# MODUL 3 Aplikasi CRUD menggunakan EJB, JPA, dan MySQL

## TUJUAN

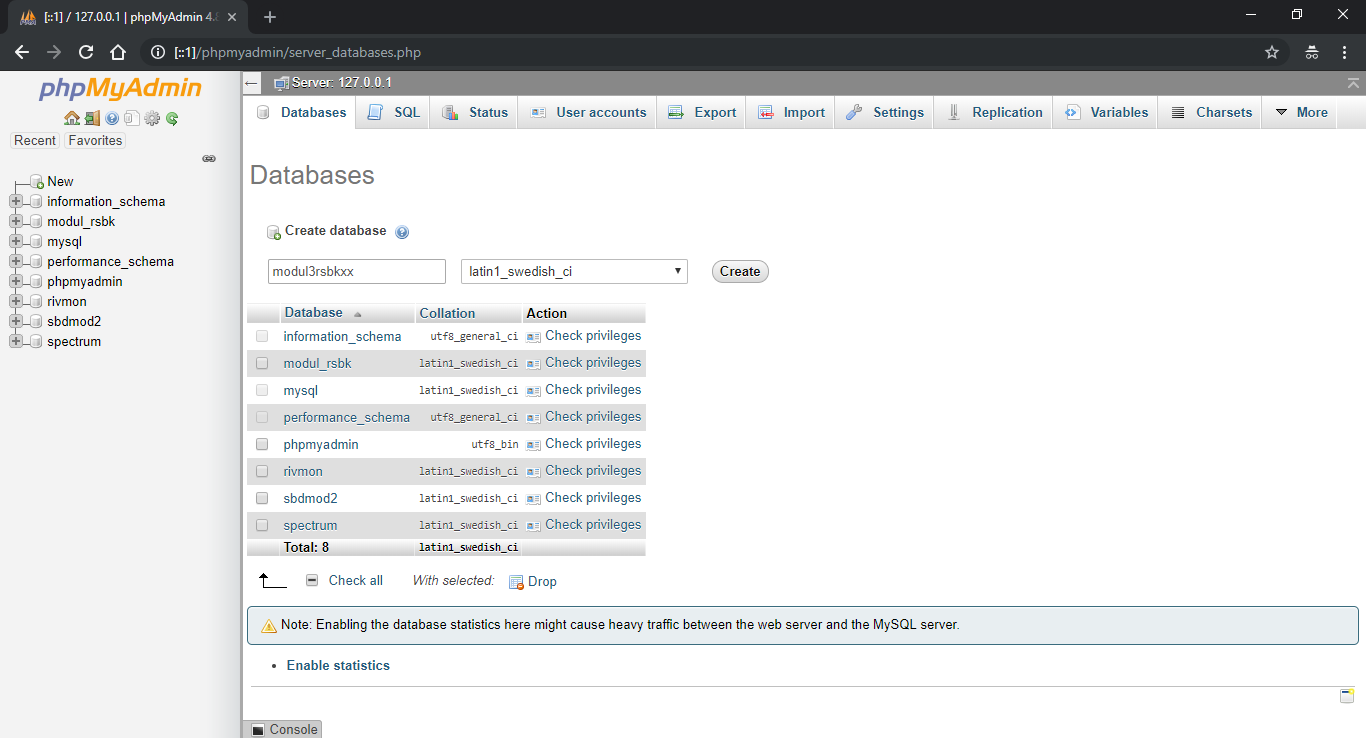
1. Praktikan dapat menggunakan operasi CRUD dengan database MySQL
2. Praktikan dapat mengimplementasikan EJB
3. Praktikan dapat mengimplementasikan JPA

## ALAT DAN BAHAN

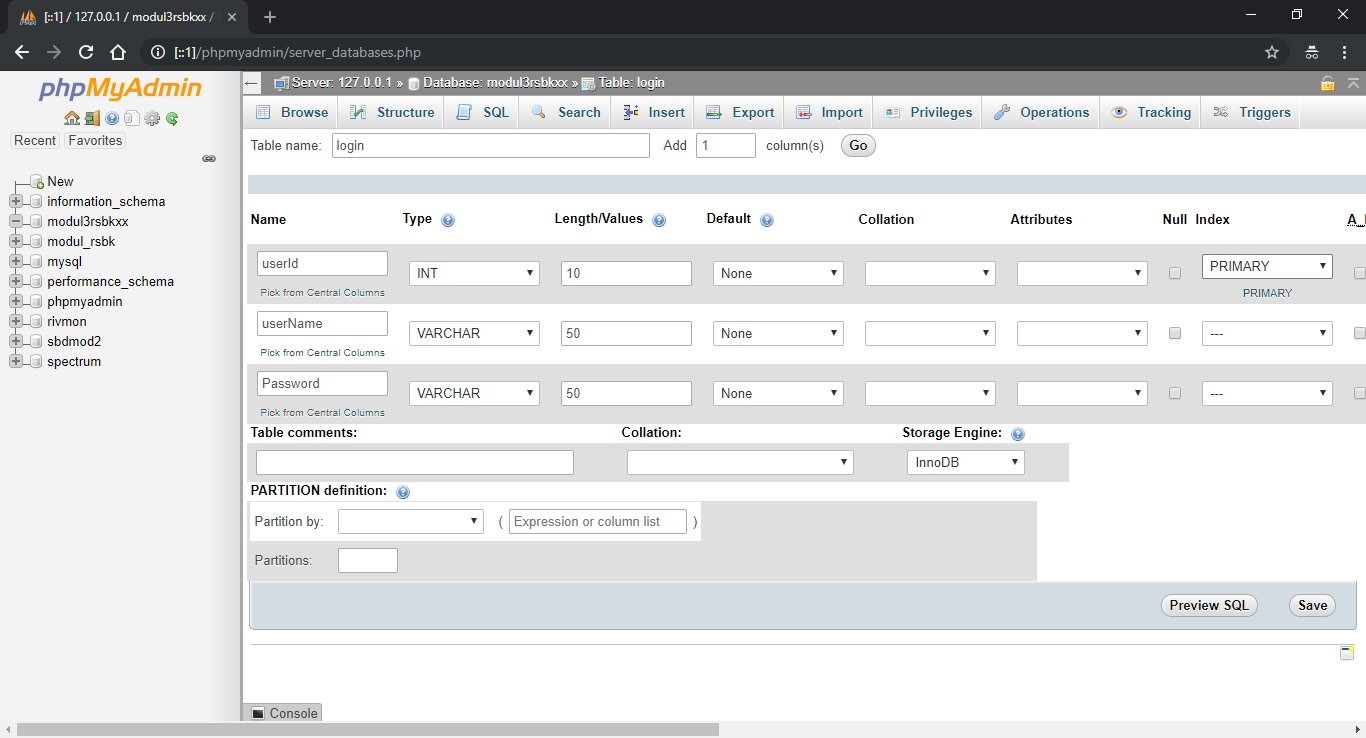
1. Laptop
2. Netbeans IDE
3. MySQL
4. XAMPP
5. Glassfish Server 4.1

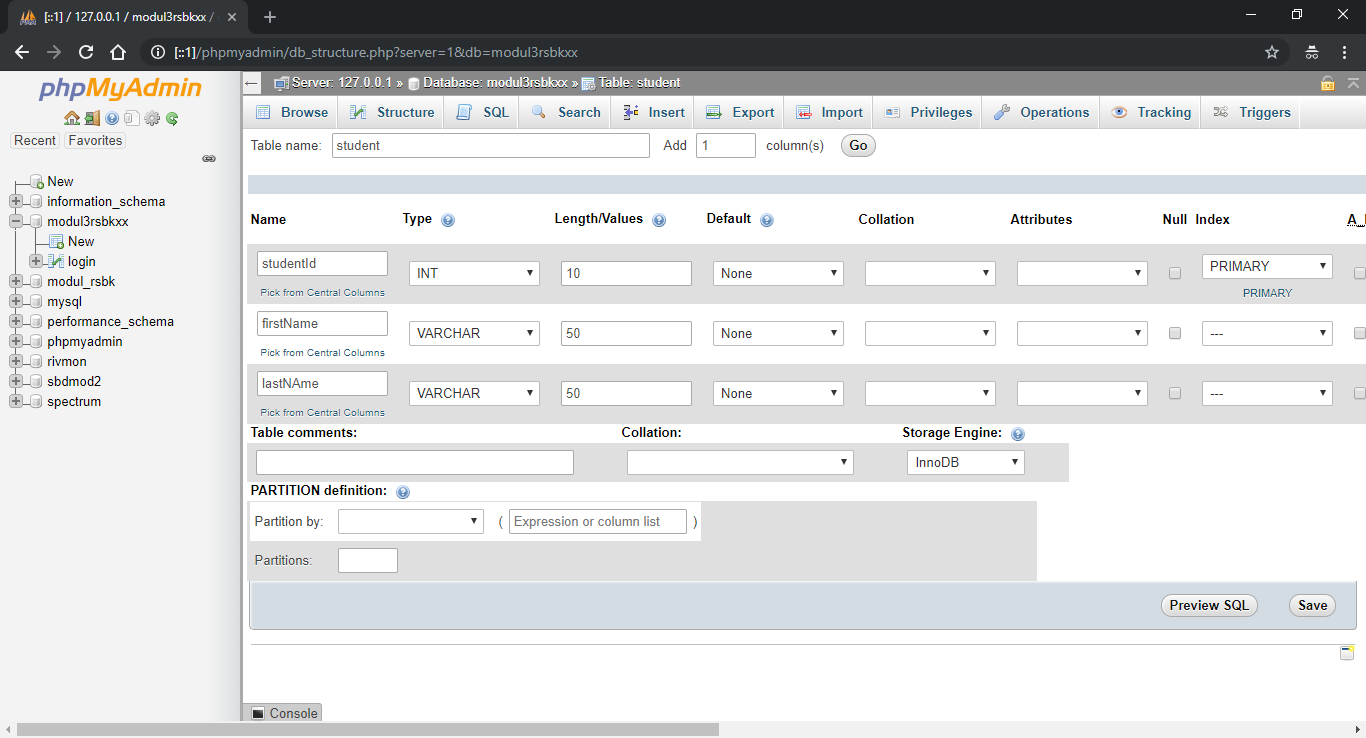
## LANGKAH KERJA

1. Jalankan XAMPP, jika ada bentrok dengan salah satu PORT, matikan PORT tersebut lalu nyalakan kembali (biasanya bentrok sama vmware/oracle)
2. Buat database di MySQL. Beri nama “modul3rsbkxx” ganti xx dengan kelompok.

