

Panduan Pendayagunaan Open Source Software :
Aplikasi Untuk Server

PENGANTAR

Perkembangan Teknologi Informasi dan Komunikasi (TIK) yang sedemikian pesat telah membawa dampak yang positif dalam era peningkatan kinerja dari berbagai kegiatan yang dilakukan. Penerapan teknologi informasi dan komunikasi telah digunakan dihampir semua sektor, mulai dari transportasi, perdagangan, ekonomi, manufaktur, pemerintahan, pendidikan, kesehatan hingga pertahanan. Dengan kemajuan teknologi informasi dan komunikasi, informasi tidak lagi mengenal batas-batas negara, berbagai transaksi ekonomi, perbankan hingga komunikasi tatap muka, dengan siapa saja di belahan bumi manapun dapat berlangsung melalui teknologi ini.

Dua aspek penting dari rekomendasi para pakar di Dewan TIK Nasional (DETIKNAS) yang telah dihasilkan dalam mempercepat pembangunan teknologi informasi dan komunikasi di tanah air. Dua aspek penting tersebut adalah *leadership* dimana kepemimpinan dalam TIK menuntut komitmen dari para pemimpin pemerintahan, baik dari jajaran pemerintah pusat maupun pemerintah daerah. Aspek kedua adalah infrastruktur informasi nasional dimana aspek ini diperlukan dalam proses pengambilan keputusan yang senantiasa didasarkan pada informasi yang relevan, terkini atau *real times*, akurat dan dapat dipertanggungjawabkan.

Menyadari akan pentingnya serta semangat untuk membangkitkan inovasi pembangunan TIK di tanah air, sangatlah tepat apabila pendeklarasian program *Indonesia, Go Open Source!* (IGOS) adalah merupakan upaya dalam meraih kemandirian dalam pemanfaatan teknologi informasi dan komunikasi yang ditopang dengan tersedianya berbagai pilihan terhadap perangkat lunak (*software*) yang pada akhirnya dapat dipergunakan oleh masyarakat. Hal ini juga sesuai dengan salah satu Tugas dari DETIKNAS yaitu “*Merumuskan kebijakan umum dan arahan strategis pembangunan nasional, melalui pemanfaatan teknologi informasi dan komunikasi (TIK)*”.

Panduan Pendayagunaan Open Source Software ini tentunya diharapkan dapat menjadi salah satu solusi dalam perumusan kebijakan umum dan arahan strategis dalam penyiapan *blue print* dan *road map* teknologi informasi dan

komunikasi di Indonesia. Dan dengan Buku Panduan ini dapat lebih memudahkan masyarakat dalam mengenal perangkat lunak berbasis open source. Buku panduan ini terdiri atas delapan dokumen yaitu *Perangkat Lunak Bebas dan Open Source*, *Petunjuk Instalasi IGOS Nusantara*, *Konfigurasi Server Linux*, *Aplikasi untuk Server*, *RDBMS (Relational Database Management System): MySQL*, *Bahasa Pemrograman Open Source*, *Aplikasi Perkantoran OpenOffice.org*, dan *CMS, CRM, dan ERP*.

Akhir kata, semoga kedelapan Dokumen ini dapat bermanfaat bagi peningkatan kemampuan masyarakat dalam penguasaan teknologi piranti lunak bebas dan open source.

Dengan semangat “**Kreativitas Tanpa Batas**” dan maju bersama *Indonesia, Go Open Source!* Kita besarkan produk aplikasi perangkat lunak Indonesia.

Jakarta, 2008

Ketua Harian Detiknas

Muhammad Nuh

DAFTAR ISI

PENGANTAR.....	iii
DAFTAR ISI.....	v
I. Pengenalan Aplikasi Server JBoss.....	1
1.1. Instalasi Jboss.....	1
1.1.1. Instalasi Java SDK.....	1
1.1.2. Variabel Environment.....	1
1.1.3. Instalasi Jboss paket jar.....	2
1.1.4. Un-instal Jboss paket instalasi	3
1.1.5. Instalasi Jboss paket zip.....	3
1.2. Administrasi JBoss	4
1.2.1. Struktur direktori instalasi Jboss	4
1.2.2. Menjalankan dan mematikan Jboss server.....	5
1.3. JDBC Data Source.....	7
1.4. Buat MySQL datasource.....	7
1.5. Aplikasi Web.....	8
1.6. Koneksi Database MySQL.....	10
1.7. Referensi.....	13
II. Integrasi Jboss-Tomcat dengan Apache web server.....	15
2.1. Instalasi web server Apache.....	15
2.2. Instal Aplikasi Jboss	16
2.3. Integrasi tomcat-Jboss dengan web server Apache.....	16
2.3.1. Download module modjk.....	16
2.3.2. Membuat module modjk.....	16
2.3.3. Konfigurasi file httpd.conf.....	16
2.3.4. Konfigurasi file mod-jk.conf.....	17
2.3.5. Konfigurasi file workers.properties	18
2.3.6. Konfigurasi file Jboss-server.xml.....	19
2.3.7. Akses aplikasi server Jbossmelalui web server apache.....	20
2.4. Clustering dan load Balancing.....	20
2.4.1. Edit filr workers.properties.....	22
2.4.2. Konfigurasi file server.xml.....	23

2.4.3. Konfigurasi file Jboss-service.xml.....	23
2.4.4. Jalankan server.....	24
2.4.5. Tes Aplikasi Server.....	24
2.4.6. Session replication.....	24
2.5. Referensi.....	25
III. Apache Tomcat.....	27
3.1. Apa itu Tomcat.....	27
3.2. Instalasi Tomcat.....	27
3.2.1. Instalasi JDK.....	27
3.2.2. Variabel Environment.....	28
3.2.3. Instalasi tomcat.....	28
3.2.4. Menjalankan dan mematikan tomcat.....	28
3.3. Tomcat direktori dan file.....	29
3.4. User manager dan Admin tomcat.....	29
3.5. Membuat Aplikasi Web.....	30
3.5.1. Ant.....	30
3.5.2. DirektoriAplikasi.....	31
3.5.3. File build.xml dan build.properties.....	32
3.5.4. File web.xml.....	33
3.5.5. Compile dan Deploy Aplikasi	33
3.6. Apache dan Tomcat	34
3.6.1. Apache paket RPM.....	34
3.6.2. Instalasi Apache dari Source.....	35
3.6.3. Membuat Module modjk.....	35
3.6.4. Konfigurasi file workers.properties.....	36
3.6.5. Konfigurasi file httpd.conf	36
3.6.6. Testing akses tomcat web server dan apache.....	37
3.7. Referensi.....	38

I. Pengenalan Aplikasi Server JBoss

Aplikasi Server JBoss tersedia dalam dua format binary dan code. Versi binary tersedia dalam bentuk format file zip dan tar.gz. Format file .jar juga tersedia untuk installasi JBoss.

Sebelum meng-install Jboss, pastikan mengecek sistem anda, apakah sudah terinstall Java-SDK. Jboss yang digunakan dalam tutorial ini adalah Jboss-4.0.5.GA dan menggunakan Java Virtual Machine 1.6 (Java-SDK 1.6).

