**BAB III**

**EJB SESSION BEAN**

**3.1 Tujuan**

1. Praktikan mampu memahami kegunaan JSP dalam Java Programming.
2. Praktikan mampu memahami kegunaan Java Sevlet dalam Java Programming.
3. Praktikan mampu memahami penggunaan Glassfish Server dalam Java Web Programming.
4. Praktikan mampu mengetahui apa itu *Component Statefull*.
5. Praktikan mampu mengetahui apa itu *Component Stateless.*

**3.2 Alat dan Bahan**

1. Laptop



Gambar 3. Laptop

Laptop digunakan untuk membuat program.

1. Netbeans IDE



Gambar 3. Netbeans

Netbeans adalah IDE yang digunakan untuk membuat Java Visual Application

1. Java EE SDK



Gambar 3. Java EE SDK

Java EE SDK adalah sebuah Software Development Kit yang diwajibkan untuk dimiliki agar dapat membuat coding dan *compiling* code Java

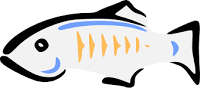
1. Java SDK 7



Gambar 3. Java SDK 7

Sebuah Environment yang digunakan untuk membuat aplikasi, komponen, dan *applets*.

1. Glassfish Server



Gambar 3. Glassfish Server

Sebuah Server *Open-Source* yang sekarang sudah di support full dan dibawahi Oracle. Digunakan untuk project yang membutuhkan server

**3.3 Dasar Teori**

**3.3.1 Java Enterprise Editional (Java EE)**

Java Enterprise, J2EE, Java EE adalah istilah-istilah yang mungkin sulit dimengerti oleh sebagian orang, bahkan programer Java (yang baru) sekalipun. Di artikel kali ini, saya akan berusaha memberikan gambaran, apa itu Java Enterprise, dan bagaimana kita bisa memanfaatkannya dalam membangun sebuah aplikasi.

J2EE adalah singkatan dari Java 2 Enterprise Edition, dan sejak versi 5 berubah menjadi Java EE atau cukup Java Enterprise Edition. Java EE adalah sebuah spesifikasi, bukan berupa product berwujud nyata. Ada banyak vendor software atau komunitas opensource membuat software berupa application server yang mengikuti standard Java EE sehingga disebut Java EE compliant application server.

Application server ini memungkinkan kita untuk membuat aplikasi berskala enterprise dengan lebih mudah karena application server sudah menyediakan berbagai fasilitas yang siap untuk digunakan sebagai pendukung aplikasi kita.  
Full Java EE compliant application server menyediakan berbagai service seperti web container, messaging, web service, mail, directory service, database connectivity, distributed transaction, remoting, persistence, dan lain lain.

Contoh application server adalah Glassfish, Oracle AS, JBoss, IBM Websphere, JRun, JOnAS, dll. Apache Tomcat adalah salah satu webserver/webcontainer untuk aplikasi web Java yang cukup terkenal, tapi bukan termasuk Java EE application server karena tidak menyediakan semua service yang ada di spesifikasi Java EE. Jika kita membuka website sun.java.com, dan menuju halaman tentang Java EE, maka kita bisa mendownload salah satu produk implementasi Java EE yang dibuat di bawah kendali Sun Microsystem, yaitu Sun Java System Application Server, yang sebenarnya adalah nama lain dari Glassfish application server.

*(Sumber: https://suhearie.wordpress.com/2008/08/26/java-enterprise-mulai-dari-mana/)*

**3.3.2 Component Statefull**

Komponen *Statefull* adalah suatu komponen yang menyimpan suatu kondisi pada suatu program, seperti *cache* dan *cookies*.

*(Sumber: https://programmingwithmosh.com/javascript/stateful-stateless-components-react/)*

**3.3.3 Component Stateless**

Komponen *Stateless* adalah suatu komponen yang tidak menyimpan *state* atau keadaan dari suatu program.

*(Sumber: https://programmingwithmosh.com/javascript/stateful-stateless-components-react/)*

**3.3.4 Java Server Page (JSP)**

JSP adalah sebuah bahasa pemrograman berbasis Java yang diperuntukkan untuk membuat sebuah website. JSP sangat sesuai dan tangguh untuk menangani kebutuhan akan sebuah website. Sama seperti PHP, bahasa yang satu ini dikhususkan untuk berkutat di bagian backend.

Artinya, bagian static yang berupa tag-tag HTML akan terpisah dari kode JSP. Kita dapat membuat halaman web static dengan html dan css seperti biasanya, kemudian akan disisipi dengan JSP untuk membuat halaman web menjadi dinamis.

*(Sumber: https://www.mahirkoding.com/pengenalan-jsp/)*

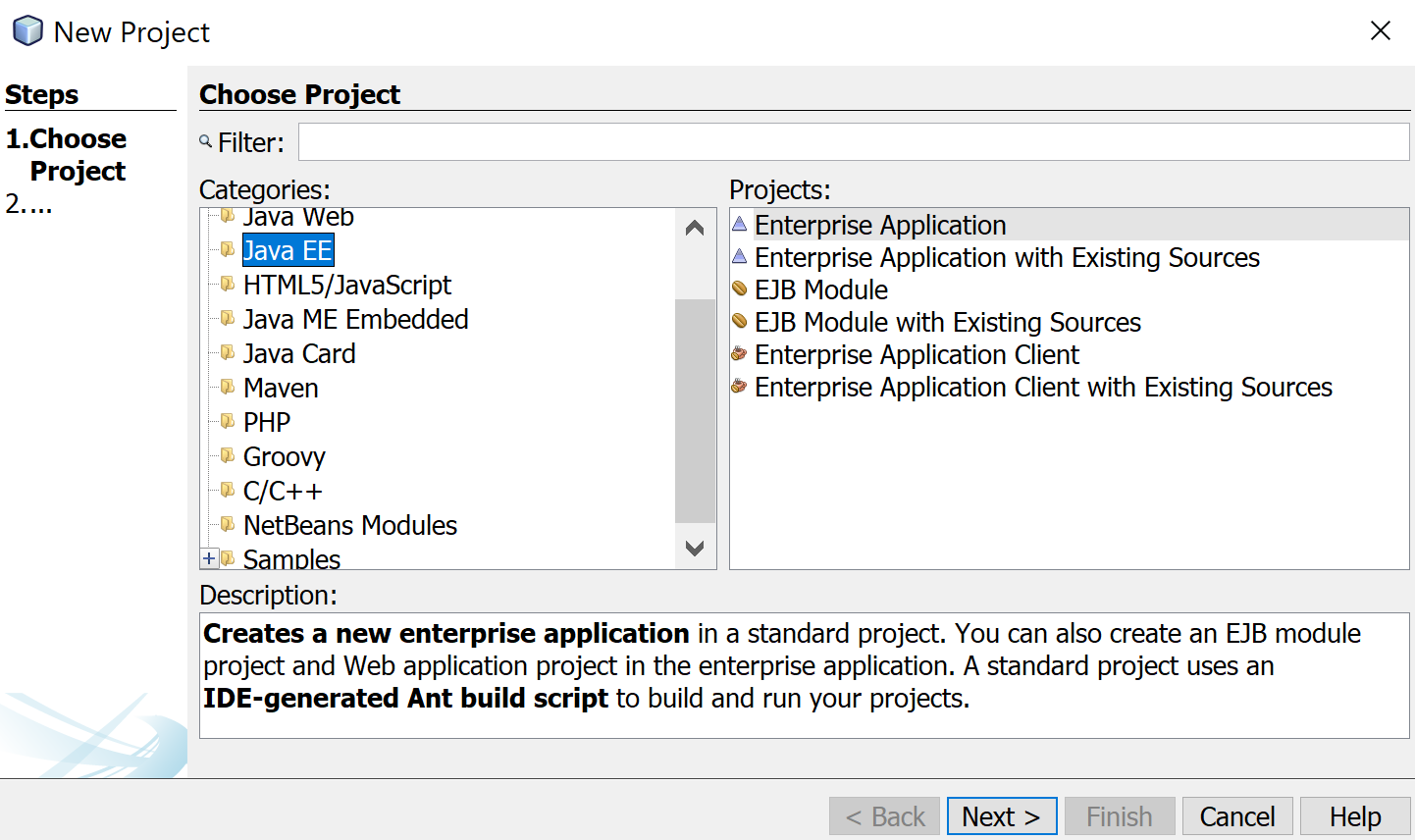
**3.3.5 Java Servlet**

Servlet adalah teknologi Java untuk aplikasi web berupa class yang digunakan untuk menerima request dan memberi respon melalui protokol http (html, xml, file dan sebagainya). Pada dasarnya Servlet merupakan file java class yang telah dikompilasi dan dijalankan oleh servlet container atau application server. Istilah application server digunakan apabila software server dapat menjalankan servlet, JSP serta teknologi J2EE utama seperti EJB (Enterprise Java Bean). Contoh Application Servlet adalah BEA Web Logic, IBM Websphere, Jboss, dsb. Servlet container biasanya juga merupakan JSP container, seperti Apache Tomcat, Macromedia Jrun, Resin.

