**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/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.p4.wismabr</groupId>  <artifactId>wisma-br</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>war</packaging>  <name>wisma-br</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <org.springframework.security.version>3.1.4.RELEASE</org.springframework.security.version>  <org.springframework.version>3.2.3.RELEASE</org.springframework.version>  </properties>  <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.11</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>opensymphony</groupId>  <artifactId>sitemesh</artifactId>  <version>2.4.2</version>  </dependency>  <dependency>  <groupId>jstl</groupId>  <artifactId>jstl</artifactId>  <version>1.1.2</version>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jstl</artifactId>  <version>1.2</version>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>javax.servlet-api</artifactId>  <version>3.0.1</version>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.23</version>  </dependency>  <dependency>  <groupId>commons-dbcp</groupId>  <artifactId>commons-dbcp</artifactId>  <version>1.4</version>  </dependency>    <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <version>3.2.2.RELEASE</version>  </dependency>    <!-- spring security-->  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-core</artifactId>  <version>3.1.4.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-web</artifactId>  <version>3.1.4.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-config</artifactId>  <version>3.1.4.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-taglibs</artifactId>  <version>3.1.4.RELEASE</version>  </dependency>    <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>3.2.1.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  <version>3.2.2.RELEASE</version>  </dependency>    <dependency>  <groupId>org.hibernate</groupId>  <artifactId>hibernate-validator</artifactId>  <version>4.3.1.Final</version>  </dependency>  <!-- QuartzJobBean in spring-context-support.jar and support jesper report -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  <version>3.2.2.RELEASE</version>  </dependency>  <!-- Spring + Quartz need transaction -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  <version>3.2.2.RELEASE</version>  </dependency>  <!-- quartz scheduler -->  <dependency>  <groupId>org.quartz-scheduler</groupId>  <artifactId>quartz</artifactId>  <version>1.8.6</version>  </dependency>  <!-- untuk xls-->  <dependency>  <groupId>org.apache.poi</groupId>  <artifactId>poi</artifactId>  <version>3.9</version>  <type>jar</type>  <scope>compile</scope>  </dependency>  <!-- Jasper -->  <dependency>  <groupId>net.sf.jasperreports</groupId>  <artifactId>jasperreports</artifactId>  <version>5.1.0</version>  </dependency>  <!-- Dependency required by Jasper -->  <dependency>  <groupId>commons-digester</groupId>  <artifactId>commons-digester</artifactId>  <version>2.1</version>  </dependency>    <!-- Dependency required by Jasper -->  <dependency>  <groupId>commons-collections</groupId>  <artifactId>commons-collections</artifactId>  <version>3.2.1</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aop</artifactId>  <version>3.2.3.RELEASE</version>  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <artifactId>maven-war-plugin</artifactId>  <version>2.3</version>  <configuration>  <webXml>src/main/webapp/WEB-INF/web.xml</webXml>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>2.0.2</version>  <configuration>  <source>1.6</source>  <target>1.6</target>  </configuration>  </plugin>  </plugins>  </build>  </project> |

**web.xml**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">    <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>  classpath\*:config-db-spring.xml,  classpath\*:config-quartz.xml,  classpath\*:spring-security.xml  </param-value>  </context-param>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>    <filter>  <filter-name>sitemesh</filter-name>  <filter-class>com.opensymphony.sitemesh.webapp.SiteMeshFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>sitemesh</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <filter>  <filter-name>springSecurityFilterChain</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>springSecurityFilterChain</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <servlet>  <servlet-name>wisma</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>  WEB-INF/jspConfig.xml,  WEB-INF/reportConfig.xml  </param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>wisma</servlet-name>  <url-pattern>/aplikasi/\*</url-pattern>  </servlet-mapping>    <welcome-file-list>  <welcome-file>aplikasi/kamar/status</welcome-file>  </welcome-file-list>  </web-app> |

