# BAB IV Aplikasi CRUD menggunakan EJB, JPA, dan MySQL

## 4.1 Tujuan

1. Praktikan dapat menggunakan operasi CRUD dengan *database* MySQL
2. Praktikan dapat mengimplementasikan EJB
3. Praktikan dapat mengimplementasikan JPA
4. Praktikan dapat menggunakan GlassFish *server* dalam pengembangan web menggunakan Bahasa pemrograman Java.

## 4.2 Dasar Teori

### 4.2.1 Java *Persistance* API

Java Persistence API merupakan *tool* untuk mengolah ataupun pengatur data relational dalam platform Java Standard *Edit*ion dan Java Enterprise *Edit*ion. JPA sendiri merupakan alat dalam pembuatan aplikasi berbentuk framework dalam pemrograman java dengan pendekatan *Object Relational Maping* (ORM). ORM sendiri merupakan sebuah konsep yang berdiri sendiri, tidak terkait dengan Java. Namun hubungan ORM dengan JPA sangat dekat karena JPA merupakan standart ORM dalam Java, dan harus diikuti oleh pengguna ORM di Java agar ada standart yang sama antara ORM di Java dengan yang diluar Java. Dengan menggunakan JPA, memungkinkan manipulasi data tanpa menggunakan query, namun bukan berarti tanpa menggunakan query sama sekali, tetap ada penggunaan query disana. Cara JPA ini dinilai lebih baik dari teknik manipulasi data dengan jdbc. Jika kita menggunakan JPA, maka cara kita terhubung ke *database* sama semua, baik pakai MySQl, SQL *Server* ataupun PostgreSQL. API JPA terdapat dalam *package* javax.persistence. Di dalamnya mengandung Query khusus yang disebut (JPQL)Java Persistence Query Language. Beberapa Library yang mengimplementasikan JPA antara lain adalah Hibernate dan EclipseLink. Kelebihan JPA yang cukup bermanfaat adalah tidak perlu membuat query untuk manipulasi data. Selain itu kita dapat dengan mudah mengelola transaksi dengan API. Kita bisa menghindari pembuatan *Data Access Object* yang rumit dan komplek sekali jika menggunakan JPA ini. Dan yang cukup bagus, kita juga dapat mengelola Plain Old Java *Object* disini.

*(Sumber :* [*https://dartoblog.wordpress.com/2012/08/01/pengenalan-jpa-java-persistence-api/*](https://dartoblog.wordpress.com/2012/08/01/pengenalan-jpa-java-persistence-api/)*)*

### 3.2.2 Session Bean

Session bean adalah EJB yang digunakan untuk mengeksekusi proses. Isi dari Session Bean ini biassanya berupa kata kerja (*transfer, pay, calculate*, updateData, dll). Stateless Session Bean (SLSB) adalah Session Bean yang tidak menyimpan state (keadaan) pada setiap kali eksekusi. Berbeda dengan Statefull Session Bean (SFSB) yang dapat menyimpan state. State ini dapat kita gunakan misalnya untuk menyimpan informasi *user* atau barang-barang yang sudah dibeli (pada kasus online shop).

*(Sumber :* [*https://suhearie.wordpress.com/2008/08/26/java-enterprise-mulai-dari-mana-part-2-netbeans-glassfish/*](https://suhearie.wordpress.com/2008/08/26/java-enterprise-mulai-dari-mana-part-2-netbeans-glassfish/)*)*

### 4.2.3 Entity Unit

Entity Bean adalah EJB yang digunakan untuk mempermudah manipulasi *database*. Konsepnya adalah *Object* – Relational Mapping (ORM) yang berarti memetakan *object* dengan data di dalam *database*. Entity Bean sebenarnya adalah spesifikasi “bawaan” dari versi EJB sebelumnya yaitu EJB 2.1. Dalam EJB3, ada API lain yang lebih sederhana yaitu Java Persistence API (JPA). Sekarang orang lebih banyak menggunakan JPA dibandingkan Entity Bean.

*(Sumber :* [*https://suhearie.wordpress.com/2008/08/26/java-enterprise-mulai-dari-mana-part-2-netbeans-glassfish/*](https://suhearie.wordpress.com/2008/08/26/java-enterprise-mulai-dari-mana-part-2-netbeans-glassfish/)*)*

### 4.2.4 Java Servlet

Servlet adalah bahasa pemrograman Java kelas digunakan untuk memperluas kemampuan dari *server* yang tuan aplikasi mengakses melalui model pemrograman *request*-respon. Meskipun servlet dapat menanggapi setiap jenis permintaan, mereka biasanya digunakan untuk memperpanjang aplikasi host oleh *server* Web. Dengan demikian, dapat dianggap sebagai Java Applet yang berjalan pada *server* bukan browser.

Servlet berbasis Java *server*-side teknologi web. Sesuai namanya, melayani permintaan klien dan menerima respon dari *server*. Secara teknis, Servlet adalah kelas Java dalam Java EE yang sesuai dengan Java Servlet API, sebuah protokol di mana kelas Java mungkin merespon permintaan. Mereka tidak terikat dengan sebuah protokol client-*server* khusus, tetapi yang paling sering digunakan dengan protokol HTTP. Oleh karena itu, kata “Servlet” sering digunakan dalam arti “Servlet HTTP”. [2] Dengan demikian, seorang pengembang perangkat lunak dapat menggunakan servlet untuk menambahkan konten dinamis ke *server* Web menggunakan platform Java. Isi dihasilkan umumnya HTML, tetapi mungkin data lain seperti XML. Servlets adalah mitra Jawa non-Jawa teknologi konten web yang dinamis seperti PHP dan ASP.NET. Servlets dapat mempertahankan negara dalam variabel sesi transaksi di *server* banyak dengan menggunakan cookie HTTP, atau menulis ulang URL.  
Untuk menyebarkan dan menjalankan Servlet, wadah Web harus digunakan. Sebuah wadah Web (juga dikenal sebagai wadah Servlet) pada dasarnya adalah komponen Web *server* yang berinteraksi dengan servlet. Wadah Web bertanggung jawab untuk mengelola siklus hidup servlets, pemetaan URL ke servlet tertentu dan memastikan bahwa URL pemohon memiliki hak akses yang benar.  
API servlet, yang terkandung dalam hirarki javax.servlet paket Java, mendefinisikan interaksi yang diharapkan dari wadah Web dan servlet [2].  
Servlet adalah sebuah objek yang menerima permintaan dan menghasilkan respon berdasarkan permintaan itu. Paket servlet mendefinisikan dasar Jawa objek untuk mewakili servlet permintaan dan tanggapan, serta sebagai objek untuk mencerminkan konfigurasi servlet parameter dan lingkungan eksekusi. Paket javax.servlet.http mendefinisikan HTTP spesifik sub*class* dari elemen servlet generik, termasuk objek manajemen sesi yang melacak beberapa permintaan dan tanggapan antara Web *server* dan klien. Servlets dapat dikemas dalam *file* WAR sebagai aplikasi Web.  
Servlets dapat dihasilkan secara otomatis dari Java*Server* *Page*s (JSP) oleh kompilator Java*Server* *Page*s. Perbedaan antara Servlets dan JSP adalah bahwa Servlets biasanya menanamkan HTML di dalam kode Java, sedangkan JSP embed kode Java dalam HTML. Sedangkan penggunaan langsung dari Servlets untuk menghasilkan HTML (seperti yang ditunjukkan pada contoh di bawah) telah menjadi langka, tingkat yang lebih tinggi MVC kerangka web di Java EE (JSF) masih secara eksplisit menggunakan teknologi Servlet untuk penanganan permintaan / tanggapan tingkat rendah melalui FacesServlet yang . Sebuah penggunaan agak lebih tua adalah dengan menggunakan servlet bersama dengan JSP dalam pola yang disebut “Model 2”, yang merupakan rasa pola model-*view*-*controlletr*.

*(Sumber :* [*https://aeroyid.wordpress.com/2012/07/16/javamemperkenalkan-java-servlet-dan-membuat-helloworld/*](https://aeroyid.wordpress.com/2012/07/16/javamemperkenalkan-java-servlet-dan-membuat-helloworld/)*)*

### 4.2.5 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-*user*. MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial. Pada saat ini MySQL merupakan *database* *server* yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses *database* yaitu SQL. SQL (Structured Query Language) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan *database* lebih *user*-friendly dibandingan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemograman seperti C, dan Delphi.

*(Sumber :* [*https://upyes.wordpress.com/2013/02/06/pengertian-dan-sejarah-mysql/*](https://upyes.wordpress.com/2013/02/06/pengertian-dan-sejarah-mysql/)*)*

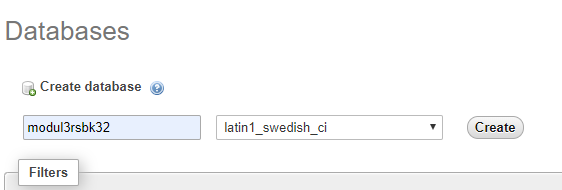
## 4.3 Langkah Kerja

1. Jalankan XAMPP, jika ada bentrok dengan salah satu *PORT*, matikan *PORT* tersebut lalu nyalakan kembali (biasanya bentrok sama vmware/oracle)

XAMPP yang di gunakan adalah MySQL dan Apache untuk konfigurasi *database*.

1. Buat *database* di MySQL. Beri nama “modul3rsbk32” ganti xx dengan kelompok.

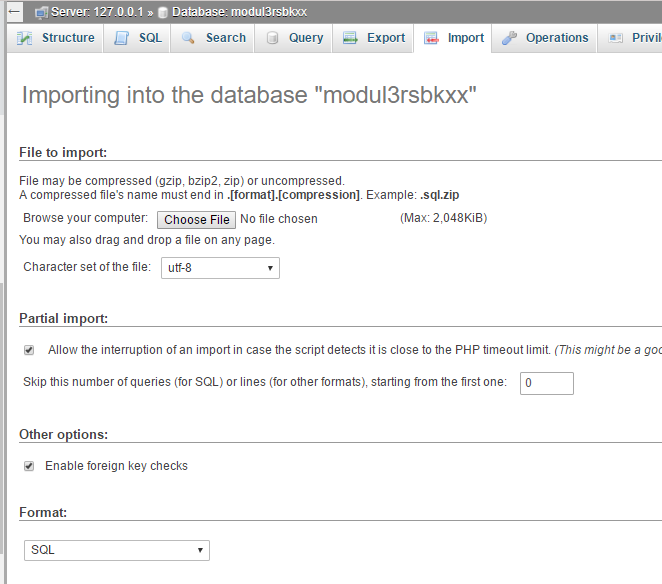
Ini adalah *database* yang akan digunakan untuk aplikasi nanti.



Gambar 4.1 Membuat *database* baru

1. im*port* *file* sql ke Mysql anda, pilih *file* .sql nya dan pilih go\

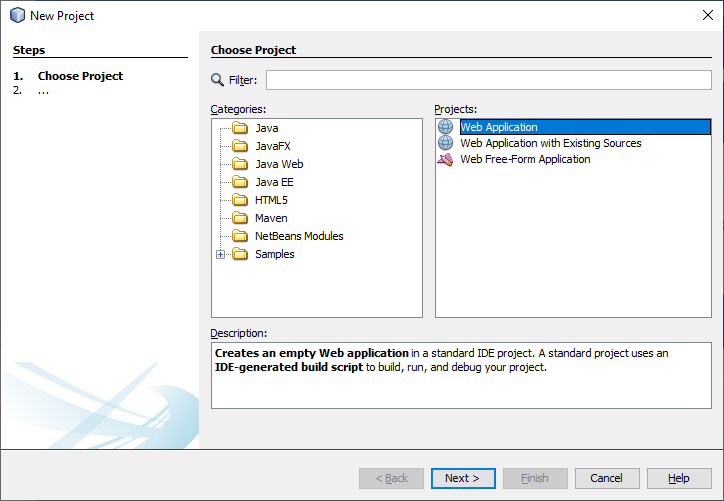
*file* sql berisi query yang akan di eksekusi ketika mengim*port*nya ke dalam *database*.



Gambar 4.2 Mengim*port* *file* sql

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

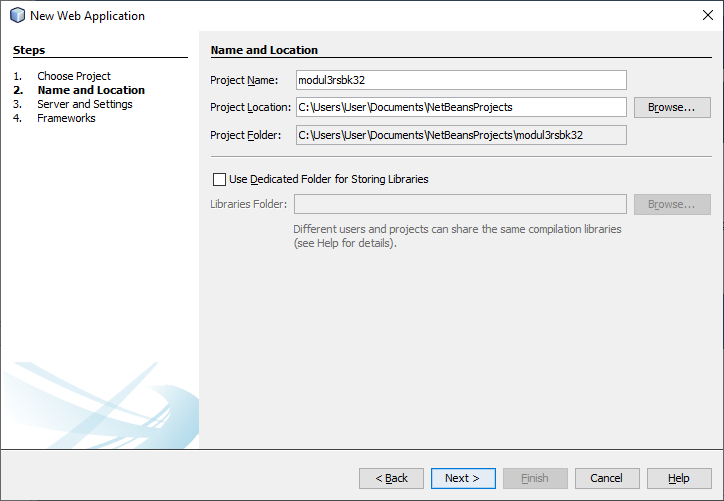
Java web application adalah sebuah project yang dapat digunakan untuk mengembangkan web berbasis Bahasa pemrograman Java.