1.1. Instalasi JBoss

1.1.1. Instalasi Java SDK

Sebagai super user (root) jalankan perintah berikut ini :

- a) Cek apakah ada Java lain yang terinstall

```
#rpm -qa | grep java
```

- b) Uninstall jika ada Java yang telah terinstall dalam sistem anda

```
#rpm -e gcc-java-3.2.3-20 --nodeps
```

- c) Download Java SDK versi terakhir, jdk-6-linux-i586.bin dari <http://java.sun.com>

- d) Install Java SDK

```
#sh jdk-6-linux-i586.bin
```

1.1.2. Variabel Environtment

Edit file /etc/profile dan tambahkan beberapa baris untuk mendefinisikan PATH dari aplikasi Java SDK yang telah terinstall dalam sistem anda.

```
JAVA_HOME="/opt/jdk1.6.0"
PATH="$JAVA_HOME/bin:$PATH"
export JAVA_HOME PATH
```

1.1.3. Instalasi Jboss paket jar

- a) Download jems-installer-1.2.0.GA.jar dari <http://www.jboss.org>
- b) Ketikan perintah berikut untuk instalasi paket jar :
`# java -jar jems-installer-1.2.0.GA.jar`

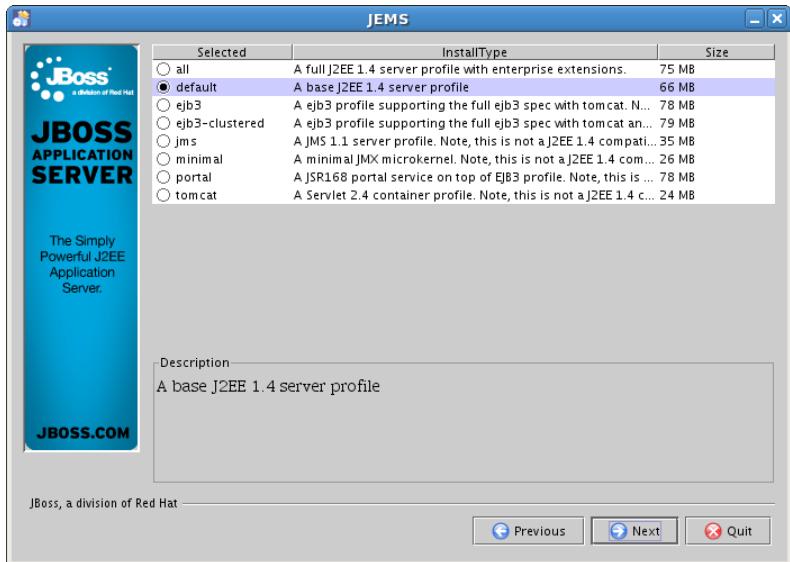
Akan muncul pilihan bahasa :



Gambar 1.0 : Pilihan Bahasa

- c) Ketik OK untuk memulai instalasi

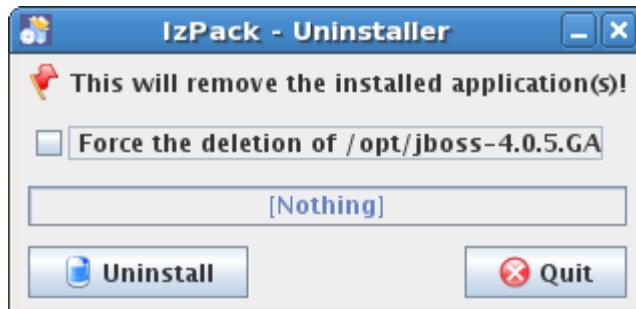
Dalam tahapan instalasi ini Anda diminta untuk menentukan lokasi instalasi server jboss, secara *default* akan diinstal di direktori **/usr/local/jboss-4.0.5.GA**, atau anda bisa tentukan sendiri direktori instalasi, misalnya di direktori **/opt/jboss-4.0.5.GA**.



Gambar 1.1 : Pilihan Paket Instalasi

1.1.4. Un-instal JBoss paket instalasi jar

- Masuk kedalam direktori instalasi Jboss
- Masuk kedalam direktori Uninstaller
- Jalankan perintah : **java -jar uninstaller.jar**



Gambar 1.2: Uninstall Jboss

1.1.5. Instalasi Jboss paket zip

- a) Download **JBoss-4.0.5.GA.zip** dari <http://www.jboss.org>
- b) Ekstrak file dalam lokasi yang anda inginkan, misalkan direktori **/opt**
#unzip JBoss-4.0.5.GA.zip
- c) Hasil ekstract file terdapat beberapa konfigurasi Jboss :
Jboss all
Jboss default
Jboss minimal

1.2. Administrasi Jboss

1.2.1. Struktur direktori Instalasi Jboss



Direktori	Keterangan
jboss-4.0.5.GA	JBOSS_HOME
/bin	Berisi skrip untuk menjalankan dan mematikan aplikasi server

Direktori	Keterangan
/client	Berisi kumpulan <i>library</i> aplikasi untuk <i>client</i>
/docs	Berisi dokumentasi jboss
/lib	Berisi kumpulan <i>library</i> jboss
/server	Berisi jenis konfigurasi server : <i>all</i> , <i>default</i> atau <i>minimal</i>
/default	Konfigurasi server default
/deploy	Direktory tempat aplikasi-aplikasi

1.2.2. Menjalankan dan Mematikan JBoss Server

Sebagai super user (root) sistem, jalankan perintah berikut :

- a) Menjalankan Server :