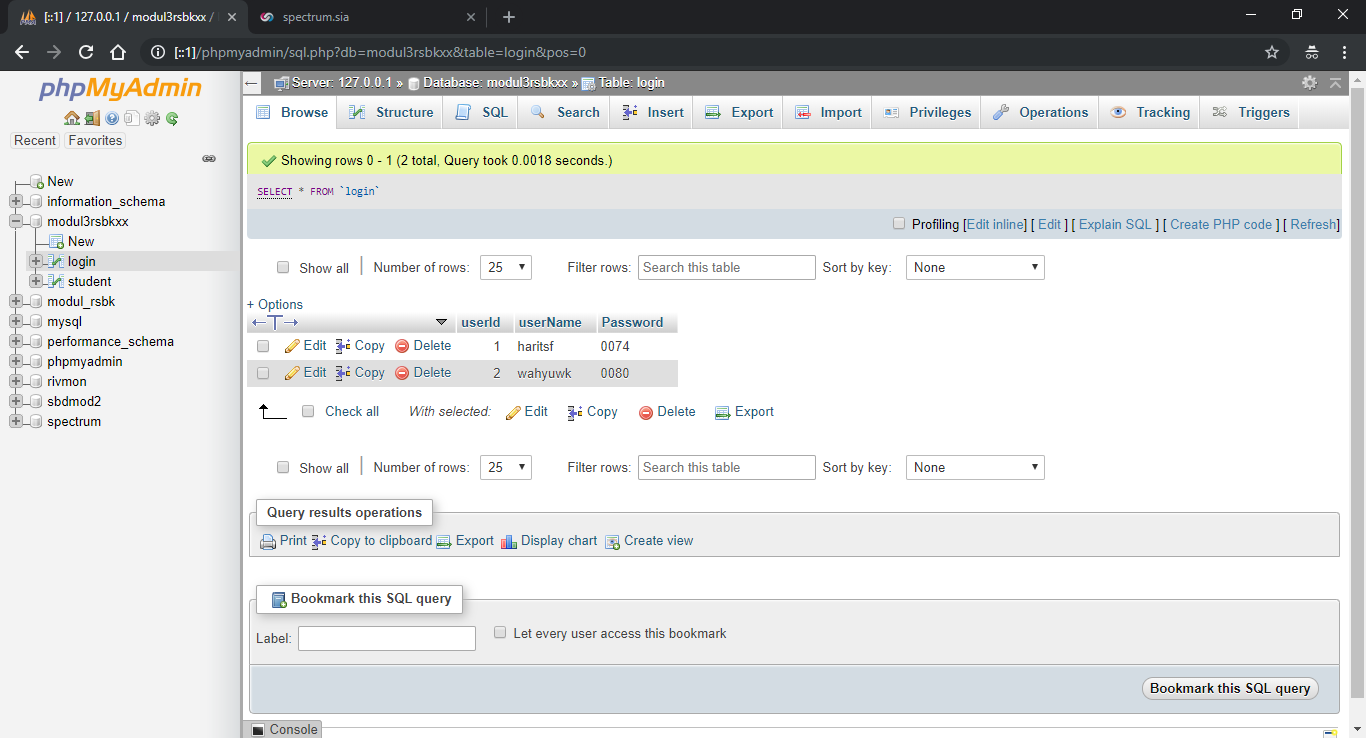


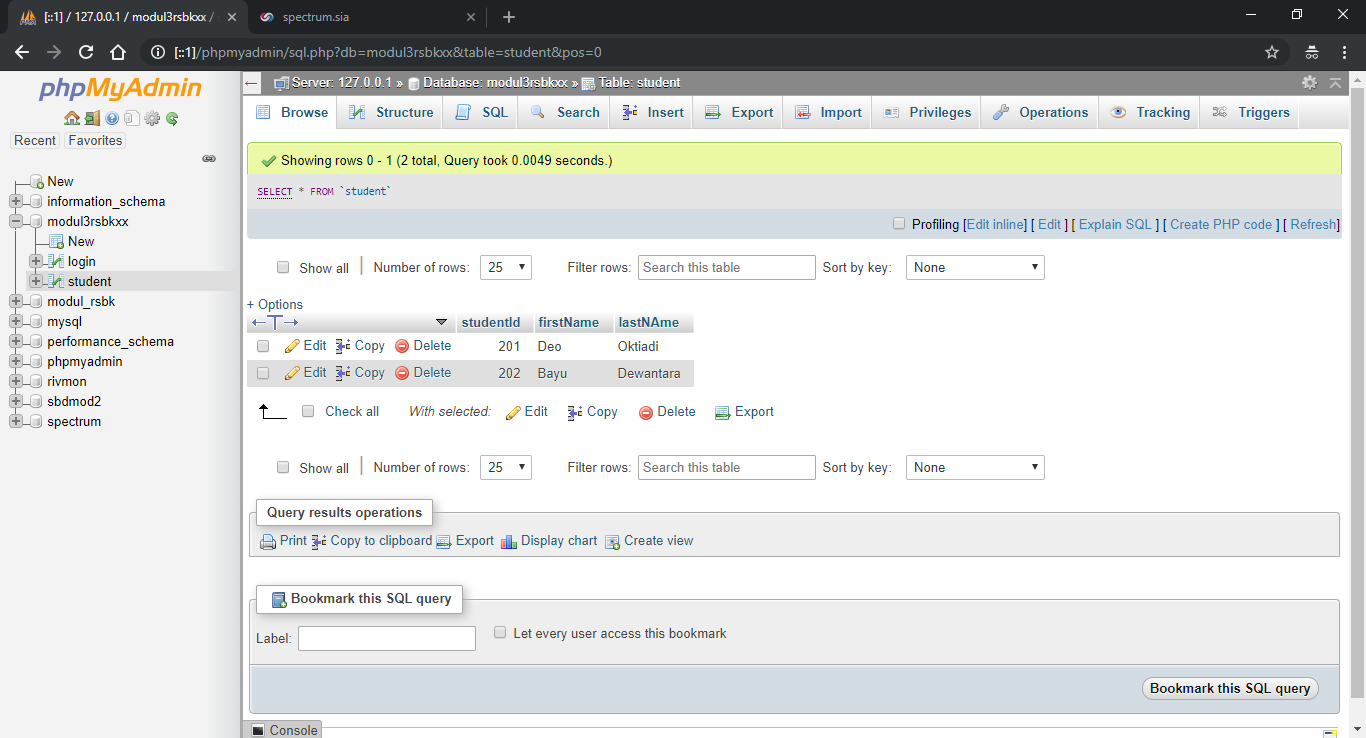
1. Buat 2 Tabel, yaitu tabel “login” dengan 3 atribut dan “student” dengan. Dengan 3 atribut juga. Untuk atribut yang ada di login yaitu userId (INT, 10, Primary Key), username (VARCHAR, 50), dan Password (VARCHAR, 50). Dan untuk atribut yang ada di student yaitu studentId (INT, 10, Primary Key), firstName (VARCHAR, 50), dan lastName (VARCHAR, 50).



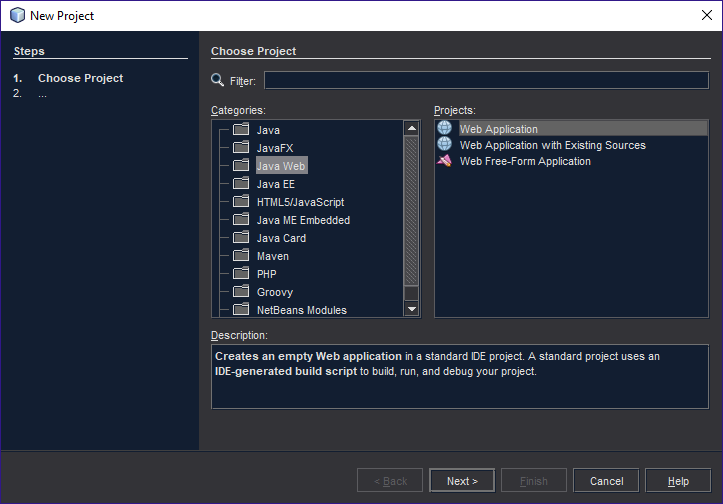


1. Isi login dan student dengan data anggota kelompok, dan student dengan nama yang berbeda.

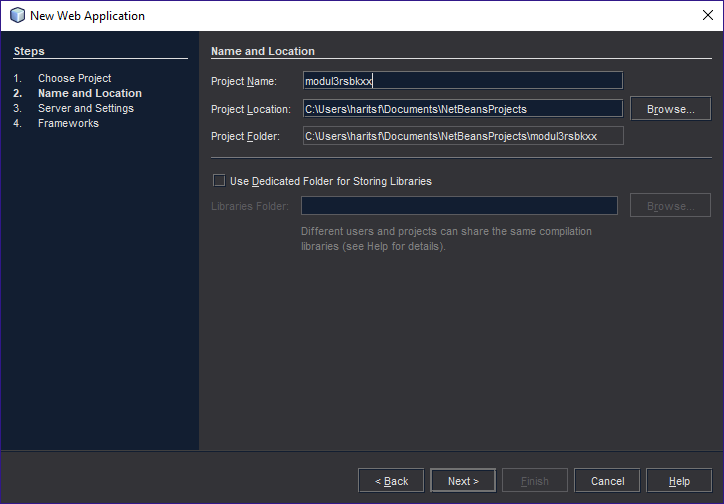


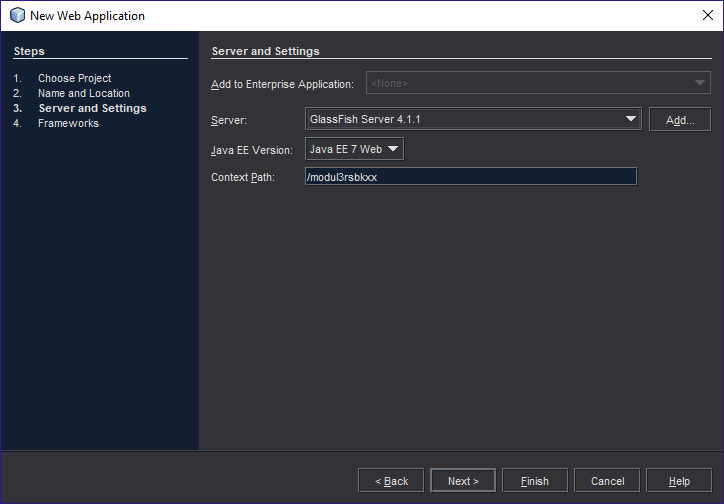


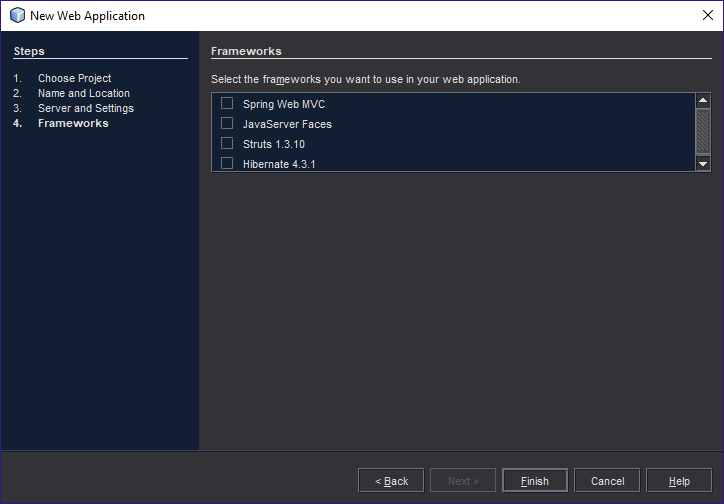
1. Buka Netbeans lalu buat project baru pada Netbeans, pilih Java Web → Web Application