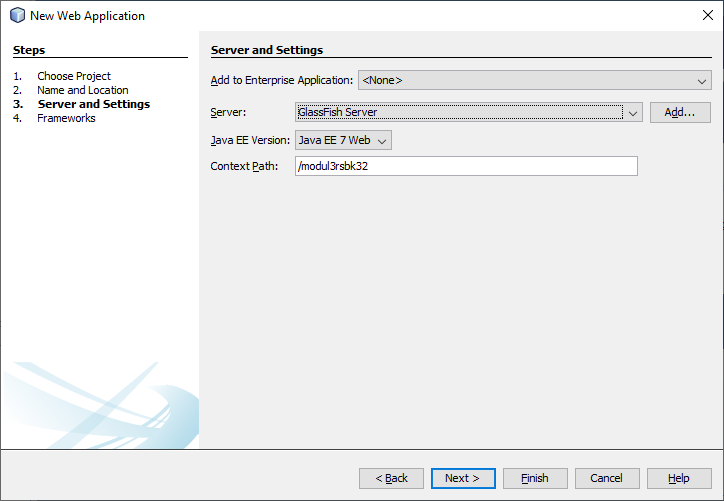
Gambar 4.3 Membuat project web application

1. Beri nama project “**modul3rsbk32**” dan pilih Glassfish *Server* (xx nomor kelompok).

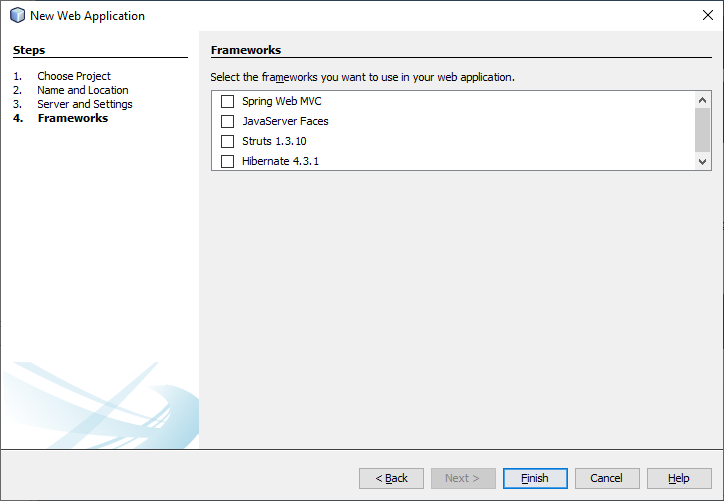
Glassfish *server* adalah sebuah web *server* yang digunakan untuk mengembangkan aplikasi berbasis web menggunakan Bahasa pemrograman Java.



Gambar 4.4 Memberi nama project



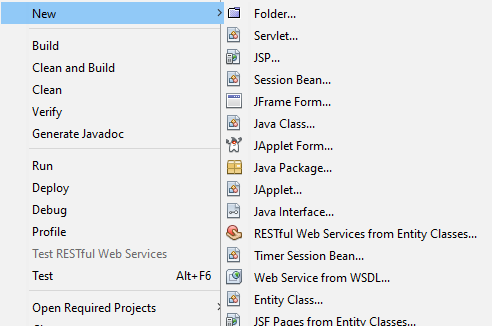
Gambar 4.5 Memilih GlassFish *Server*



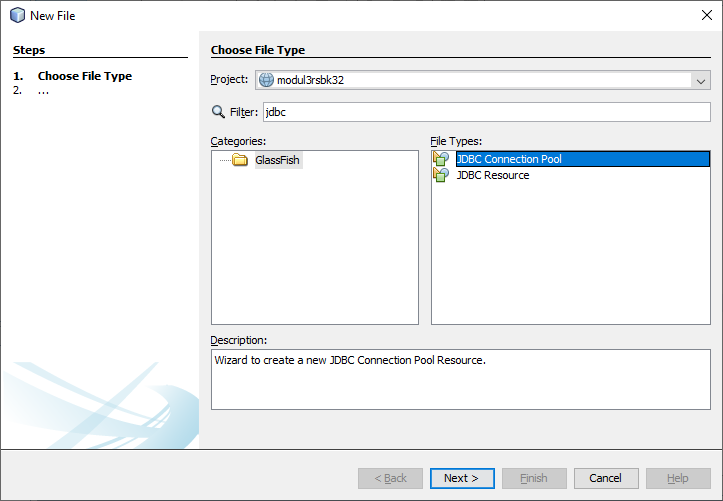
Gambar 4.6 *Finish* pembuatan project

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

JDBC Connection Pool adalah koneksi untuk menghubungkan Netbeans dengan *Database*.

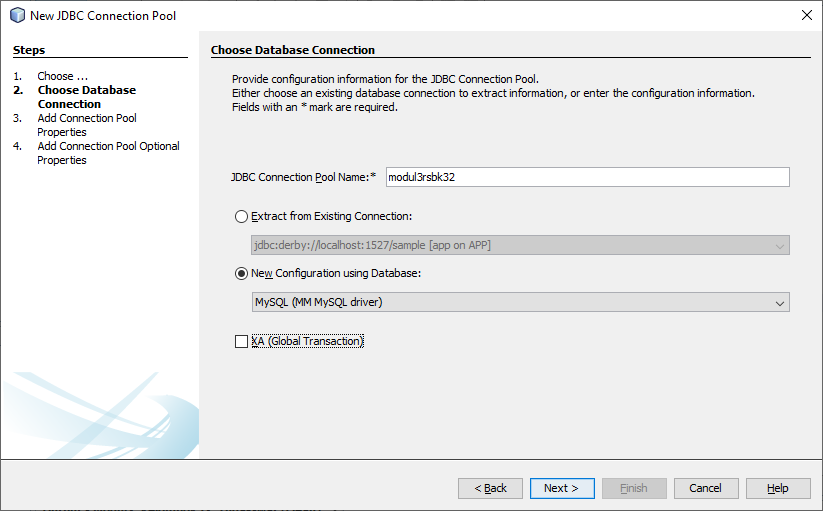


Gambar 4.7 Memilih other untuk menambahkan *file* connection pool



Gambar 4.8 Membuat connection pool baru

1. Beri nama connection pool (sama seperti nama projectnya, misal modul3rsbk32) dan pilih ‘*New* Configuration using *Database*’ → ‘MySQL (MM MySQL driver)

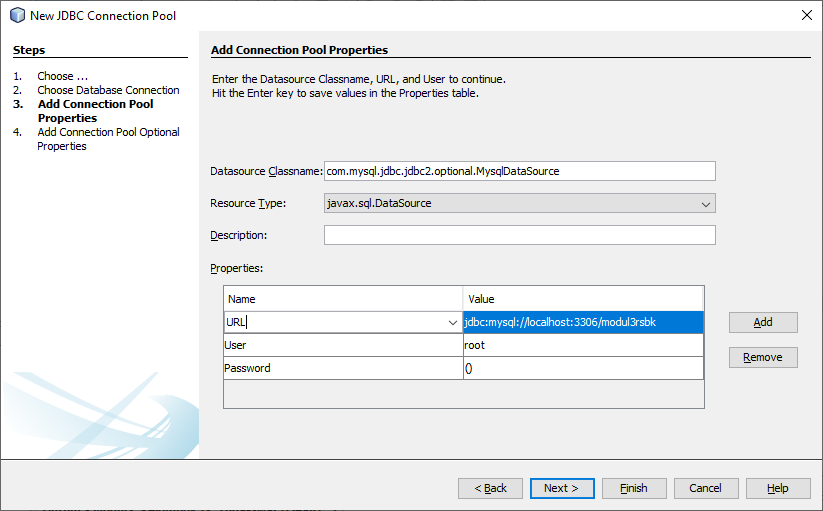


Gambar 4.9 Memilih koneksi untuk *database* MySQL

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*.

Ini adalah konfigurasi untuk mengakses *database* MySQL.

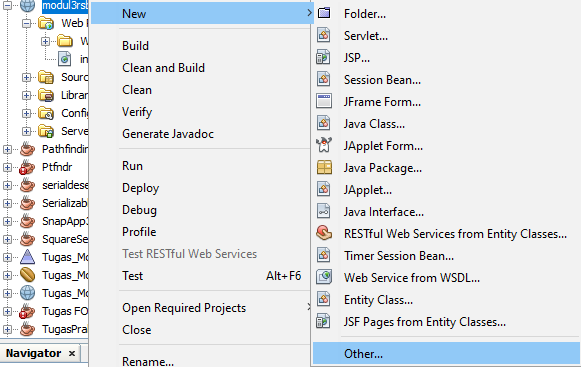
jdbc:mysql://localhost:3306/modul3rsbkxx



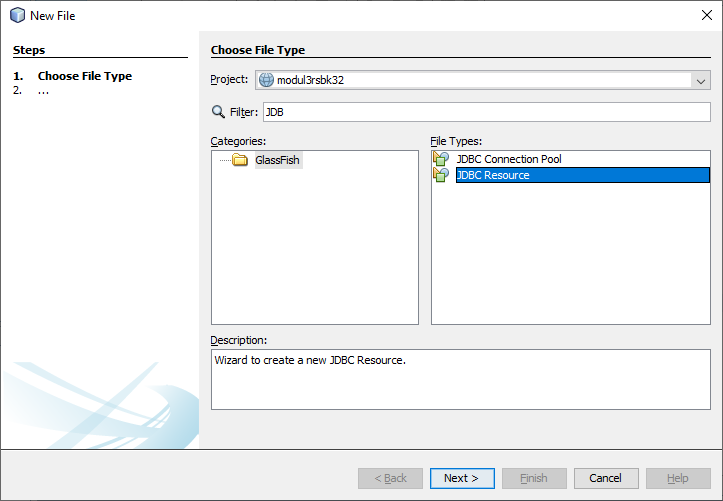
Gambar 4.10 Mengkonfigurasi koneksi

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

JDBC Resource bisa dianggap sebagai alat untuk memilih *database* yang di maksud dari koneksi yang sudah dibuat.



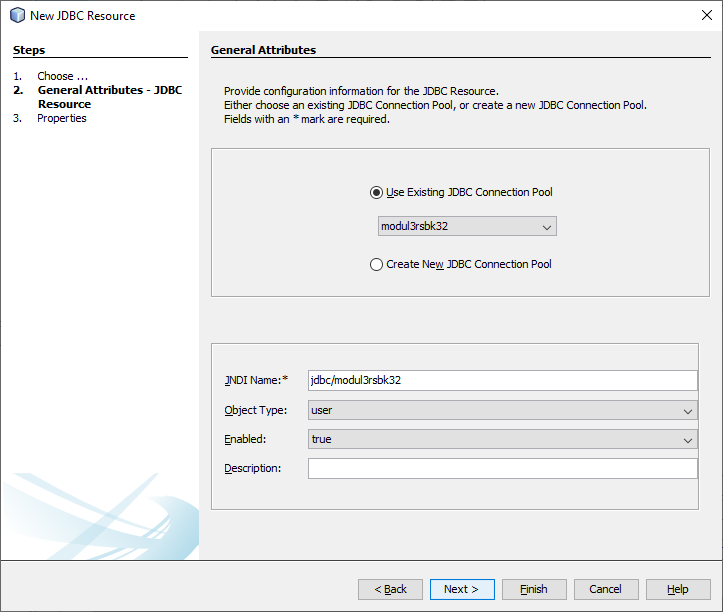
Gambar 4.11 Memilih other untuk membuat *file* JDBC Resource



Gambar 4.12 Membuat JDBC Resource baru

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*.

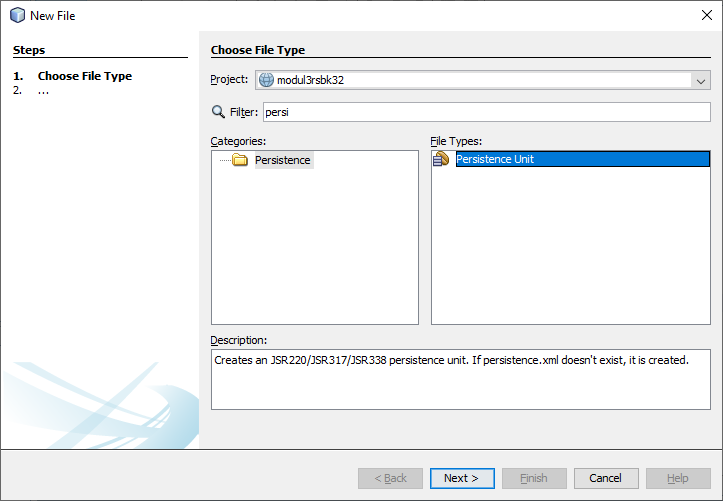
Memilik koneksi yang sudah dibuat. Disini yang dipilih adalah koneksi untuk MySQL yang sudah dibuat di langkah sebelumnya.



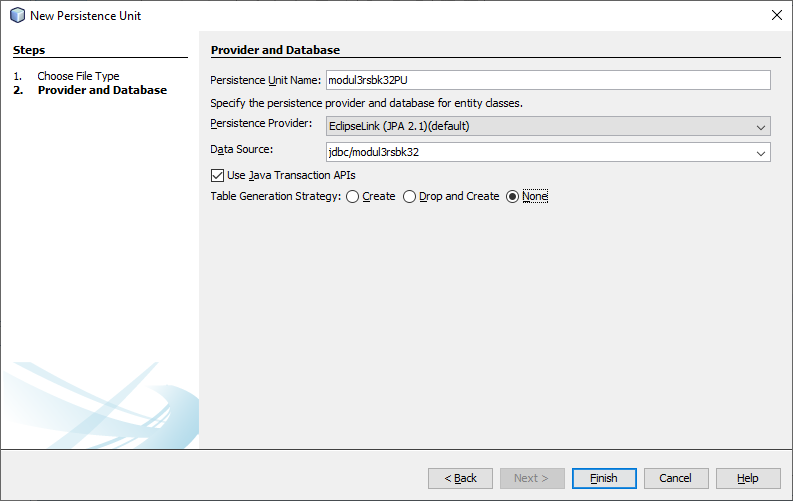
Gambar 4.13 Memilih Connecetion pool yang sudah dibuat

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’

Setelah membuat koneksi, presistance unit digunakan untuk memilih *database* yang akan digunakan dan menghubungkannya dengan Glassfish *server*.