/opt/jboss-4.0.5.GA/bin/run.sh

Akan muncul informasi Startup dan konfigurasi Jboss Server

```
=====
JBoss_HOME: /opt/jboss-4.0.5.GA

JAVA: /opt/jdk1.6.0/bin/java

JAVA_OPTS: -Dprogram.name=run.sh -server -Xms128m -Xmx512m
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000

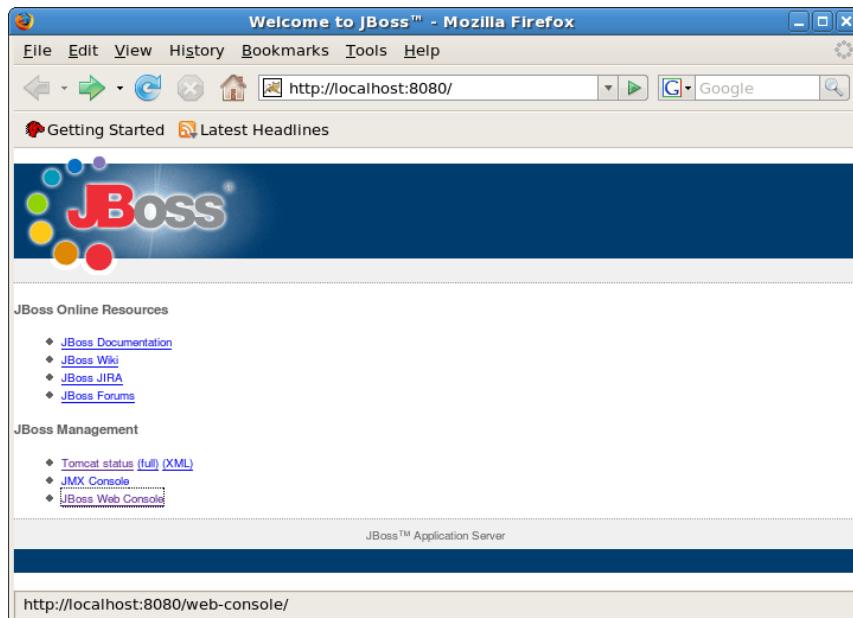
CLASSPATH: /opt/jboss-4.0.5.GA/bin/run.jar:/opt/jdk1.6.0/lib/tools.jar

=====
15:09:52,003 INFO [Server] Starting JBoss (MX MicroKernel)...
15:09:52,005 INFO [Server] Release ID: JBoss [Zion] 4.0.5.GA (build:
CVSTag=Branch_4_0 date=200610162339)
15:09:52,008 INFO [Server] Home Dir: /opt/jboss-4.0.5.GA
15:09:52,008 INFO [Server] Home URL: file:/opt/jboss-4.0.5.GA/
15:09:52,010 INFO [Server] Patch URL: null
```

```
15:09:52,011 INFO [Server] Server Name: default
15:09:52,011 INFO [Server] Server Home Dir: /opt/jboss-4.0.5.GA/server/default
15:09:52,012 INFO [Server] Server Home URL:
file:/opt/jboss-4.0.5.GA/server/default/
15:09:52,012 INFO [Server] Server Log
Dir: /opt/jboss-4.0.5.GA/server/default/log
15:09:52,012 INFO [Server] Server Temp
Dir: /opt/jboss-4.0.5.GA/server/default/tmp
15:09:52,013 INFO [Server] Root Deployment Filename: jboss-service.xml
15:09:53,275 INFO [ServerInfo] Java version: 1.6.0,Sun Microsystems Inc.
```

- b) Menjalankan Server dengan pilihan konfigurasi minimal :
/opt/jboss-4.0.5.GA/bin/run.sh -c minimal
- c) Mematikan Server :
/opt/jboss-4.0.5.GA/bin/shutdown.sh -S
- d) Halaman web jboss server :

Buka browser (misal : firefox), dan arahkan URL ke : <http://localhost:8080>



1.3. JDBC DataSource

- 1) Untuk melakukan koneksi database, kita harus menambahkan sebuah *datasource* pada Jbos
- 2) Buat sebuah database di MySQL berikut pemiliknya
Contoh datasource ada di dalam direktori :
/opt/jboss-4.0.5.GA/docs/example/jc
- 3) Copy file konfigurasi datasource **mysql-ds.xml** dan letakan ke dalam direktori **/opt/JBoss-4.0.5.GA/server/default/deploy**
- 4) Edit file konfigurasi datasource sesuai dengan variabel koneksi database yang digunakan

1.4. Buat MySQL datasource

- 1) Anda memerlukan file MySQL Connector untuk koneksi JDBC ke MySQL, dapat di download di :
<http://dev.mysql.com/downloads/connector/j/>
- 2) Ekstrak file **mysql-connector-java-3.1.12.zip**
- 3) Copy file JAR : **/mysql-connector-java-3.1.12/mysql-connector-java-3.1.12-bin.jar** dan diletakan dalam direktori
/opt/jboss-4.0.5.GA/server/default/lib/

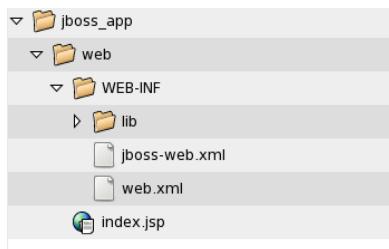
Berikut isi file mysql-ds.xml :

```
<!-- Nama Unik JNDI -->
<jndi-name>MySqlDS</jndi-name>
<!-- URL koneksi database -->
<connection-url>jdbc:mysql://localhost:3306/databasiku</connection-url>
<!-- Driver Class -->
<driver-class>com.mysql.jdbc.Driver</driver-class>
<!-- Nama user database -->
<user-name>rojul</user-name>
<!-- Password user database -->
<password>r4h4s14</password>
```

1.5. Aplikasi Web

Aplikasi web JSP (Java Server Pages)

- 1) Buat struktur direktori :



Direktori	Keterangan
jboss_app	ROOT_APLIKASI
/web	Berisi file aplikasi web (jsp, html)
/WEB-INF	Berisi file-file konfigurasi (*.xml), dan binary class
/lib	Berisi file jar library aplikasi

- 2) Buat file **web.xml** dan letakan dalam direktori **jboss_app/web/WEB-INF**

Berikut isi file web.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <display-name>Welcome To My Web</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

- 3) Buat file **jboss-web.xml** dan letakan dalam direktori
jboss_app/web/WEB-INF/

Berikut isi file jboss-web.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
    <context-root>myweb</context-root>
    <context-priority-classloader>
        false
    </context-priority-classloader>
</jboss-web>
```

- 4) Buat file **index.jsp**, dan letakan dalam direktori **jboss_app/web** .

Berikut isi file index.jsp :

```
<h1>Selamat Datang</h1>
<%!
    java.util.Calendar cal = java.util.Calendar.getInstance();
%>
Sekarang Tanggal : <%= cal.getTime() %>
```

- 5) Buat file web archive (war) .

Masuk kedalam direktori jboss_app/web, dan jalankan perintah berikut :
jar cvf webtes.war .

```
[web]$ jar cvf webtes.war .
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/jboss-web.xml(in = 197) (out= 122)(deflated 38%)
adding: WEB-INF/web.xml(in = 406) (out= 228)(deflated 43%)
adding: WEB-INF/lib/(in = 0) (out= 0)(stored 0%)
adding: index.jsp(in = 135) (out= 107)(deflated 20%)
```

- 6) Copy file web archive **webtes.war** ke dalam direktori
/opt/jboss-4.0.5.GA/server/default/deploy
- 7) Buka browser anda dan arahkan ke URL: <http://localhost:8080/myweb>



Gambar 1.4 : Aplikasi Salam di Tomcat - JBoss Server

1.6. Koneksi Database MySQL

- 1) Buat table pada database yang telah didefinisikan sesuai dengan **mysql-ds.xml**,

Misal telah dibuat table member dengan skema berikut ini :

```
mysql> desc member;
+-----+-----+-----+-----+
| Field | type   | Null | Key | Default | Extra      |
+-----+-----+-----+-----+
| id   | int(11) | NO  | PRI | NULL    | auto_increment |
| nama | varchar(32)| YES |     | NULL    |             |
| email| varchar(32) | YES |     | NULL    |             |
+-----+-----+-----+-----+
```

- 2) Membuat Koneksi ke datasource

Akses datasource yang telah didefinisikan pada **mysql-ds.xml**