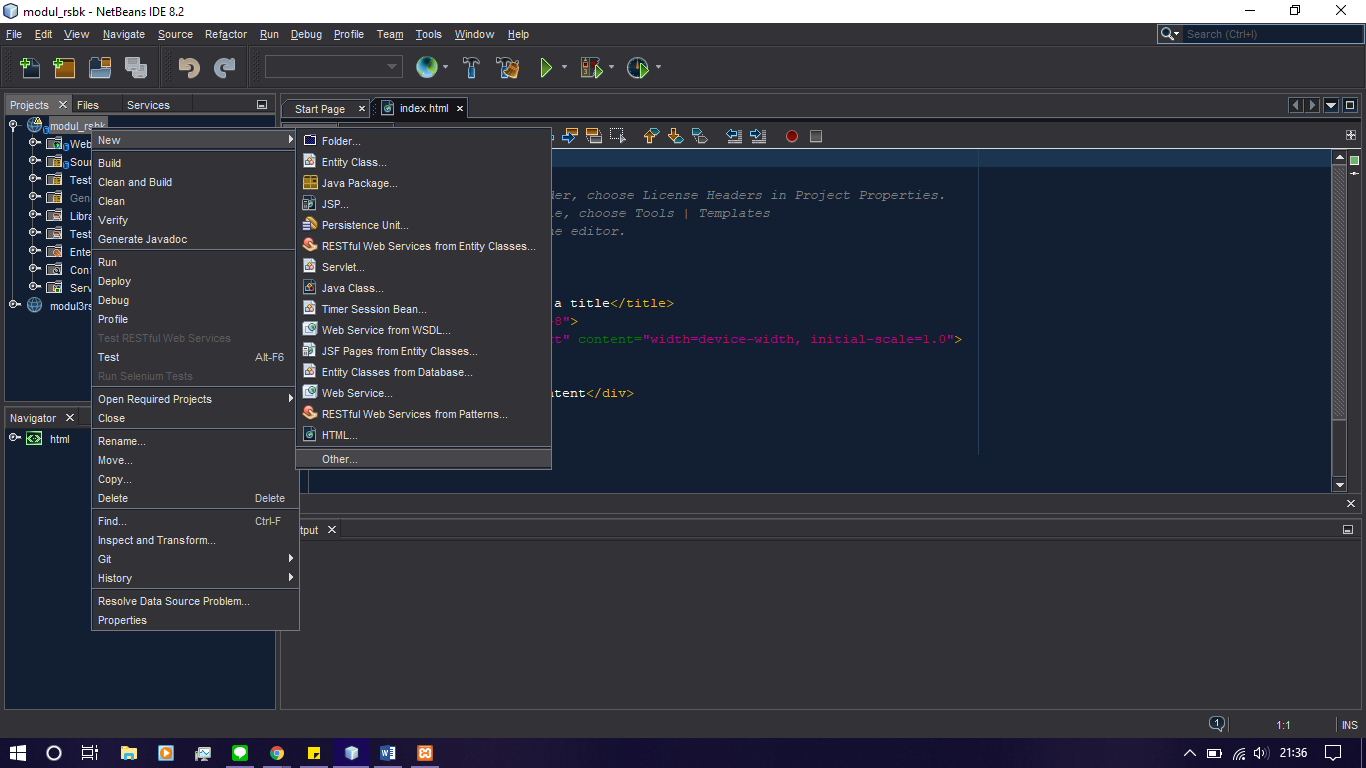
1. Beri nama project “**modul3rsbkxx**” dan pilih Glassfish Server (xx nomor kelompok).

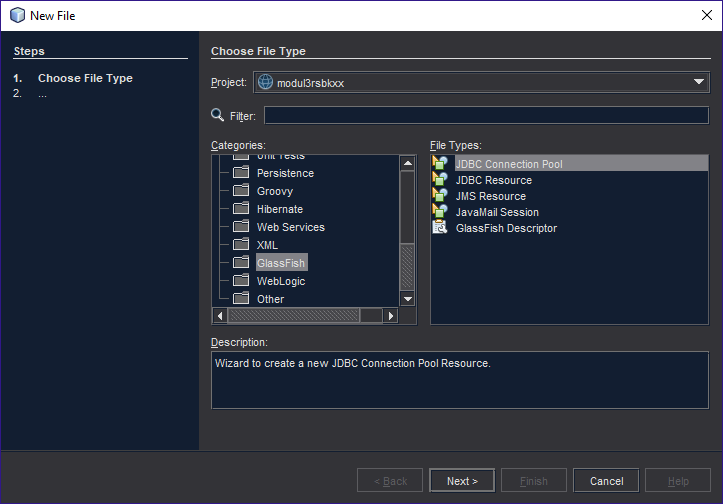




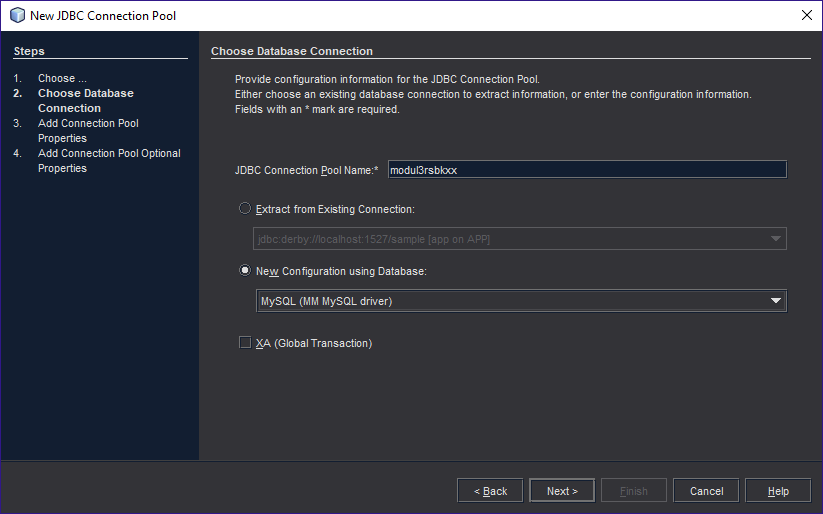


1. Buat JDBC Connection Pool, klik kanan pada project, new file, other. Lalu masuk ke kategori GlassFish, pilih tipe file JDBC Connection Pool.

