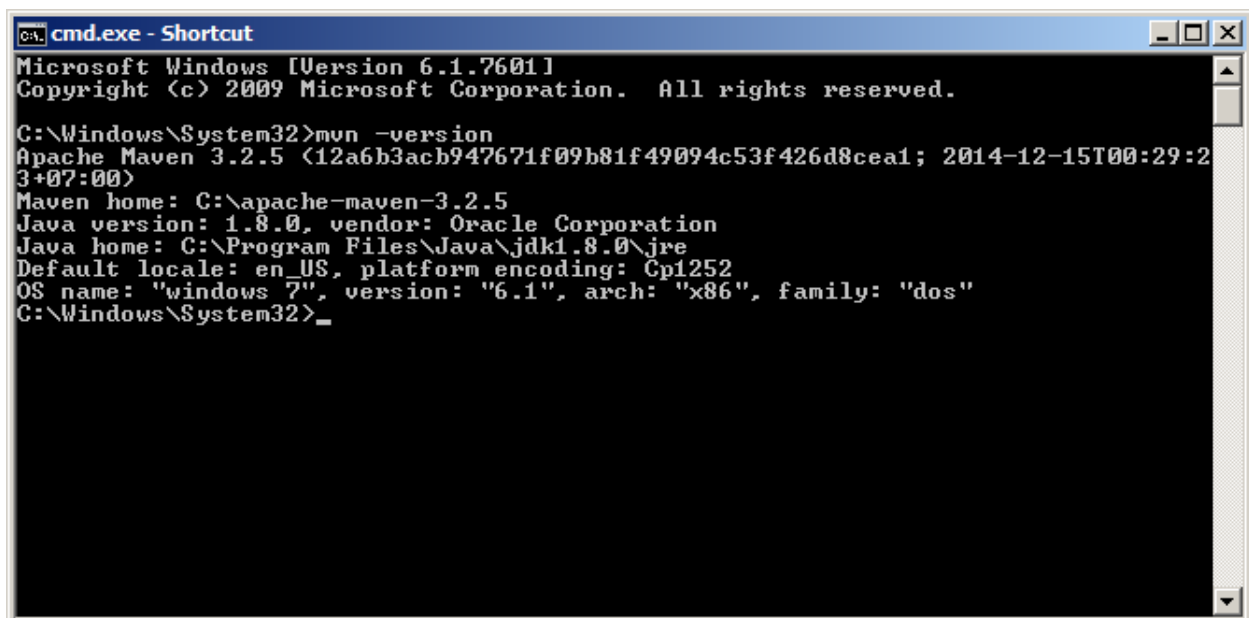


6. Simpan semua hasil setting.

Untuk menguji hasil instalasi Maven, silakan buka *command prompt* dan jalankan perintah berikut ini:

```
mvn -version
```

Jika instalasi dilakukan dengan benar maka tampilannya akan seperti di bawah ini.



Melihat Daftar Project Template

Setelah Maven terinstal dengan baik maka pada bagian berikut ini akan diperlihatkan cara membuat sebuah project Java menggunakan Maven.

Cara programmer membuat sebuah project Java adalah dengan memilih *project template* yang sudah disediakan oleh Maven. Maven menyediakan daftar template yang sangat banyak dan bisa dipilih sesuai kebutuhan programmer.

Untuk melihat daftar template tersebut silakan menjalankan perintah ini pada command prompt.

```
mvn archetype:generate
```

Perintah tersebut digunakan untuk membuat sebuah project Java. Tunggu beberapa saat sampai Maven memperlihatkan semua daftar template yang ada. Setelah daftar template ditampilkan, silakan melihat sepintas tipe-tipe yang disediakan oleh Maven dan bisa dipilih oleh programmer.

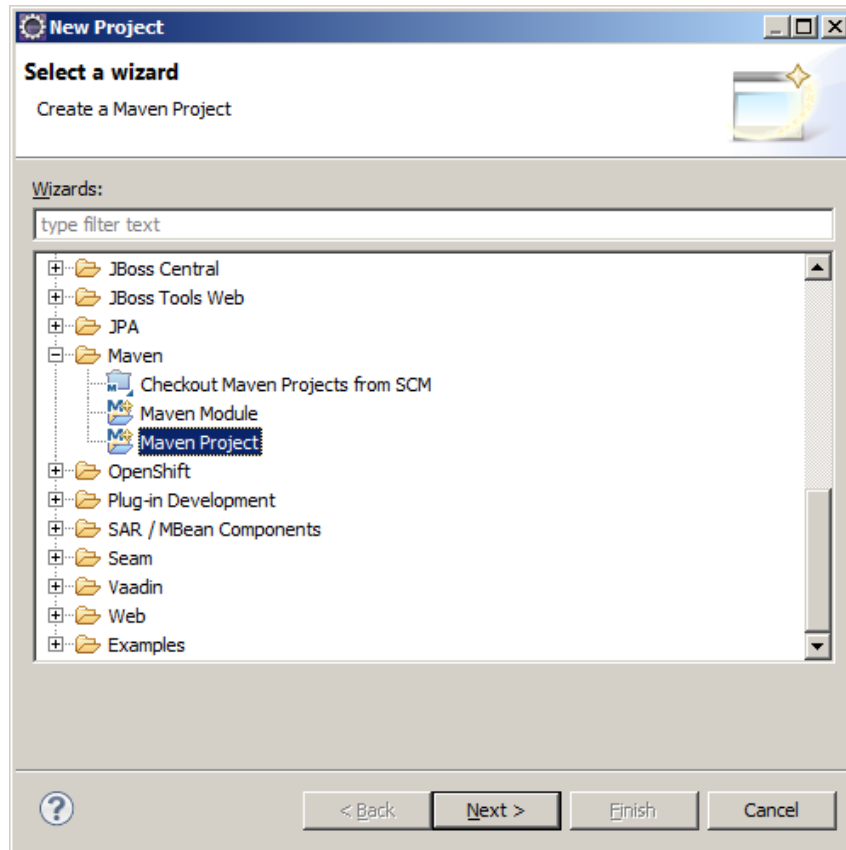
Untuk saat ini kita tidak akan membuat project dengan perintah command prompt, sehingga kita tidak akan melanjutkan. Tekan tombol Ctrl+C dan jawab "Y" pada pertanyaan "Terminate batch job?". Kita akan membuat project dengan langkah-langkah yang lebih mudah sekaligus memberikan pemahaman lebih baik kepada pembaca.