*(Sumber: https://ptwfighter.wordpress.com/2011/09/28/servlet-jsp-dan-web-browser/)*

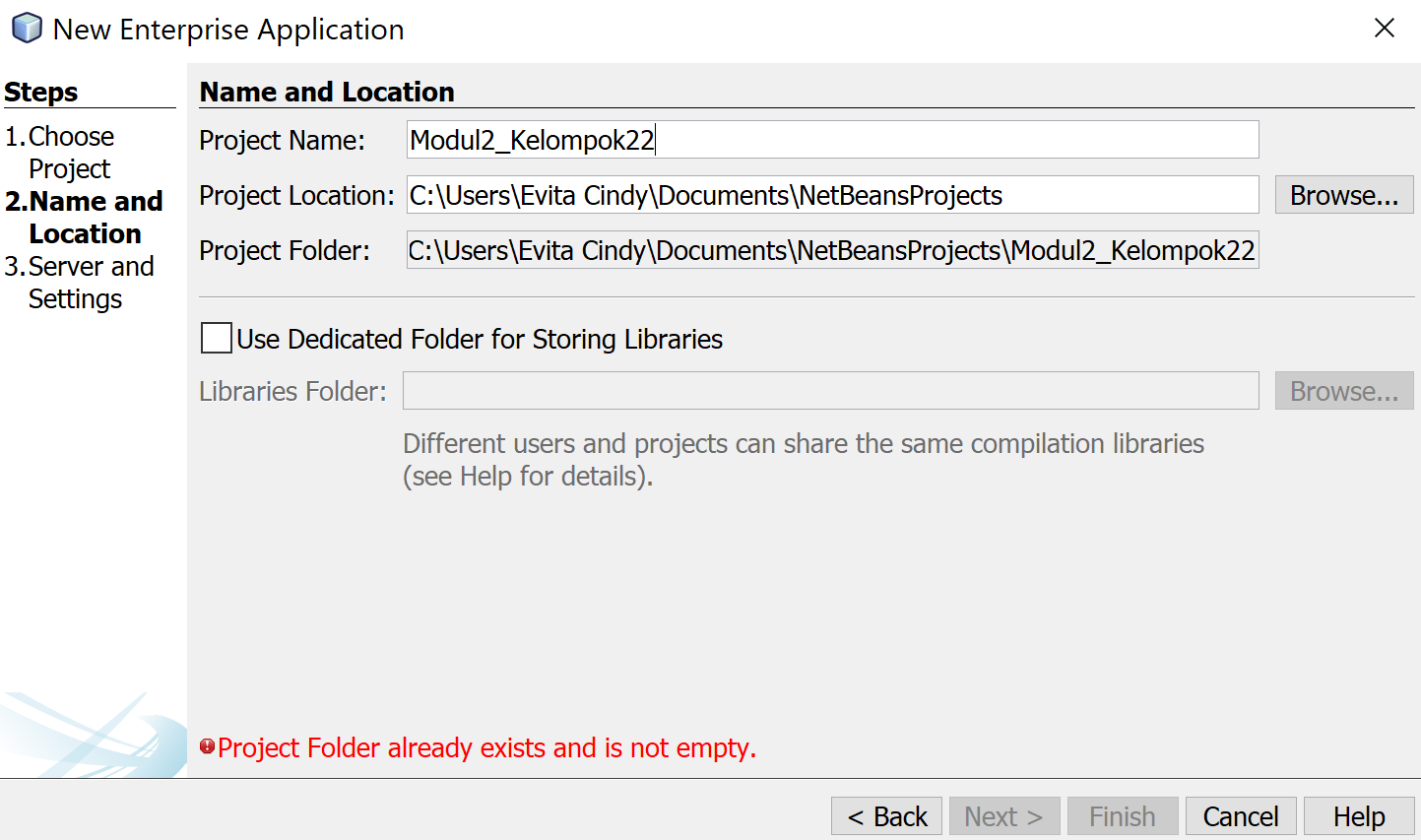
**3.4 Langkah Kerja**

1. Buka aplikasi **NETBEANS IDE**.
2. Buat project baru pilih Java EE > Enterprise Application lalu klik next



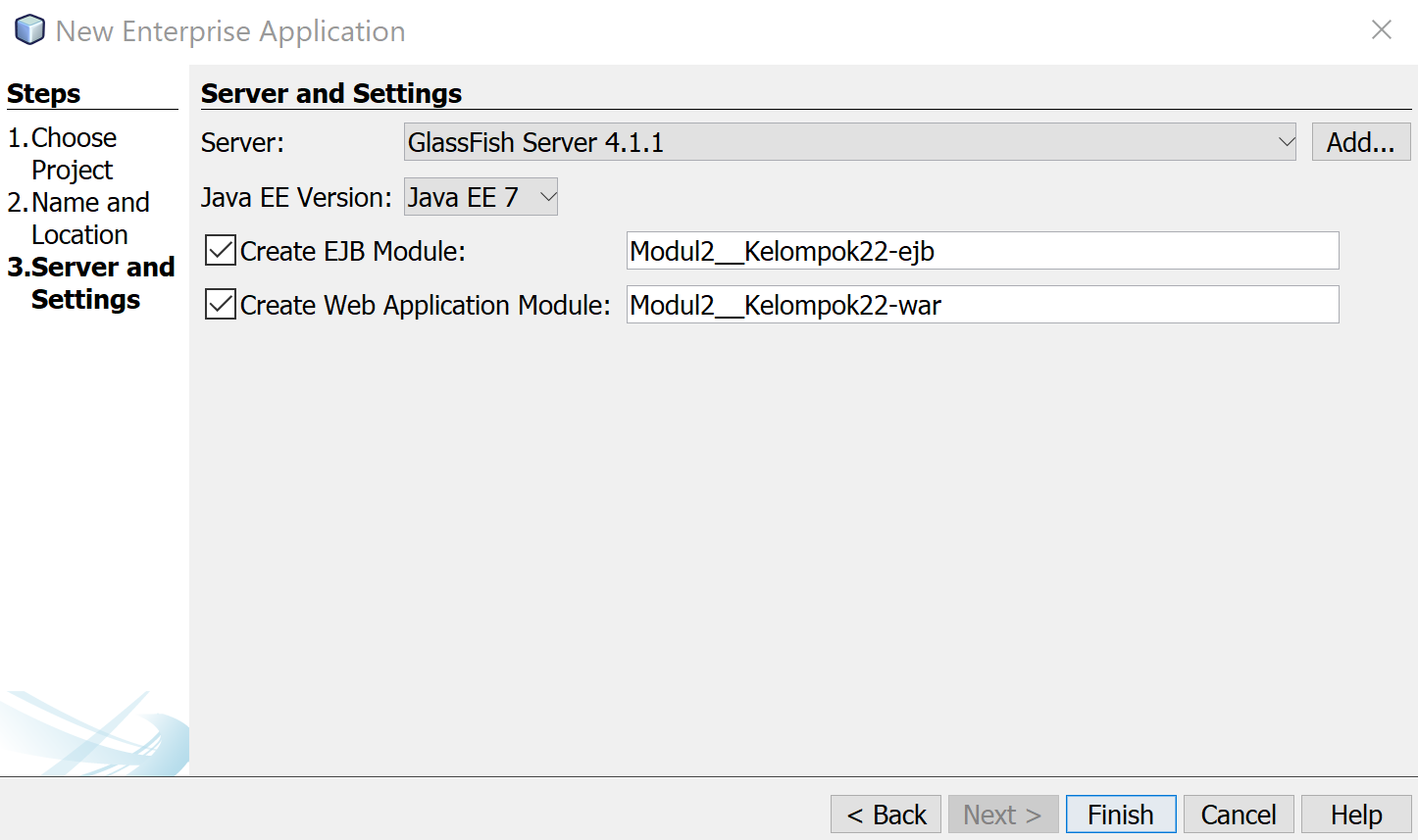
Gambar 3. New Project

1. Pilih location project lalu klik next



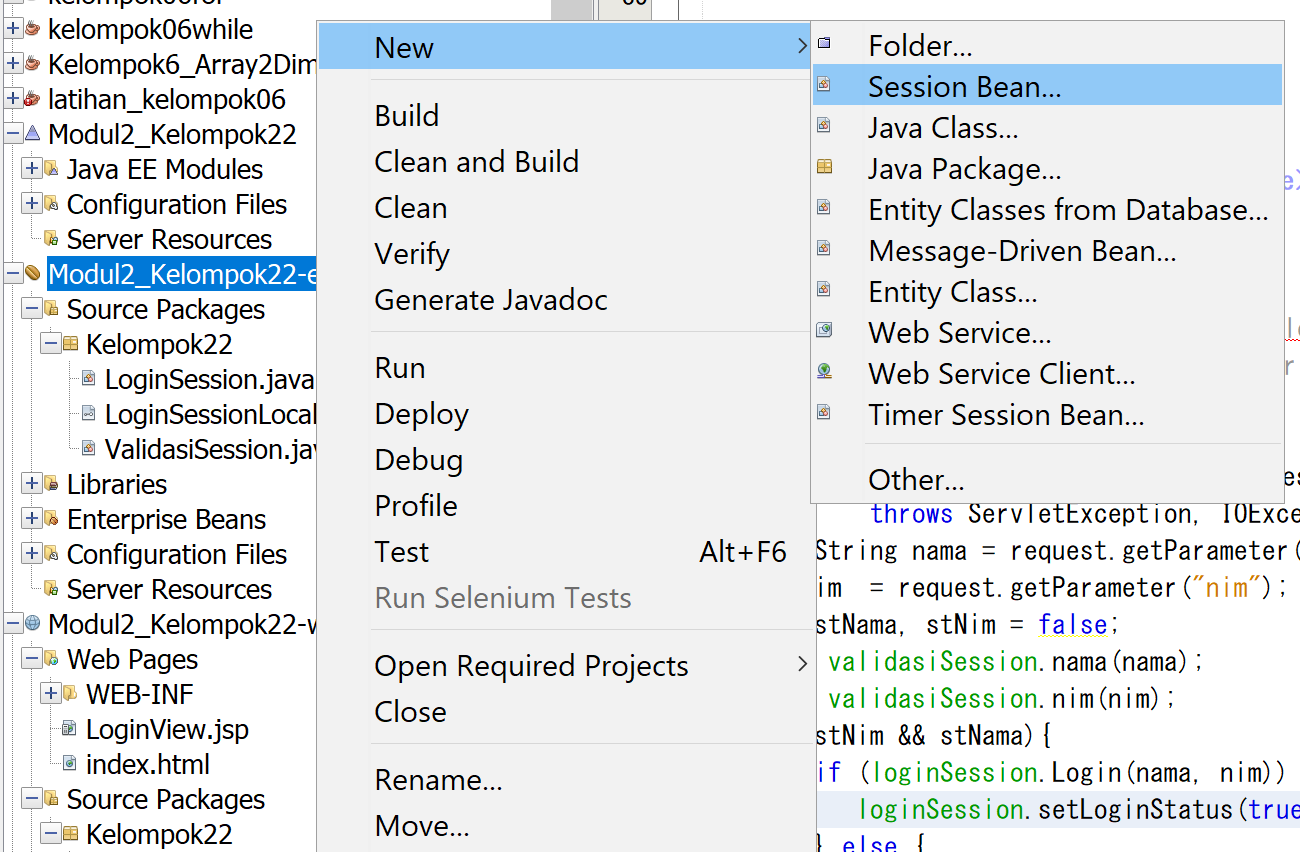
Gambar 3. Pilih Location

1. Pilih server glashfish server yang, ikuti seperti gambar.



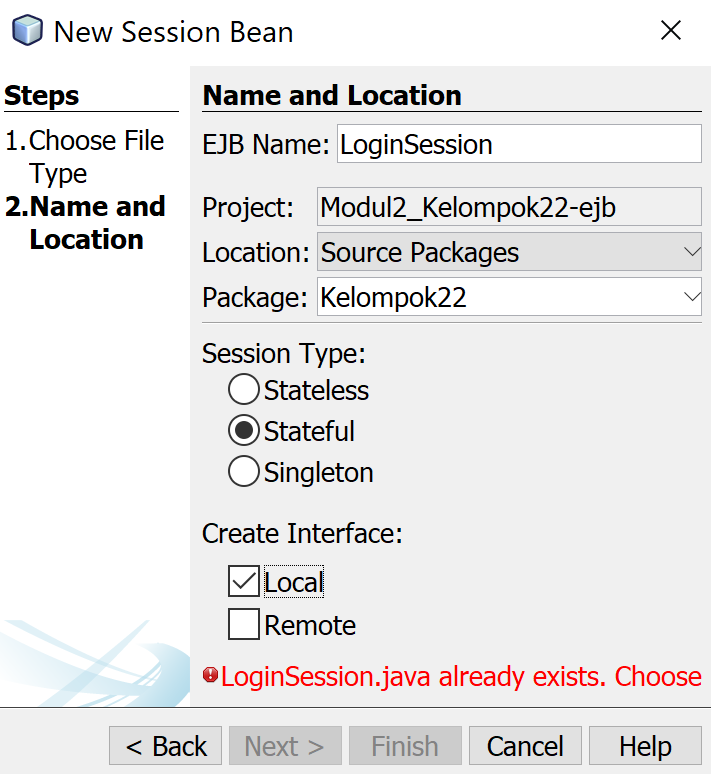