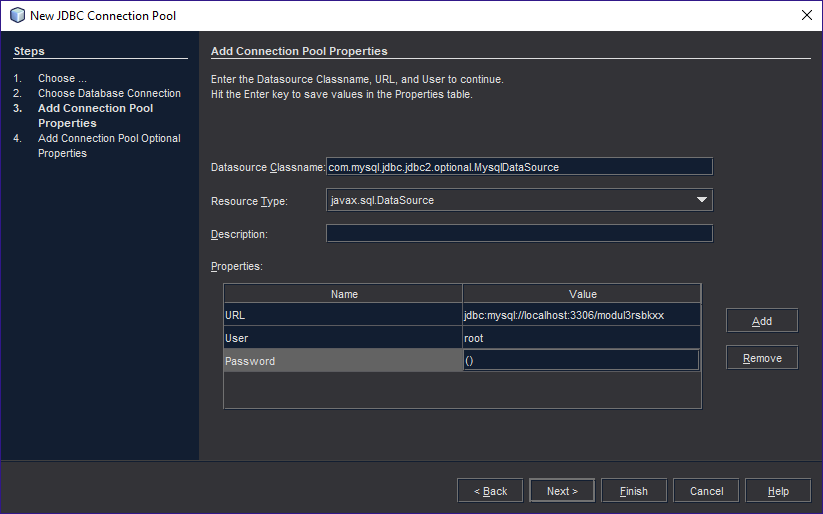




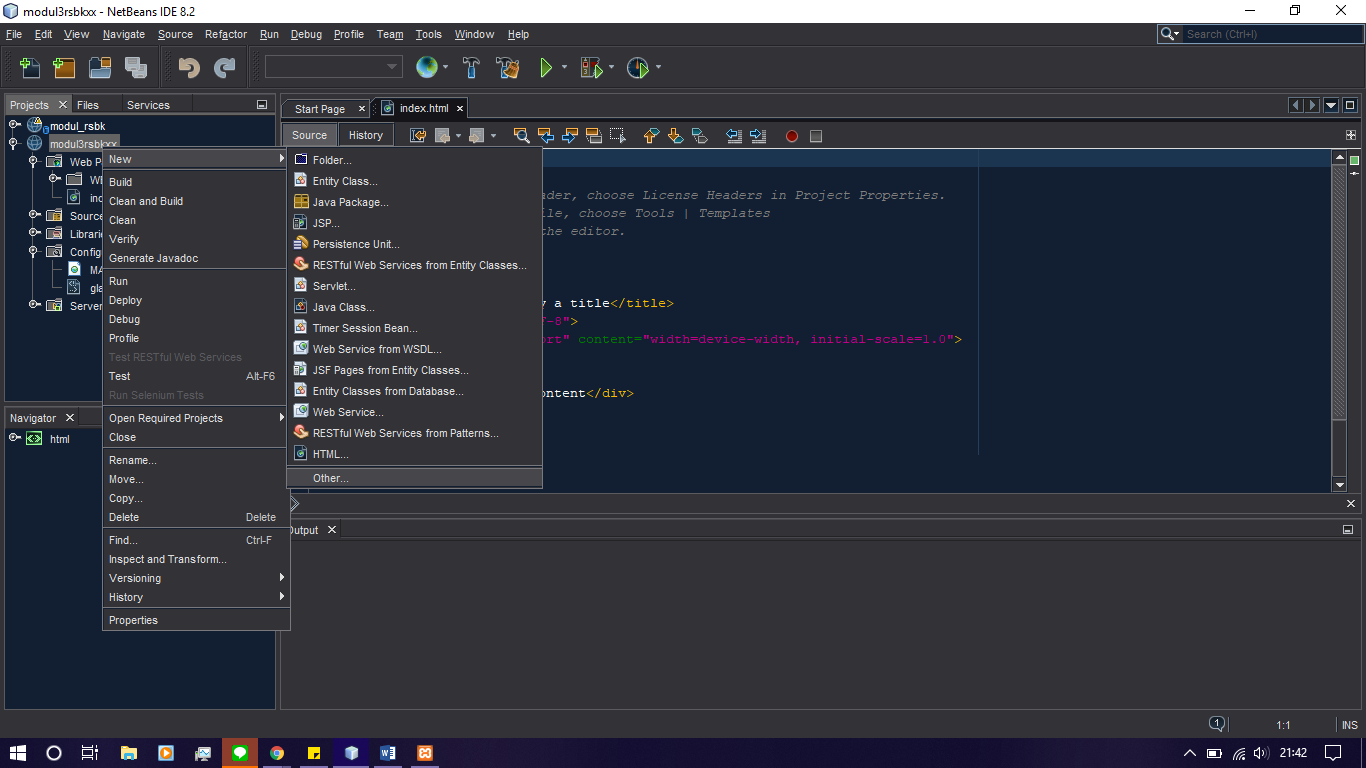
1. Beri nama connection pool (sama seperti nama projectnya, misal modul3rsbkxx) dan pilih ‘New Configuration using Database’ → ‘MySQL (MM MySQL driver).

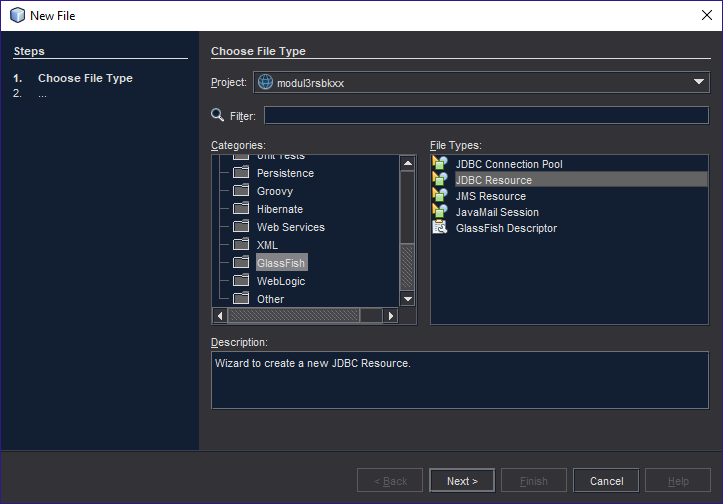


1. Masukkan URL, User, dan Password MySQL yang sudah dibuat. URL ke ***localhost*** dengan port default MySQL **3306**, lalu nama database yang akan kita gunakan (missal **modul3rsbkxx**). Untuk user pakai **root** dan passwordnya **()**. Kemudian pilih Finish.

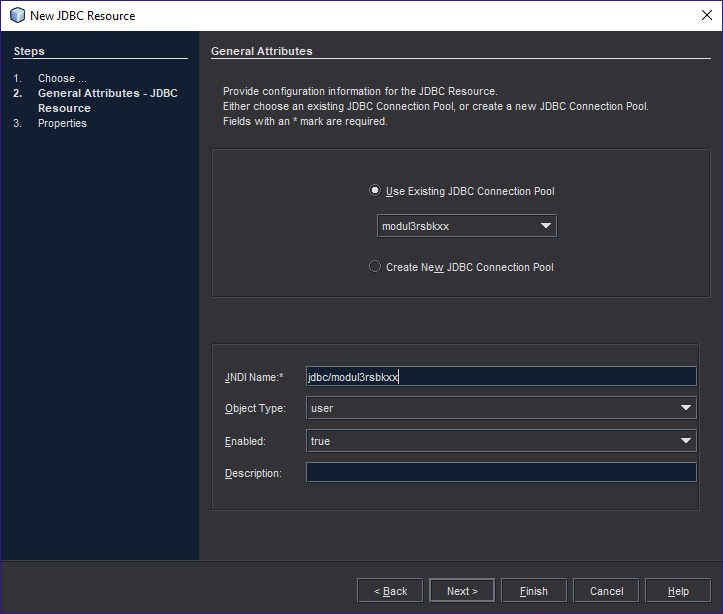


1. Buat JDBC Resource. Dengan klik kanan pada Project, New File, Other. Lalu pada kategori pilih GlassFish, kemudian pilih JDBC Resource.

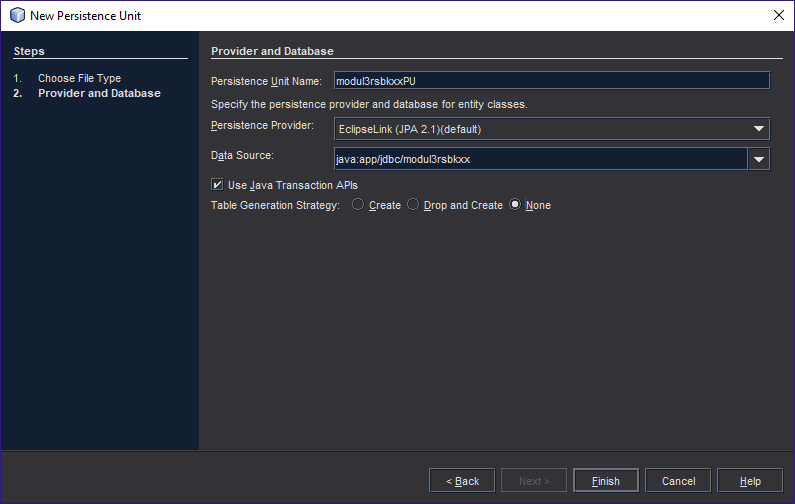




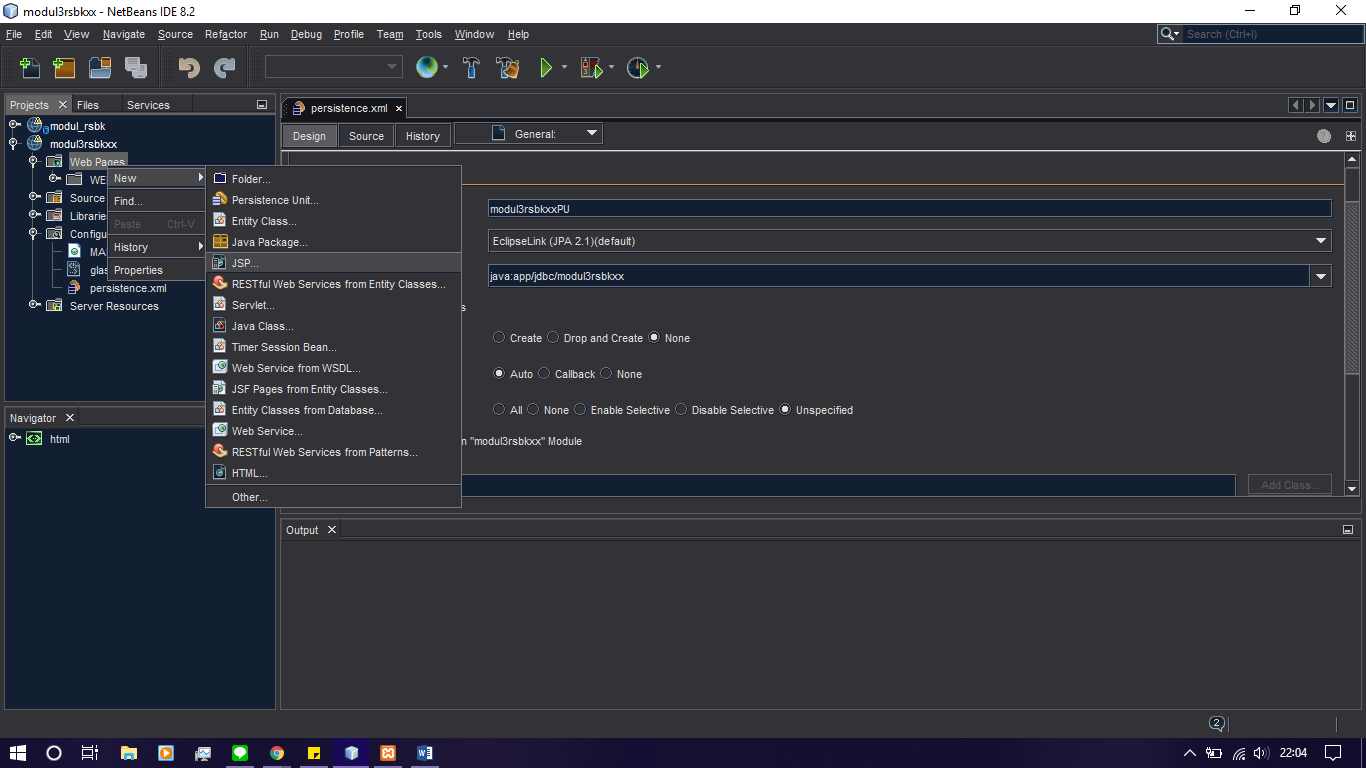
1. Pilih ‘Use Existing JDBC Connection Pool’, pilih List JDBC yang baru kita buat tadi. Kemudian isikan JNDI Name seperti yang tadi. Misal modul3rsbkxx. Lalu pilih Finish.