Membuat Project Baru

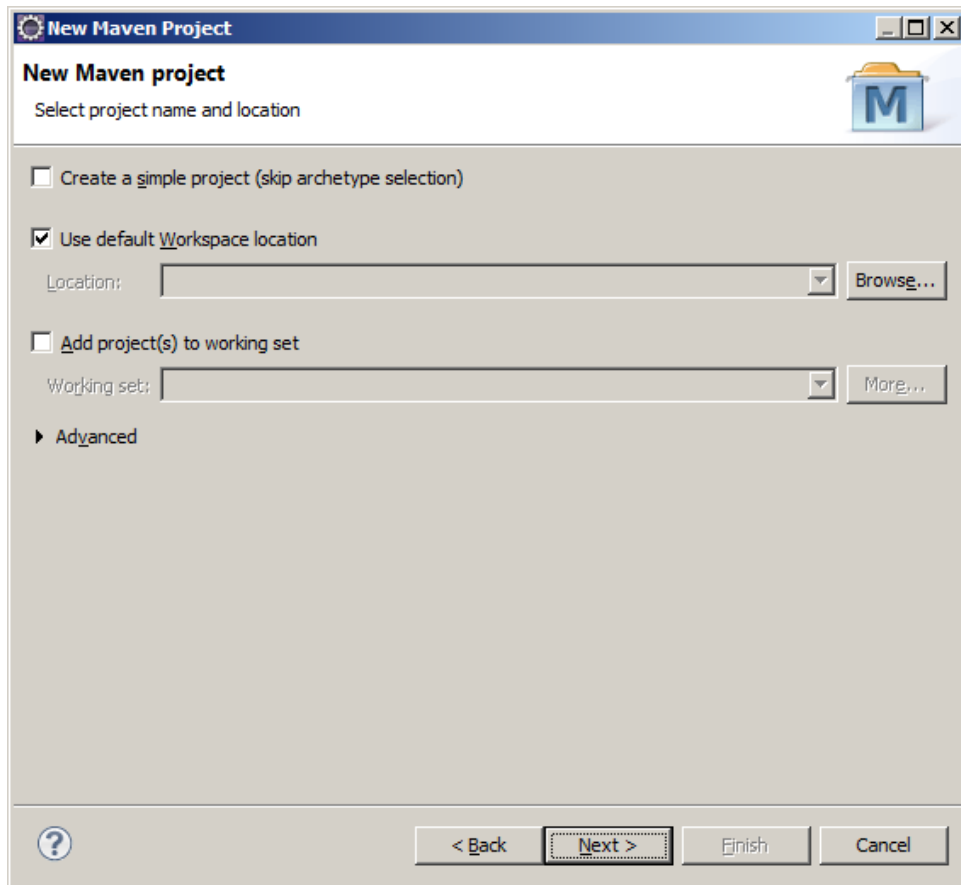
Pada pembahasan ini *Java IDE* yang digunakan adalah **Eclipse Java EE dengan versi Kepler**. Alasan penulis menggunakan IDE ini adalah karena ia terintegrasi dengan Maven. Pembaca dipersilakan menggunakan Java IDE yang lain selama fungsi Maven tercakup di dalamnya.

Pada saat programmer menggunakan Maven untuk membuat sebuah project baru atau mereferensikan dengan external library, pastikan komputer tersebut terhubung ke Internet karena Maven membutuhkan koneksi Internet untuk berkomunikasi dengan *Maven repository*. Maven akan mendownload file-file external library secara otomatis sesuai yang dibutuhkan oleh programmer dari servernya.