Gambar 3. GlassFish Server

1. Buat file Session Been baru dengan cara klik kanan project EJB pilih New>Session Bean. Ini dikarenakan kita ingin membuat sebuah Session yang nanti di aplikasikan pada Web.



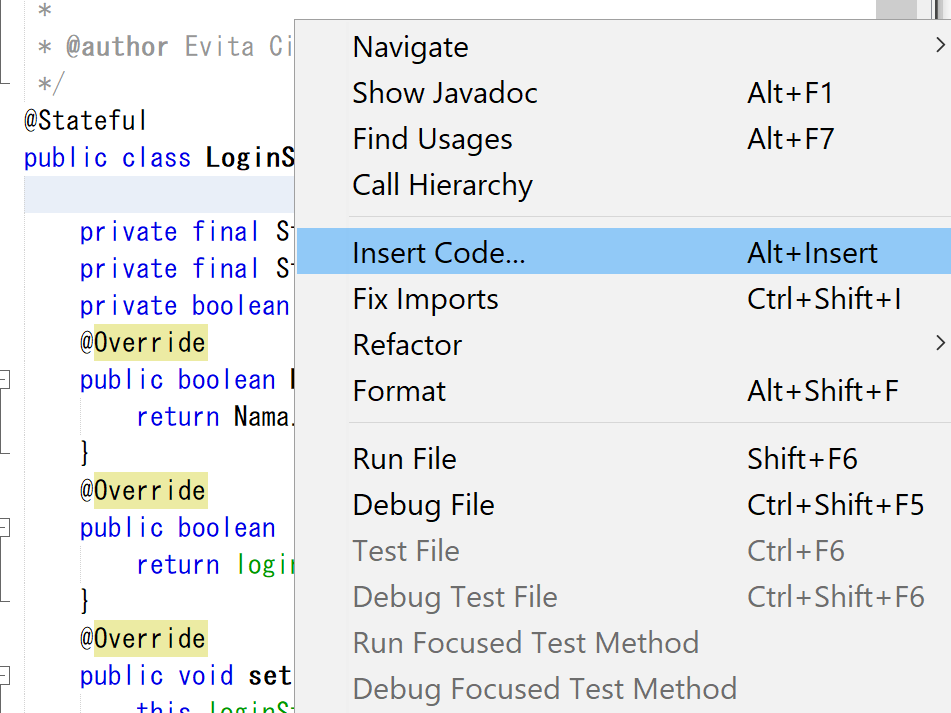
Gambar 3. New Session Bean

1. Pada Isikan LoginSession pada **EJB Name**. Pada **Session Type** pilih Stateful dan centang pilihan Local pada **Create Interface**. Interface ini nantinya akan berupa Stateful karena kita ingin membawa informasi yang diterima dari interface ke session berikutnya.