**reportconfig.xml**

|  |
| --- |
| <?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-3.0.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-3.0.xsd">  <context:component-scan base-package="com.p4.wismabr"/>  <bean class="org.springframework.web.servlet.view.XmlViewResolver">  <property name="location" value="/WEB-INF/jasper-view.xml"/>  <property name="order" value="0"/>  </bean>  </beans> |

**jesperView.xml**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:p="http://www.springframework.org/schema/p"  xmlns:util="http://www.springframework.org/schema/util"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  http://www.springframework.org/schema/util  http://www.springframework.org/schema/util/spring-util-3.0.xsd">  <!--here all the url value should contains the valid path for the jrxml file-->  <bean id="pdfReport" class="org.springframework.web.servlet.view.jasperreports.JasperReportsPdfView">  <property name="url" value="/WEB-INF/templates/jrxml/laporan-inap.jrxml" />  <property name="reportDataKey" value="datasource" />  </bean>    <bean id="xlsReport" class="org.springframework.web.servlet.view.jasperreports.JasperReportsXlsView">  <property name="url" value="/WEB-INF/templates/jrxml/laporan-inap.jrxml" />  <property name="reportDataKey" value="datasource" />  </bean>      </beans> |

**prak7 uploadservlet.xml**

|  |
| --- |
| <?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">  <context:component-scan base-package="com.p4.upload"/>  <mvc:annotation-driven />    <bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">  <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>  <property name="prefix" value="/WEB-INF/template/jsp/"/>  <property name="suffix" value=".jsp"/>  </bean>    <bean id="multipartResolver"  class="org.springframework.web.multipart.commons.CommonsMultipartResolver">  <property name="maxUploadSize" value="1000000" />  </bean>  </beans> |

**FileUpload.java**

|  |
| --- |
| package com.p4.upload;  public class FileUpload {  private String namaFile;  private Long ukuranFile;  public String getNamaFile() {  return namaFile;  }  public void setNamaFile(String namaFile) {  this.namaFile = namaFile;  }  public Long getUkuranFile() {  return ukuranFile;  }  public void setUkuranFile(Long ukuranFile) {  this.ukuranFile = ukuranFile;  }    } |

**UploadController.java**

|  |
| --- |
| package com.p4.upload;  import java.io.File;  import java.util.ArrayList;  import java.util.List;  import org.springframework.stereotype.Controller;  import org.springframework.ui.ModelMap;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestMethod;  import org.springframework.web.bind.annotation.RequestParam;  import org.springframework.web.multipart.MultipartFile;  @Controller  public class UploadController {    @RequestMapping(value="upload/upload-gambar", method= RequestMethod.GET)  public void tampilForm(){  System.out.println("menampilkan form");  }    @RequestMapping(value="upload/upload-gambar", method= RequestMethod.POST)  public String prosesForm(@RequestParam("photofile") MultipartFile fileUpload )throws Exception{  String namaFile = fileUpload.getOriginalFilename();  File file = new File("src/main/webapp/upload-folder/"+namaFile);  fileUpload.transferTo(file);  return "redirect:/aplikasi/upload/uploaded-file";  }    @RequestMapping("upload/uploaded-file")  public ModelMap tampilFile(){  List<FileUpload> lFile = new ArrayList<FileUpload>();  String direktori="src/main/webapp/upload-folder/";  File folder=new File(direktori);  File[] lisGambar = folder.listFiles();  for (File file : lisGambar){  if(file.isFile()){  FileUpload ofu = new FileUpload();  ofu.setNamaFile(file.getName());  ofu.setUkuranFile(file.length()/1024);  lFile.add(ofu);  }  }  ModelMap mm = new ModelMap();  mm.addAttribute("listFile", lFile);  return mm;  }    @RequestMapping("upload/hapus-file")  public String hapusFile(@RequestParam("namaFile") String namaFile){  File file = new File("src/main/webapp/upload-folder/"+namaFile);  boolean hapus = file.delete();  if(hapus){  System.out.println("hapus berhasil");  }else{  System.out.println("hapus gagal");  }    return "redirect:uploaded-file";  }    @RequestMapping("upload/hapus-semua-file")  public String hapusSemuaFile(){  String direktori="src/main/webapp/upload-folder/";  File folder=new File(direktori);  File[] lisGambar = folder.listFiles();  for (File file : lisGambar){  if(file.isFile()){  File namaFile = new File(file.getName());  boolean hapus = file.delete();  }  }    return "redirect:uploaded-file";  }    @RequestMapping(value="upload/hapus-semua-file",method= RequestMethod.POST)  public String hapusPilihanFile(@RequestParam("filePilihan") String[] filePilihan){  if(filePilihan==null){  return "redirect:uploaded-file";  }    for (String nama : filePilihan) {  File nameFile = new File("src/main/webapp/upload-folder/"+nama);  Boolean hapus = nameFile.delete();  }  return "redirect:uploaded-file";  }  } |