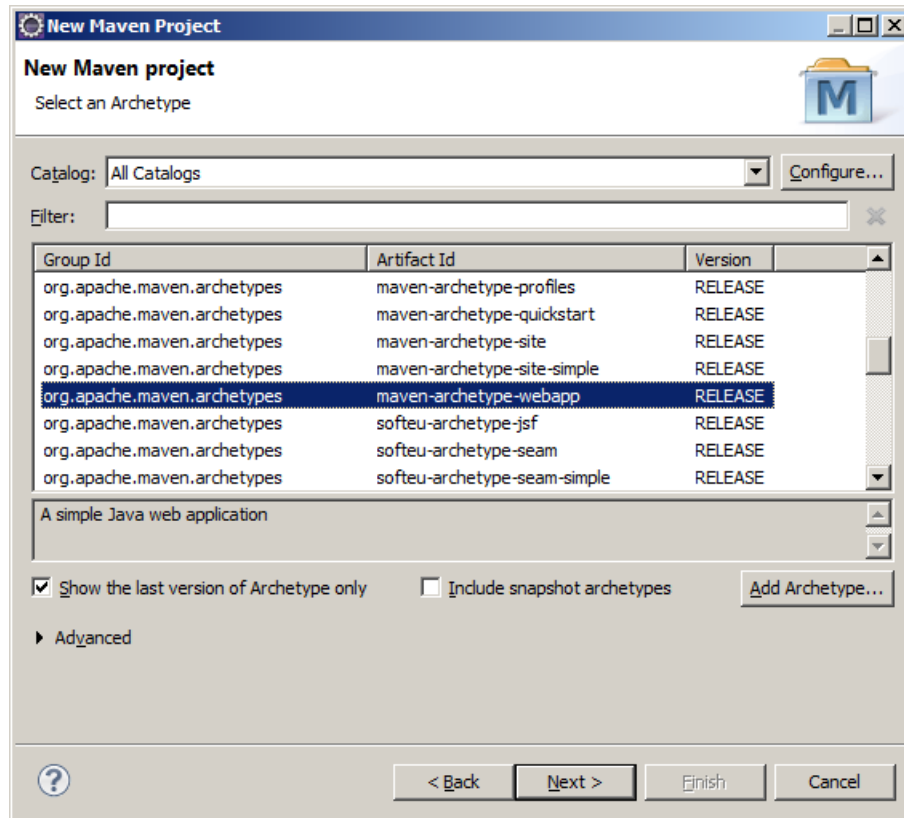
1. Silakan membuka Eclipse dan jalankan menu **File → New → Project**.
2. Pilih **Maven → Maven Project**.



3. Pilih folder dimana project akan ditempatkan, atau gunakan lokasi yang default.

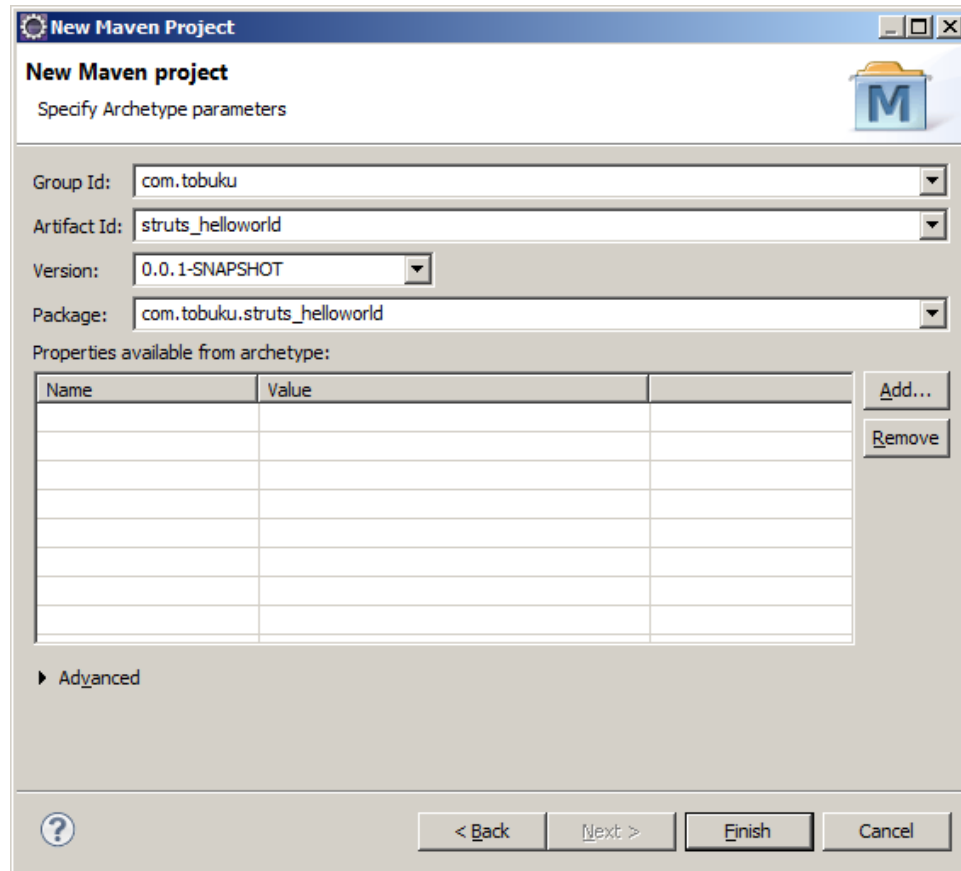


4. Pilih project template yang akan digunakan untuk project ini. Pada umumnya ada 2 template yang paling sering digunakan, yaitu:
 - **maven-archetype-webapp** – untuk aplikasi web (WAR)
 - **maven-archetype-quickstart** – untuk aplikasi Java biasa (JAR)