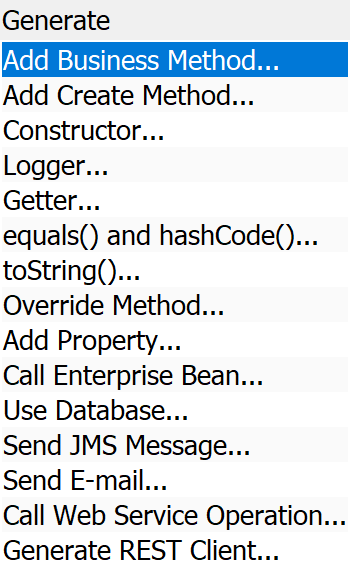
Gambar 3. Create Interface

1. Klik kanan pada text editor dan pilih **Insert Code**



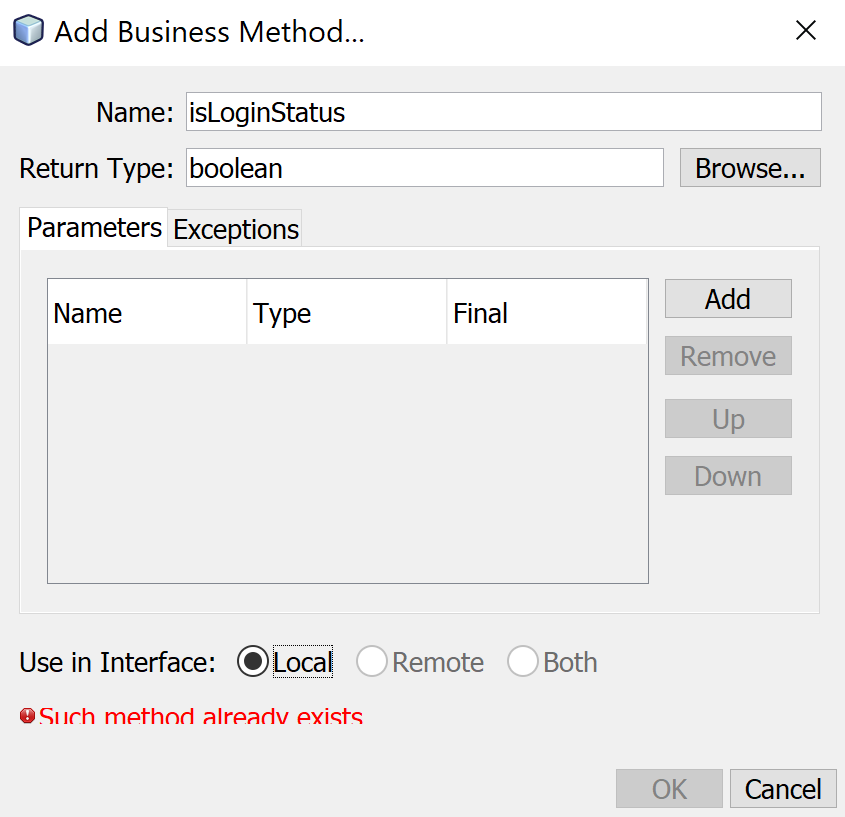
Gambar 3. Insert Code

1. Pilih **Add Business Methode.**

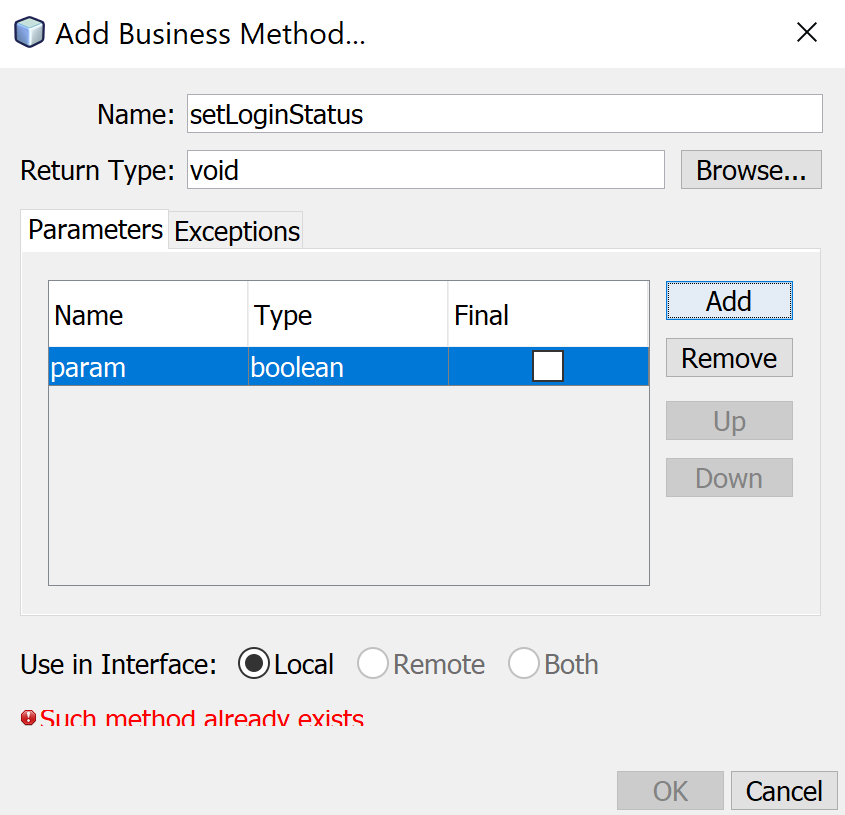


Gambar 3. Add Business Method

1. Masukan nama Login dengan **Return Type** boolean. Tambahkan 2 parameter dengan nama **Nama** dan **Nim**. Pastikan parameter sesuai dengan yang terdapat pada Gambar. Nama dan NIM ini nantinya akan digunakan sebagai inputtan pada website yang akan dibuat.
2. Lakukan hal yang sama untuk method **isLoginStatus** dan **setLoginStatus**



Gambar 3. Pada isLoginStatus

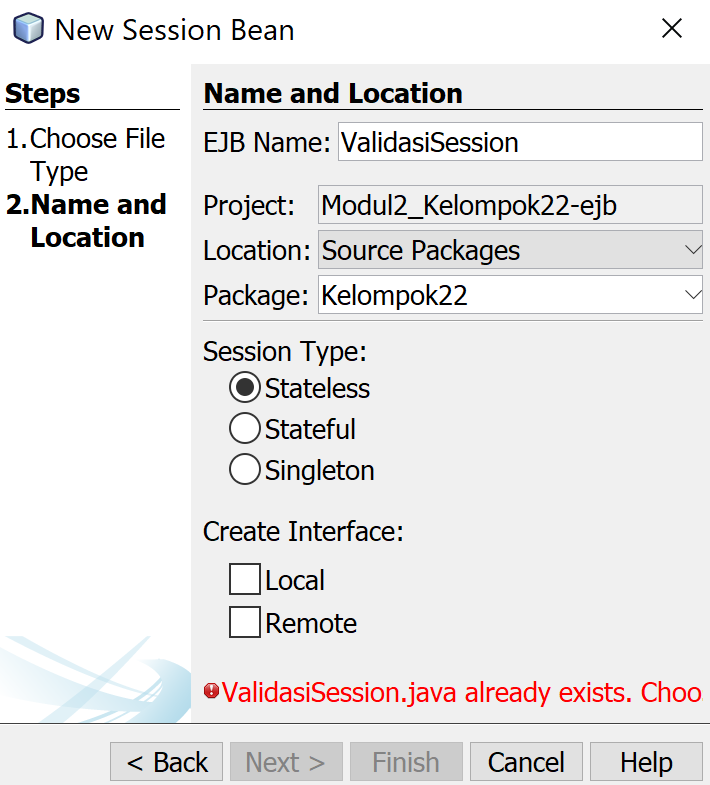


Gambar 3. Pada setLoginStatus

1. Hapus source code yanga ada didalam class LoginSession lalu tambahkan source code berikut.

|  |
| --- |
| private final String [] Nama = {"Praktikum RSBK"};  private final String [] Nim = {"21120116000001"};  private boolean loginStatus = false;  @Override  public boolean Login(String Nama, String Nim) {  return Nama.equals(this.Nama[0]) && Nim.equals(this.Nim[0]);  }  @Override  public boolean isLoginStatus() {  return loginStatus;  }  @Override  public void setLoginStatus(boolean loginStatus) {  this.loginStatus = loginStatus;  } |

1. Ikuti langkah 5 untuk membuat Session Bean baru
2. Pada Isikan ValidasiSession pada **EJB Name**. Pada **Session Type** pilih Stateless.



Gambar 3. ValidasiSession

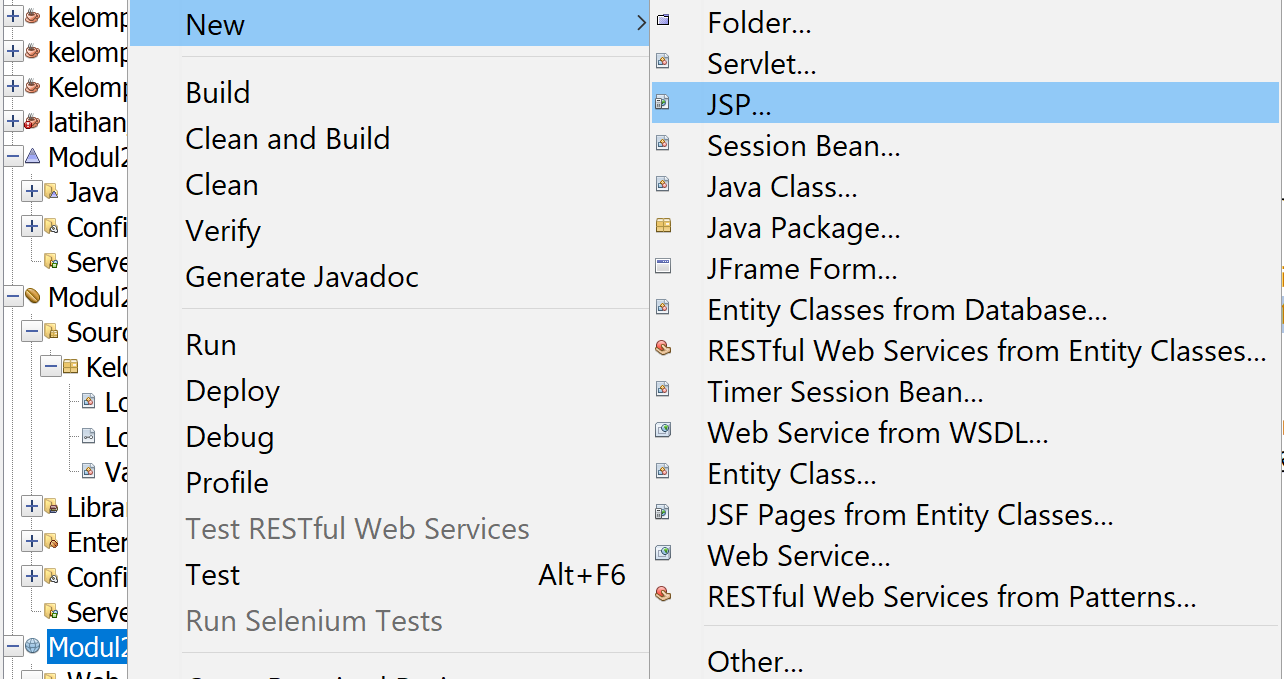
1. Tambakan source code berikut didalam class **ValidasiSession**.

|  |
| --- |
| public boolean nama(String param) {  try{  if(param.isEmpty()){  return false;  }  if(param.length()<=5){  return false;  }  }catch(NullPointerException e){  return false;  }  return true;  }  public boolean nim(String param) {  try{  Long.parseLong(param);  }catch(NumberFormatException e){  return false;  }  if(param.isEmpty()){  return false;  }  if(param.length()<14){  return false;  }  return true;  } |

1. Buka file index.html di directory **Modul2\_KelompokXX-war>Web Page** dan timpa source code yang sudah ada dengan source code berikut.