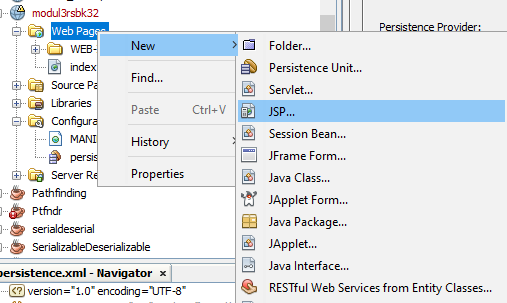
Gambar 4.14 Membuat Persistance Unit baru



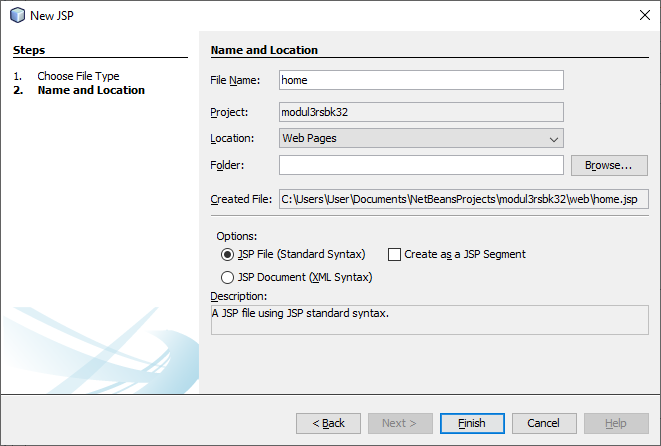
Gambar 4.15 Memilih JDBC resource yang sudah dibuat

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

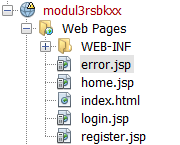
JSP bisa di bilang sebagai *file* tampilan yang menggunakan HTML sebagai bahasanya. Seperti pada pemrograman web.



Gambar 4.16 Membuat *file* JSP baru



Gambar 4.17 Contoh membuat JSP *home*



Gambar 4.18 *File* JSP yang telah dibuat

***register*.jsp**

|  |
| --- |
| <%--  Document : *register*  Created on : Sep 22, 2019, 2:48:05 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="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>*Register* 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">  <a href="./*login*.jsp" *class*="btn btn-md btn-success" style="float:right" />*Login*</a><br>  <center><h2><b>Data Mahasiswa</b></h2>  <h4>**Modul RSBK – Kelompok32**</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 Up</p>  <form method="POST" action="./*Register*sServlet">  <div *class*="form-group">  <label>*User*name</label>  <input type="text" *class*="form-control" placeholder="*User*name" name="*user*Name" 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="*register*" value="*Register*" *class*="btn btn-success" />  </form>  </div>  </div>  </div>  </div>  </body>  </html> |

***login*.jsp**

|  |
| --- |
| <%--  Document : *login*  Created on : Sep 22, 2019, 12:45:49 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="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">  <a href="./*register*.jsp" *class*="btn btn-md btn-success" style="float:right" />*Register*</a><br>  <center><h2><b>Data Mahasiswa</b></h2>  <h4>**Modul RSBK – Kelompok32**</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="./*Login*Servlet">  <div *class*="form-group">  <label>*User*name</label>  <input type="text" *class*="form-control" placeholder="*User*name" name="*user*Name" 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**

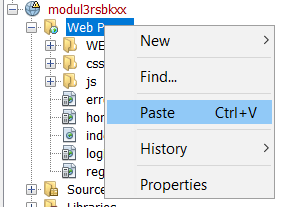
|  |
| --- |
| <%--  Document : *home*  Created on : Sep 22, 2019, 12:45:58 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="UTF-8"%>  <%@taglib prefix="s" uri="http://java.sun.com/jsp/jstl/core" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; 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("*login*Name")%></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="./*login*.jsp">Logout</a></li>  </ul>  </div>  <div *class*="col-md-9 daftar">  <form action="./*Student*Servlet" method="POST">  <table *class*="table table-bordered">  <tr>  <td>*Student* ID</td>  <td><input *class*="form-control" type="text" name="*student*Id" value="${*student*.*student*Id}" /></td>  </tr>  <tr>  <td>First Name</td>  <td><input *class*="form-control" type="text" name="firstname" value="${*student*.firstName}" /></td>  </tr>  <tr>  <td>Last Name</td>  <td><input *class*="form-control" type="text" name="lastname" value="${*student*.lastName}" /></td>  </tr>  <tr>  <td colspan="2">  <input type="submit" *class*="btn btn-primary btn-sm" name="action" value="Add" />  <input type="submit" *class*="btn btn-default btn-sm" name="action" value="*Edit*" />  <input type="submit" *class*="btn btn-danger btn-sm" name="action" value="*Delete*" />  <input type="submit" *class*="btn btn-warning btn-sm" name="action" value="Search" />  </td>  </tr>  </table>  </form>  <h3 align="center">Informasi Data</h3>  <table *class*="table table-bordered table-hover">  <thead>  <tr>  <th>No. ID</th>  <th>First Name</th>  <th>Last Name</th>  </tr>  </thead>  <tbody>  <s:forEach items="${all*Student*s}" var="stud">  <tr>  <td>${stud.*student*Id}</td>  <td>${stud.firstName}</td>  <td>${stud.lastName}</td>  </tr>  </s:forEach>  </tbody>  </table>  </div>  </div>  </div>  </html> |

***error*.jsp**

|  |
| --- |
| <%@*page* contentType="text/html" *page*Encoding="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. Masukkan asset dengan copy dan paste folder css dan js ke web *page*s.

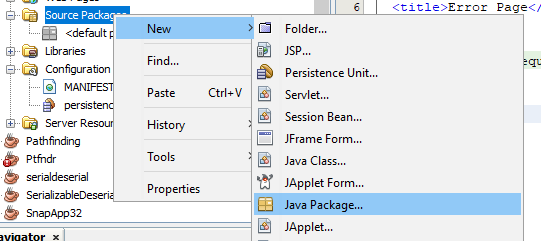
Asset ini berisi css untuk tampilan web.



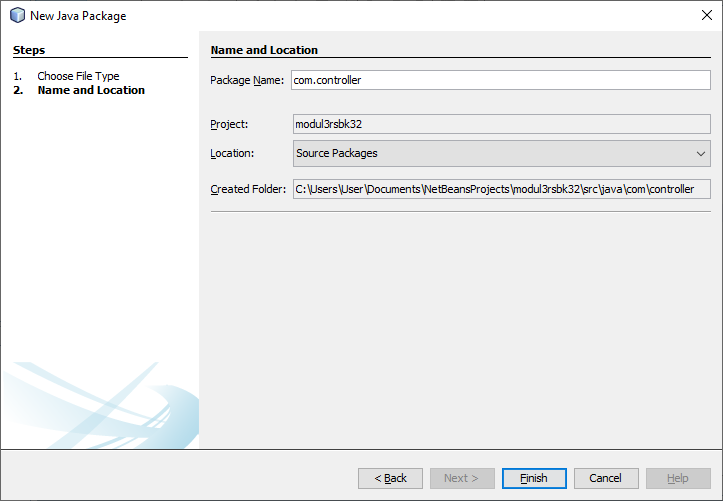
Gambar 4.19 Menambahkan Asset ke project

1. Buat 3 java *package* pada folder “Source *Package*s”, beri nama: “com.*controlletr*”, “com.dao”, dan “com.model”.

Ini adalah *package* yang digunakan untuk penerapan MVC.



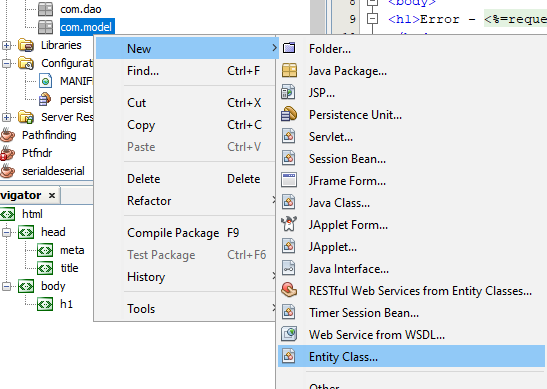
Gambar 4. 20 Membuat Java *Package* baru



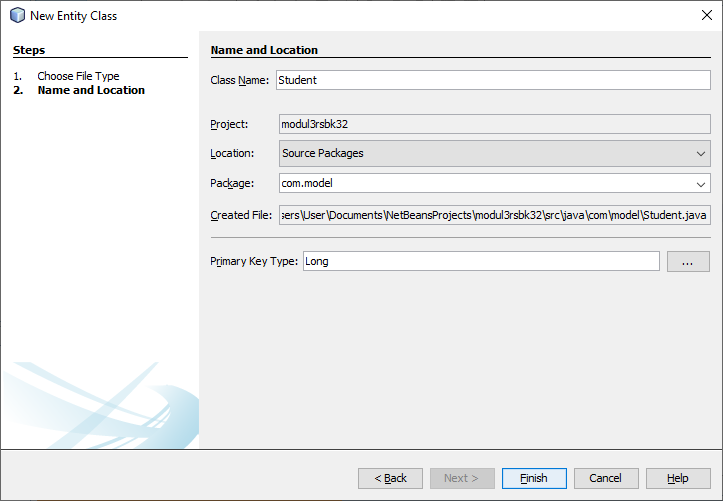
Gambar 4. 21 Contoh membuat Java *Package* com.*controlletr*

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

Ini adalah bagian model yang berisi *object* yang akan diolah pada aplikasi.



Gambar 4. 22 Membuat Entity *Class* baru



Gambar 4. 23 Contoh membuat Entity *Class* *Student*

***User*.java**

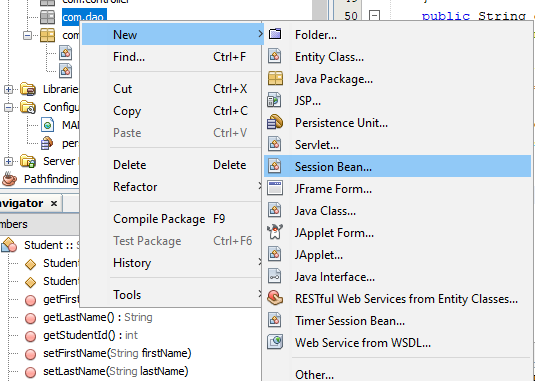
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.model;  im*port* java.io.Serializable;  im*port* javax.persistence.Entity;  im*port* javax.persistence.GeneratedValue;  im*port* javax.persistence.GenerationType;  im*port* javax.persistence.Id;  im*port* javax.persistence.NamedQueries;  im*port* javax.persistence.NamedQuery;  im*port* javax.persistence.Table;  im*port* javax.persistence.Column;  /\*\*  \*  \* @author *USER*  \*/  @Entity  @Table  @NamedQueries({@NamedQuery(name="*User*.getAll",query="SELECT e FROM *User* e")})  public *class* *User* implements Serializable {  private static final long serialVersionUID = 1L;  @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  private int *user*Id;  @Column  private String *user*Name;  @Column  private String password;  public *User*() {  }  public *User*(String *user*Name, String password) {  this.*user*Name = *user*Name;  this.password = password;  }  public int get*User*Id() {  return *user*Id;  }  public void set*User*Id(int *user*Id) {  this.*user*Id = *user*Id;  }  public String get*User*Name() {  return *user*Name;  }  public void set*User*Name(String *user*Name) {  this.*user*Name = *user*Name;  }  public String getPassword() {  return password;  }  public void setPassword(String password) {  this.password = password;  }  } |

***Student*.java**

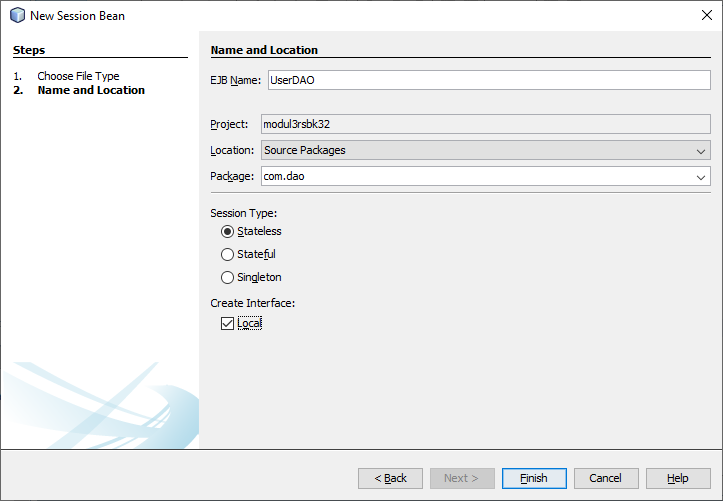
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.model;  im*port* java.io.Serializable;  im*port* javax.persistence.Entity;  im*port* javax.persistence.GeneratedValue;  im*port* javax.persistence.GenerationType;  im*port* javax.persistence.Id;  im*port* javax.persistence.NamedQueries;  im*port* javax.persistence.NamedQuery;  im*port* javax.persistence.Table;  im*port* javax.persistence.Column;  /\*\*  \*  \* @author *USER*  \*/  @Entity  @Table  @NamedQueries({@NamedQuery(name="*Student*.getAll",query="SELECT e FROM *Student* e order by e.*student*Id")})  public *class* *Student* implements Serializable {  @Id  @GeneratedValue(strategy=GenerationType.AUTO)  private int *student*Id;  @Column  private String firstName;  @Column  private String lastName;  public *Student*(int *student*Id, String firstName, String lastName) {  this.*student*Id = *student*Id;  this.firstName = firstName;  this.lastName = lastName;  }  public *Student*() {  }  public void set*Student*Id(int *student*Id) {  this.*student*Id = *student*Id;  }  public int get*Student*Id() {  return *student*Id;  }  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 : “*User*DAO” dan “*Student*DAO”. Ketika membuat session bean, pilih Session Type ‘**Stateless**’ dan Create Interface ‘**Local’**. Masukkan source code yang tersedia.