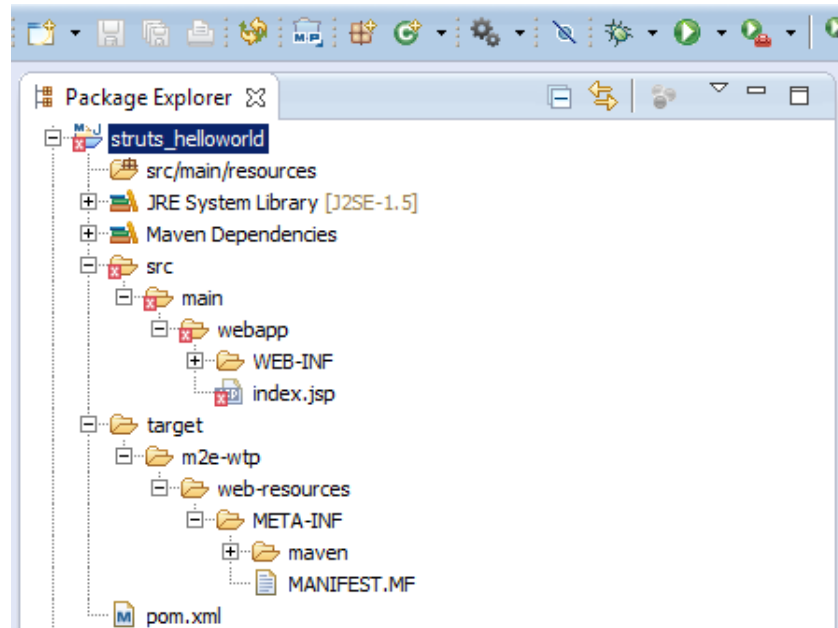


Pada contoh ini dipilih *maven-archetype-webapp* karena kita akan membuat sebuah aplikasi web.

5. Pada layar berikutnya Maven menanyakan beberapa parameter yang harus diisi.
 - **Group Id** adalah domain atau namespace, umumnya diisi dengan identitas organisasi.
 - **Artifact Id** adalah ID aplikasi. Silakan mengisi dengan kode yang unik dan spesifik yang mewakili kode aplikasi tersebut.
 - **Version**, umumnya menggunakan nilai default atau bebas ditentukan sendiri.
 - **Package** secara otomatis akan berformat <Group Id> + >Artifact Id>



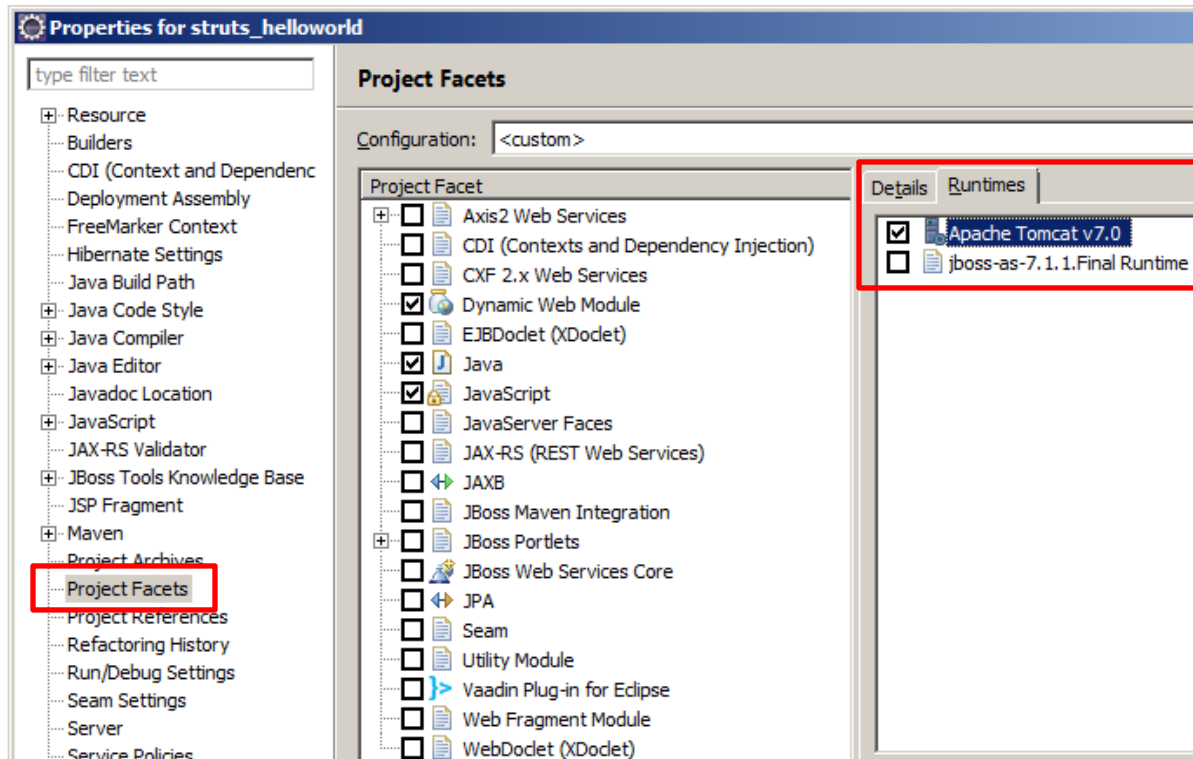
6. Klik **Finish** untuk menyelesaikan. Selanjutnya Maven akan segera membuat project tersebut termasuk struktur folder dan file-file konfigurasi yang dibutuhkan. Perhatikan pada gambar berikut ini bahwa pada **Project Explorer** akan muncul project baru dengan nama yang disebutkan di atas. Buka project tersebut untuk melihat isi di dalamnya.