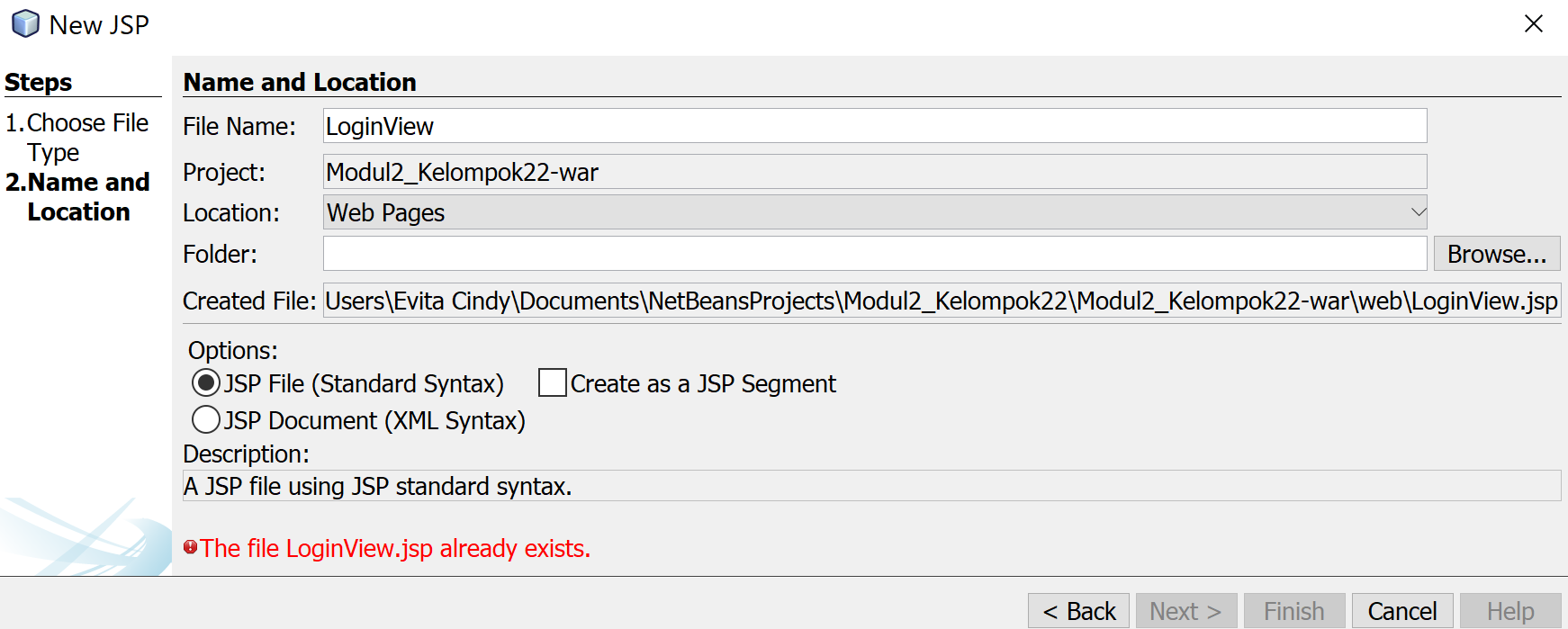
|  |
| --- |
| <!DOCTYPE html>  <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <title>2020 Gelar ST</title>  </head>  <body>  <h1 style="text-align:center"> 2020 GELAR ST</h1>  <h3 style="text-align:center">Tahun depan kita wisuda</h3>  <h3 style="text-align:center">Pelajari program berikut biar bisa mewujudkannya</h3>  <h3 style="text-align:center;"><a href="Login" style="text-decoration:none"><< Klik Disini >></a></h3>  <h4 style="text-align:center">&copy; Copyright Praktikum Rekayasa Perangkat Lunak 2019</h4>  </body>  </html> |