```
Connection con = null;
```

```
InitialContext ctxt = new InitialContext();
DataSource ds = (DataSource) ctxt.lookup("java:MySqlIDS");
con = ds.getConnection();
```

- 3) Berikut program JSP untuk menampilkan data dari table, Misal nama file **koneksi.jsp**

```
<%@page import="java.sql.Connection" %>
<%@page import="java.sql.ResultSet" %>
<%@page import="java.sql.Statement" %>
<%@page import="javax.sql.DataSource" %>
<%@page import="javax.naming.InitialContext" %>
<%
    Connection con = null;
    InitialContext ctxt = new InitialContext();
    DataSource ds = (DataSource) ctxt.lookup("java:MySqlIDS");
    con = ds.getConnection();

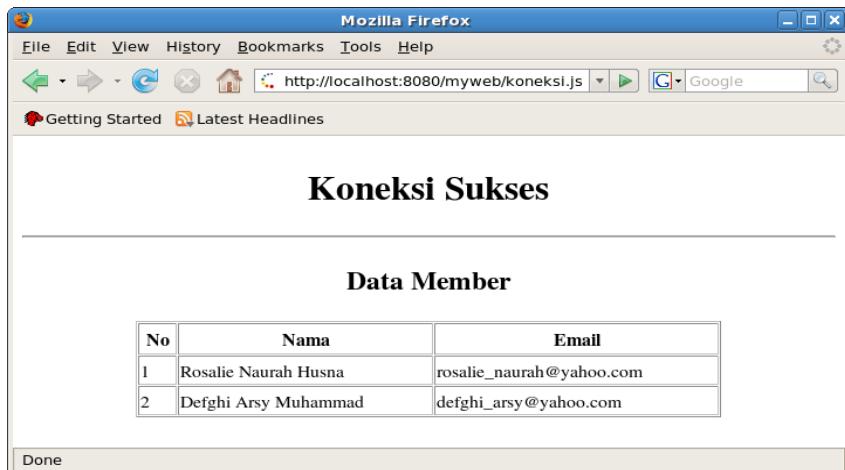
    if (con == null )
    {
%>
        <h1>Gagal Koneksi</h1>
<%      }
        else
        {
%>
        <center>
            <h1>Koneksi Sukses</h1>
            <hr>
            <h2>Data Member</h2>
<%
    ResultSet rs = null;
    String sql = "SELECT * FROM member";
    Statement st = con.createStatement();
    rs = st.executeQuery( sql );
%>
```

```

<table border="1" width="500px" cellspacing="1" cellpadding="1">
  <tr><th>No</th><th>Nama</th><th>Email</th></tr>
<%
  int num = 1;
  while (rs.next())
  {
<%
    <tr><td><%= num %></td>
    <td><%= rs.getString("nama") %></td>
    <td><%= rs.getString("email") %></td></tr>
<%
    num++;
  }
<%
</table>
</center>
<% } %>

```

- 4) Buat file web archive dan **deploy** ke JBoss server, dan akses koneksi.jsp.



1.7. Referensi

- 1) Jboss.ORG: <http://www.jboss.org>
- 2) Apache Software Foundation : <http://www.apache.org>

II. Integrasi Jboss-Tomcat dengan Apache Web Server

2.1. Instalasi Web Server Apache

Pastikan telah terinstal web server Apache (HTTP server) dalam komputer server anda, jika belum terinstal, versi terakhir web server Apache dapat di download di website <http://www.apache.org/>.

Jika web server Apache yang telah terinstal di komputer server anda adalah menggunakan paket rpm, maka pastikan paket source atau development juga telah terinstal. Anda bisa mengecek dengan menjalankan perintah : **rpm -qa | grep httpd-devel**.

Paket source web server apache (*httpd-devel*) berisi tools development APXS dari web server Apache yang diperlukan pada saat kompilasi **modjk**.

Pada tulisan ini, web server apache yang digunakan adalah hasil *compile* dari *source*.

- 1) Download Apache web server terakhir (*httpd-2.2.4.tar.gz*) dari
<http://httpd.apache.org/download.cgi>
- 2) Ekstrak hasil download dalam direktori **/usr/local**
tar -xvzf httpd-2.2.4.tar.gz
- 3) Masuk ke dalam direktori hasil ekstrak, dan lakukan konfigurasi instal apache web server
./configure --prefix=/usr/local/apache2 --enable-so
Apache akan di instal dalam direktori /usr/local/apache2
- 4) Jalankan perintah *make* untuk compile source apache
make
- 5) Jalankan perintah *make install* untuk instalasi apache
make install
- 6) Aktifkan server apache hasil instalasi dengan jalankan :
/usr/local/apache2/bin/apachectl start
- 7) Untuk mematikan serer apache jalankan perintah berikut :
/usr/local/apache2/bin/apachectl stop

2.2. Instal Aplikasi JBoss

- 1) Download aplikasi server Jboss, tersedia di <http://www.jboss.org/>
- 2) Jboss membutuhkan Java, download Java SDK atau JRE dari <http://java.sun.com/>

2.3. Integrasi Tomcat-JBoss dengan web server Apache

Berikut ini tahap-tahap untuk meng-integrasikan Jboss yang didalamnya terdapat aplikasi server tomcat dengan web server Apache.

2.3.1. Download module modjk (tomcat connector)

Download module modjk dari website Tomcat-Apache (<http://tomcat.apache.org>), dalam penulisan tutorial ini menggunakan tomcat-connectors versi terakhir, yaitu **tomcat-connectors-1.2.23-src.tar.gz**.

2.3.2. Membuat module modjk

- 1) Ekstrak file source **tomcat-connectors-1.2.23-src.tar.gz**.
\$ tar -xvzf tomcat-connectors-1.2.23-src.tar.gz
- 2) Masuk kedalam direktori hasil ekstrak
\$ cd tomcat-connectors-1.2.23/native
- 3) Compile source : pastikan file apxs dari web server apache, anda telah ketahui. Dalam penulisan ini web server Apache hasil instalasi dari source terletak dalam direktori : **/usr/local/apache2**, dan APXS (file apxs) terletak dalam direktori **/usr/local/apache2/bin**
- 4) Jalankan perintah berikut ini untuk memulai meng-compile modjk
\$./configure --with-apxs=/usr/local/apache2/bin/apxs
\$ make
\$ su -c make install
- 5) Hasil instal *tomcat-connectors* akan menghasilkan file **mod_jk.so**, dan file ini langsung diletakan dalam : **/usr/local/apache2/modules**

2.3.3. Konfigurasi file httpd.conf

Edit file **httpd.conf**, dan tambahkan baris berikut di akhir file, agar web server Apache dapat me-load module JK yang diperlukan untuk integrasi Jboss-Tomcat dengan web server Apache