Pada nama project tersebut tampak sebuah tanda silang merah yang menandakan adanya error. Tidak perlu khawatir karena ini adalah hal yang wajar. Ketika sebuah aplikasi web dibuat, *runtime*-nya harus ditentukan sedangkan langkah tersebut tidak dilakukan di dalam Maven. Runtime yang dimaksud disini adalah jenis *application server* yang diasosiasikan dengan project ini, misalnya Apache Tomcat atau JBoss AS.

Untuk menentukan runtime untuk sebuah project aplikasi web, silakan ikuti langkah-langkah berikut ini:

1. Klik-kanan pada nama project dan pilih **Properties**.
2. Pilih **Project Facets** kemudian pilih tab **Runtimes**.



Pilih salah satu Application Server sesuai dengan yang terinstall pada komputer. Perhatikan bahwa daftar runtime ini mungkin berbeda antar komputer. Pada contoh ini dipilih *Apache Tomcat*.

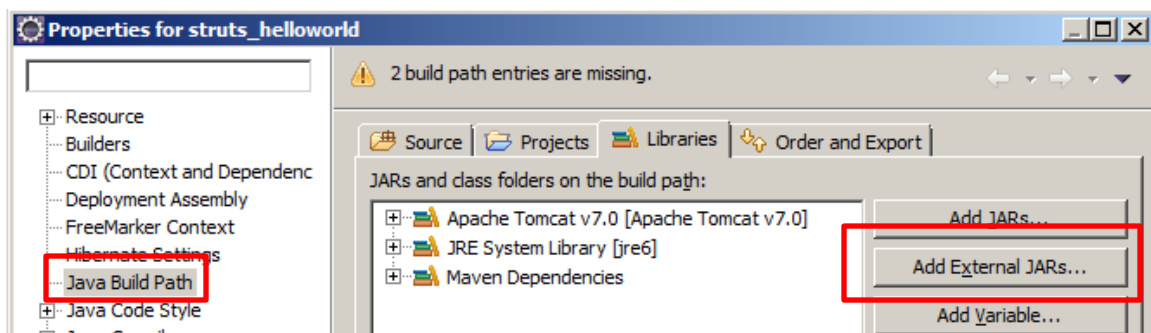
3. Klik **Ok** dan perhatikan bahwa tanda error telah hilang.

Mengenal pom.xml

Maven menyimpan konfigurasi project yang dibuat dalam sebuah file yang bernama **pom.xml**. POM adalah kependekan dari **Project Object Model**. Silakan temukan file pom.xml di dalam folder project dan membukanya. Contoh sebuah file pom.xml diperlihatkan di bawah ini.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.tobuku</groupId>
  <artifactId>struts_helloworld</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>struts_helloworld Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>struts_helloworld</finalName>
  </build>
</project>
```

Hingga saat ini project tersebut belum memiliki referensi ke external library (external JAR). Pada umumnya penambahan referensi dilakukan dengan menu **Build Path** → **Configure Build Path** kemudian menambahkan masing-masing JAR satu per satu dengan tombol **Add External JARs**. Ini adalah langkah manual untuk membuat referensi.



Kekurangan dari langkah ini adalah:

- Programmer harus mendapatkan sendiri setiap file JAR yang dibutuhkan, misalnya dengan cara men-download dari situs pembuatnya secara satu demi satu.

- Sulitnya untuk memastikan file JAR mana yang harus direferensikan secara lengkap. Sebagai contoh adalah Struts2. Framework ini terdiri dari banyak file JAR dan programmer harus memilih dengan cermat mana saja file JAR yang harus direferensikan. Cara ini cukup menyulitkan dan menyita waktu jika tidak tersedia dokumentasi yang memadai karena beberapa file JAR bisa terlewatkan dan mengakibatkan kesalahan ketika aplikasi dijalankan.

Kesulitan tersebut ditangani oleh Maven yang melakukannya dengan sangat baik. Di dalam Maven repository sudah terdapat berbagai macam daftar library berikut file-file JAR yang berkaitan. Yang programmer perlu lakukan adalah menyebutkan library yang dibutuhkannya (pada Maven disebut sebagai *dependency*) di dalam file pom.xml, dan selanjutnya Maven akan men-download dan mereferensikan dengan project yang bersangkutan.

Untuk memudahkan pembaca mencerna penjelasan ini, silakan mengikuti contoh berikut ini dimana kita akan menambahkan Struts2 framework ke dalam project.

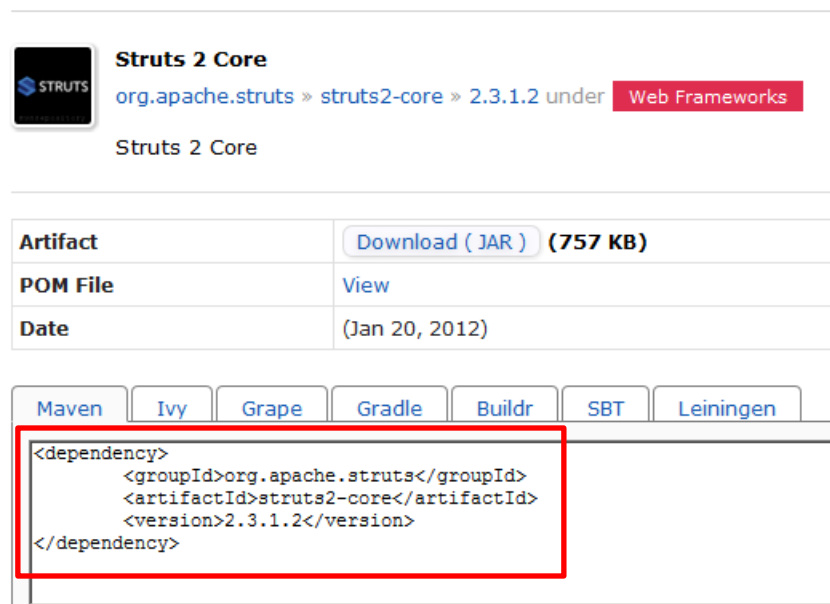
1. Dengan menggunakan Internet browser, buka situs:

<http://www.mvnrepository.com/>

Masukkan kata kunci “struts2” dan lakukan pencarian.

2. Jika ditemukan maka akan muncul item “Struts 2 Core”. Buka item tersebut dan akan ditampilkan daftar berbagai versi. Pada contoh ini kita akan menggunakan versi 2.3.1.2. Pembaca bebas memilih versi yang lebih baru jika diinginkan.
3. Pada saat versi tersebut dipilih, tampilan halaman akan seperti berikut ini.

Struts 2 Core » 2.3.1.2



Struts 2 Core
org.apache.struts » struts2-core » 2.3.1.2 under **Web Frameworks**

Struts 2 Core

| | |
|-----------------|--|
| Artifact | Download (JAR) (757 KB) |
| POM File | View |
| Date | (Jan 20, 2012) |

[Maven](#) [Ivy](#) [Grape](#) [Gradle](#) [Buildr](#) [SBT](#) [Leiningen](#)