**uploaded-file.jsp**

|  |
| --- |
| <%--  Document : uploaded-file  Created on : May 17, 2013, 3:27:45 PM  Author : Muhammad Sidratul M  --%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <%@page contentType="text/html" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Daftar Foto</title>  </head>  <body>  <h1>Isi Folder</h1>  <a href="<%= request.getContextPath() %>/aplikasi/upload/upload-gambar">Upload Foto</a>  <c:choose>  <c:when test="${empty listFile}">  <h2>Folder Empty</h2>  </c:when>  <c:otherwise>  <table border="1">  <thead>  <tr>  <th><input type="checkbox" disabled></th>  <th>Nama File</th>  <th>Ukuran(KB)</th>  <th colspan="2">Aksi</th>  </tr>  </thead>  <form action="hapus-semua-file" method="POST" id="checkForm">  <tbody>  <c:forEach items="${listFile}" var="lf">  <tr>  <td><input type="checkbox" name="filePilihan" value="${lf.namaFile}"></td>  <td>${lf.namaFile}</td>  <td>${lf.ukuranFile}</td>  <td><a href="<%= request.getContextPath() %>/upload-folder/${lf.namaFile}">lihat</a></td>  <td><a href="hapus-file?namaFile=${lf.namaFile}">hapus</a></td>  </tr>  </c:forEach>  </tbody>  </form>  </table>  <a href="hapus-semua-file">hapus semua</a> <a href="#" onclick="submitForm()">hapus terpilih</a>  </c:otherwise>  </c:choose>  <script>  function submitForm(){  var a = document.getElementById("checkForm");  a.submit();  }  </script>  </body>  </html> |

**config-db.xml**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:context="http://www.springframework.org/schema/context"  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-3.2.xsd">  <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">  <property name="driverClassName" value="com.mysql.jdbc.Driver" />  <property name="url" value="jdbc:mysql://localhost/p4" />  <property name="username" value="root" />  <property name="password" value="" />  </bean>    <context:component-scan base-package="com.p4.springmvc" />  </beans> |

**produk-servlet.xml**

|  |
| --- |
| <?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">  <context:component-scan base-package="com.p4.springmvc"/>  <mvc:annotation-driven />    <bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">  <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>  <property name="prefix" value="/WEB-INF/templates/jsp/"/>  <property name="suffix" value=".jsp"/>  </bean>  </beans> |

**validator**

|  |
| --- |
| import javax.validation.constraints.Min;  import javax.validation.constraints.NotNull;  import javax.validation.constraints.Size;  import org.hibernate.validator.constraints.NotEmpty;  @NotNull  @NotEmpty  @Size(min=1,max=5) |