```
#include mod-jk specific configuration file  
Include conf/mod-jk.conf
```

File **httpd.conf** terletak dalam direktori **/usr/local/apache2/conf/** atau jika httpd atau web server Apache di install lewat paket rpm, file httpd.conf terletak dalam direktori **/etc/httpd/conf**.

2.3.4. Konfigurasi file mod-jk.conf

Buat file **mod-jk.conf** dan letakan dalam direktori **/usr/local/apache2/conf** atau jika instal lewat paket rpm letakan di **/etc/httpd/conf**.

Berikut isi file mod-jk.conf :

```
# Load mod_jk module  
# Specify the filename of the mod_jk lib  
LoadModule jk_module modules/mod_jk.so  
  
# Where to find workers.properties  
JkWorkersFile conf/workers.properties  
  
# Where to put jk logs  
JkLogFile logs/mod_jk.log  
  
# Set the jk log level [debug/error/info]  
JkLogLevel info  
  
# Select the log format  
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"  
  
# JkOptions indicates to send SSKEY SIZE  
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
```

```
# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
#JkMount /application/* loadbalancer

#JkMout /nama_aplikasi_dalam_jboss_tomcat_server
JkMount /myweb/* loadbalancer
#JkMount/*.jsp loadbalancer
#JkMount/*.html loadbalancer

# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
#JkMountFile conf/uriworkermap.properties

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
# JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
    JkMount status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```

2.3.5. Konfigurasi file workers.properties

Dalam direktori yang sama dengan **mod-jk.conf** buat file **workers.properties**,

berikut isi filenya :

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=[Isi dengan No IP atau hostname]
worker.node1.type=ajp13
worker.node1.lbfactor=1
#worker.node1.local_worker=1 (1)
worker.node1.cachesize=10
#ndex.jspworker.node1.cachesize=10

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1
worker.loadbalancer.sticky_session=1
#worker.loadbalancer.local_worker_only=1
#worker.list=loadbalancer

# Status worker for managing load balancer
worker.status.type=status
```

2.3.6. Konfigurasi file jboss-server.xml

Edit file **jboss-server.xml** yang terletak dalam

[root_jboss]/server/default/deploy/jbossweb-tomcat55.sar/META-INF.

Ubahlah pada tag atribut “UseJK” dari false menjadi true

```
<attribute name="UseJK">false</attribute>
```

ubah menjadi

```
<attribute name="UseJK">true</attribute>
```

2.3.7. Akses aplikasi server Jboss melalui web server Apache

Setelah aplikasi server JBoss dan web server Apache dinyalakan kembali, dan buat aplikasi web (web archive) dan letakan dalam JBoss server, kemudian akses aplikasi melalui web server Apache.



Gambar 1.0 : Aplikasi Salam

Untuk lebih mengetahui lebih detail tentang aplikasi web dengan menggunakan aplikasi server JBoss, bisa digunakan panduan manual yang disediakan oleh jboss.org yaitu "*Jboss Installation And Administration*".

2.4. Clustering dan Load Balancing

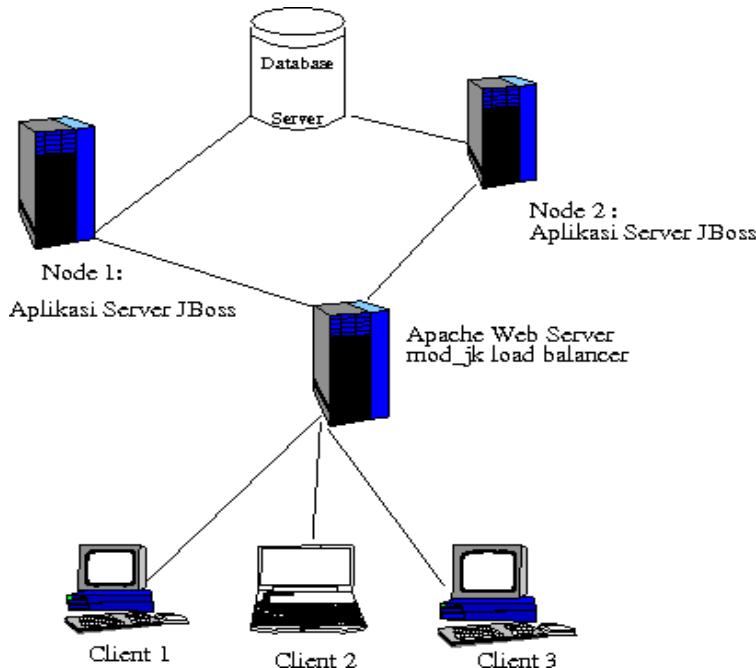
Anda dapat membuat dua atau lebih server aplikasi JBoss menjadi sebuah server (*single server*) yang saling membagi akses (*load*), hal ini dinamakan *cluster JBoss*. Dalam cluster Jboss terdapat beberapa aplikasi server Jboss yang dinamakan *node*.

Masing-masing node dapat berjalan dalam sebuah jaringan komputer seperti LAN, misalnya ingin membuat dua aplikasi server jboss yang bekerja pada server yang berbeda IP atau hostname, dan masing-masing node dapat diakses secara bergantian melalui sebuah server cluster, berdasarkan load / beban akses sebuah aplikasi dalam

masing-masing node.

Tujuan utama dari clustering service HTTP adalah untuk replikasi dari session, dimana session ini dilakukan oleh Jboss itu sendiri, sedangkan untuk load balancing harus di implementasikan oleh software lain.

Module mod_jk pada web server Apache berperan sebagai software untuk load balancing.



Gambar 1.1: JBoss Cluster dan Load Balancing dengan mod_jk

Agar bisa sebuah **mod_jk** pada web server Apache menjalankan fungsi load balancer, anda hanya mengubah step ke 5 (3.6) pada integrasi JBoss-Tomcat server dengan web server Apache.

Berikut langkah-langkahnya :

2.4.1. Edit file workers.properties

Edit file *workers.properties* seperti baris berikut ini.

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=[Isi dengan IP atau Hostname Node 1]
worker.node1.type=ajp13
worker.node1.lbfactor=1
#worker.node1.local_worker=1 (1)
worker.node1.cachesize=10

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=[Isi dengan IP atau Hostname Node 2]
worker.node2.type=ajp13
worker.node2.lbfactor=1
#worker.node2.local_worker=1 (1)
worker.node2.cachesize=10

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
#worker.loadbalancer.local_worker_only=1
#worker.list=loadbalancer

# Status worker for managing load balancer
worker.status.type=status
```

2.4.2. Konfigurasi file server.xml

Pada setiap node JBoss cluster, edit file **server.xml** pada direktori :
JBOSS_HOME/server/all/deploy/jbossweb-tomcat55.sar/

Ubah pada tag *<Engine>* dengan memasukan atribut *jvmRoute* :
dengan isi sesuai dengan alamat IP atau Hostname yang telah didefinisikan
dalam server load balancer (pada file **workers.properties**).

