
 <b>ULBI</b> Universitas Logistik & Bisnis Internasional	Tugas Algoritma dan Struktur Data 2
Nama :	Aditya Nur Khaerullah
NPM :	714242005
Matakuliah :	Algoritma dan Struktur Data 2

## 1. Konfigurasi awal:

- Menggunakan Maven untuk membuat project aplikasi

← ↻ <https://maven.apache.org/download.cgi>



[Apache](#) / [Maven](#) / Download Apache Maven

Welcome

License

ABOUT MAVEN

[What is Maven?](#)

[Installation](#)

**[Downloads](#)**

[Use](#)

[Run](#)

[Configure](#)

[Release Notes](#)

DOCUMENTATION

[Maven Plugins](#)

[Maven Extensions](#)

[Maven Tools](#)

[Maven Daemon](#)

## Downloading Apache Maven 3.9.11

Apache Maven 3.9.11 is the latest release: it is the recommended version for all users.

### System Requirements


<b>Java Development Kit (JDK)</b>	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.
<b>Memory</b>	No minimum requirement
<b>Disk</b>	Approximately 10MB is required for the Maven installation itself. In addition to that, c repository will vary depending on usage but expect at least 500MB.
<b>Operating System</b>	No minimum requirement. Start up scripts are included as shell scripts (tested on m

### Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive


- Tomcat untuk running aplikasi

← ↻ <https://tomcat.apache.org/download-10.cgi>



## Apache Tomcat®

Search...



**Apache Tomcat**

[Home](#)

[Taglibs](#)

[Maven Plugin](#)

**Download**

Which version?

[Tomcat 11](#)

[Tomcat 10](#)

### Tomcat 10 Software Downloads

Welcome to the Apache Tomcat® 10.x software download page. This page pro releases.

Unsure which version you need? Specification versions implemented, minimu

Users of Tomcat 10 onwards should be aware that, as a result of the move froi implemented APIs has changed from `javax.*` to `jakarta.*`. This will almost c [migration tool](#) has been developed to aid this process.

#### Quick Navigation

[KEYS](#) | [10.1.43](#) | [Browse](#) | [Archives](#)

#### Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP sig keys of Tomcat's Release Managers. We also provide `SHA-512` checksums for e

## 2. Membuat Aplikasi:

- Inialisasi Aplikasi dengan Maven

```
mvn archetype:generate -DgroupId=com.aditya.webapp -DartifactId=TvShowApp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

- Mengisi file pom.xml

```
<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.aditya.webapp</groupId>
  <artifactId>TvShowApp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>TvShowApp 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>
    <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20240303</version>
    </dependency>
    <dependency>
      <groupId>jakarta.servlet</groupId>
      <artifactId>jakarta.servlet-api</artifactId>
      <version>6.0.0</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>TvShowApp</finalName>
  </build>
</project>
```

- File web.xml

```
<!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>
  <welcome-file-list>
    <welcome-file>shows</welcome-file> <!-- Servlet akan forward ke index.jsp -->
  </welcome-file-list>
  <servlet>
    <servlet-name>TvMazeServlet</servlet-name>
    <servlet-class>com.tvshowapp.TvMazeServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>TvMazeServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>TvMazeServlet</servlet-name>
    <url-pattern>/shows</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>TvShowDetailServlet</servlet-name>
    <servlet-class>com.tvshowapp.TvShowDetailServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>TvShowDetailServlet</servlet-name>
    <url-pattern>/detail</url-pattern>
  </servlet-mapping>
</web-app>
```

- Membuat File src/main/java/TvMazeServlet.java

```
package com.tvshowapp;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

Windsurf: Refactor | Explain
public class TvMazeServlet extends HttpServlet {
    Windsurf: Refactor | Explain | Generate Javadoc | X
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String query = request.getParameter("q");
        String apiUrl;

        if (query != null && !query.trim().isEmpty()) {
            apiUrl = "https://api.tvmaze.com/search/shows?q=" + URLEncoder.encode(query, "UTF-8");
            request.setAttribute("isSearch", true);
        } else {
            apiUrl = "https://api.tvmaze.com/shows";
            request.setAttribute("isSearch", false);
        }

        URL url = new URL(apiUrl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");