```
<dependency>  
  <groupId>org.apache.struts</groupId>  
  <artifactId>struts2-core</artifactId>  
  <version>2.3.1.2</version>  
</dependency>
```

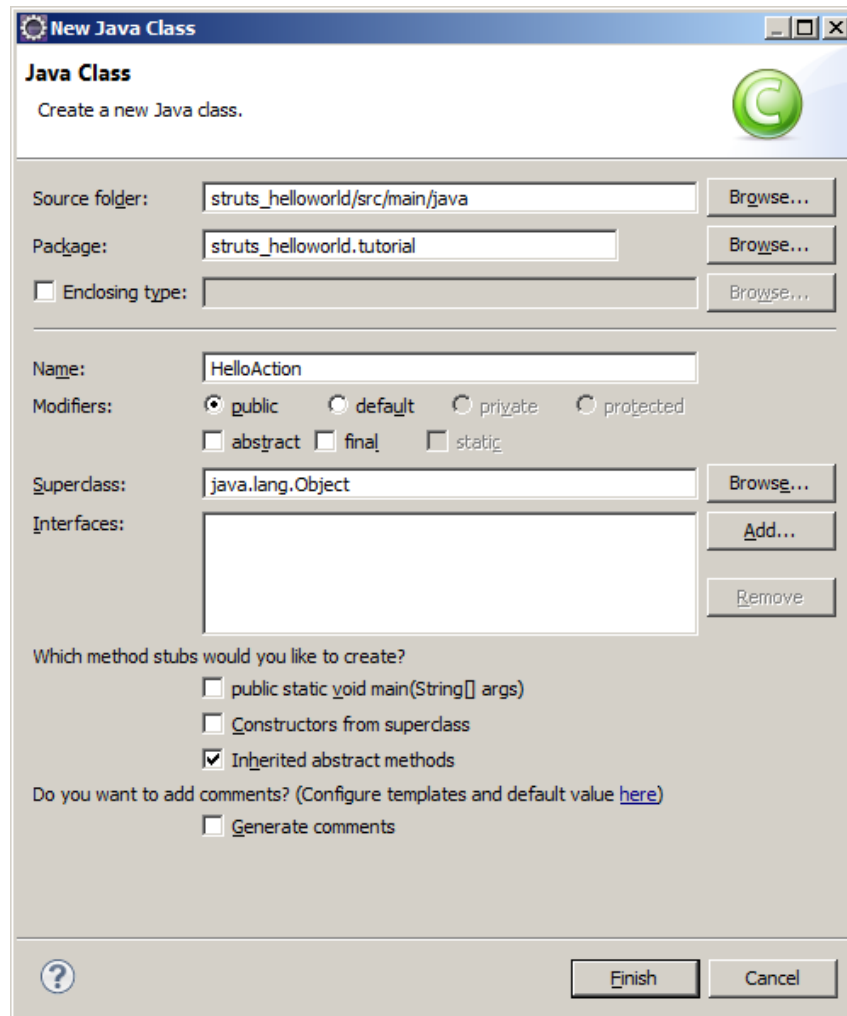
4. Salin seluruh teks pada bagian yang ditandai tersebut. Itu adalah definisi dari dependency Struts2.
5. Sekarang kembali ke Eclipse dan buka file pom.xml dari project yang sedang dibuat. Cari bagian yang ditandai dengan tag `<dependencies>`.
6. Masukkan teks yang disalin tadi ke dalam tag `<dependencies>` seperti berikut ini.