Pada node 1 :

```
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="node1">
```

Pada node 2 :

```
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="node2">
```

2.4.3. Konfigurasi file jboss-service.xml

Pada masing-masing node, edit file **jboss-server.xml** yang terletak dalam :
[root_jboss]/server/default/deploy/jbossweb-tomcat55.sar/META-INF

Ubahlah pada tag atribut “UseJK” dari false menjadi true

```
<attribute name="UseJK">false</attribute>
```

ubah menjadi

```
<attribute name="UseJK">true</attribute>
```

Lakukan hal yang sama pada file **jboss-server.xml** yang terletak dalam
[root_jboss]/server/all/deploy/jbossweb-tomcat55.sar/META-INF

Ubahlah pada tag atribut “UseJK” dari false menjadi true

```
<attribute name="UseJK">false</attribute>
```

Ubah menjadi

```
<attribute name="UseJK">true</attribute>
```

2.4.4. Jalankan Server

- 1) Aktifkan server aplikasi JBoss pada masing-masing node dengan perintah :
`# /JBoss_HOME/bin/run.sh -c all`
- 2) Aktikan atau restart service web server apache pada server load balancer
`# /usr/local/apache2/bin/apachectl start`
atau kalau anda menggunakan paket httpd rpm
`# service httpd start`

2.4.5. Tes Aplikasi Server

Untuk menguji apakah server menjalankan load balancer, buatlah sebuah aplikasi dan bundle dalam sebuah file web archive misalkan **webtes.war**, dimana didalamnya telah dibuat sebuah file **test.jsp**.

```
<!-- test.jsp dalam node 1 -->
<% out.print("Node 1" ); %>

<!-- test.jsp dalam node 1 -->
<% out.print("Node 1" ); %>
```

Dan deploy aplikasi dengan meletakan file webtes.war pada direktori :

JBOSS_HOME/server/all/deploy/

Selanjutnya anda bisa akses aplikasi myweb pada web server Apache , dengan alamat `http://IP_WEB_SERVER/myweb/tes.jsp`. Dan bisa mengecek apakah Apache telah menjalankan *load-balancing* dengan benar.

2.4.6. Session Replication

Berikut ini adalah konfigurasi optional, tambahkan baris berikut pada file **web.xml** aplikasi anda agar session replikasi berjalan.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">  
<!-- tambahkan baris ini -->  
<distributable/>  
  
<display-name>Welcome To My Web</display-name>  
<welcome-file-list>  
    <welcome-file>index.jsp</welcome-file>  
</welcome-file-list>  
</web-app>
```

Anda juga bisa mengatur konfigurasi session replication dengan mengedit file **jboss-web.xml** dan gunakan konfigurasi session replication berikut ini :

```
<?xml version="1.0" encoding="UTF-8"?>  
<jboss-web>  
    <replication-config>  
        <replication-trigger>  
            SET_AND_NON_PRIMITIVE_GET  
        </replication-trigger>  
        <replication-granularity>SESSION</replication-granularity>  
    </replication-config>  
</jboss-web>
```

2.5. Referensi

- 1) Jboss.ORG: <http://www.jboss.org>
- 2) The Apache Software Foundation : <http://www.apache.org>

III. Apache Tomcat

3.1. Apa itu Tomcat

Tomcat adalah sebuah aplikasi web server untuk memproses Servlet/JSP (*Servlet/JSP container*). Tomcat dikembangkan dan dikekola oleh Apache Sofware Foundation. Servlet atau JSP (Java Server Pages) sendiri adalah sebuah program yang ditulis dengan bahasa pemrograman Java.

Web server Apache tidak secara otomatis bisa memproses suatu halaman dinamik yang mengandung program Servlet atau JSP, namun dengan melakukan konfigurasi web server Apache, program Servlet/JSP yang diproses oleh Tomcat dapat di forward ke web server Apache.

Tidak seperti apache yang dikembangkan dengan C/C++, Tomcat dikembangkan dengan bahasa pemrograman Java, karenanya untuk instalasi Tomcat diperlukan Java. Tomcat yang digunakan dalam tulisan ini adalah Apache Tomcat versi 5.5 yang menggunakan spesifikasi Servlet 2.4 dan Java Server Pages 2.0 yang dikeluarkan oleh Java (java.sun.com).

3.2. Instalasi Tomcat

Aplikasi Tomcat tersedia dalam format kode program (source) atau format binary. Dalam tulisan ini untuk instalasi Tomcat digunakan format binary.

3.2.1. Instalasi JDK

Sebagai super user (root) jalankan perintah berikut ini :

- 1) Cek apakah ada JDK lain yang terinstall

```
#rpm -qa | grep java
```

- 2) Uninstall jika ada JDK yang telah terinstall dalam sistem anda

```
#rpm -e gcc-java-3.2.xxx --nodeps
```

- 3) Download JDK versi terakhir, **jdk-6-linux-i586.bin** dari <http://java.sun.com> Install JDK

```
#sh jdk-6-linux-i586.bin
```

3.2.2. Variabel Environment

Edit file **/etc/profile** dan tambahkan beberapa baris untuk mendefinisikan PATH dari aplikasi JDK yang telah terinstall dalam sistem anda.

```
JAVA_HOME="/opt/jdk1.6.0"  
PATH="$JAVA_HOME/bin:$PATH"  
export JAVA_HOME PATH
```

3.2.3. Instalasi Tomcat

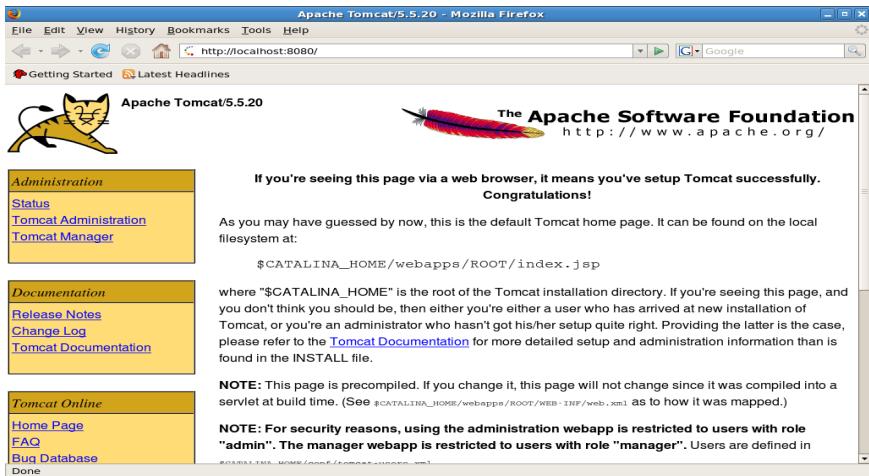
- 1) Download apache tomcat versi stabil dari : <http://tomcat.apache.org/>
- 2) Ekstrak paket tar hasil download dalam direktori **/opt**
#tar -xvzf apache-tomcat-5.5.20.tar.gz
- 3) Hasil ekstrak akan menghasilkan direktori apache-tomcat-5.5.20, untuk mempermudah mengingat versi tomcat, ganti nama direktori hasil ekstrak dengan tomcat552
#mv apache-tomcat-5.5.20 tomcat552

3.2.4. Menjalankan dan Mematikan Tomcat

- 1) Menjalankan tomcat :
/opt/tomcat552/bin/startup.sh