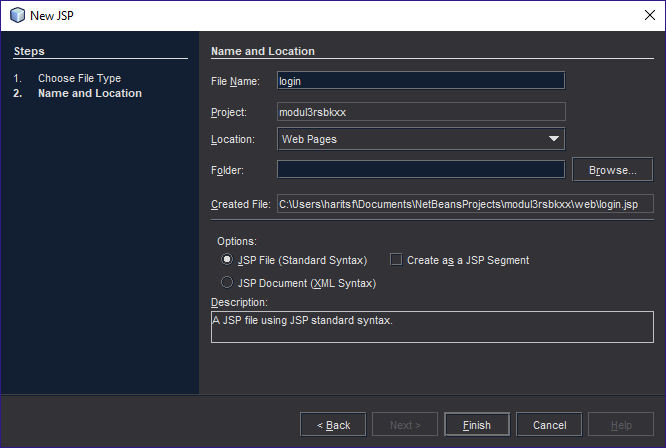


1. Buat Persistence Unit dengan klik kanan pada project, New File, pilih Persistence Unit (Jika tidak ketemu, pilih other, lalu pada kolom filter ketik persistence unit). Nama Persistence Unit akan otomatis sesuai dengan Project, lalu pilih data resource yang tadi sudah dibuat. Table Generation Strategy pilih ‘None’



1. Buat 3 JSP page pada folder “Web Pages” dengan klik kanan folder lalu new file. JSP. Beri nama: home.jsp, login.jsp, dan error.jsp. Masukkan source code yang tersedia.





**login.jsp**

<%--

Document : login

Created on : Oct 10, 2018, 11:03:47 PM

Author : haritsf

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Login Data</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<script src="js/bootstrap.min.js"></script>

<style>

.menu {

margin-left: -15px;

margin-right: 15px;

}

.daftar{

border: 2px solid #e5e5e5;

border-radius: 10px;

padding: 20px;

}

.daftar a{

margin-top: 2%;

}

.detail{

padding: 10px 0px;

}

.nav{

padding: 0px;

border: 1px solid #e5e5e5;

border-radius: 5px;

}

.nav li{

border-bottom: 1px solid #e5e5e5;

border-radius: 5px;

}

</style>

</head>

<body>

<div class="container">

<div class="jumbotron row">

<center><h2><b>Data Mahasiswa</b></h2>

<h4>Modul RSBK - Kelompokxx</h4></center>

</div>

<div class="row content">

<div class="col-md-12">

<div class="col-md-4 col-md-offset-4 daftar">

<p class="form-title">Sign In</p>

<form method="POST" action="/**modul3rsbkxx**/LoginServlet" class="login">

<div class="form-group">

<label>Identitas</label>

<input type="text" class="form-control" placeholder="Username" name="userName" type="text" autofocus />

</div>

<div class="form-group">

<label>Password</label>

<input type="password" class="form-control" placeholder="Password" name="password" value="" required />

</div>

<input type="submit" name="login" value="Login" class="btn btn-md btn-success" />

</form>

</div>

</div>

</div>

</div>

</body>

</html>

**home.jsp**

<%--

Document : home

Created on : Oct 10, 2018, 11:02:49 PM

Author : haritsf

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="s" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Home Page</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<script src="js/bootstrap.min.js"></script>

<style>

.menu {

margin-left: -15px;

margin-right: 15px;

}

.daftar{

border: 2px solid #e5e5e5;

border-radius: 5px;

padding: 5px;

}

.table th, .table td{

text-align: center;

}

.nav{

padding: 5px;

border: 2px solid #e5e5e5;

border-radius: 5px;

}

.nav li{

border-bottom: 2px solid #e5e5e5;

border-radius: 5px;

}

.daftar h3{

margin-top: 50px;

margin-bottom: 25px;

}

</style>

</head>

<div class="container">

<div class="jumbotron row">

<center><h2><b>Data Mahasiswa</b></h2>

<p>Modul RSBK - Kelompokxx</p>

<h5>Selamat Datang, <%=session.getAttribute("loginName")%></h6></center>

</div>

<div class="row content col-md-8 col-md-offset-2">

<div class="col-md-3 menu">

<ul class="nav nav-pills nav-stacked" style="">

<li><a href="#">Home</a></li>

<li><a href="/**modul3rsbkxx**/login.jsp">Logout</a></li>

</ul>

</div>

<div class="col-md-9 daftar">

<form action="./StudentServlet" method="POST">

<table class="table table-bordered">

<tr>

<td>Student ID</td>

<td><input class="form-control" type="text" name="studentId" value="${student.studentId}" /></td>

</tr>

<tr>

<td>First Name</td>

<td><input class="form-control" type="text" name="firstname" value="${student.firstName}" /></td>

</tr>

<tr>

<td>Last Name</td>

<td><input class="form-control" type="text" name="lastname" value="${student.lastName}" /></td>

</tr>

<tr>

<td colspan="2">

<input type="submit" class="btn btn-primary btn-sm" name="action" value="Add" />

<input type="submit" class="btn btn-default btn-sm" name="action" value="Edit" />

<input type="submit" class="btn btn-danger btn-sm" name="action" value="Delete" />

<input type="submit" class="btn btn-warning btn-sm" name="action" value="Search" />