**produkDao.java**

|  |
| --- |
| package com.p4.springmvc;  import java.math.BigDecimal;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import javax.sql.DataSource;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Repository;  @Repository  public class ProdukDao {  @Autowired private DataSource dataSource;    private String sqlInsert = "INSERT INTO `produk`(`kode`, `nama`, `harga`, `terakhir\_update`) VALUES (?,?,?,?)";  private String sqlUpdate = "UPDATE `produk` SET `kode`=?,`nama`=?,`harga`=?,`terakhir\_update`=? WHERE id=?";  private String sqlTampil = "SELECT \* FROM produk";  private String sqlCariById = "SELECT \* FROM produk where id=?";  private String sqlDelete = "DELETE FROM produk where id=?";    public void simpan(Produk p)throws Exception{  Connection c = dataSource.getConnection();  if(p.getId()==null){  PreparedStatement psInsert = c.prepareStatement(sqlInsert);  psInsert.setString(1,p.getKode());  psInsert.setString(2,p.getNama());  psInsert.setBigDecimal(3,p.getHarga());  psInsert.setDate(4,new java.sql.Date(p.getTerakhirUpdate().getTime()));  psInsert.executeUpdate();  }else{  PreparedStatement psUpdate = c.prepareStatement(sqlUpdate);    psUpdate.setString(1,p.getKode());  psUpdate.setString(2,p.getNama());  psUpdate.setBigDecimal(3,p.getHarga());  psUpdate.setDate(4,new java.sql.Date(p.getTerakhirUpdate().getTime()));  psUpdate.setInt(5,p.getId());  psUpdate.executeUpdate();  }  c.close();  }    public List<Produk> tampilProduk() throws Exception{  Connection c = dataSource.getConnection();  PreparedStatement psTampil = c.prepareStatement(sqlTampil);    List<Produk> hasil = new ArrayList<Produk>();  ResultSet rs = psTampil.executeQuery();  while(rs.next()){  Produk p = konversiResultSetProduk(rs);  hasil.add(p);  }  c.close();  return hasil;  }    public Produk cariById(Integer id) throws Exception{  if(id==null){  return null;  }  Connection c = dataSource.getConnection();  PreparedStatement psCariById = c.prepareStatement(sqlCariById);  psCariById.setInt(1, id);  ResultSet rs = psCariById.executeQuery();  if(!rs.next()){  return null;  }  Produk p =konversiResultSetProduk(rs);  c.close();  return p;  }    private Produk konversiResultSetProduk(ResultSet rs) throws SQLException {  Produk p = new Produk();  p.setId(rs.getInt("id"));  p.setKode(rs.getString("kode"));  p.setNama(rs.getString("nama"));  p.setHarga(rs.getBigDecimal("harga"));  p.setTerakhirUpdate(rs.getDate("terakhir\_update"));  return p;  }    public void hapus(Integer id)throws Exception{  if(id==null){  return;  }  Connection c = dataSource.getConnection();  PreparedStatement psDelete = c.prepareStatement(sqlDelete);  psDelete.setInt(1, id);  psDelete.executeUpdate();  c.close();  }    } |

**daftarproduk.jsp**

|  |
| --- |
| <%@page contentType="text/html" pageEncoding="windows-1252"%>  <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">  <title>tampil produk</title>  </head>  <body>  <a href="input-produk">input produk</a>  <h1>Produk</h1>  <c:choose>  <c:when test="${empty daftarProduk}">  <h2>Record Empty</h2>  </c:when>  <c:otherwise>  <table border="1">  <thead>  <tr>  <td>#</td>  <td>Kode</td>  <td>Nama</td>  <td>Harga</td>  <td>Terakhir Update</td>  <td>&nbsp;</td>  </tr>  </thead>  <tbody>  <c:forEach items="${daftarProduk}" var="p" varStatus="i">  <tr>  <td>${i.count}</td>  <td>${p.kode}</td>  <td>${p.nama}</td>  <td>  <fmt:formatNumber value="${p.harga}" type="number"></fmt:formatNumber>  </td>  <td>  <fmt:formatDate value="${p.terakhirUpdate}" pattern="EEE, dd MMM yyyy"></fmt:formatDate>  </td>  <td>  <a href="input-produk?id=${p.id}">edit</a>  <a href="delete?id=${p.id}">hapus</a>  </td>  </tr>  </c:forEach>  </tbody>  </table>  </c:otherwise>  </c:choose>  </body>  </html> |