        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        StringBuilder apiResponse = new StringBuilder();
        String line;

        while ((line = reader.readLine()) != null) {
            apiResponse.append(line);
        }
        reader.close();

        request.setAttribute("apiResult", apiResponse.toString());
        RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
        dispatcher.forward(request, response);
    }
}
```

- Membuat File src/main/java/TvShowDetailServlet.java

```
package com.tvshowapp;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;

Windsurf: Refactor | Explain
public class TvShowDetailServlet extends HttpServlet {
    Windsurf: Refactor | Explain | Generate Javadoc | X
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String id = request.getParameter("id");
        if (id == null || id.trim().isEmpty()) {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "ID TV Show tidak valid");
            return;
        }

        String apiUrl = "https://api.tvmaze.com/shows/" + id;

        URL url = new URL(apiUrl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");

        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        StringBuilder apiResponse = new StringBuilder();
        String line;

        while ((line = reader.readLine()) != null) {
            apiResponse.append(line);
        }
        reader.close();

        request.setAttribute("detailResult", apiResponse.toString());
        RequestDispatcher dispatcher = request.getRequestDispatcher("detail.jsp");
        dispatcher.forward(request, response);
    }
}
```

- Mengisi file index.jsp

```
<form method="get" action="shows">
  <input type="text" name="q" placeholder="Cari TV Show..." value="<%= request.getParameter("q") != null ? request.getParameter("q") : "" %>" />
  <button type="submit">Cari</button>
</form>

<div class="container">
  <%
    String result = (String) request.getAttribute("apiResult");
    boolean isSearch = request.getAttribute("isSearch") != null && (Boolean) request.getAttribute("isSearch");

    if (result != null) {
      try {
        JSONArray array = new JSONArray(result);
        int maxToShow = 30;

        <div class="grid">
          <%
            for (int i = 0; i < Math.min(array.length(), maxToShow); i++) {
              JSONObject show = isSearch ? array.getJSONObject(i).getJSONObject("show") : array.getJSONObject(i);
              String name = show.optString("name", "Tanpa Nama");
              int id = show.optInt("id");
              String imageUrl = "#";
              if (!show.isNull("image")) {
                JSONObject imageObj = show.getJSONObject("image");
                imageUrl = imageObj.optString("medium", "#");
              }
              String rating = show.has("rating") && !show.isNull("rating") ? show.getJSONObject("rating").optString("average", "-") : "-";
              String genreList = show.has("genres") ? show.getJSONArray("genres").join(", ").replace("\\", "") : "-";

              <div class="card-small">
                <% if (!imageUrl.equals("#")) { %>
                  <a href="detail?id=<%= id %>">
                    
                  </a>
                <% } %>
                <div class="card-meta">
                  <div class="show-title"><a href="detail?id=<%= id %>"><%= name %></a></div>
                  <div><strong>Rating:</strong> <%= rating %></div>
                  <div><strong>Genre:</strong> <%= genreList %></div>
                </div>
              </div>
            } // end for
          </div>
          <%
            } catch (Exception e) {
              <div class="card-small">Terjadi kesalahan saat parsing data: <%= e.getMessage() %></div>
            }
          <%
            } else {
              <div class="card-small">Belum ada data. Akses <a href="shows">/shows</a> untuk melihat daftar show dari API.</div>
            }
          <%
        }
      }
    }
  </div>
```

Mengambil data TV Show dari TvMaze API. Data yang ditampilkan adalah Judul, Gambar, Rating, dan Genre. Terdapat juga fitur search untuk mencari judul TV Show.

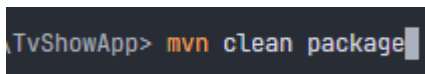
- Menambah file detail.jsp