</td>

</tr>

</table>

</form>

<h3 align="center">Informasi Data</h3>

<table class="table table-bordered table-hover">

<thead>

<tr>

<th>No. ID</th>

<th>First Name</th>

<th>Last Name</th>

</tr>

</thead>

<tbody>

<s:forEach items="${allStudents}" var="stud">

<tr>

<td>${stud.studentId}</td>

<td>${stud.firstName}</td>

<td>${stud.lastName}</td>

</tr>

</s:forEach>

</tbody>

</table>

</div>

</div>

</div>

</html>

**error.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Error Page</title>

</head>

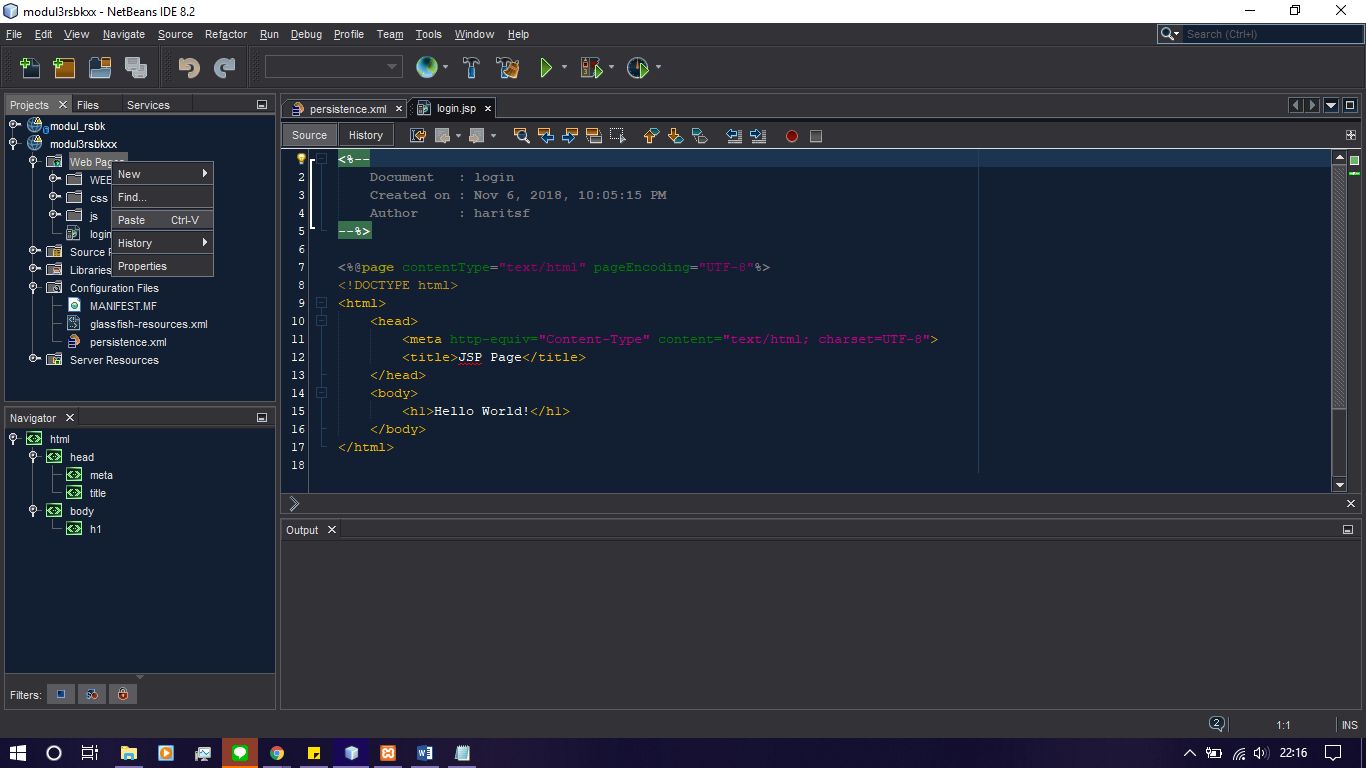
<body>

<h1>Error - <%=request.getAttribute("error")%> </h1>

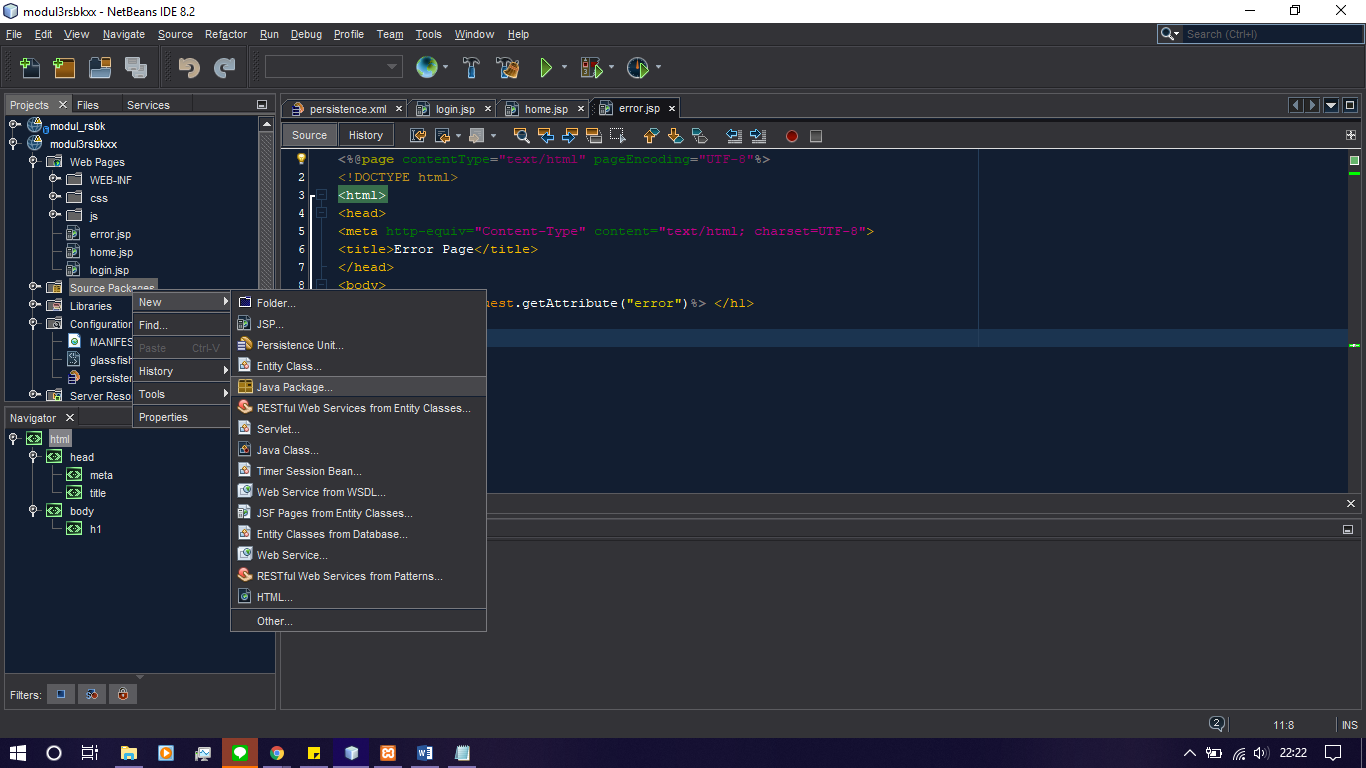
</body>

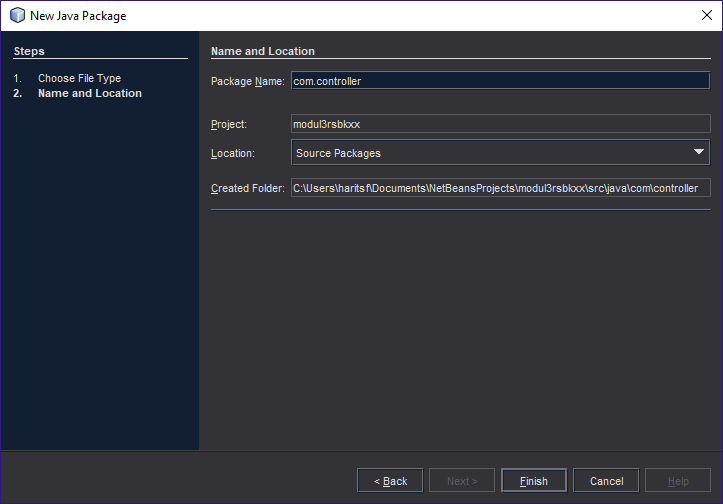
</html>

1. Cek “login.jsp” dan “home.jsp” pada font yang di **bold merah** disesuaikan dengan project kelompok.
2. Masukkan asset dengan copy dan paste folder css dan js ke web pages

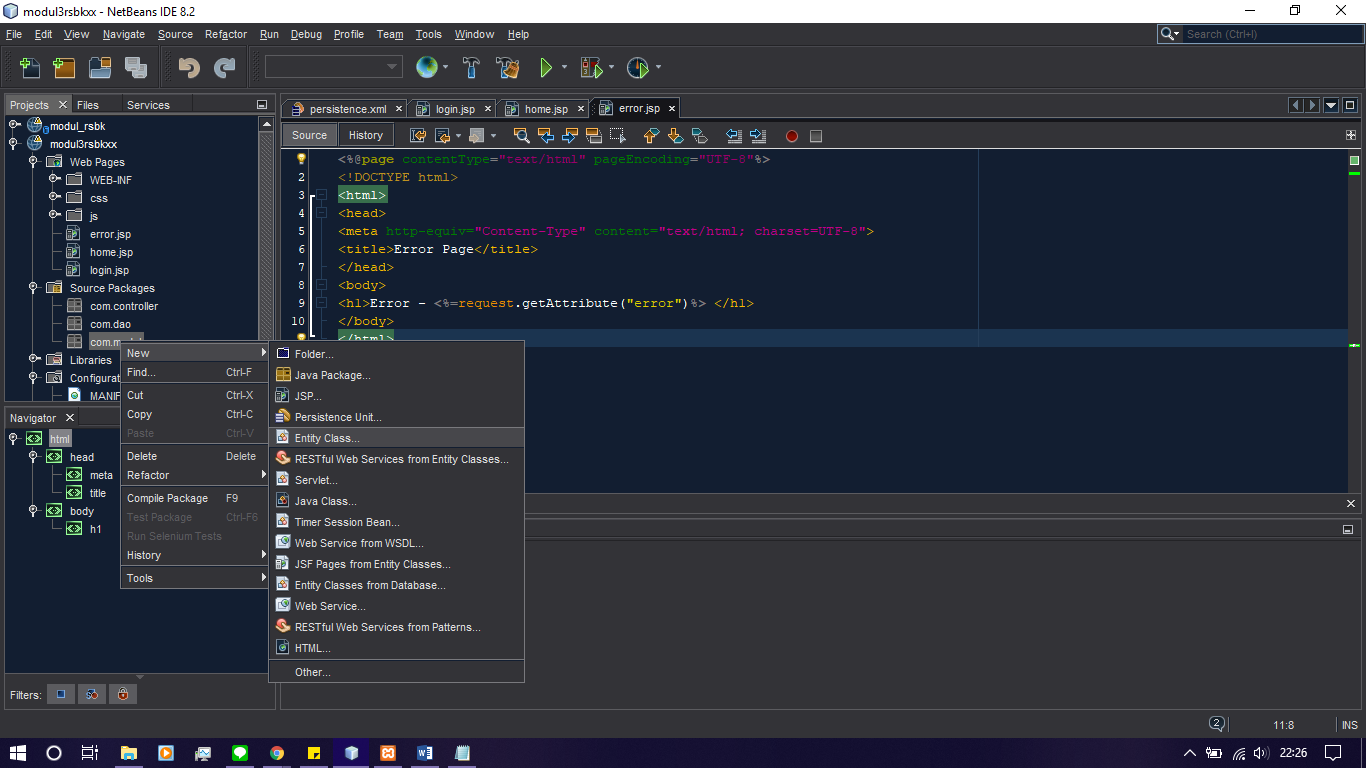


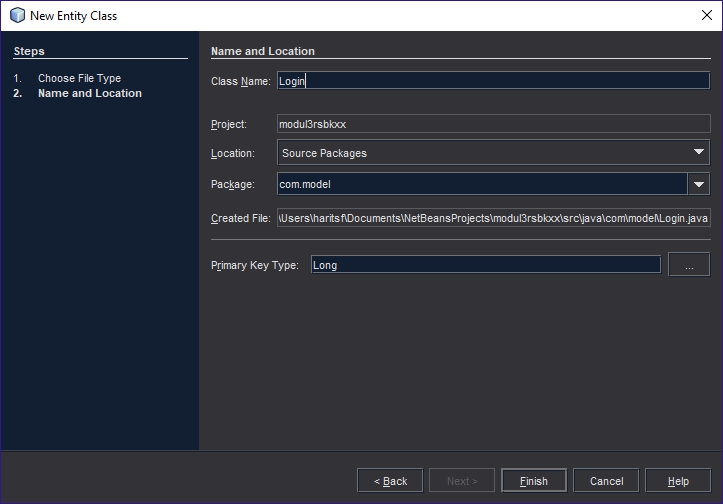
1. Buat 3 java package pada folder “Source Packages”, beri nama: “com.controller”, “com.dao”, dan “com.model”.





1. Buat 2 Entity Class pada package “com.model”, beri nama : “Login” dan “Student”. Masukkan source code yang tersedia.





**Login.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

import javax.persistence.Column;

/\*\*

\*

\* @author USER

\*/

@Entity

@Table

@NamedQueries({@NamedQuery(name="Login.getAll",query="SELECT e FROM Login e")})

public class Login implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column

private int userId;

@Column

private String userName;

@Column

private String password;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

**Student.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

import javax.persistence.Column;

/\*\*

\*

\* @author USER

\*/

@Entity

@Table

@NamedQueries({@NamedQuery(name="Student.getAll",query="SELECT e FROM Student e order by e.studentId")})

public class Student implements Serializable {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column

private int studentId;

@Column

private String firstName;

@Column

private String lastName;

public Student(int studentId, String firstName, String lastName) {

this.studentId = studentId;

this.firstName = firstName;

this.lastName = lastName;

}

public Student() {

}

public void setStudentId(int studentId) {

this.studentId = studentId;

}

public int getStudentId() {

return studentId;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getFirstName() {