Pada *package* DAO berisi sebuah java session bean yang digunakan untuk interface antara model dengan *controlletr* nantinya.



Gambar 4. 24 Membuat session bean baru



Gambar 4. 25 Contoh membuat Session Bean *User*DAO

***User*DAO.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* javax.ejb.Stateless;  im*port* com.model.*User*;  im*port* java.util.List;  im*port* javax.persistence.EntityManager;  im*port* javax.persistence.PersistenceContext;  /\*\*  \*  \* @author WIN 10  \*/  @Stateless  public *class* *User*DAO implements *User*DAOLocal {  @PersistenceContext  private EntityManager em;  @Override  public boolean credential(String *user*Name, String password) {  List<*User*> s = (List<*User*>)em.createQuery("select e from *User* e where e.*user*Name='"+*user*Name+"' and e.password='"+password+"'").getResultList();  System.out.println("is list empty ?"+s.isEmpty()+" for the"+*user*Name+" and "+password);  if(!s.isEmpty())  return true;  else  return false;  }  @Override  public boolean check*User*(String *user*Name) {  List<*User*> s = (List<*User*>)em.createQuery("select e from *User* e where e.*user*Name='"+*user*Name+"'").getResultList();  if(s.isEmpty())  return true;  else  return false;  }  @Override  public void add*User*(*User* *user*){  em.merge(*user*);  em.flush();  }  // Add business logic below. (Right-click in *edit*or and choose  // "Insert Code > Add Business Method")  } |

***User*DAOLocal.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*User*;  im*port* javax.ejb.Local;  /\*\*  \*  \* @author WIN 10  \*/  @Local  public interface *User*DAOLocal {  public boolean credential(String *user*Name, String password);  public boolean check*User* (String *user*Name);  void add*User* (*User* *user*);  } |

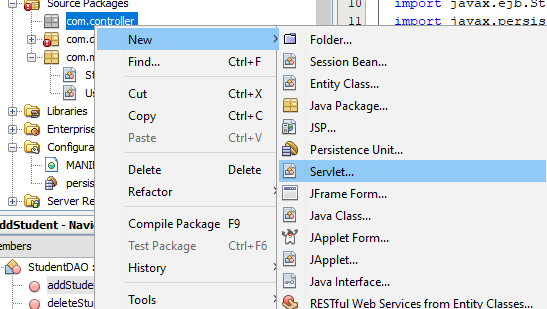
***Student*DAO.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*Student*;  im*port* java.util.List;  im*port* javax.ejb.Stateless;  im*port* javax.persistence.EntityManager;  im*port* javax.persistence.PersistenceContext;  /\*\*  \*  \* @author WIN 10  \*/  @Stateless  public *class* *Student*DAO implements *Student*DAOLocal {  @PersistenceContext  private EntityManager em;  @Override  public void add*Student*(*Student* *student*) {  em.merge(*student*);  em.flush();  }  @Override  public void *editStudent*(*Student* *student*) {  em.merge(*student*);  em.flush();  }  @Override  public void *deleteStudent*(int *student*Id) {  em.remove(get*Student*(*student*Id));  em.flush();  }  @Override  public *Student* get*Student*(int *student*Id) {  em.flush();  return em.find(*Student*.*class*, *student*Id);  }  @Override  public List<*Student*> getAll*Student*s() {  em.flush();  return em.createNamedQuery("*Student*.getAll").getResultList();  }  } |

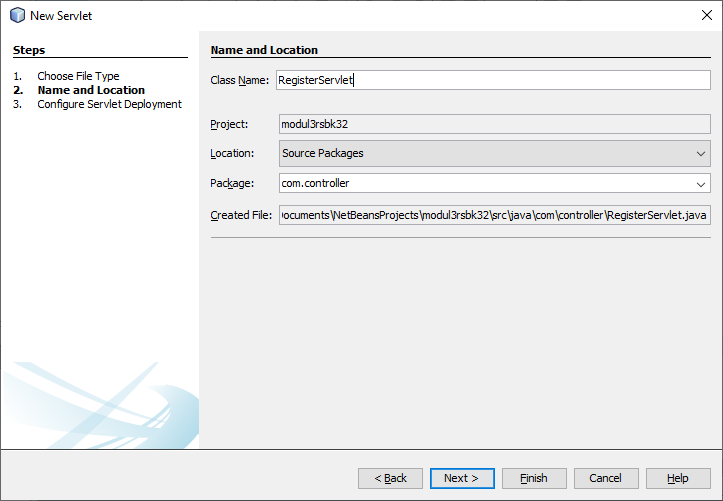
***Student*DAOLocal.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*Student*;  im*port* java.util.List;  im*port* javax.ejb.Local;  /\*\*  \*  \* @author WIN 10  \*/  @Local  public interface *Student*DAOLocal {  void add*Student*(*Student* *student*);  void *editStudent*(*Student* *student*);  void *deleteStudent*(int *student*Id);  *Student* get*Student*(int *student*Id);  List<*Student*> getAll*Student*s();  } |

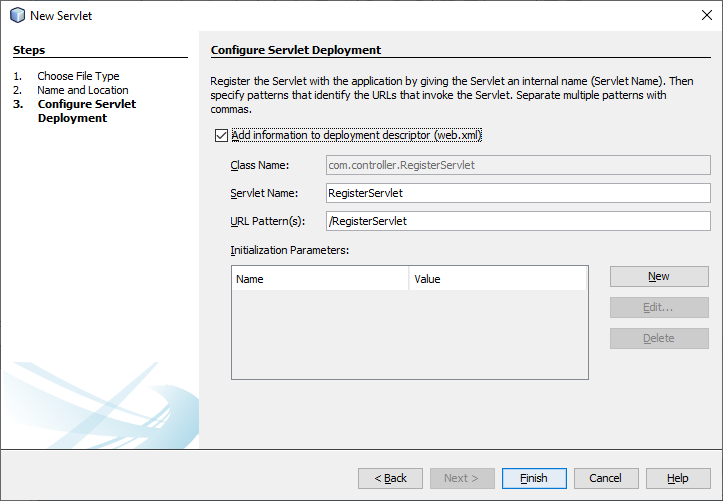
1. Buat 3 Servlet pada *package* “com.*controlletr*”, beri nama : “*Login*Servlet”, “*Register*sServlet”, “*Student*Servlet”. Ketika membuat servlet, centang pada ‘Add information to deployment descriptor (web.xml)’. Kemudian masukkan source code yang tersedia.



Gambar 4. 26 Membuat Servlet baru



Gambar 4. 27 Contoh membuat *Register*Servlet



Gambar 4. 28 Mengubah pattern url *Register*Servlet

***Login*Servlet.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.annotation.WebServlet;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  im*port* javax.servlet.http.HttpSession;  im*port* com.dao.*User*DAOLocal;  /\*\*  \*  \* @author WIN 10  \*/  @WebServlet(name = "*Login*Servlet", urlPatterns = {"/*Login*Servlet"})  public *class* *Login*Servlet extends HttpServlet {  @EJB  private *User*DAOLocal *user*DAO;  boolean check = false;    /\*\*  \* Processes *request*s 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 process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  String *user*Name = *request*.getParameter("*user*Name");  String password = *request*.getParameter("password");  HttpSession session = *request*.getSession();  check = *user*DAO.credential(*user*Name,password);  System.out.println("check is"+check+" "+*user*Name);  if(check)  {  session.setAttribute("*user*Name", *user*Name);  *request*.get*Request*Dispatcher("./*Student*Servlet").forward(*request*, response);  } else {  *request*.setAttribute("*error*", "Wrong *User*name or Password");  *request*.get*Request*Dispatcher("*error*.jsp").forward(*request*, response);  }  }  // <*edit*or-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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  /\*\*  \* Returns a short description of the servlet.  \*  \* @return a String containing servlet description  \*/  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

***Register*Servlet.java**

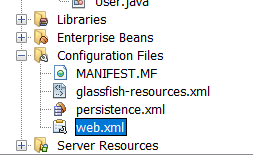
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* com.model.*User*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  im*port* com.dao.*User*DAOLocal;  /\*\*  \*  \* @author WIN 10  \*/  public *class* *Register*sServlet extends HttpServlet {  @EJB  private *User*DAOLocal *User*DAO;  boolean check = true;  /\*\*  \* Processes *request*s 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 process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  String *user*Name = *request*.getParameter("*user*Name");  String password = *request*.getParameter("password");  check = *User*DAO.check*User*(*user*Name);  System.out.println("check is"+check+" "+*user*Name);  if(check){  *User* *user* = *new* *User*(*user*Name, password);  *User*DAO.add*User*(*user*);  *request*.setAttribute("*user*", *user*);  *request*.get*Request*Dispatcher("*login*.jsp").forward(*request*, response);  }else{  *request*.setAttribute("*error*", "*User*name already taken");  *request*.get*Request*Dispatcher("*error*.jsp").forward(*request*, response);  }    }  // <*edit*or-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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  /\*\*  \* Returns a short description of the servlet.  \*  \* @return a String containing servlet description  \*/  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

***Student*Servlet.java**

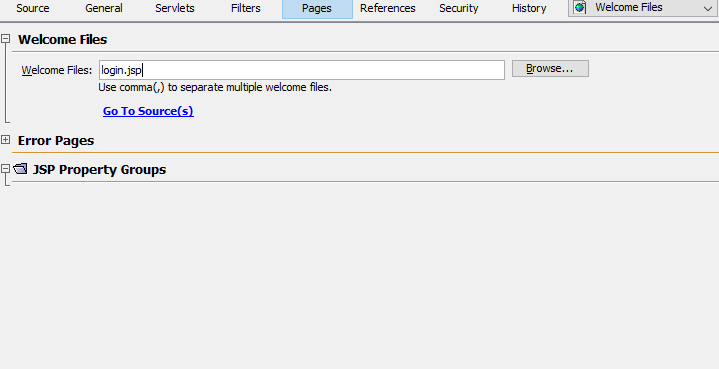
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* com.dao.*Student*DAOLocal;  im*port* com.model.*Student*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.annotation.WebServlet;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  /\*\*  \*  \* @author WIN 10  \*/  @WebServlet(name = "*Student*Servlet")  public *class* *Student*Servlet extends HttpServlet {  @EJB  private *Student*DAOLocal *student*Dao;  protected void process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  String action = *request*.getParameter("action");  String *student*IdStr = *request*.getParameter("*student*Id");  int *student*Id=0;  if(*student*IdStr!=null && !*student*IdStr.equals("")){  *student*Id=Integer.parseInt(*student*IdStr);  }  String firstname = *request*.getParameter("firstname");  String lastname = *request*.getParameter("lastname");  *Student* *student* = *new* *Student*(*student*Id, firstname, lastname);  if("Add".equalsIgnoreCase(action)){  *student*Dao.add*Student*(*student*);  }else if("*Edit*".equalsIgnoreCase(action)){  *student*Dao.*editStudent*(*student*);  }else if("*Delete*".equalsIgnoreCase(action)){  *student*Dao.*deleteStudent*(*student*Id);  }else if("Search".equalsIgnoreCase(action)){  *student* = *student*Dao.get*Student*(*student*Id);  }  *request*.setAttribute("*student*", *student*);  *request*.setAttribute("all*Student*s", *student*Dao.getAll*Student*s());  *request*.get*Request*Dispatcher("*home*.jsp").forward(*request*, response);  }  @Override  protected void doGet(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  @Override  protected void doPost(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

1. Buka “web.xml” di folder ‘Configuration *File*s’. Pada tab ‘*Page*s’, isikan Welcome *File*s dengan “*login*.jsp”.

Ini adalah *file* yang digunakan untuk meng*edit* tampilan awal.



Gambar 4. 29 Mengubah isi dari *file* web.xml



Gambar 4.30 Mengganti halaman awal web menjadi *login*

1. Lakukan ‘Clean and Build’ (Shift + F11), kemudian ‘Run’ (F6)

## 4.4 Hasil Percobaan

***Package* dao**

***User*DAO.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* javax.ejb.Stateless;  im*port* com.model.*User*;  im*port* java.util.List;  im*port* javax.persistence.EntityManager;  im*port* javax.persistence.PersistenceContext;  /\*\*  \*  \* @author WIN 10  \*/  @Stateless  public *class* *User*DAO implements *User*DAOLocal {  @PersistenceContext  private EntityManager em;  @Override  public boolean credential(String *user*Name, String password) {  List<*User*> s = (List<*User*>)em.createQuery("select e from *User* e where e.*user*Name='"+*user*Name+"' and e.password='"+password+"'").getResultList();  System.out.println("is list empty ?"+s.isEmpty()+" for the"+*user*Name+" and "+password);  if(!s.isEmpty())  return true;  else  return false;  }  @Override  public boolean check*User*(String *user*Name) {  List<*User*> s = (List<*User*>)em.createQuery("select e from *User* e where e.*user*Name='"+*user*Name+"'").getResultList();  if(s.isEmpty())  return true;  else  return false;  }  @Override  public void add*User*(*User* *user*){  em.merge(*user*);  em.flush();  }  // Add business logic below. (Right-click in *edit*or and choose  // "Insert Code > Add Business Method")  } |

Pada baris code di *User*DAO berguna untuk penghubung dari *database* ke aplikasi untuk mengolah *database* berupa pengeksekusian query terhadap *user*. Di dalamnya terdapat fungsi pengambilan data dan memasukannya kedalam arraylist model *User*.