1. Buat file JSP dengan cara klik kanan pada project war pilih **New>JSP**



Gambar 3. JSP

1. Beri nama LoginView lalu klik finish

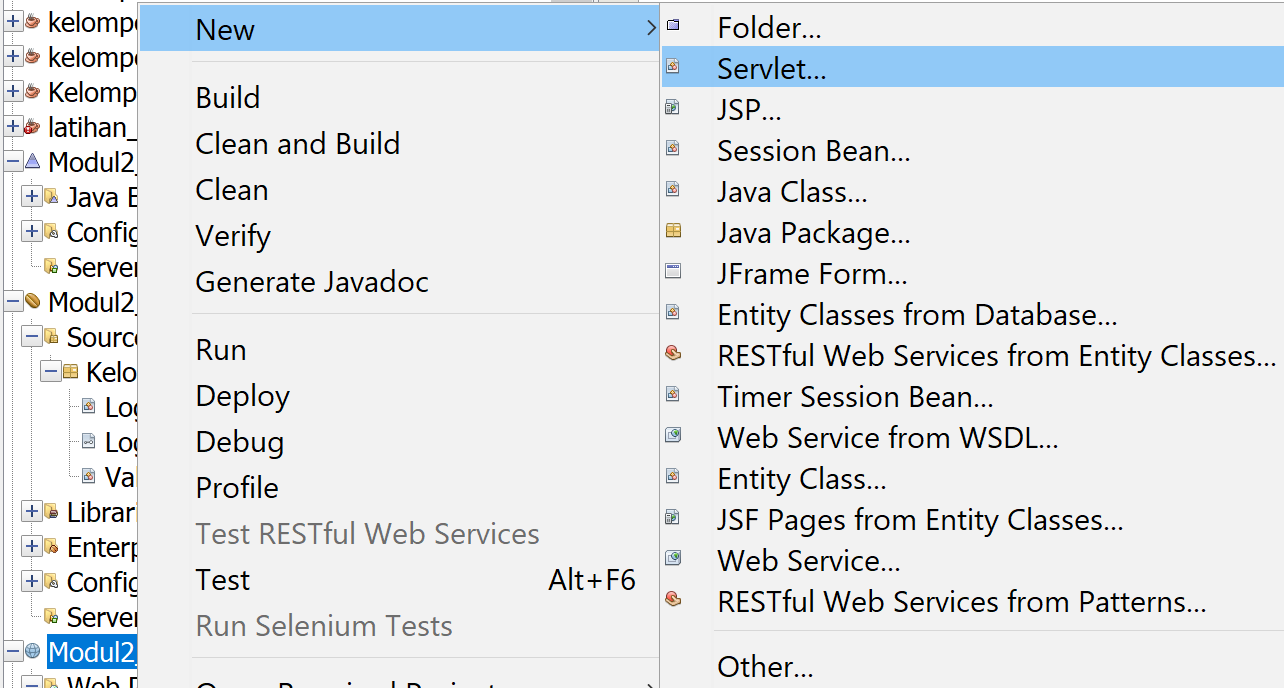


Gambar 3. Buat LoginView

1. Paste source code berikut ke dalam file JSP yang dibuat.

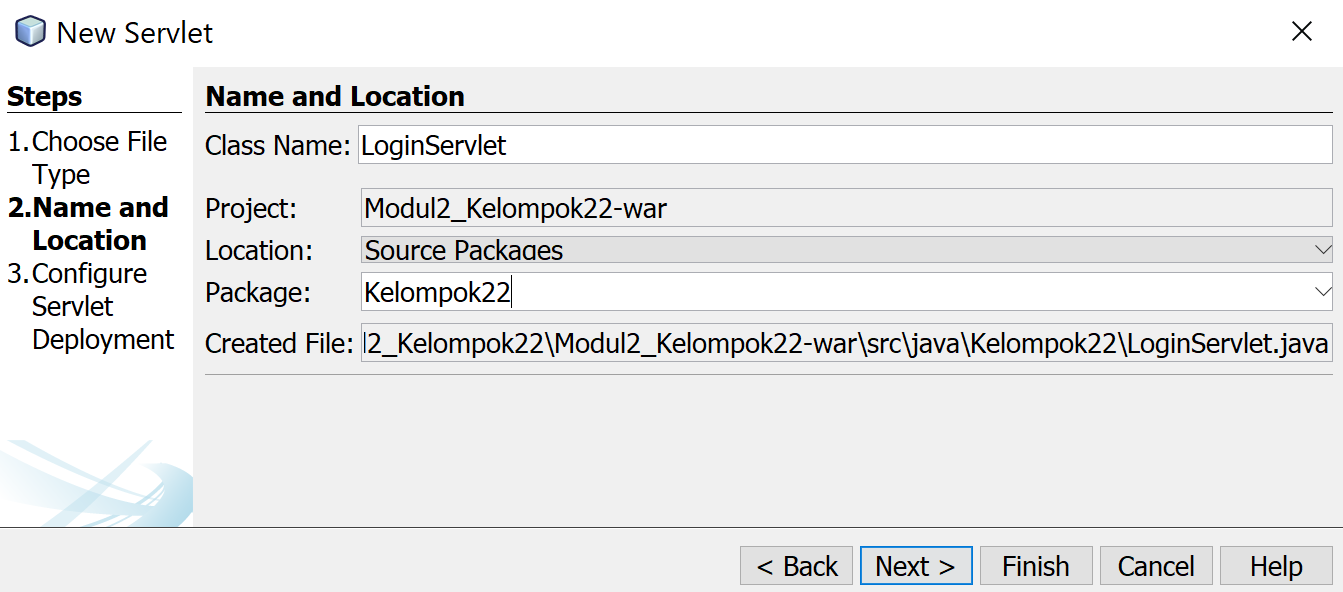
|  |
| --- |
| <%@page contentType="text/html" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <title>Tahun Depan Wisuda</title>  <meta name="viewport" content="width=device-width, initial-scale=1.0">  </head>  <body>  <header>  <h1 style="text-align:center">SEMANGAT, TAHUN DEPAN WISUDA</h1>  </header>  <main style="display:flex;justify-content:center;flex-direction: column;">  <% String stat = (String)request.getAttribute("status");  if(stat != "true"){%>  <form action="Login" method="post" style="display:flex;justify-content:center;flex-direction: column;align-items: center;">  <label for="nama" style="margin-bottom:5px">Nama Mahasiswa</label>  <input id="nama" type="text" name="nama" placeholder="Nama Mahasiswa">  <span style="color:red">${namaEr}</span>  <br>  <label for="nim" style="margin-bottom:5px">NIM Mahasiswa</label>  <input id="nim" type="text" name="nim" placeholder="Nim Mahasiswa">  <span style="color:red">${nimEr}</span>  <span style="color:red">${userEr}</span>  <br>  <input type="submit" name="submit" value="Cari" style="width:150px">  </form>  <%} else {%>  <h2 style="text-align: center">Hallo ${namaMahasiswa}, Semoga cepat lulus ya</h2>  <form action="Login" method="get" style="display:flex;justify-content:center;flex-direction: column;align-items: center;">  <input type="submit" name="keluar" value="Kembali" style="width:150px">  </form>  <%}%>  </main><br>  <footer style="text-align:center">&copy; Copyright Praktikum Rekayasa Perangkat Lunak 2019</footer>  </body>  </html> |