**filter.java**

|  |
| --- |
| package com.muhardin.endy.training.web;  import java.io.IOException;  import javax.servlet.Filter;  import javax.servlet.FilterChain;  import javax.servlet.FilterConfig;  import javax.servlet.ServletException;  import javax.servlet.ServletRequest;  import javax.servlet.ServletResponse;  public class HaloFilter implements Filter{  public void init(FilterConfig fc) throws ServletException {  System.out.println("Filter diinisialisasi");  }  public void doFilter(ServletRequest request, ServletResponse response, FilterChain fc) throws IOException, ServletException {  System.out.println("Mencegat request");  fc.doFilter(request, response);  System.out.println("Mencegat response");  }  public void destroy() {  System.out.println("Filter di-destroy");  }    } |

**listener.java**

|  |
| --- |
| /\*  \* To change this template, choose Tools | Templates  \* and open the template in the editor.  \*/  package com.muhardin.endy.training.web;  import javax.servlet.ServletContextEvent;  import javax.servlet.ServletContextListener;  import javax.servlet.http.HttpSessionEvent;  import javax.servlet.http.HttpSessionListener;  public class HaloListener implements ServletContextListener, HttpSessionListener{  public void contextInitialized(ServletContextEvent sce) {  System.out.println("Aplikasi start");  }  public void contextDestroyed(ServletContextEvent sce) {  System.out.println("Aplikasi stop");  }  public void sessionCreated(HttpSessionEvent hse) {  System.out.println("Ada session baru");  }  public void sessionDestroyed(HttpSessionEvent hse) {  System.out.println("Session di-destroy");  }    } |

**servlet.java**

|  |
| --- |
| package com.muhardin.endy.training.web;  import java.io.IOException;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class HaloServlet extends HttpServlet {  public void doGet(HttpServletRequest request, HttpServletResponse response) {  try {    System.out.println("Menjalankan method doGet");  System.out.println("Context Path : "+req.getContextPath());  System.out.println("Path Info : "+req.getPathInfo());  System.out.println("Path Translated : "+req.getPathTranslated());  System.out.println("Query String : "+req.getQueryString());  System.out.println("Request URI : "+req.getRequestURI());  System.out.println("Isi parameter nama : "+req.getParameter("nama"));  System.out.println("Isi parameter email : "+req.getParameter("email"));    String asal = request.getRemoteAddr();  String nama = request.getParameter("nama");    String output = "<html>";  output += "<body>";  output += "<h1>Halo "+nama+"</h1>";  output += "<h2>Anda datang dari "+asal+"</h2>";  output += "</body>";  output += "</html>";    // content type = html  response.setContentType("text/html");    // supaya filenya didownload  // resp.setHeader("Content-Disposition", "attachment; filename=halo.txt");    response.getWriter().print(output);  } catch (IOException ex) {  Logger.getLogger(HaloServlet.class.getName()).log(Level.SEVERE, null, ex);  }  }  } |

**web.xml**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>  /WEB-INF/konfig-spring-security.xml  </param-value>  </context-param>    <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>    <listener>  <listener-class>com.muhardin.endy.training.web.HaloListener</listener-class>  </listener>    <filter>  <filter-name>springSecurityFilterChain</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>springSecurityFilterChain</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <filter>  <filter-name>sitemesh</filter-name>  <filter-class>com.opensymphony.sitemesh.webapp.SiteMeshFilter</filter-class>  </filter>    <filter-mapping>  <filter-name>sitemesh</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <filter>  <filter-name>haloFilter</filter-name>  <filter-class>com.muhardin.endy.training.web.HaloFilter</filter-class>  </filter>    <filter-mapping>  <filter-name>haloFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>    <servlet>  <servlet-name>halo</servlet-name>  <servlet-class>com.muhardin.endy.training.web.HaloServlet</servlet-class>  </servlet>    <servlet-mapping>  <servlet-name>halo</servlet-name>  <url-pattern>/halo.php</url-pattern>  </servlet-mapping>    </web-app> |