***User*DAOLocal.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*User*;  im*port* javax.ejb.Local;  /\*\*  \*  \* @author WIN 10  \*/  @Local  public interface *User*DAOLocal {  public boolean credential(String *user*Name, String password);  public boolean check*User* (String *user*Name);  void add*User* (*User* *user*);  } |

Pada baris code di *User*DAOLocal berguna untuk penghubung dari aplikasi ke *database*. *File* ini khusus untuk menghubungkan antar fungsi yang ada pada *controlletr* *user* sehingga dapat diolah dalam pemrograman menggunakan Bahasa java.

***Student*DAO.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*Student*;  im*port* java.util.List;  im*port* javax.ejb.Stateless;  im*port* javax.persistence.EntityManager;  im*port* javax.persistence.PersistenceContext;  /\*\*  \*  \* @author WIN 10  \*/  @Stateless  public *class* *Student*DAO implements *Student*DAOLocal {  @PersistenceContext  private EntityManager em;  @Override  public void add*Student*(*Student* *student*) {  em.merge(*student*);  em.flush();  }  @Override  public void *editStudent*(*Student* *student*) {  em.merge(*student*);  em.flush();  }  @Override  public void *deleteStudent*(int *student*Id) {  em.remove(get*Student*(*student*Id));  em.flush();  }  @Override  public *Student* get*Student*(int *student*Id) {  em.flush();  return em.find(*Student*.*class*, *student*Id);  }  @Override  public List<*Student*> getAll*Student*s() {  em.flush();  return em.createNamedQuery("*Student*.getAll").getResultList();  }  } |

Pada baris code di *Student*DAO berguna untuk penghubung dari *database* ke aplikasi untuk mengolah *database* berupa pengeksekusian query terhadap *student*. Di dalamnya terdapat fungsi pengambilan data dan memasukannya ke dalam arraylist model *Student*.

***Student*DAOLocal.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.dao;  im*port* com.model.*Student*;  im*port* java.util.List;  im*port* javax.ejb.Local;  /\*\*  \*  \* @author WIN 10  \*/  @Local  public interface *Student*DAOLocal {  void add*Student*(*Student* *student*);  void *editStudent*(*Student* *student*);  void *deleteStudent*(int *student*Id);  *Student* get*Student*(int *student*Id);  List<*Student*> getAll*Student*s();  } |

Pada baris code di *Student*DAOLocal berguna untuk penghubung dari aplikasi ke *database*. *File* ini khusus untuk menghubungkan antar fungsi yang ada pada *controlletr* *student* sehingga dapat diolah dalam pemrograman menggunakan Bahasa java.

***Package* *controlletr***

***Login*Servlet.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.annotation.WebServlet;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  im*port* javax.servlet.http.HttpSession;  im*port* com.dao.*User*DAOLocal;  /\*\*  \*  \* @author WIN 10  \*/  @WebServlet(name = "*Login*Servlet", urlPatterns = {"/*Login*Servlet"})  public *class* *Login*Servlet extends HttpServlet {  @EJB  private *User*DAOLocal *user*DAO;  boolean check = false;    /\*\*  \* Processes *request*s 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 process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  String *user*Name = *request*.getParameter("*user*Name");  String password = *request*.getParameter("password");  HttpSession session = *request*.getSession();  check = *user*DAO.credential(*user*Name,password);  System.out.println("check is"+check+" "+*user*Name);  if(check)  {  session.setAttribute("*user*Name", *user*Name);  *request*.get*Request*Dispatcher("./*Student*Servlet").forward(*request*, response);  } else {  *request*.setAttribute("*error*", "Wrong *User*name or Password");  *request*.get*Request*Dispatcher("*error*.jsp").forward(*request*, response);  }  }  // <*edit*or-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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  /\*\*  \* Returns a short description of the servlet.  \*  \* @return a String containing servlet description  \*/  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

*File* ini berguna untuk menghubungkan pengolahan aplikasi dengan tampilan web. Disini berisi pengkondisian dan pengaturan terhadap data pada model, kemudian menampilkannya pada halaman *login*, ataupun sebaliknya pengolahan data yang dimasukan melalui halaman jsp *login* ke *database* melalui *file* interface DAO.

***Register*sServlet.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* com.model.*User*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  im*port* com.dao.*User*DAOLocal;  /\*\*  \*  \* @author WIN 10  \*/  public *class* *Register*sServlet extends HttpServlet {  @EJB  private *User*DAOLocal *User*DAO;  boolean check = true;  /\*\*  \* Processes *request*s 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 process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  String *user*Name = *request*.getParameter("*user*Name");  String password = *request*.getParameter("password");  check = *User*DAO.check*User*(*user*Name);  System.out.println("check is"+check+" "+*user*Name);  if(check){  *User* *user* = *new* *User*(*user*Name, password);  *User*DAO.add*User*(*user*);  *request*.setAttribute("*user*", *user*);  *request*.get*Request*Dispatcher("*login*.jsp").forward(*request*, response);  }else{  *request*.setAttribute("*error*", "*User*name already taken");  *request*.get*Request*Dispatcher("*error*.jsp").forward(*request*, response);  }    }  // <*edit*or-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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*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(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  /\*\*  \* Returns a short description of the servlet.  \*  \* @return a String containing servlet description  \*/  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

*File* ini berguna untuk menghubungkan pengolahan aplikasi dengan tampilan web. Disini berisi pengkondisian dan pengaturan terhadap data pada model, kemudian menampilkannya pada halaman *register*, ataupun sebaliknya pengolahan data yang dimasukan melalui halaman jsp *register* ke *database* melalui *file* interface DAO.

***Student*Servlet.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.*controlletr*;  im*port* com.dao.*Student*DAOLocal;  im*port* com.model.*Student*;  im*port* java.io.IOException;  im*port* javax.ejb.EJB;  im*port* javax.servlet.ServletException;  im*port* javax.servlet.annotation.WebServlet;  im*port* javax.servlet.http.HttpServlet;  im*port* javax.servlet.http.HttpServlet*Request*;  im*port* javax.servlet.http.HttpServletResponse;  /\*\*  \*  \* @author WIN 10  \*/  @WebServlet(name = "*Student*Servlet")  public *class* *Student*Servlet extends HttpServlet {  @EJB  private *Student*DAOLocal *student*Dao;  protected void process*Request*(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  String action = *request*.getParameter("action");  String *student*IdStr = *request*.getParameter("*student*Id");  int *student*Id=0;  if(*student*IdStr!=null && !*student*IdStr.equals("")){  *student*Id=Integer.parseInt(*student*IdStr);  }  String firstname = *request*.getParameter("firstname");  String lastname = *request*.getParameter("lastname");  *Student* *student* = *new* *Student*(*student*Id, firstname, lastname);  if("Add".equalsIgnoreCase(action)){  *student*Dao.add*Student*(*student*);  }else if("*Edit*".equalsIgnoreCase(action)){  *student*Dao.*editStudent*(*student*);  }else if("*Delete*".equalsIgnoreCase(action)){  *student*Dao.*deleteStudent*(*student*Id);  }else if("Search".equalsIgnoreCase(action)){  *student* = *student*Dao.get*Student*(*student*Id);  }  *request*.setAttribute("*student*", *student*);  *request*.setAttribute("all*Student*s", *student*Dao.getAll*Student*s());  *request*.get*Request*Dispatcher("*home*.jsp").forward(*request*, response);  }  @Override  protected void doGet(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  @Override  protected void doPost(HttpServlet*Request* *request*, HttpServletResponse response)  throws ServletException, IOException {  process*Request*(*request*, response);  }  @Override  public String getServletInfo() {  return "Short description";  }// </*edit*or-fold>  } |

*File* ini berguna untuk menghubungkan pengolahan aplikasi dengan tampilan web. Disini berisi pengkondisian dan pengaturan terhadap data pada model, kemudian menampilkannya pada halaman *home*, ataupun sebaliknya pengolahan data yang dimasukan melalui halaman jsp *home* ke *database* melalui *file* interface DAO.

***Package* model**

***User*.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.model;  im*port* java.io.Serializable;  im*port* javax.persistence.Entity;  im*port* javax.persistence.GeneratedValue;  im*port* javax.persistence.GenerationType;  im*port* javax.persistence.Id;  im*port* javax.persistence.NamedQueries;  im*port* javax.persistence.NamedQuery;  im*port* javax.persistence.Table;  im*port* javax.persistence.Column;  /\*\*  \*  \* @author *USER*  \*/  @Entity  @Table  @NamedQueries({@NamedQuery(name="*User*.getAll",query="SELECT e FROM *User* e")})  public *class* *User* implements Serializable {  private static final long serialVersionUID = 1L;  @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  private int *user*Id;  @Column  private String *user*Name;  @Column  private String password;  public *User*() {  }  public *User*(String *user*Name, String password) {  this.*user*Name = *user*Name;  this.password = password;  }  public int get*User*Id() {  return *user*Id;  }  public void set*User*Id(int *user*Id) {  this.*user*Id = *user*Id;  }  public String get*User*Name() {  return *user*Name;  }  public void set*User*Name(String *user*Name) {  this.*user*Name = *user*Name;  }  public String getPassword() {  return password;  }  public void setPassword(String password) {  this.password = password;  }  } |

*File* ini adalah model *user* yang berisi variabel-variabel pembentuk *user*. Di dalamnya terdapat *setter* dan *getter* serta constructor untuk membangun *object* *user*. Disinilah penggunaan komponen pada aplikasi ini.

***Student*.java**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this *template* *file*, choose *Tool*s | *Template*s  \* and open the *template* in the *edit*or.  \*/  *package* com.model;  im*port* java.io.Serializable;  im*port* javax.persistence.Entity;  im*port* javax.persistence.GeneratedValue;  im*port* javax.persistence.GenerationType;  im*port* javax.persistence.Id;  im*port* javax.persistence.NamedQueries;  im*port* javax.persistence.NamedQuery;  im*port* javax.persistence.Table;  im*port* javax.persistence.Column;  /\*\*  \*  \* @author *USER*  \*/  @Entity  @Table  @NamedQueries({@NamedQuery(name="*Student*.getAll",query="SELECT e FROM *Student* e order by e.*student*Id")})  public *class* *Student* implements Serializable {  @Id  @GeneratedValue(strategy=GenerationType.AUTO)  private int *student*Id;  @Column  private String firstName;  @Column  private String lastName;  public *Student*(int *student*Id, String firstName, String lastName) {  this.*student*Id = *student*Id;  this.firstName = firstName;  this.lastName = lastName;  }  public *Student*() {  }  public void set*Student*Id(int *student*Id) {  this.*student*Id = *student*Id;  }  public int get*Student*Id() {  return *student*Id;  }  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;  }  } |

*File* ini adalah model *student* yang berisi variabel-variabel pembentuk *student*. Di dalamnya terdapat *setter* dan *getter* serta constructor untuk membangun *object* *student*. Disinilah penggunaan komponen pada aplikasi ini.

***register*.jsp**

|  |
| --- |
| <%--  Document : *register*  Created on : Sep 22, 2019, 2:48:05 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="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>*Register* 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">  <a href="./*login*.jsp" *class*="btn btn-md btn-success" style="float:right" />*Login*</a><br>  <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 Up</p>  <form method="POST" action="./*Register*sServlet">  <div *class*="form-group">  <label>*User*name</label>  <input type="text" *class*="form-control" placeholder="*User*name" name="*user*Name" 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="*register*" value="*Register*" *class*="btn btn-success" />  </form>  </div>  </div>  </div>  </div>  </body>  </html> |

*File* jsp ini digunakan untuk menampilkan tampilan web pada *register*.

***login*.jsp**

|  |
| --- |
| <%--  Document : *login*  Created on : Sep 22, 2019, 12:45:49 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="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">  <a href="./*register*.jsp" *class*="btn btn-md btn-success" style="float:right" />*Register*</a><br>  <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="./*Login*Servlet">  <div *class*="form-group">  <label>*User*name</label>  <input type="text" *class*="form-control" placeholder="*User*name" name="*user*Name" 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> |

*File* jsp ini digunakan untuk menampilkan tampilan web pada *login*.