return firstName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

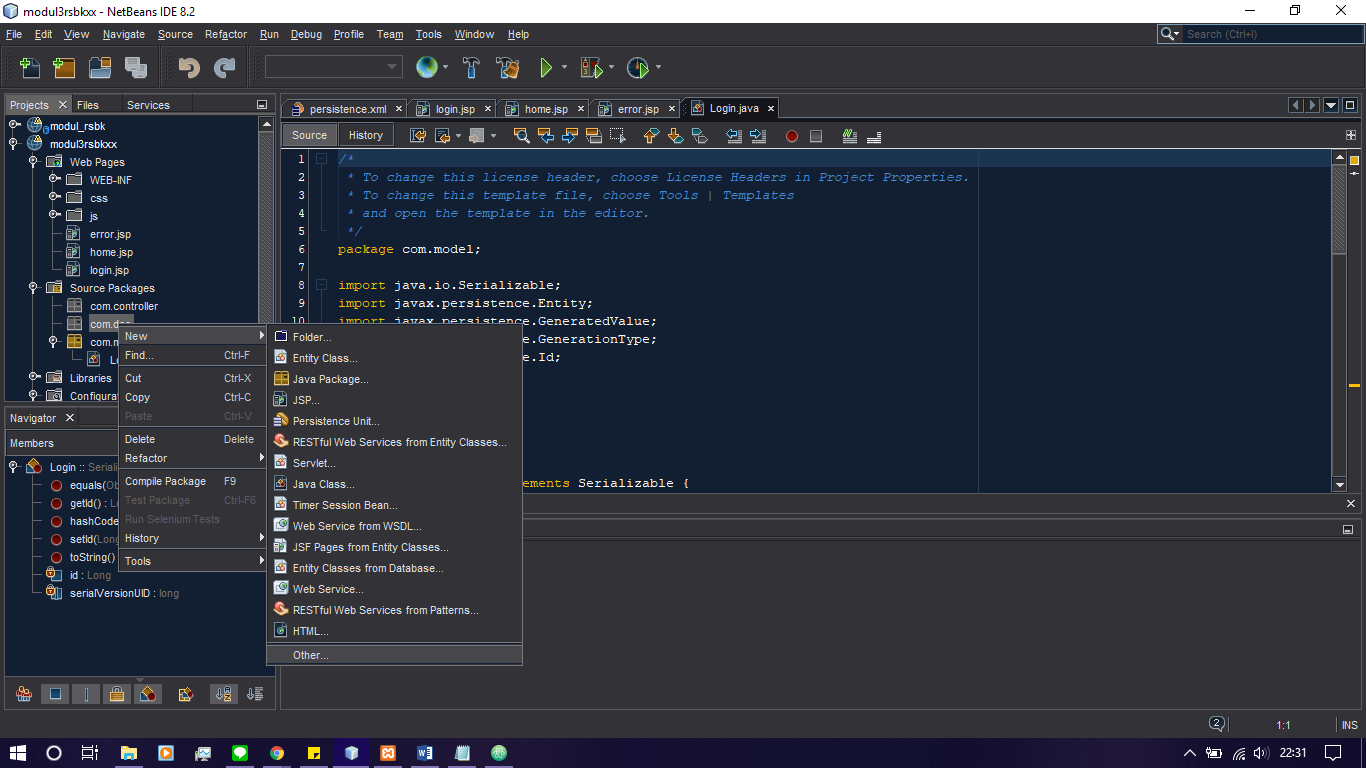
public String getLastName() {

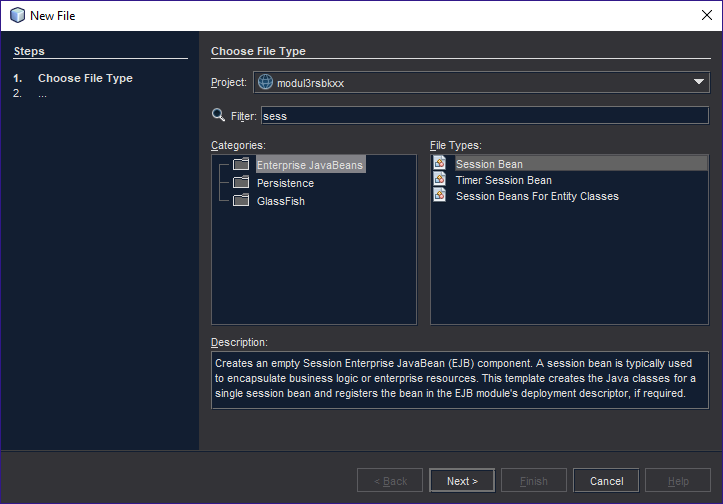
return lastName;

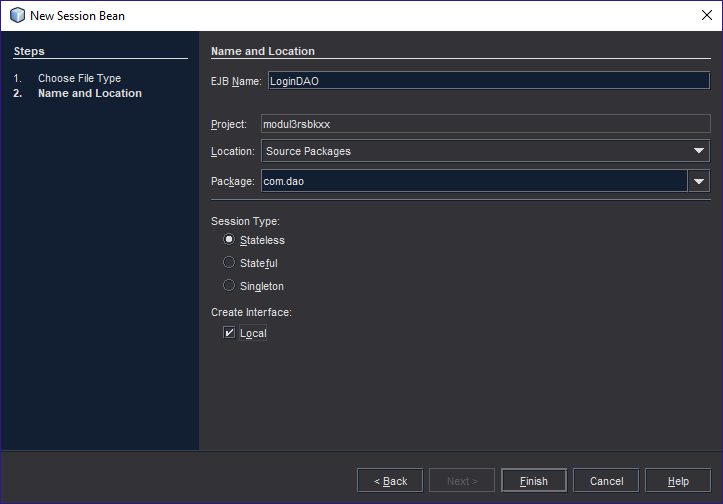
}

}

1. Buat 2 Session Beans pada package “com.dao”, beri nama : “LoginDAO” dan “StudentDAO”. Ketika membuat session bean, pilih Session Type ‘**Stateless**’ dan Create Interface ‘**Local’**. Masukkan source code yang tersedia.







**LoginDAO.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dao;

import javax.ejb.Stateless;

import com.model.Login;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

/\*\*

\*

\* @author USER

\*/

@Stateless

public class LoginDAO implements LoginDAOLocal {

@PersistenceContext

private EntityManager em;

@Override

public boolean checkUser(String userName, String password) {

List<Login> s = (List<Login>)em.createQuery("select e from Login e where e.userName='"+userName+"' and e.password='"+password+"'").getResultList();

System.out.println("is list empty ?"+s.isEmpty()+" for the"+userName+" and "+password);

if(!s.isEmpty())

return true;

else

return false;

}

// Add business logic below. (Right-click in editor and choose

// "Insert Code > Add Business Method")

}

**LoginDAOLocal.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dao;

import javax.ejb.Local;

/\*\*

\*

\* @author USER

\*/

@Local

public interface LoginDAOLocal {

public boolean checkUser(String username, String password);

}

**StudentDAO.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dao;

import com.model.Student;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

/\*\*

\*

\* @author USER

\*/

@Stateless

public class StudentDAO implements StudentDAOLocal {

@PersistenceContext

private EntityManager em;

@Override

public void addStudent(Student student) {

em.merge(student);

em.flush();

}

@Override

public void editStudent(Student student) {

em.merge(student);

em.flush();

}

@Override

public void deleteStudent(int studentId) {

em.remove(getStudent(studentId));

em.flush();

}

@Override

public Student getStudent(int studentId) {

em.flush();

return em.find(Student.class, studentId);

}

@Override

public List<Student> getAllStudents() {

em.flush();

return em.createNamedQuery("Student.getAll").getResultList();

}

}

**StudentDAOLocal.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dao;

import com.model.Student;

import java.util.List;

import javax.ejb.Local;

/\*\*

\*

\* @author USER

\*/

@Local

public interface StudentDAOLocal {

void addStudent(Student student);

void editStudent(Student student);

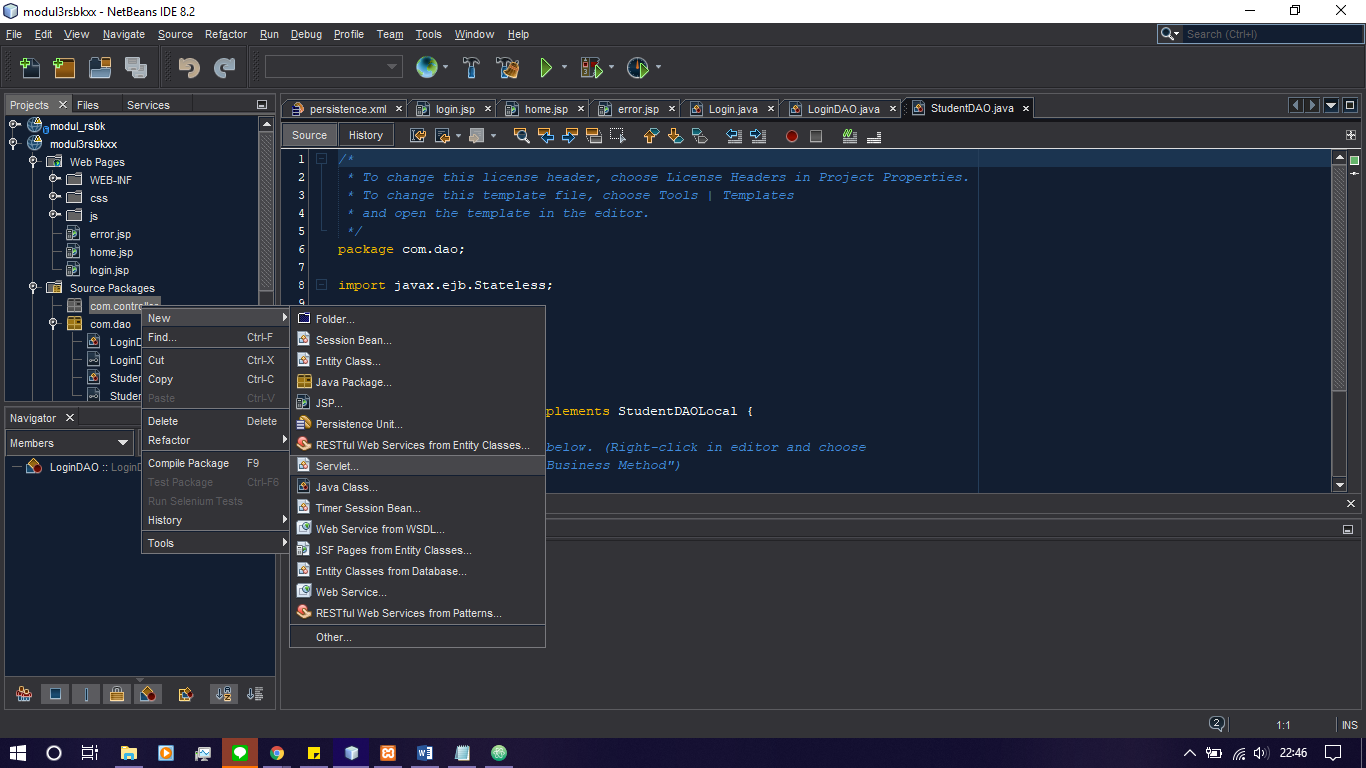
void deleteStudent(int studentId);

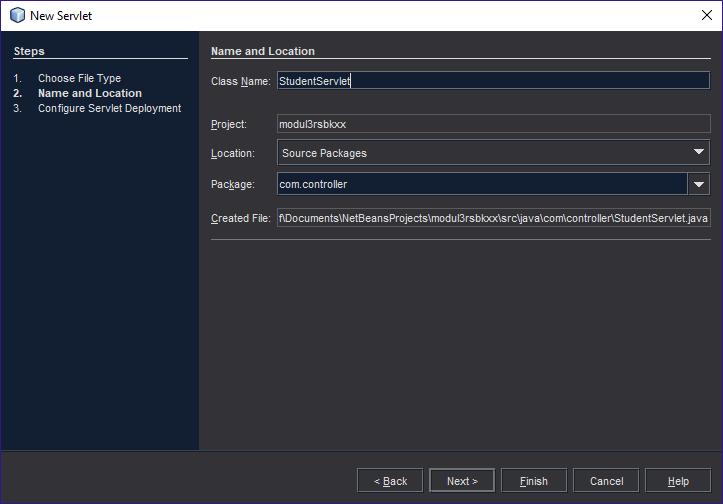
Student getStudent(int studentId);