```
...
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.struts</groupId>
    <artifactId>struts2-core</artifactId>
    <version>2.3.1.2</version>
  </dependency>
</dependencies>
...
```

7. Simpan file pom.xml. Pada saat file itu disimpan, Maven secara otomatis men-download semua JAR yang berkaitan dengan dependency itu sekaligus mereferensikannya dengan project. Pastikan computer terhubung dengan Internet ketika hal ini dilakukan. Perhatikan juga *status bar* di bagian bawah window Eclipse ketika proses sedang berlangsung.

Jika tidak ditemukan kesalahan selanjutnya kita akan melanjutkan pembuatan aplikasi ini dengan Struts framework sekaligus menguji-cobanya.

1. Buat sebuah folder dengan menu **New → Folder** dan beri nama sebagai **src/main/java**. Folder ini akan digunakan untuk menampung semua class (file-file dengan ekstensi JAVA).
2. Di dalam folder baru tersebut, buat sebuah package dengan menu **New → Package** dan beri nama sebagai **struts_helloworld.tutorial**.
3. Di dalam package tersebut, buat sebuah class dengan menu **New → Class** dan bernama **HelloAction**.



4. Ketikkan baris-baris program di dalam class **HelloAction.java**.

```
package struts_helloworld.tutorial;

import com.opensymphony.xwork2.ActionSupport;

public class HelloAction extends ActionSupport {
    private static final long serialVersionUID = 1L;
    public static final String MESSAGE = "Saya dipanggil dari dalam
Struts";
    private String message="";
    public String execute() {
        setMessage(MESSAGE);
        return SUCCESS;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

```
}  
}
```

5. Bukalah file **web.xml** di dalam folder **src/main/webapp/WEB-INF** dan ubah isinya menjadi seperti berikut ini.

```
<!DOCTYPE web-app PUBLIC  
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
"http://java.sun.com/dtd/web-app_2_3.dtd" >  
  
<web-app>  
  <display-name>Archetype Created Web Application</display-name>  
  <filter>  
    <filter-name>struts2</filter-name>  
    <filter-  
class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>  
  </filter>  
  <filter-mapping>  
    <filter-name>struts2</filter-name>  
    <url-pattern>/*</url-pattern>  
  </filter-mapping>  
  <welcome-file-list>  
    <welcome-file>index.jsp</welcome-file>  
  </welcome-file-list>  
</web-app>
```

6. Buatlah sebuah file XML di dalam folder **src/main/resources** dengan nama **struts.xml**, dan ketikkan baris-baris program berikut ini.

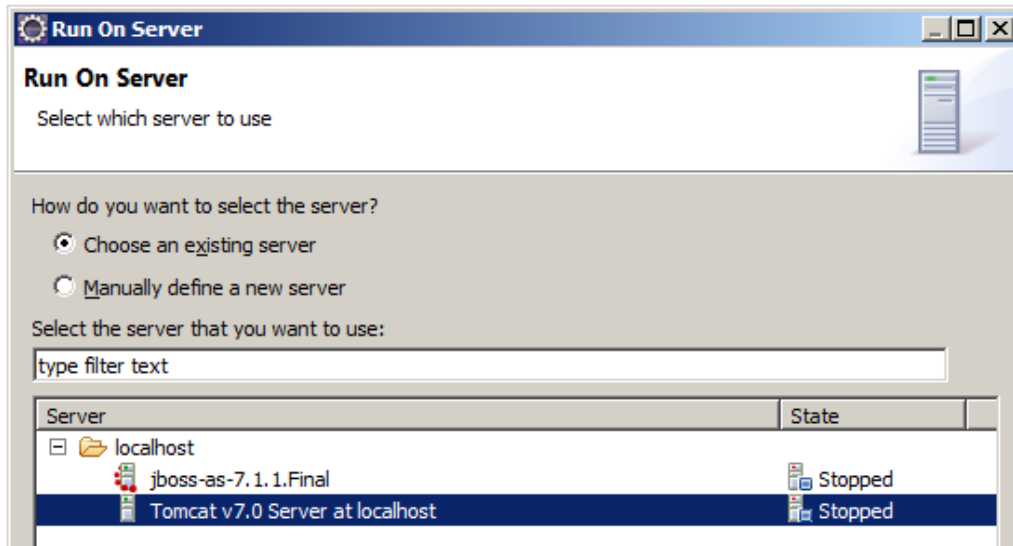
```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE struts PUBLIC  
  "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"  
  "http://struts.apache.org/dtds/struts-2.0.dtd">  
<struts>  
  <package name="tutorial" namespace="/" extends="struts-default">  
    <action name="hello"  
class="struts_helloworld.tutorial.HelloAction">  
      <result>/index.jsp</result>  
    </action>  
  </package>  
</struts>
```

7. Bukalah file **index.jsp** di dalam folder **src/main/webapp** dan ubah isinya menjadi seperti berikut ini.

```
<%@ taglib prefix="s" uri="/struts-tags"%>  
<html>  
<body>  
  <h1>Selamat Datang</h1>  
  <h2>  
    <s:property value="message" />  
  </h2>  
</body>  
</html>
```

```
</h2>  
</body>  
</html>
```

8. Jalankan aplikasi dengan menu **Run As** → **Run on Server** dan pilih Apache Tomcat sebagai server-nya.



Tampilan aplikasi diperlihatkan pada gambar di bawah ini.



Tambahkan teks **/hello.action** pada *address bar* untuk memanggil class HelloAction dan memastikan Struts2 Framework berjalan dengan baik.