***home*.jsp**

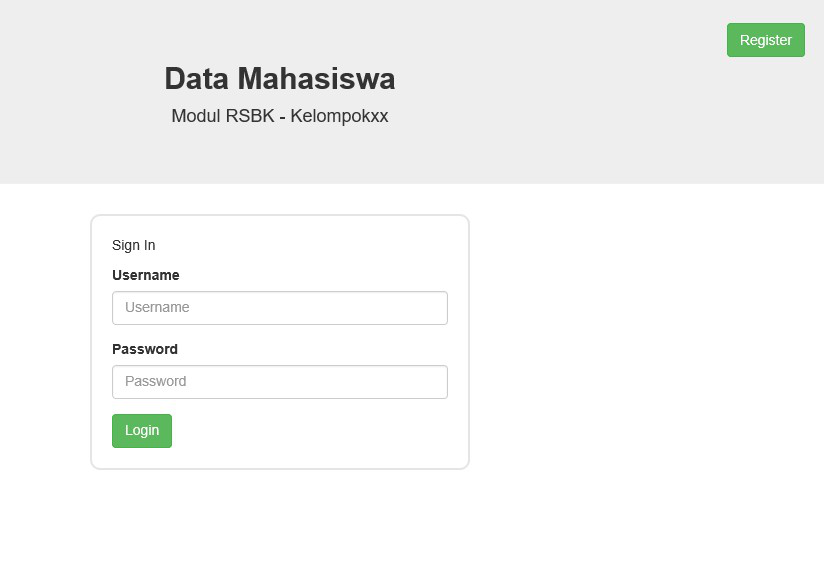
|  |
| --- |
| <%--  Document : *home*  Created on : Sep 22, 2019, 12:45:58 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="UTF-8"%>  <%@taglib prefix="s" uri="http://java.sun.com/jsp/jstl/core" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; 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("*login*Name")%></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="./*login*.jsp">Logout</a></li>  </ul>  </div>  <div *class*="col-md-9 daftar">  <form action="./*Student*Servlet" method="POST">  <table *class*="table table-bordered">  <tr>  <td>*Student* ID</td>  <td><input *class*="form-control" type="text" name="*student*Id" value="${*student*.*student*Id}" /></td>  </tr>  <tr>  <td>First Name</td>  <td><input *class*="form-control" type="text" name="firstname" value="${*student*.firstName}" /></td>  </tr>  <tr>  <td>Last Name</td>  <td><input *class*="form-control" type="text" name="lastname" value="${*student*.lastName}" /></td>  </tr>  <tr>  <td colspan="2">  <input type="submit" *class*="btn btn-primary btn-sm" name="action" value="Add" />  <input type="submit" *class*="btn btn-default btn-sm" name="action" value="*Edit*" />  <input type="submit" *class*="btn btn-danger btn-sm" name="action" value="*Delete*" />  <input type="submit" *class*="btn btn-warning btn-sm" name="action" value="Search" />  </td>  </tr>  </table>  </form>  <h3 align="center">Informasi Data</h3>  <table *class*="table table-bordered table-hover">  <thead>  <tr>  <th>No. ID</th>  <th>First Name</th>  <th>Last Name</th>  </tr>  </thead>  <tbody>  <s:forEach items="${all*Student*s}" var="stud">  <tr>  <td>${stud.*student*Id}</td>  <td>${stud.firstName}</td>  <td>${stud.lastName}</td>  </tr>  </s:forEach>  </tbody>  </table>  </div>  </div>  </div>  </html> |

*File* jsp ini digunakan untuk menampilkan tampilan web pada *home*.

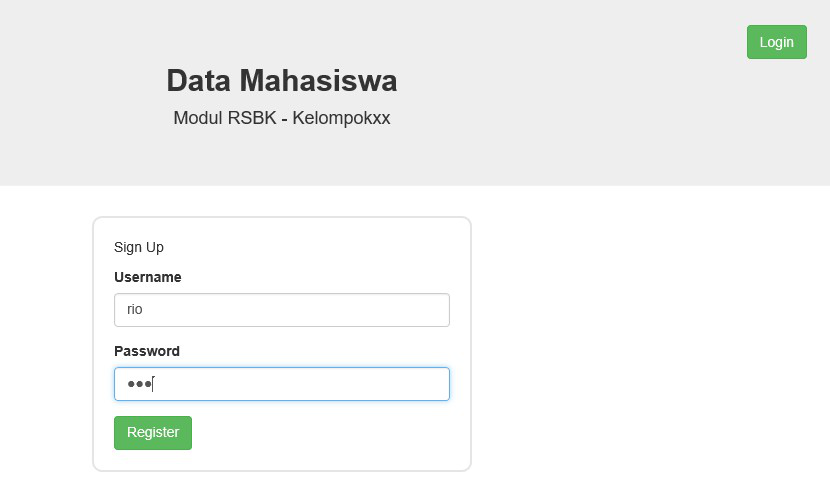
***error*.jsp**

|  |
| --- |
| <%@*page* contentType="text/html" *page*Encoding="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> |

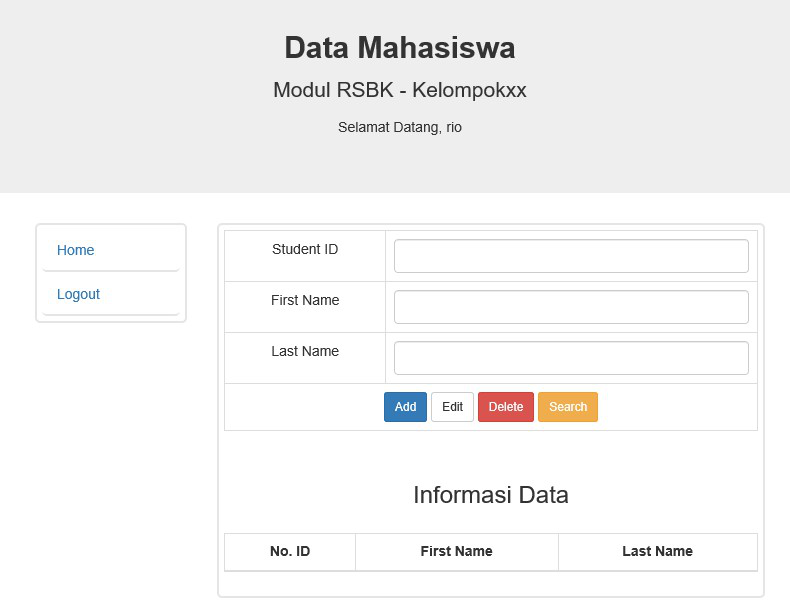
*File* jsp ini digunakan untuk menampilkan tampilan web saat *error* *login*.



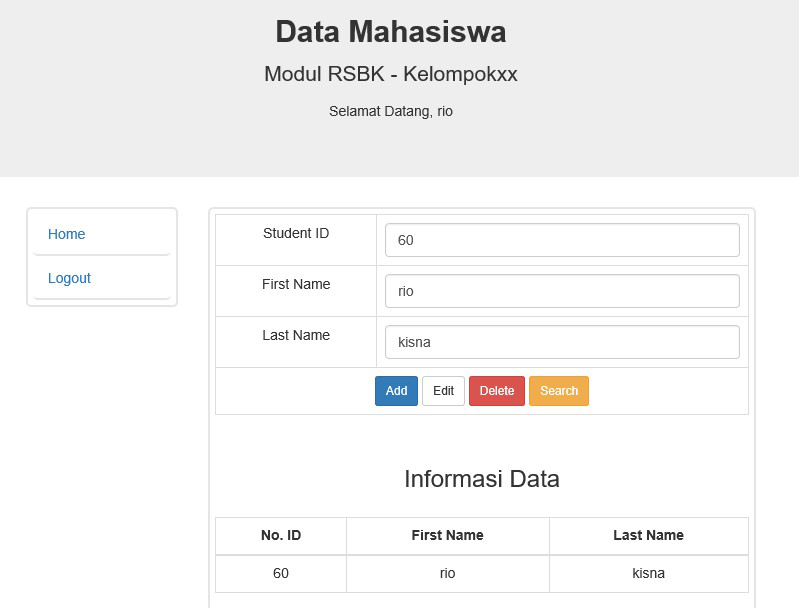
Gambar 4.31 Hasil halaman *Login*



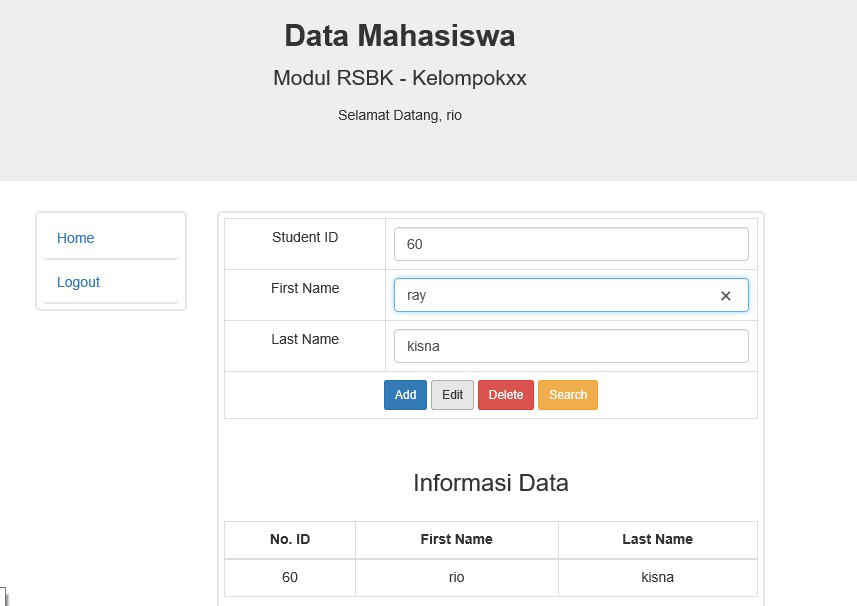
Gambar 4.32 Hasil halaman *Register*



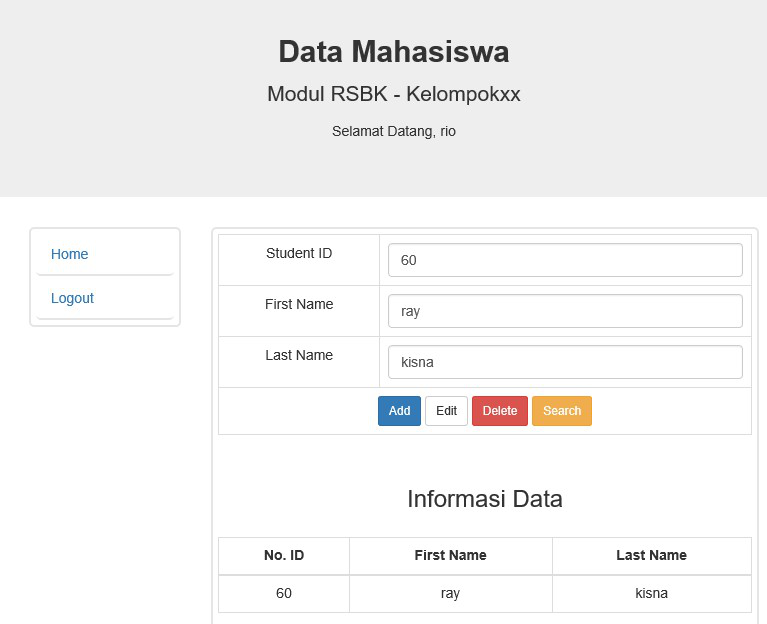
Gambar 4.33 Hasil halaman *Home*



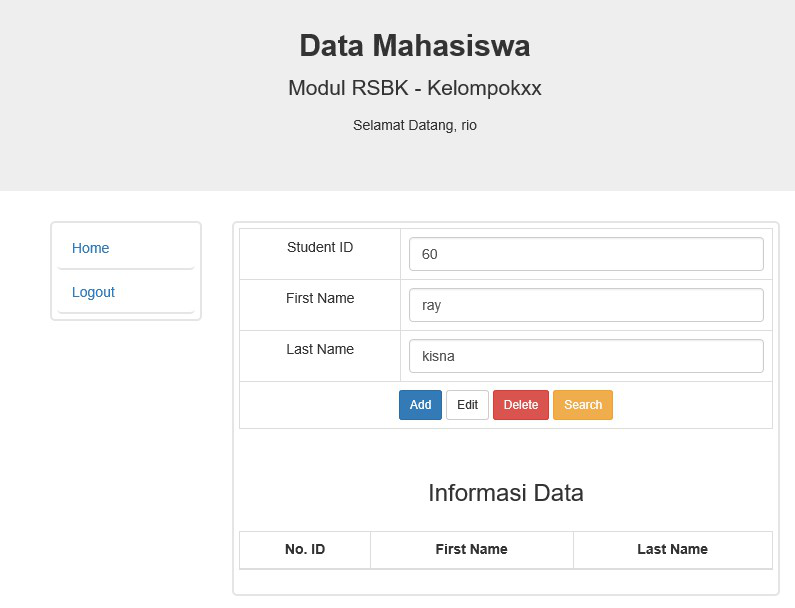
Gambar 4.34 Hasil percobaan tambah data



Gambar 4.35 Hasil percobaan *edit* data



Gambar 4.36 Hasil percobaan search data



Gambar 4.37 Hasil percobaan *delete* data

## 4.5 Tugas dan Pembahasan

Pada tugas kali ini adalah memperbaiki *error* yang terjadi saat praktikum. Karena saat praktikum kelompok kami tidak mengalami *error*, sehingga kami tidak mengubah aplikasi yang sudah kami buat saat praktikum.

Tugas kedua adalah mempercantik tampilan. Tampilan yang kami buat sebagian besar menggunakan *template*. Pertama kita membuat *file* css dan resource lainnya yang dibutuhkan untuk menghasilkan tampilan yang lebih menarik. Kemudian memasukan *file* html pada jsp tampilan yang diperlukan.