```
<body>
<div class="container">
    <%
        String detailJson = (String) request.getAttribute("detailResult");
        if (detailJson != null) {
            try {
                JSONObject show = new JSONObject(detailJson);
                String name = show.optString("name", "Tanpa Nama");
                String summary = show.optString("summary", "Tidak ada deskripsi");
                String imageUrl = show.isNull("image") ? null : show.getJSONObject("image").optString("medium");
                String language = show.optString("language", "-");
                String status = show.optString("status", "-");
                String genreList = show.has("genres") ? show.getJSONArray("genres").join(", ").replace("\\", "") : "-";
                String rating = show.has("rating") && !show.isNull("rating") ? show.getJSONObject("rating").optString("average", "-") : "-";
                String premiered = show.optString("premiered", "-");
                String ended = show.optString("ended", "-");
                String runtime = show.optString("runtime", "-");
                String officialSite = show.optString("officialSite", "-");
                String network = "-";
                if (!show.isNull("network")) {
                    network = show.getJSONObject("network").optString("name", "-");
                } else if (!show.isNull("webChannel")) {
                    network = show.getJSONObject("webChannel").optString("name", "-");
                }
            } catch (Exception e) {
                // Handle exception
            }
        }
    %>
    <h2><%= name %></h2>
    <% if (imageUrl != null) { %>
        
    <% } %>
    <p><strong>Bahasa:</strong> <%= language %></p>
    <p><strong>Status:</strong> <%= status %></p>
    <p><strong>Genre:</strong> <%= genreList %></p>
    <p><strong>Rating:</strong> <%= rating %></p>
    <p><strong>Tayang Perdana:</strong> <%= premiered %></p>
    <p><strong>Tamat:</strong> <%= ended %></p>
    <p><strong>Durasi per episode:</strong> <%= runtime %> menit</p>
    <p><strong>Jaringan/Platform:</strong> <%= network %></p>
    <% if (officialSite != null && !officialSite.equals("null") && !officialSite.isEmpty()) { %>
        <p><strong>Situs Resmi:</strong> <a href="<%= officialSite %>" target="_blank"><%= officialSite %></a></p>