Membuat Installation Package

Setelah aplikasi web ini berjalan dengan baik, langkah selanjutnya adalah membuat installation package yang berbentuk WAR.

Cara membuat package melalui Eclipse adalah dengan menjalankan menu **Run As → Maven Install**.

Namun dari pengalaman penulis, langkah yang dijalankan ini menghasilkan kesalahan:

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
```

```
SLF4J: Defaulting to no-operation (NOP) logger implementation
```

```
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

Dari referensi yang didapatkan oleh penulis, hal tersebut berikaitan dengan bug.

<http://stackoverflow.com/questions/11916706/slf4j-failed-to-load-class-org-slf4j-impl-staticloggerbinder-error>

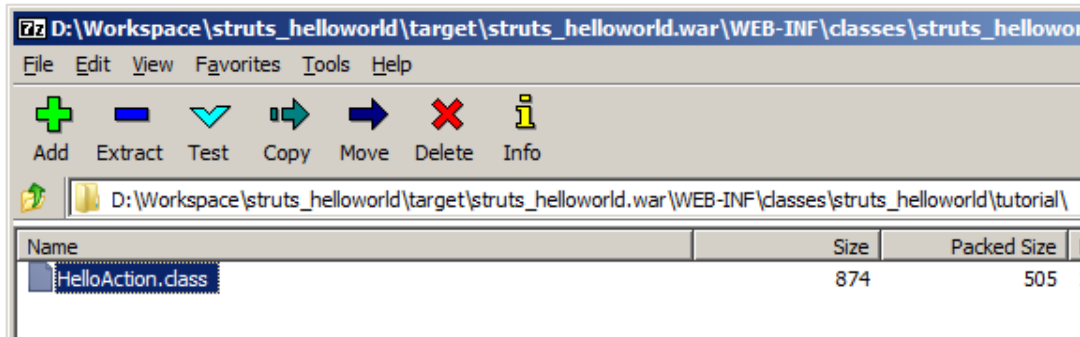
Oleh karena itu kita tidak akan menggunakan Eclipse untuk membuat file WAR namun langsung menjalankan perintah Maven.

1. Buka command prompt dan masuk ke dalam folder dimana project tersebut disimpan. Pada contoh ini, lokasi project berada di folder D:\Workspace\struts_helloworld. Perhatikan bahwa di folder tersebut terdapat file pom.xml.
2. Jalankan perintah berikut ini untuk melakukan compiling dan pembuatan package WAR.

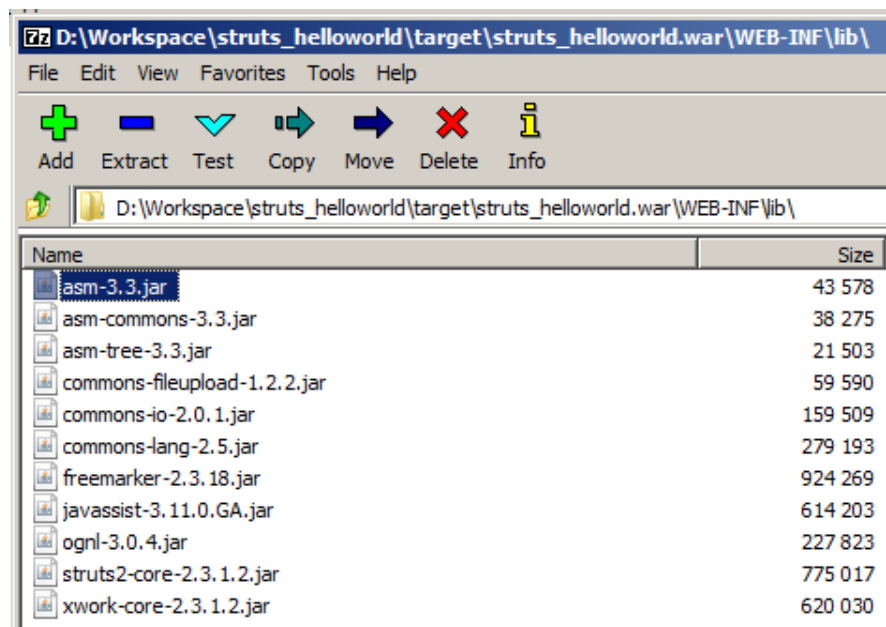
```
mvn clean package
```
3. Tunggu beberapa saat sampai Maven menyelesaikan semua proses.
4. Hasil compiling dan package dapat ditemukan di dalam folder **../target**. Pastikan di dalam folder itu terdapat file **struts_helloworld.war** yang menandakan proses berjalan baik tanpa kesalahan.

Selanjutnya jika pembaca ingin mengetahui lebih jauh mengenai apa yang ada di dalam file WAR ini, silakan membukanya menggunakan program ekstrak seperti **7-zip**.

Hasil compiling terdapat di dalam folder ../WEB-INF/classes.



File-file JAR yang di-download oleh Maven dikemas di dalam folder ../WEB-INF/lib termasuk file-file yang digunakan oleh Struts2 Framework.



Proses otomatisasi oleh Maven sungguh sangat membantu, bukan?

Referensi:

- http://en.wikipedia.org/wiki/Apache_Maven
- <http://www.tutorialspoint.com/maven/>
- <http://www.mkyong.com/tutorials/maven-tutorials/>

Proverbs 1:7. The fear of the LORD is the beginning of knowledge: but fools despise wisdom and instruction.