*File* *login*.jsp

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>*Login* V17</title>  <meta charset="UTF-8">  <meta name="*viewport*" content="width=device-width, initial-scale=1">  <!--===============================================================================================-->  <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-font.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/animsition/css/animsition.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="css/util.css">  <link rel="stylesheet" type="text/css" href="css/main.css">  <!--===============================================================================================-->  </head>  <body>    <div *class*="limiter">  <div *class*="container-*login*100">  <div *class*="wrap-*login*100">  <form *class*="*login*100-form validate-form" method="POST" action="./*Login*Servlet">  <span *class*="*login*100-form-title p-b-34">  Account *Login*  </span>    <div *class*="wrap-input100 rs1-wrap-input100 validate-input m-b-20" data-validate="Type *user* name">  <input id="first-name" *class*="input100" type="text" name="*user*Name" placeholder="*User* name">  <span *class*="focus-input100"></span>  </div>  <div *class*="wrap-input100 rs2-wrap-input100 validate-input m-b-20" data-validate="Type password">  <input *class*="input100" type="password" name="password" placeholder="Password">  <span *class*="focus-input100"></span>  </div>    <div *class*="container-*login*100-form-btn">  <button *class*="*login*100-form-btn">  Sign in  </button>  </div>  <div *class*="w-full text-center p-t-27 p-b-239">  <span *class*="txt1">  Forgot  </span>  <a href="#" *class*="txt2">  *User* name / password?  </a>  </div>  <div *class*="w-full text-center">  <a href="./*register*.jsp" *class*="txt3">  Sign Up  </a>  </div>  </form>  <div *class*="*login*100-more" style="background-image: url('images/bg-01.jpg');"></div>  </div>  </div>  </div>      <div id="dropDownSelect1"></div>    <!--===============================================================================================-->  <script src="vendor/jquery/jquery-3.2.1.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/animsition/js/animsition.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/bootstrap/js/popper.js"></script>  <script src="vendor/bootstrap/js/bootstrap.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/select2/select2.min.js"></script>  <script>  $(".selection-2").select2({  minimumResultsForSearch: 20,  dropdownParent: $('#dropDownSelect1')  });  </script>  <!--===============================================================================================-->  <script src="vendor/daterangepicker/moment.min.js"></script>  <script src="vendor/daterangepicker/daterangepicker.js"></script>  <!--===============================================================================================-->  <script src="vendor/countdowntime/countdowntime.js"></script>  <!--===============================================================================================-->  <script src="js/main.js"></script>  </body>  </html> |

*Register*.jsp

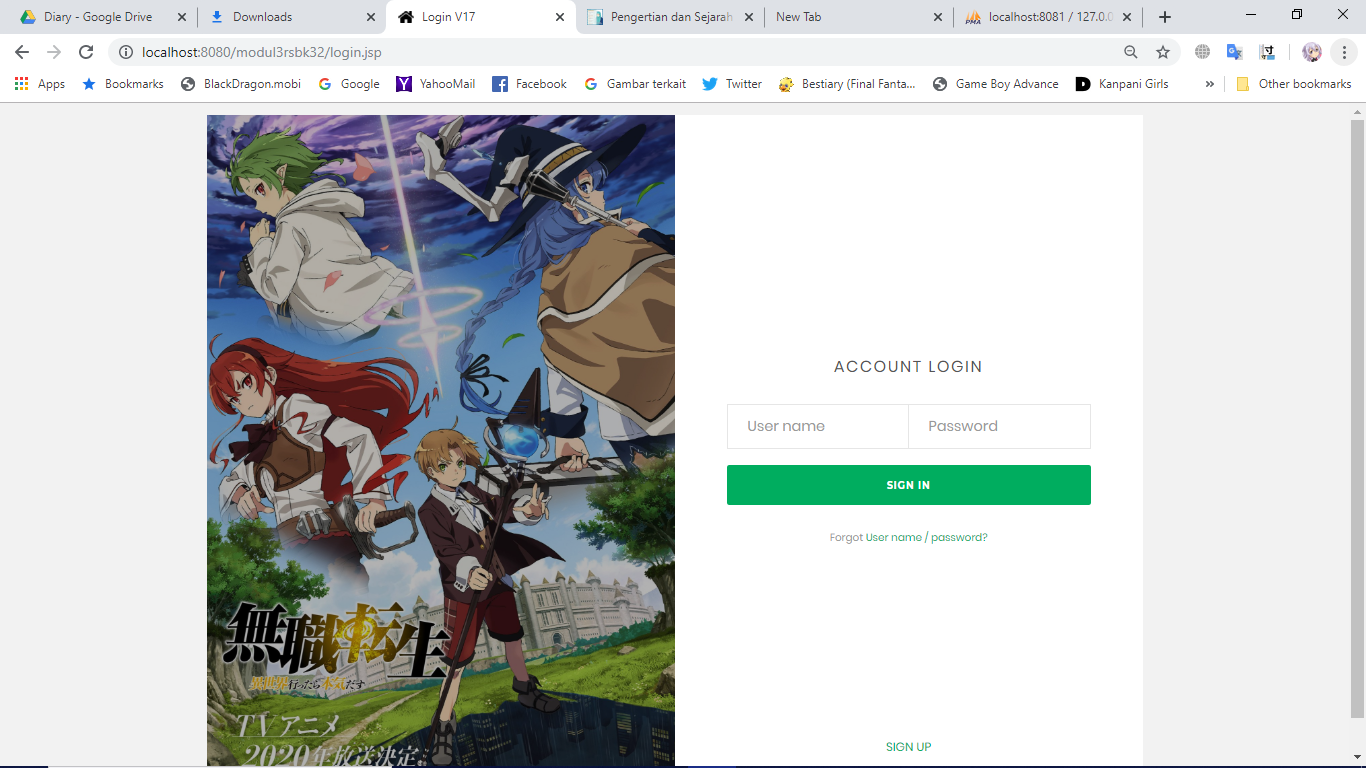
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>*Login* V17</title>  <meta charset="UTF-8">  <meta name="*viewport*" content="width=device-width, initial-scale=1">  <!--===============================================================================================-->  <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-font.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/animsition/css/animsition.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">  <!--===============================================================================================-->  <link rel="stylesheet" type="text/css" href="css/util.css">  <link rel="stylesheet" type="text/css" href="css/main.css">  <!--===============================================================================================-->  </head>  <body>    <div *class*="limiter">  <div *class*="container-*login*100">  <div *class*="wrap-*login*100">  <form *class*="*login*100-form validate-form" method="POST" action="./*Register*Servlet">  <span *class*="*login*100-form-title p-b-34">  Account *Register*  </span>  <span *class*="txt1">  Pastikan anda benar-benar pengguna baru!  </span>  <div *class*="wrap-input100 rs1-wrap-input100 validate-input m-b-20" data-validate="Type *user* name">  <input id="first-name" *class*="input100" type="text" name="*user*Name" placeholder="*User* name">  <span *class*="focus-input100"></span>  </div>  <div *class*="wrap-input100 rs2-wrap-input100 validate-input m-b-20" data-validate="Type password">  <input *class*="input100" type="password" name="password" placeholder="Password">  <span *class*="focus-input100"></span>  </div>    <div *class*="container-*login*100-form-btn">  <button *class*="*login*100-form-btn">  Sign up  </button>  </div>  <div *class*="w-full text-center p-t-27 p-b-239">  </div>  <div *class*="w-full text-center">  <a href="./*login*.jsp" *class*="txt3">  Sudah Punya Akun?  </a>  </div>  </form>  <div *class*="*login*100-more" style="background-image: url('images/bg-01.jpg');"></div>  </div>  </div>  </div>      <div id="dropDownSelect1"></div>    <!--===============================================================================================-->  <script src="vendor/jquery/jquery-3.2.1.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/animsition/js/animsition.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/bootstrap/js/popper.js"></script>  <script src="vendor/bootstrap/js/bootstrap.min.js"></script>  <!--===============================================================================================-->  <script src="vendor/select2/select2.min.js"></script>  <script>  $(".selection-2").select2({  minimumResultsForSearch: 20,  dropdownParent: $('#dropDownSelect1')  });  </script>  <!--===============================================================================================-->  <script src="vendor/daterangepicker/moment.min.js"></script>  <script src="vendor/daterangepicker/daterangepicker.js"></script>  <!--===============================================================================================-->  <script src="vendor/countdowntime/countdowntime.js"></script>  <!--===============================================================================================-->  <script src="js/main.js"></script>  </body>  </html> |

*Home*.jsp

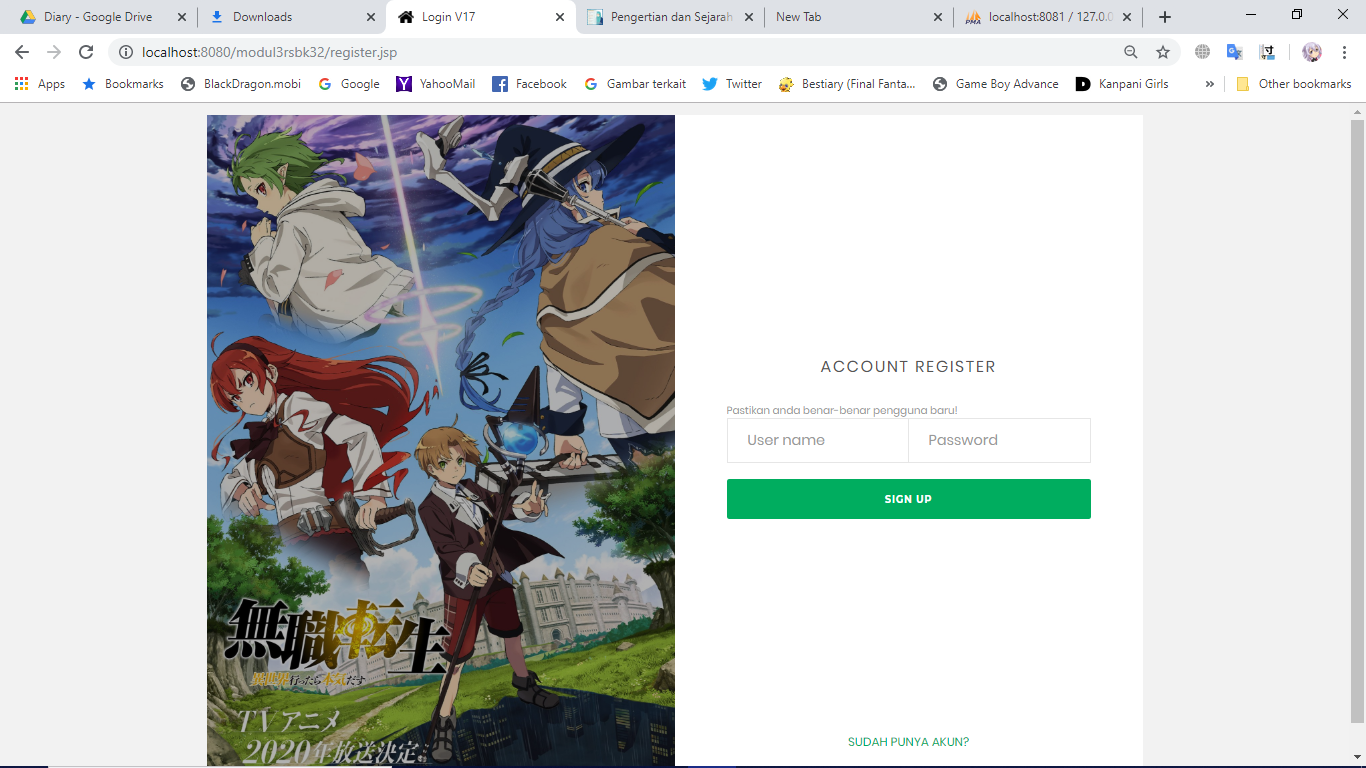
|  |
| --- |
| <%--  Document : *home*  Created on : Sep 22, 2019, 12:45:58 PM  Author : WIN 10  --%>  <%@*page* contentType="text/html" *page*Encoding="UTF-8"%>  <%@taglib prefix="s" uri="http://java.sun.com/jsp/jstl/core" %>  <!doctype html>  <html lang="en">  <head>  <!-- Required meta tags -->  <meta charset="utf-8">  <meta name="*viewport*" content="width=device-width, initial-scale=1, shrink-to-fit=no">  <title>Sistem Informasi Mahasiswa TEKKOM</title>  <!-- Bootstrap CSS -->  <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css" >  <style type="text/css">  .nav-link {  color: #fff;  }  li a:hover {  color: #fff;  }  .navbar{  background-color: #4A314D;  z-index: 100;  }  .container{  margin-top: 20px;  }  .jumbotron {  background-color: #6B6570;  margin-left: 5%;  width: 90%;  }  .jumbotron hr {  color: #A8BA9A;  border: none;  }  .jumbotron p {  color: #FFF;  padding-left: 7.5%;  font-size: 26px;  }  .btn {  float: right;  background-color: #6fc94b;  color: #FFF;  width: 10%;  height: 42px;  margin-left: 2.5%;  }  .btn:hover {  opacity: 0.8;  }  .table1 {  margin-left: 5%;  width: 90%;  }  .table1 td {  width: 20%;  }  .table1 inputtext {  width: 80%;  }  </style>  </head>  <body>  <nav *class*="navbar navbar-expand-lg">  <a *class*="navbar-brand" href="index.php">Mahasiswa TEKKOM</a>  <button *class*="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSup*port*edContent" aria-controls="navbarSup*port*edContent" aria-expanded="false" aria-label="Toggle navigation">  <span *class*="navbar-toggler-icon"></span>  </button>  <div *class*="collapse navbar-collapse" id="navbarSup*port*edContent">  <ul *class*="navbar-nav mr-auto">  <li *class*="nav-item active">  <a *class*="nav-link" href="#">*HOME* <span *class*="sr-only">(current)</span></a>  </li>  </ul>  <ul *class*="nav navbar-nav navbar-right">  <li *class*="nav-item">  <a href="./about.html" *class*="nav-link" ><%=session.getAttribute("*user*Name")%></a>  </li>  <li *class*="nav-item">  <a href="./*login*.jsp" *class*="nav-link" >Logout</a>  </li>  </ul>  </div>  </nav>    <div *class*="container">  <div *class*="jumbotron">  <p>Data Mahasiswa</p><hr>  <p>Modul RSBK - Kelompok32</p><hr>  <p>Selamat Datang, <%=session.getAttribute("*user*Name")%></p>  </div>    <div *class*=" daftar">  <form action="./*Student*Servlet" method="POST">  <table *class*="table table1 table-bordered">  <tr>  <td>*Student* ID</td>  <td><input *class*="form-control inputtext" type="text" name="*student*Id" value="${*student*.*student*Id}" /></td>  </tr>  <tr>  <td>First Name</td>  <td><input *class*="form-control inputtext" type="text" name="firstname" value="${*student*.firstName}" /></td>  </tr>  <tr>  <td>Last Name</td>  <td><input *class*="form-control inputtext" type="text" name="lastname" value="${*student*.lastName}" /></td>  </tr>  <tr>  <td>Alamat</td>  <td><input *class*="form-control inputtext" type="text" name="alamat" value="${*student*.alamat}" /></td>  </tr>  <tr>  <td colspan="2">  <input type="submit" *class*="btn btn-sm" name="action" value="Search" />  <input type="submit" *class*="btn btn-sm" name="action" value="*Delete*" />  <input type="submit" *class*="btn btn-sm" name="action" value="*Edit*" />  <input type="submit" *class*="btn btn-sm" name="action" value="Add" />  </td>  </tr>  </table>  </form>  <h3 align="center">Informasi Data</h3>  <table *class*="table table1 table-bordered table-hover">  <thead>  <tr>  <th>No. ID</th>  <th>First Name</th>  <th>Last Name</th>  <th>Alamat</th>  </tr>  </thead>  <tbody>  <s:forEach items="${all*Student*s}" var="stud">  <tr>  <td>${stud.*student*Id}</td>  <td>${stud.firstName}</td>  <td>${stud.lastName}</td>  <td>${stud.alamat}</td>  </tr>  </s:forEach>  </tbody>  </table>  </div>  </div>  </div>  </html> |