1. Buat file Sevlet dengan cara klik kanan pada project web lalu pilih **New>Sevlet.**



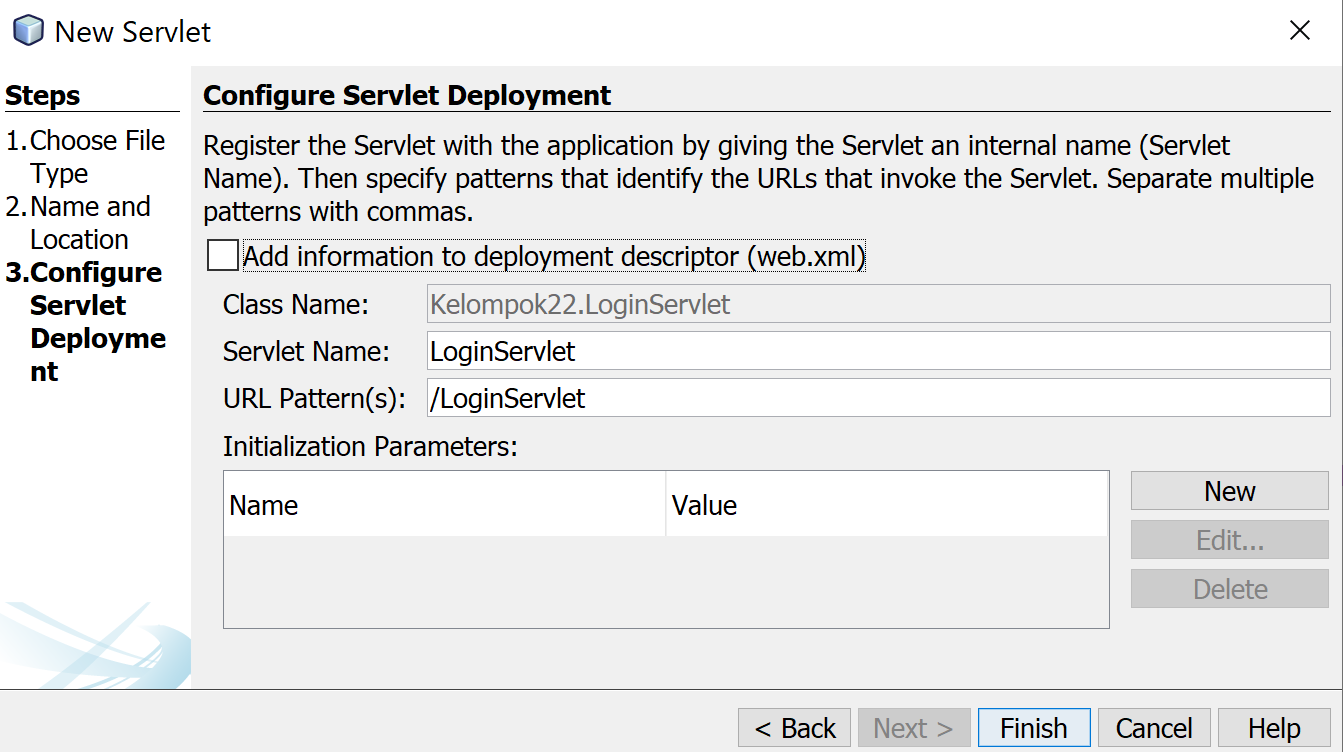
Gambar 3. Buat Servlet

1. Beri nama LoginSevlet pada kolom **Class Name** dan KelompokXX **( XX = No. Kelompok )** padapackage.



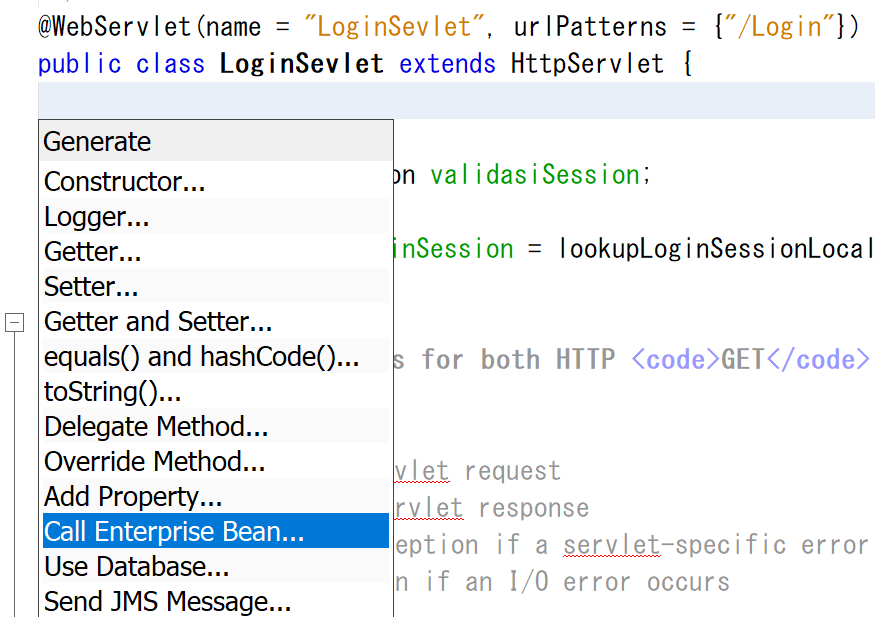
Gambar 3. Buat LoginServlet

1. Pastikan **Add information** di uncheck dan ganti URL Pattern(s) menjadi /Login



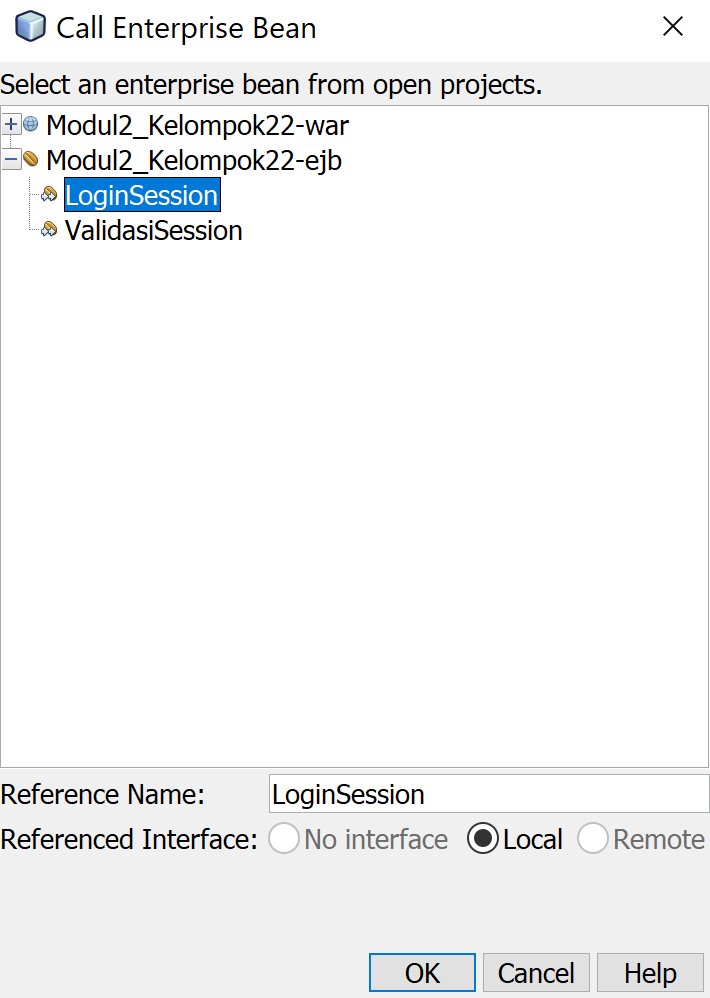
Gambar 3. Ganti URL

1. Import LoginSession dan ValidasiSession dengan cara klik kanan pada text editor dan pilih Insert Code lalu pilih Call Enterprise Bean. Note: Klik kanan tepat public class LoginSevlet, untuk lebih jelasnya lihat pada gambar.

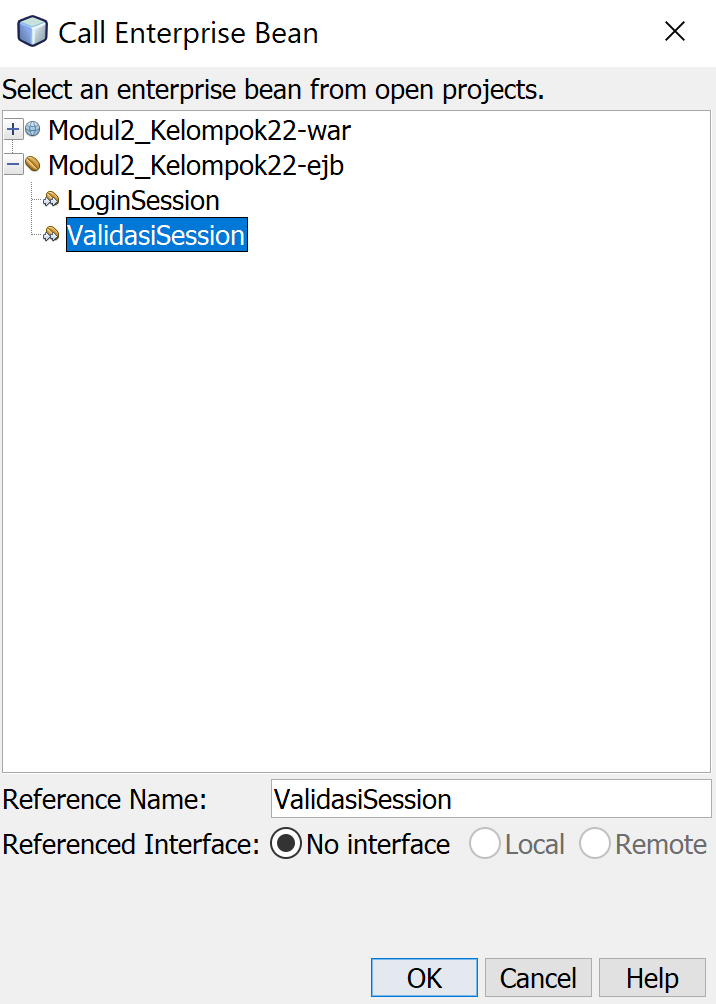


Gambar 3. Call Enterprise Bean

1. Pilih LoginSession lalu klik OK dan ulang dari langkah 22 untuk impor ValidasiSession.



Gambar 3. Pilih LoginSession



Gambar 3. Pilih ValidasiSession

1. Hapus source code pada method **processRequest,** lalutambahkan source code berikut.

|  |
| --- |
| validasiSession = new ValidasiSession();    request.setAttribute("status", "false");  RequestDispatcher rd = getServletContext().getRequestDispatcher("/LoginView.jsp");  rd.forward(request, response); |

1. Tambahkan source code berikut didalam method **doGet**.

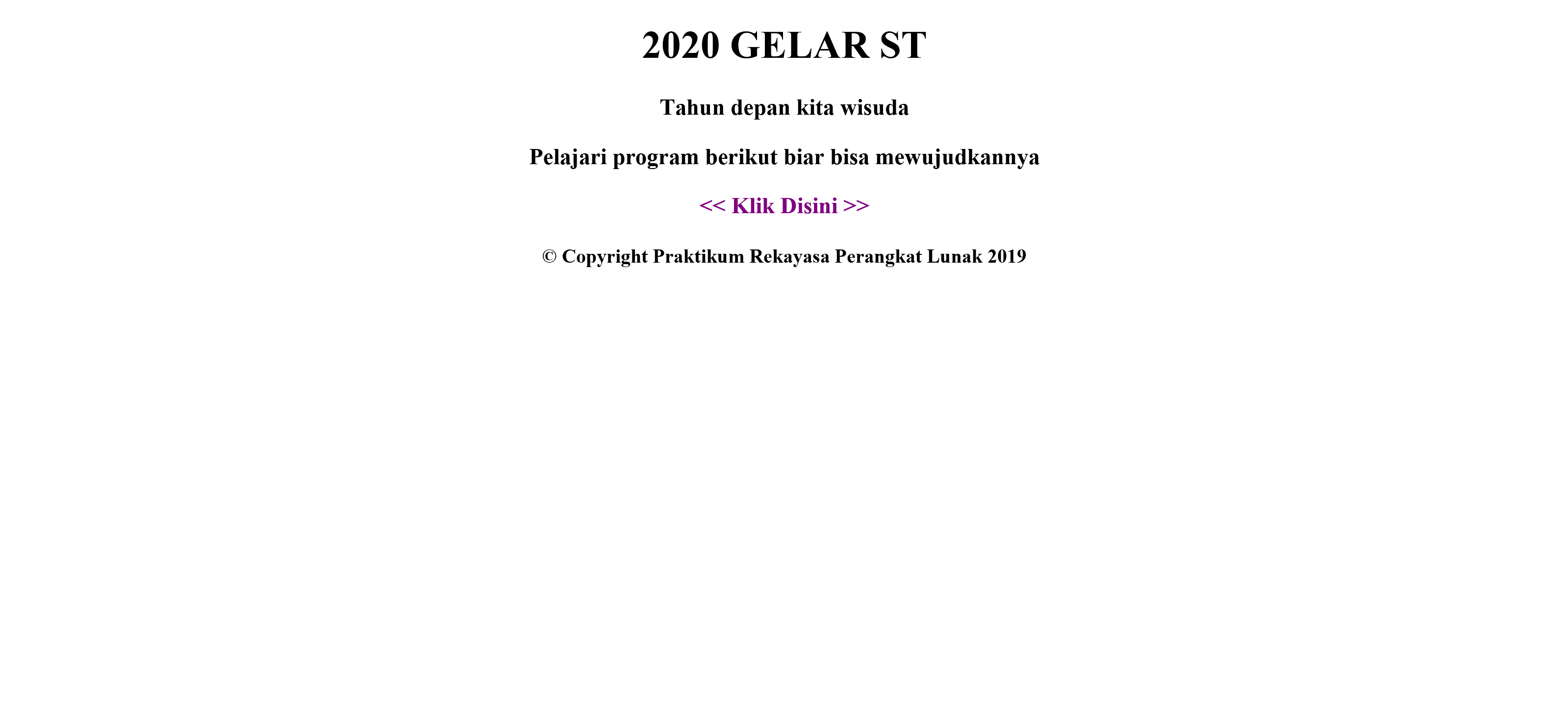
|  |
| --- |
| if("Keluar".equals(request.getParameter("keluar"))){  loginSession.setLoginStatus(false);  request.setAttribute("nimEr", "");  request.setAttribute("namaEr", "");  } |

1. Tambahkan source code berikut didalam method **doPost**.

|  |
| --- |
| String nama = request.getParameter("nama");  String nim = request.getParameter("nim");  boolean stNama, stNim = false;  stNama = validasiSession.nama(nama);  stNim = validasiSession.nim(nim);  if (stNim && stNama){  if (loginSession.Login(nama, nim)) {  loginSession.setLoginStatus(true);  } else {  request.setAttribute("userEr", "Mahasiswa tidak terdaftar");  }  }  else{  if (!stNim) request.setAttribute("nimEr", "Inputan Salah");  if (!stNama) request.setAttribute("namaEr", "Inputan Salah");  }    if (loginSession.isLoginStatus()) {  request.setAttribute("status", "true");  RequestDispatcher rd = getServletContext().getRequestDispatcher("/LoginView.jsp");  rd.forward(request, response);  } else {  request.setAttribute("status", "false");  RequestDispatcher rd = getServletContext().getRequestDispatcher("/LoginView.jsp");  rd.forward(request, response);  } |

1. Jalankan project enterprise dengan cara klik kanan project enterprise lalu klik pilih run.

**3.5 Hasil Percobaan**

****

Gambar 3. Tampilan awal

*Source code* tampilan diatas berasal dari *Web Page* yang ada pada NamaProject.war.