```
# /opt/tomcat552/bin/startup.sh  
Using CATALINA_BASE: /opt/tomcat552  
Using CATALINA_HOME: /opt/tomcat552  
Using CATALINA_TMPDIR: /opt/tomcat552/temp  
Using JRE_HOME: /opt/jdk1.6.0
```

- 2) Mematikan tomcat :
/opt/tomcat552/bin/shutdown.sh
- 3) Melihat halaman tomcat manager :
Buka browser anda dan arahkan URL ke : <http://localhost:8080>



Gambar 1.0 : Tomcat Web Server

3.3. Tomcat direktori dan file

Dalam direktori tomcat (**/opt/tomcat552**) terdapat direktori :

- 1) **/bin** : direktori berisi skrip file untuk mengaktifkan web server tomcat, (*startup.sh* dan *shutdown.sh*).
- 2) **/conf** : direktori ini berisi file-file konfigurasi tomcat
- 3) **/logs** : berisi file-file log
- 4) **/webapps** : berisi tempat direktori-direktori aplikasi

3.4. User Manager dan Admin Tomcat

Edit file **tomcat-user.xml** dalam direktori **/opt/tomcat552/conf**, Misalkan ditentukan bahwa user **rosa** dengan password **ros4** sebagai manager dan admin tomcat server.

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="manager"/>
    <role rolename="admin"/>
    <user username="rosa" password="ros4" roles="admin,manager"/>
```

```
</tomcat-users>
```

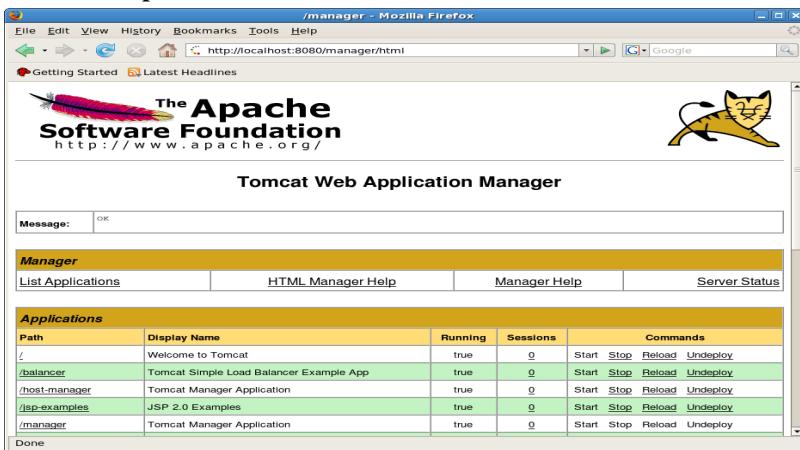
Lakukan *restart* tomcat server, kemudian buka browser anda dengan URL:
<http://localhost:8080>

untuk masuk ke halaman manager klik link Tomcat Manager

Gambar 1.1 : Tomcat Manager

Dalam halaman Tomcat Manager, terdapat informasi aplikasi-aplikasi yang berada dalam tomcat server, user yang mempunyai role “manager” dapat mengatur aplikasi (*start,stop,reload, undeploy*) atau men-*deploy* aplikasi.

3.5. Membuat Aplikasi Web



3.5.1. Ant

Aplikasi ant dibutuhkan untuk membantu dalam proses *compile* program, Ant adalah aplikasi berbasis XML, ant dikembangkan dan dikelola oleh Apache Software Foundation.

- 1) Download source ant (**apache-ant-1.7.0-bin.zip**) di :
<http://ant.apache.org/> dan unzip file hasil download dalam direktori /opt
unzip apache-ant-1.7.0-bin.zip
- 2) Ubah nama direktori ant untuk memudahkan penggunaanya,

```
# mv apache-ant-1.7.0 ant170
```

- 3) Edit file `/etc/profile`, dan tambahkan PATH aplikasi ant.

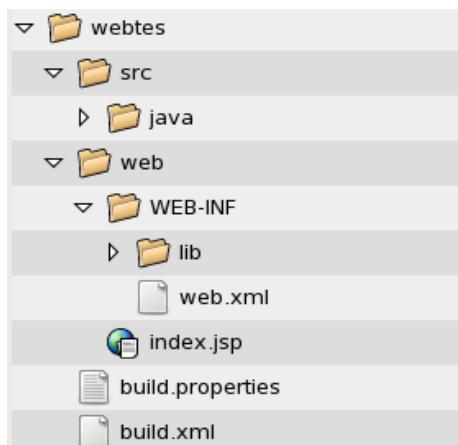
```
ANT_HOME="/opt/ant170"  
PATH="$ANT_HOME/bin:$PATH"  
export ANT_HOME PATH
```

- 4) Jalankan perintah berikut untuk mengaktifkan perubahan file `/etc/profile`:

```
$ source /etc/profile
```

3.5.2. Direktori Aplikasi

Misalkan direktori webtes adalah ROOT_APLIKASI yang ingin kita buat, dan buatlah direktori lainnya dalam direktori webtes seperti gambar berikut :



Direktori	Keterangan
webtes	ROOT_APLIKASI
/web	Berisi file aplikasi web (jsp, html)
/WEB-INF	Berisi file-file konfigurasi (*.xml), dan binary class

Direktori	Keterangan
/lib	Berisi file jar library aplikasi
/src	Berisi file-file program Java (Servlet)

3.5.3. File build.xml dan build.properties

- 1) Copy file build.xml dari template pada direktori Tomcat Server.
File **build.xml** digunakan untuk konfigurasi pada saat compile dengan Ant, file template **build.xml** bisa dicopy dari direktori tomcat :
/opt/tomcat552/webapps/tomcat-docs/appdev/build.xml.txt
dan diletakan dalam direktori : **ROOT_APLIKASI/build.xml**
- 2) Buat file **build.properties**
File build.properties digunakan sebagai konfigurasi aplikasi pada saat proses kompilasi dengan ant, file diletakan dalam direktori **ROOT_APLIKASI**.
Berikut isi file build.properties

```
#nama aplikasi dan versi aplikasi
app.name=webtes
app.version=1.0

#Letak tomcat server
catalina.home=/opt/tomcat552

#manager tomcat server
manager.username=rosa
manager.password=ros4
manager.url=http://localhost:8080/manager
```

- 3) Edit file build.xml

Gantilah baris berikut pada file build.xml :

```
<!--property name="catalina.home" value="..../.."<!----&gt; &lt;!-- UPDATE THIS!
--&gt;</pre>

```

dengan

```
<property name="catalina.home" value="${catalina.home}" />
```

3.5.4. File web.xml

File **web.xml** digunakan untuk konfigurasi aplikasi web, seperti mendaftarkan file servlet, mendaftarkan varibel konfigurasi koneksi database dan sebagainya.