*Error*.jsp

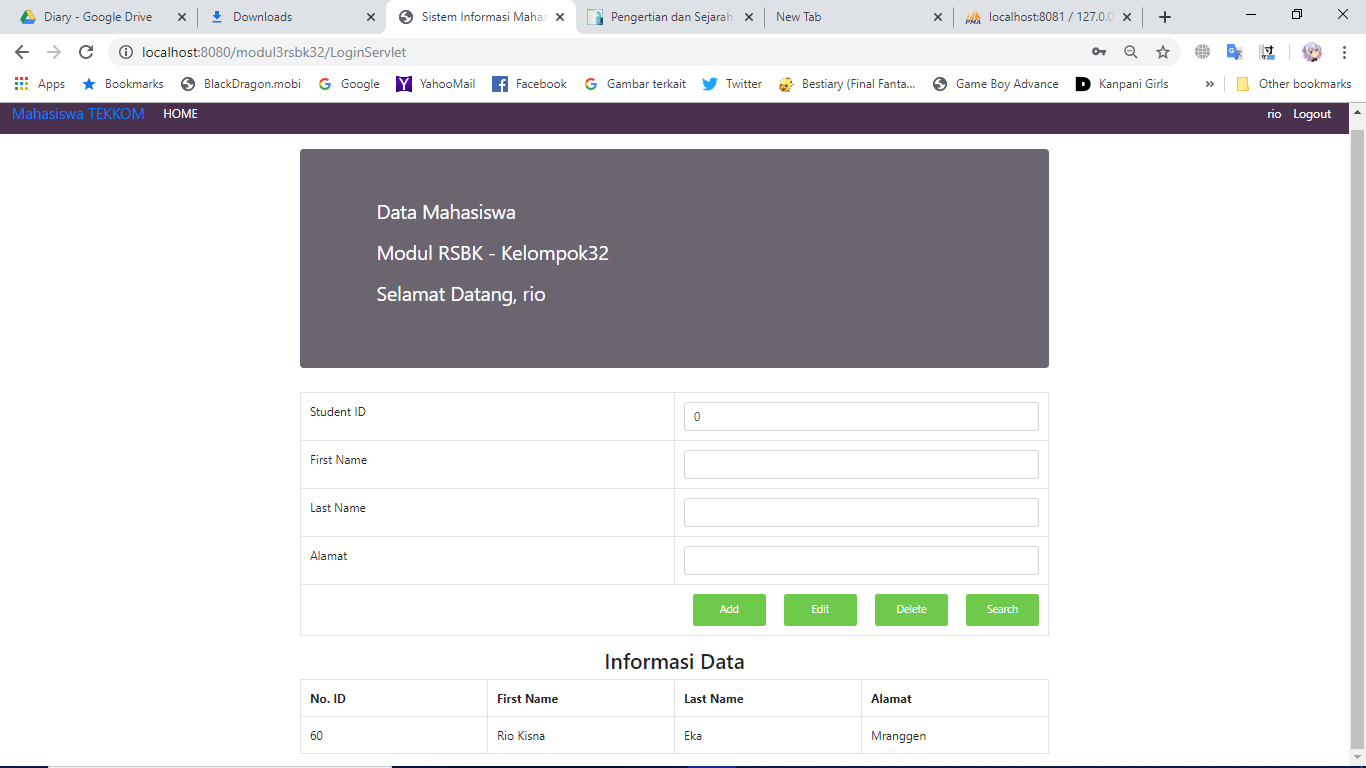
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="*viewport*" content="width=device-width, initial-scale=1">  <!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->  <title>404 HTML Tempalte by Colorlib</title>  <!-- Google font -->  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,900" rel="stylesheet">  <!-- Custom stlylesheet -->  <link type="text/css" rel="stylesheet" href="css/style.css" />  <!-- HTML5 shim and Respond.js for IE8 sup*port* of HTML5 elements and media queries -->  <!-- WARNING: Respond.js doesn't work if you *view* the *page* via *file*:// -->  <!--[if lt IE 9]>  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>  <![endif]-->  </head>  <body>  <div id="notfound">  <div *class*="notfound">  <div *class*="notfound-404">  <h1>Oops!</h1>  </div>  <h2>*User* tidak terdaftar!</h2>  <p>Mohon untuk segera mendaftar kembali.</p>  <a href="./*login*.jsp">Go To *Homepage*</a>  </div>  </div>  </body><!-- This *template*s was made by Colorlib (https://colorlib.com) -->  </html> |



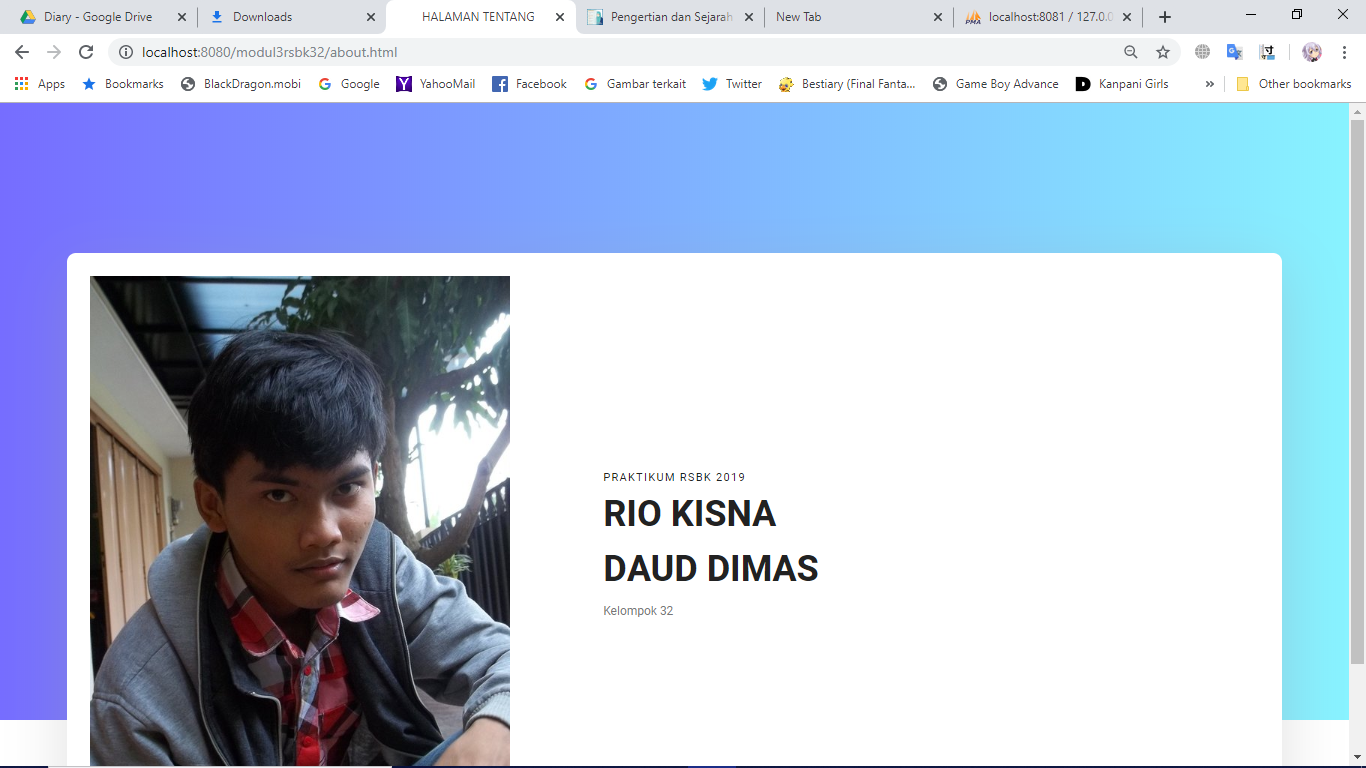
Gambar 4.38 Tampilan *login*



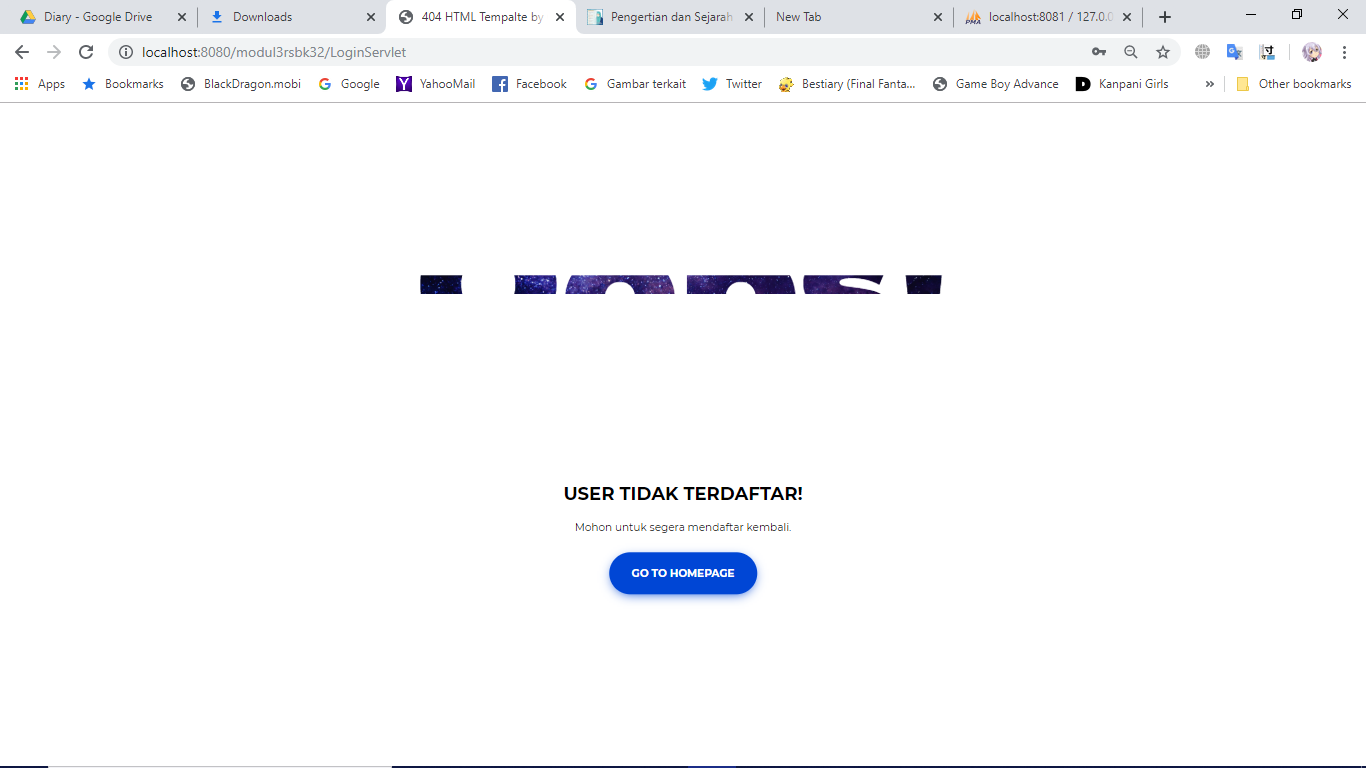
Gambar 4.39 Tampilan *register*



Gambar 4.40 Tampilan Halaman *Home*



Gambar 4.41 Tampilan halaman *About*



Gambar 4.42 Tampilan halaman *error*

## 4.6 Kesimpulan

1. Servlet pada praktikum digunakan sebagai *controlletr* yang menghubungkan antara *view* pada jsp dengan perintah yang terdapat di model.
2. JSP pada praktikum digunakan untuk membuat halaman web yang terdiri dari baris HTML dan fungsi JSP itu sendiri.
3. JPA digunakan untuk menghubungkan aplikasi yang dibuat dengan *database* dan memanipulasi data yang di dalamnya.
4. Entity *Class* berisi atribut dan berfungsi untuk diambil dan diberi nilai pada atribut tersebut.
5. Session Bean berisikan perintah – perintah yang dijalankan di dalam aplikasi, seperti perintah CRUD pada aplikasi, cek *user*, tambah *user*, dan credential *user*.