****

Gambar 3. Tampilan login

Untul LoginView diatas dibuat dengan JSP, dimana tampilan ini berfungsi sebagai Login dari web yang dibuat. Setelah pembuatan ini, akan dibuat Servlet yang mengatur proses *backend* dari sistem login web.

****

Gambar 3. Tampilan jika inputtan salah

Inputtan salah jika Nama Mahasiswa dan NIM Mahasiswa tidak sesuai dengan apa yang di input pada *source code* di Class LoginSession. Sehingga, Nama Mahasiswa dan NIM yang diambil harus sesuai.

Untuk Class yang memastikan Nama Mahasiswa dan NIM sama, ada pada class Validasi Session.

Semua aktivitas dilakukan oleh Servlet yang sudah mengimport method-method pada masing-masing Class yang tersedia (contoh:Validasi Session)

****

Gambar 3. Tampilan jika inputtan benar

**3.6 Tugas dan Pembahasan**

**3.6.1 Tugas**

Dalam tugas, terdapat beberapa ketentuan yaitu ketentuan umum dan ketentuan tambahan. Pada ketentuan umum, kami diminta untuk menambahkan mahasiswa yang dapat dicari menjadi minimal sebanyak 5 orang, menggunakan array untuk menyimpan data nama dan nim mahasiswa, dan menggunakan 1 method untuk enkapsulasi public pada Session Bean. Berikut adalah source code LoginSession.java untuk tugas utama.

package Kelompok22;

import javax.ejb.Stateful;

/\*\*

\*

\* @author Evita Cindy S

\*/

@Stateful

public class LoginSession implements LoginSessionLocal {

private final String [] Nama = {"Evita Cindy Septiviani", "Rio Lutfi Adipradana", "Saitama", "Naruto", "Sasuke"};

private final String [] Nim = {"21120116120004", "21120116140069", "21120116100000", "21120116100001", "21120116100002"};

private String checkNama (String param){

for (int i = 0; i< Nama.length; i++){

if (param.equals (Nama[i])){

return "Nama Praktikan : " + Nama[i]+"("+Nim[i]+")";

}

}

return null;

}

private String checkNim (String param){

for (int i = 0; i< Nim.length; i++){

if (param.equals (Nim[i])){

return "Nama Praktikan : " + Nama[i]+"("+Nim[i]+")";

}

}

return null;

}

public String search (String param){

if (checkNama(param) != null) return checkNama(param);

else if (checkNim(param) != null) return checkNim(param);

return null;

}

private boolean loginStatus = false;

@Override

public boolean Login(String Nama, String Nim) {

return Nama.equals(this.Nama[0]) && Nim.equals(this.Nim[0]);

}

@Override

public boolean isLoginStatus() {

return loginStatus;

}

@Override

public void setLoginStatus(boolean loginStatus) {

this.loginStatus = loginStatus;

}

}



Pada source code tersebut, kami telah menambahkan jumlah mahasiswa yang dapat dicari sebanyak 5 orang dan menggunakan array untuk menyimpan data nama dan NIM mahasiswa dalam code private final String [] Nama = {"Evita Cindy Septiviani", "Rio Lutfi Adipradana", "Saitama", "Naruto", "Sasuke"}; dan private final String [] Nim = {"21120116120004", "21120116140069", "21120116100000", "21120116100001", "21120116100002"}; untuk memasukkan nama mahasiswa dan NIM, digunakan array sebagai kumpulan variabel dengan tipe data yang sama sehingga apabila ingin menggunakan fungsi pencarian, data akan diambil dan disimpan melalui array tersebut.

Kemudian kami juga membuat source code menjadi 1 method dengan enkapsulasi publik saja. Fungsi dari pengubahan menjadi 1 method ini yaitu agar pada pengecekan nama maupun NIM menjadi lebih ringkas dan apabila ada kesalahan yang dilakukan saat pengisian nama maupun NIM, sistem dapat memberi tahu bagian manakah yang salah, apakah hanya nama, apakah hanya NIM, atau keduanya.

Selain itu, dengan menulis 1 method public saja dibandingkan 2 method membuat kerja sistem tidak berat dan coding menjadi lebih ringkas karena dikerjakan pada method yang sama.

Kemudian untuk tugas tambahan kami memperbaiki tampilan pada index.html dan loginview.jsp. Berikut adalah source code pada index.html:

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>2020 Gelar ST</title>

</head>

<body background="5086.jpg">

<h1 style="text-align:center; font-family:Arial Black, Gadget, sans-serif; font-size:44px;"> 2020 GELAR ST</h1>

<center><img src="undip.png" alt="Universitas Diponegoro" style="width:160px;height:200px"></center>

<h3 style="text-align:center; font-family:Arial, Helvetica, sans-serif; font-size:24px;">Tahun depan kita wisuda</h3>

<h3 style="text-align:center; font-family:Arial, Helvetica, sans-serif; font-size:24px;">Pelajari program berikut biar bisa mewujudkannya</h3>

<h3 style="text-align:center;; font-family:Arial Black, Gadget, sans-serif; font-size:30px;"><a href="Login" style="text-decoration:none"><< Klik Disini >></a></h3>

<h4 style="text-align:center; font-family:Arial, Helvetica, sans-serif; font-size:24px;">&copy; Copyright Praktikum Rekayasa Perangkat Lunak 2019</h4>

</body>

</html>



Pada source tersebut, kami menambahkan background melalui file eksternal yang di-copy ke dalam project. Kemudian kami juga mengganti jenis font dan ukuran font pada semua teks yang terdapat dalam index.

Selanjutnya berikut adalah source code pada loginview.jsp.

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

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Tahun Depan Wisuda</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body background="5086.jpg">

<h1 style="text-align:center; font-family:Arial Black, Gadget, sans-serif; font-size:44px;"> 2020 GELAR ST</h1>

<header>

<h1 style="text-align:center; font-family:Arial Black, Gadget, sans-serif; font-size:26px;">SEMANGAT, TAHUN DEPAN WISUDA</h1>

<center><img src="undip.png" alt="Universitas Diponegoro" style="width:160px;height:200px"></center>

</header>

<br>

<main style="display:flex;justify-content:center;flex-direction: column;">

<% String stat = (String)request.getAttribute("status");

if(stat != "true"){%>

<form action="Login" method="post" style="display:flex;justify-content:center;flex-direction: column;align-items: center;">

<label for="nama" style="margin-bottom:5px; font-family:Arial, Helvetica, sans-serif; font-size:16px;">Nama Mahasiswa</label>

<input id="nama" type="text" name="nama" placeholder="Nama Mahasiswa">

<span style="color:red">${namaEr}</span>

<br>

<label for="nim" style="margin-bottom:5px; font-family:Arial, Helvetica, sans-serif; font-size:16px;">NIM Mahasiswa</label>

<input id="nim" type="text" name="nim" placeholder="Nim Mahasiswa">

<span style="color:red">${nimEr}</span>

<span style="color:red">${userEr}</span>

<br>

<input type="submit" name="submit" value="Cari" style="width:150px">

</form>

<%} else {%>

<h2 style="text-align: center">Hallo ${namaMahasiswa}, Semoga cepat lulus ya</h2>

<form action="Login" method="get" style="display:flex;justify-content:center;flex-direction: column;align-items: center;">

<input type="submit" name="keluar" value="Kembali" style="width:150px">

</form>

<%}%>

</main><br>

<footer style="text-align:center; font-family:Arial, Helvetica, sans-serif; font-size:14px;">&copy; Copyright Praktikum Rekayasa Perangkat Lunak 2019</footer>

</body>

</html>



Pada source tersebut, kami menambahkan background yang sama juga seperti pada index serta mengubah font pada teks yang ada sehingga tampilan menjadi terlihat lebih rapi dan tertata.

Hasilnya, tampilan yang kami buat menjadi seperti ini.







**3.7 Kesimpulan**

1. Komponen *Stateful* menyimpan keadaan dalam program.
2. Komponen *Stateless* tidak menyimpan keadaan apapun.
3. Servlet digunakan untuk menerima respon dari HTTP ke program Java.
4. Glassfish Server pada praktikum digunakan sebagai localhost untuk mengecek *progress* dari website.
5. Java EE digunakan karena *support* untuk Java Web Programming, tidak seperti Java SE.