    <% } %>
    <p><strong>Deskripsi:</strong></p>
    <div><%= summary %></div>
    <div style="clear:both;">
    <a href="#shows" class="back-link">Kembali ke daftar</a>
    <%
        } catch (Exception e) {
    %>
    <p>Gagal memuat detail: <%= e.getMessage() %></p>
    <%
        }
    } else {
    %>
    <p>Data tidak tersedia.</p>
    <%
        }
    %>
</div>
</body>
```

Menampilkan detail data dari satu TV Show yang dipilih, diantaranya data Bahasa, Status, Genre, Rating, Tayang Perdana, Tanggal Tamat, Durasi, Platform, Situs Resmi (jika tertera), dan Deskripsi.

### 3. Running Aplikasi:

- Build dan Update Aplikasi: **mvn clean package**



- Mengambil hasil build yaitu file .war dan menyimpannya pada Tomcat

This PC > Windows-SSD (C:) > Program Files > Apache > Tomcat > apache-tomcat-10.1.43 > webapps >

Name	Date modified	Type	Size
docs	01/07/2025 21:30	File folder	
examples	01/07/2025 21:30	File folder	
host-manager	01/07/2025 21:30	File folder	
manager	01/07/2025 21:30	File folder	
ROOT	01/07/2025 21:30	File folder	
TvShowApp	18/07/2025 03:49	File folder	
TvShowApp.war	18/07/2025 03:48	WAR File	81 KB

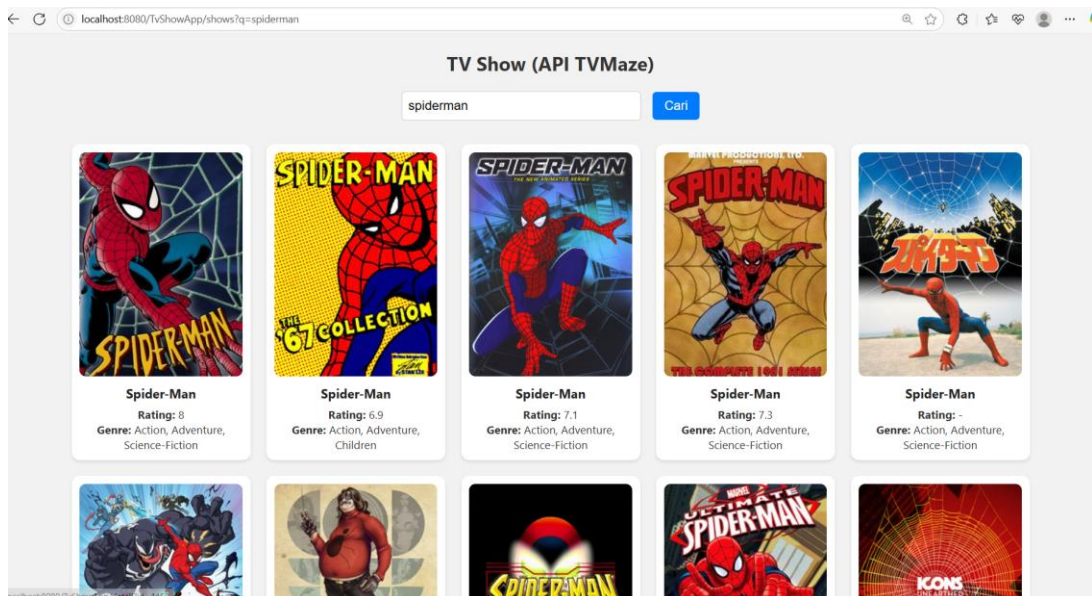
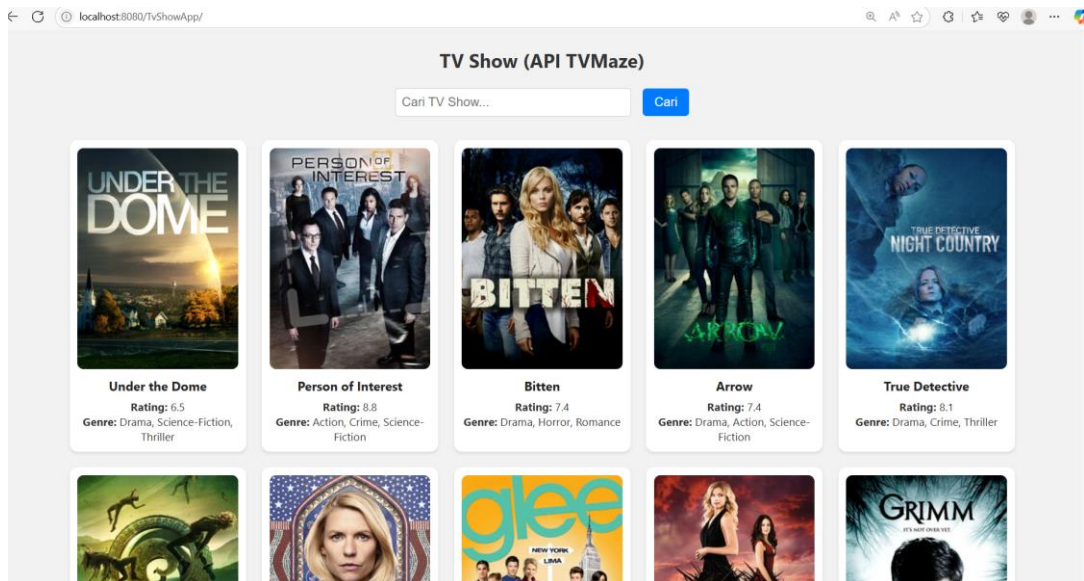
- Jalankan Apache Tomcat: `startup.bat`

```
C:\Program Files\Apache\Tomcat\apache-tomcat-10.1.43\bin>startup.bat
```

- Melihat aplikasi yang sudah dibuat di localhost

```
http://localhost:8080/TvShowApp
```

#### 4. Hasil



## Spider-Man

**Bahasa:** English

**Status:** Ended

**Genre:** Action, Adventure, Science-Fiction

**Rating:** 8

**Tayang Perdana:** 1994-11-19

**Tamat:** 1998-01-31

**Durasi per episode:** 30 menit

**Jaringan/Platform:** FOX

**Situs Resmi:** <https://www.disneyplus.com/en-nl/series/spider-man/SIQvHdSZO9kv>

### Deskripsi:

**Spider-Man** (1994) tells the story of a 19-year-old Peter Parker in his first year at Empire State University, and his alter-ego Spider-Man. The show features Spider-Man's classic villains, including the Kingpin, the Green Goblin, the Lizard, the Scorpion, Doctor Octopus, Mysterio, the Rhino, the Shocker, The Vulture and the Chameleon, as well as more recent villains such as Venom, Carnage and the Hobgoblin.

[← Kembali ke daftar](#)