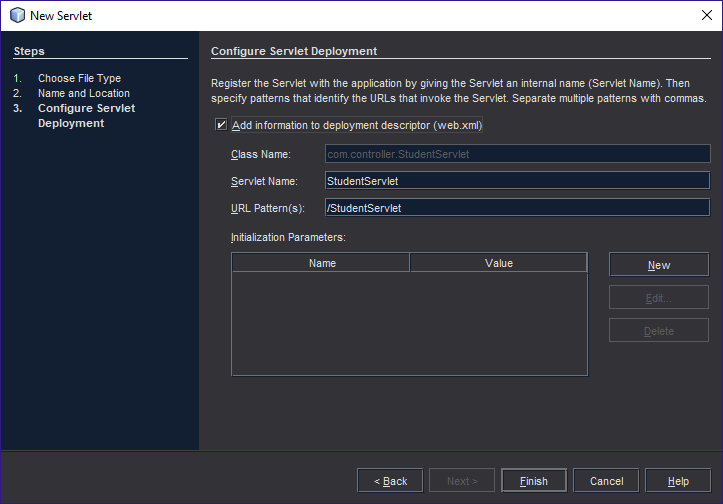
List<Student> getAllStudents();

}

1. Buat 2 Servlet pada package “com.controller”, beri nama : “LoginServlet” dan “StudentServlet”. Ketika membuat servlet, centang pada ‘Add information to deployment descriptor (web.xml)’. Kemudian masukkan source code yang tersedia.







**StudentServlet.java**

package com.controller;

import com.dao.StudentDAOLocal;

import com.model.Student;

import java.io.IOException;

import javax.ejb.EJB;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author USER

\*/

@WebServlet(name = "StudentServlet")

public class StudentServlet extends HttpServlet {

@EJB

private StudentDAOLocal studentDao;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String action = request.getParameter("action");

String studentIdStr = request.getParameter("studentId");

int studentId=0;

if(studentIdStr!=null && !studentIdStr.equals("")){

studentId=Integer.parseInt(studentIdStr);

}

String firstname = request.getParameter("firstname");

String lastname = request.getParameter("lastname");

Student student = new Student(studentId, firstname, lastname);

if("Add".equalsIgnoreCase(action)){

studentDao.addStudent(student);

}else if("Edit".equalsIgnoreCase(action)){

studentDao.editStudent(student);

}else if("Delete".equalsIgnoreCase(action)){

studentDao.deleteStudent(studentId);

}else if("Search".equalsIgnoreCase(action)){

student = studentDao.getStudent(studentId);

}

request.setAttribute("student", student);

request.setAttribute("allStudents", studentDao.getAllStudents());

request.getRequestDispatcher("home.jsp").forward(request, response);

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**LoginServlet.java**

package com.controller;

import com.dao.LoginDAOLocal;

import java.io.IOException;

import javax.ejb.EJB;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\*

\* @author USER

\*/

@WebServlet(name = "LoginServlet", urlPatterns = {"/LoginServlet"})

public class LoginServlet extends HttpServlet {

@EJB

private LoginDAOLocal loginDAO;

boolean check = false;

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

\* methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

String userName = request.getParameter("userName");

String password = request.getParameter("password");

HttpSession session = request.getSession();

check = loginDAO.checkUser(userName,password);

System.out.println("check is"+check+" "+userName);

if(check)

{

session.setAttribute("loginName", userName);

request.getRequestDispatcher("./StudentServlet").forward(request, response);

} else {

request.setAttribute("error", "Wrong Username or Password");

request.getRequestDispatcher("error.jsp").forward(request, response);

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

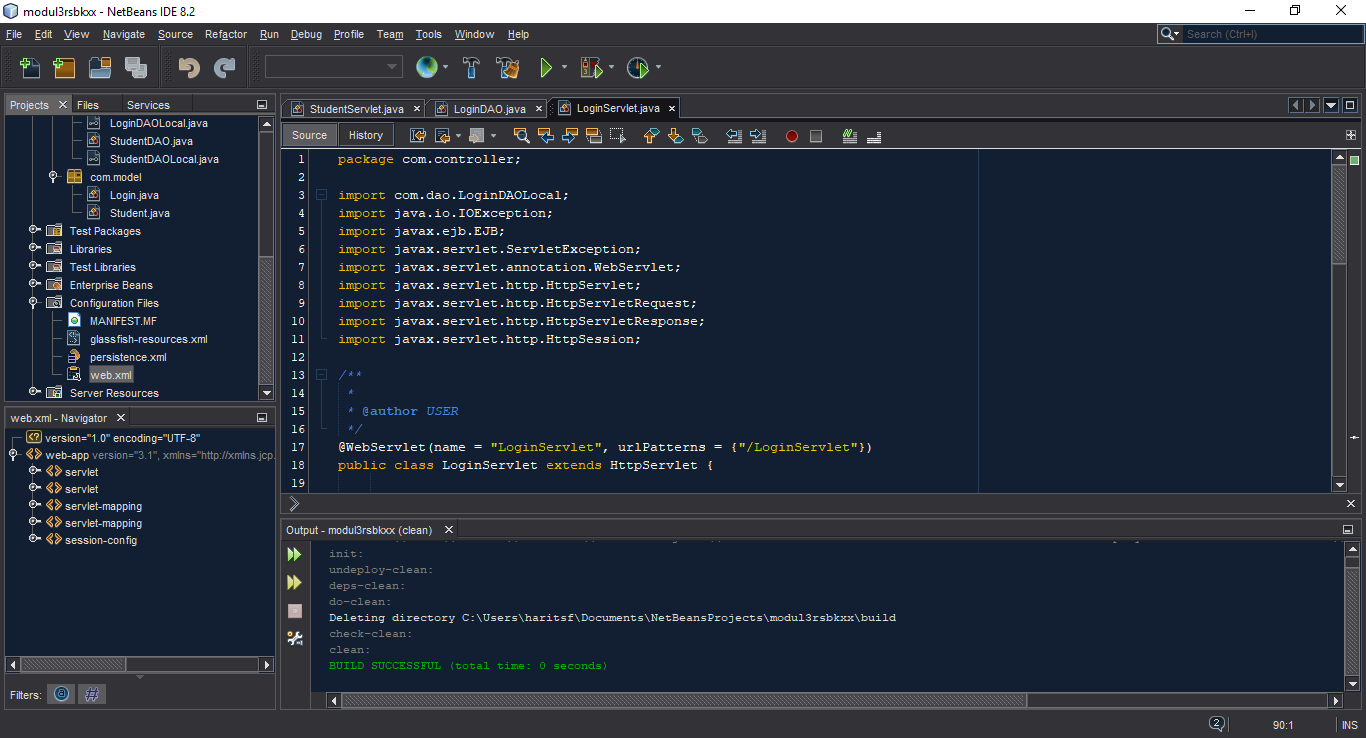
public String getServletInfo() {

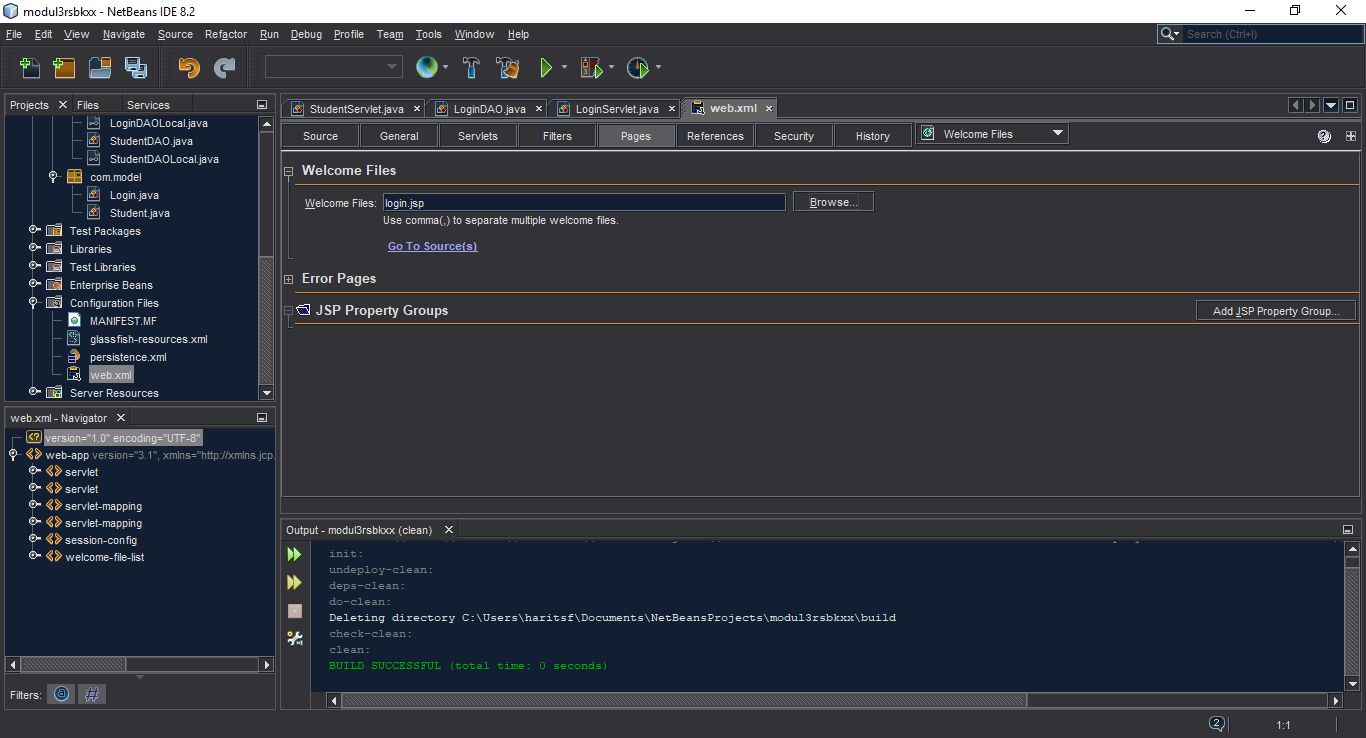
return "Short description";

}// </editor-fold>

}

1. Buka “web.xml” di folder ‘Configuration Files’. Pada tab ‘Pages’, isikan Welcome Files dengan “login.jsp”.





1. Lakukan ‘Clean and Build’, kemudian run http://localhost:8080/modul3rsbk**xx**