Berikut isi file web.xml, dalam file ini didefinisikan file index.jsp yang akan pertama kali akan diakses oleh aplikasi.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>Welcome To My Web</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Berikut isi file index.jsp :

```
<h1>Selamat Belajar Web Java</h1>
<%!
  java.util.Calendar cal = java.util.Calendar.getInstance();
%>
Sekarang Tanggal : <%= cal.getTime() %>
```

3.5.5. Compile dan Deploy Aplikasi

Anda dalam direktori ROOT_APLIKASI

- 1) Compile source :

ant

- 2) Deploy aplikasi ke tomcat server

- ```
ant install
```
- 3) Undeploy aplikasi dari tomcat server

```
ant remove
```

  - 4) Task-task yang ada dalam file build.xml

```
ant -projecthelp
```



Gambar 1.2 : Aplikasi Web dengan Tomcat

### 3.6. Apache dan Tomcat

Server Tomcat secara default berjalan di port 8080, sedangkan Apache berjalan secara default pada port 80. Untuk bisa mengakses Server Tomcat, kita harus secara jelas menyebutkan port pada URL (<http://localhost:8080/>). Namun demikian kita bisa menggabungkan Tomcat dan Apache Web Server sehingga user tidak perlu menuliskan lagi port Tomcat, yaitu dengan mengarahkan request pada Tomcat ke Apache Web Server.

#### 3.6.1. Apache Paket RPM

Jika web server Apache (HTTP Server) yang telah terinstall di komputer server anda adalah menggunakan paket rpm, maka pastikan paket source development juga telah terinstall. Anda bisa mengecek dengan jalankan perintah : **rpm -qa | grep httpd-devel**. Jika belum terinstall saatnya Anda install paket development Apache Web Server (httpd-devel).

Paket source web server apache berisi tools development APXS dari web server Apache yang diperlukan pada saat kompilasi modjk

### 3.6.2. Instalasi Apache dari Source

Pada tulisan ini, web server apache yang digunakan adalah hasil *compile* dari *source*

- 1) Download Apache web server versi terakhir (`httpd-2.2.4.tar.gz`) dari <http://httpd.apache.org/download.cgi>
- 2) Ekstrak hasil download dalam direktori **/usr/local**  
`# tar -xvzf httpd-2.2.4.tar.gz`
- 3) Masuk kedalam direktori hasil ekstrak, dan lakukan konfigurasi instal apache web server  
`# ./configure --prefix=/usr/local/apache2 --enable-so`  
Apache akan di instal dalam direktori **/usr/local/apache2**
- 4) Jalankan perintah make untuk compile source apache  
`# make`
- 5) Jalankan perintah make install untuk instal apache  
`# make install`
- 6) Aktifkan server apache hasil instal dengan jalankan :  
`# /usr/local/apache2/bin/apachectl start`
- 7) Untuk mematikan server apache jalankan perintah berikut :  
`# /usr/local/apache2/bin/apachectl stop`

### 3.6.3. Membuat module modjk

- 1) Download module modjk dari website apache tomcat (<http://tomcat.apache.org>), dalam penulisan tutorial ini menggunakan tomcat-connectors versi terakhir, yaitu tomcat-connectors-1.2.23-src.tar.gz.
- 2) Ekstrak file source tomcat-connectors-1.2.23-src.tar.gz.  
`$ tar -xvzf tomcat-connectors-1.2.23-src.tar.gz`
- 3) Masuk kedalam direktori hasil ekstrak  
`$ cd tomcat-connectors-1.2.23/native`
- 4) Compile source : pastikan file apxs dari web server apache, anda telah ketahui. Dalam penulisan ini web server Apache hasil instalasi dari source terletak dalam direktori : **/usr/local/apache2**, dan APXS (file apxs) terletak dalam direktori **/usr/local/apache2/bin/**

- 5) Jalankan perintah berikut ini untuk memulai meng-compile modjk  
    \$ ./configure --with-apxs=/usr/local/apache2/bin/apxs  
    \$ make  
    \$ su -c make install
- 6) Hasil instalasi akan menghasilkan file mod\_jk.so, dan file ini langsung diletakan dalam : /usr/local/apache2/modules

### 3.6.4. Konfigurasi file workers.properties

Dalam direktori /usr/local/apache2/conf buat file **workers.properties**, berikut isi filenya :

```
Define list of workers that will be used

workers.tomcat_home=$CATALINA_HOME
workers.java_home=$JAVA_HOME
ps=/
worker.list=default

worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
worker.default.lbfactor=1
```

### 3.6.5. Konfigurasi file httpd.conf

Edit file httpd.conf

- 1) Tambahkan baris berikut pada bagian LoadModule

```
LoadModule jk_module modules/mod_jk.so
```

- 2) Pastikan port httpd berjalan dalam port 80
- 3) Tambahkan baris berikut sebelum bagian Virtual Host

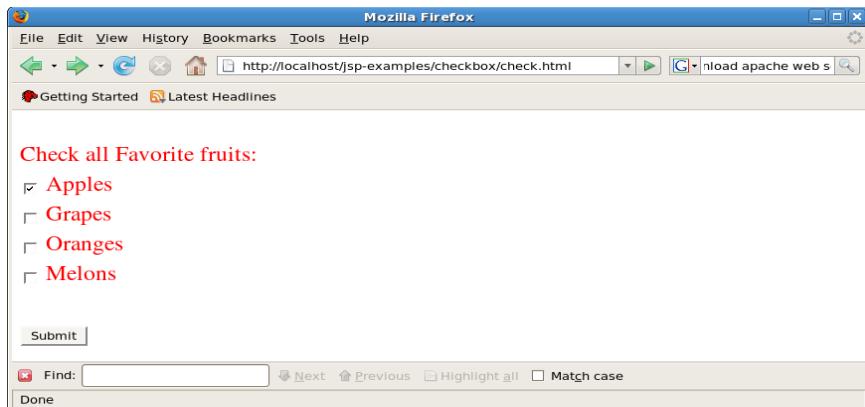
```
JkWorkersFile "conf/workers.properties"
```

```
JkLogFile "logs/mod_jk.log"
JkLogLevel error

aplikasi pada tomcat yang ingin diakses
JkMount /webtes default
JkMount /webtes/* default
JkMount /jsp-examples default
JkMount /jsp-examples/* default
```

### 3.6.6. Testing Akses Tomcat Web Server dan Apache

- 1) Jalankan Tomcat  
**# /opt/tomcat552/bin/startup.sh**
- 2) Jalankan Apache web server  
**# /usr/local/apache2/bin/apachectl start**
- 3) Buka browser anda, dan arahkan URL ke <http://localhost> dan <http://localhost:8080> untuk memastikan dua server berjalan dengan baik
- 4) Buka browser dan arahkan URL ke <http://localhost/jsp-examples> , web server apache akan secara langsung mengarahkan ke tomcat server.



Gambar 1.3 : Aplikasi Java Web diakses lewat Apache Web Server

### **3.7. Referensi**

- 1) Jakarta Apache : <http://jakarta.apache.org>
- 2) Jboss.ORG: <http://www.jboss.org>
- 3) The Apache Software Foundation : <http://www.apache.org>