# BAB X – Pengenalan Database pada OOP.

Bab X, Praktikan akan memahami sistem database yang ada dalam Pemrograman Berorientasi Object dimana akan di beri perkenalan database MySQL menggunakan localhost dengan software XAMPP.

Tujuan setelah mempelajari bab 10 ini adalah:

* 1. Memahami Pegertian dasar Database MySQL dalam OOP.
  2. Memahami tentang bahasa yang digunakan dalam Database.
  3. Memahami prasyarat yang harus di lakukan untuk menghubungkan Database ke program dalam OOP.

1. **Pengenalan Database Dalam OOP**
2. **Basis Data (*Database*)**

Basis data (database) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan.

1. **XAMPP**

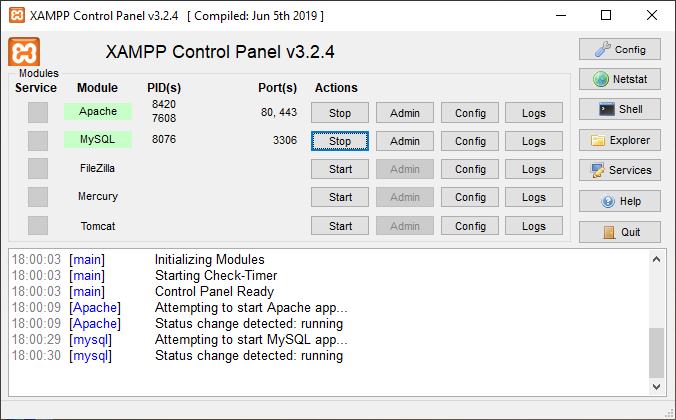
Xampp itu kepanjangan dari Apache, PHP, MySQL dan phpMyAdmin. XAMPP merupakan tool yang menyeKdiakan paket perangkat lunak ke dalam satu buah paket. XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Sifatnya hampir mirip dengan Web Server yang ada di internet, hanya bedanya Xampp tidak perlu terkoneksi ke internet alias dapat berdiri sendiri atau localhost.

1. **Koneksi java ke mysql**

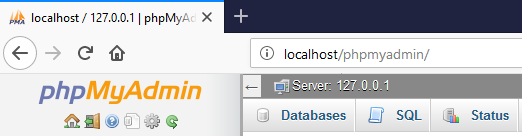
MySQL adalah salah satu DBMS (Database Management System) yang menggunakan bahasa SQL (Structured Query Language) untuk mengelola informasi dalam basis data. MySQL memungkinkan kita untuk membuat dan mengelola basis data serta kontennya, termasuk melakukan operasi seperti menambahkan, mengubah, dan menghapus data dalam basis data tersebut.

Dalam pengembangan aplikasi menggunakan bahasa pemrograman Java, kita dapat menghubungkan aplikasi tersebut dengan basis data MySQL. Untuk melakukan koneksi antara Java dan basis data MySQL, kita memerlukan JDBC (Java Database Connectivity) driver. JDBC API (Java Database Connectivity Application Programming Interface) adalah antarmuka pemrograman dalam bahasa Java yang memungkinkan kita berinteraksi dengan informasi dalam basis data.

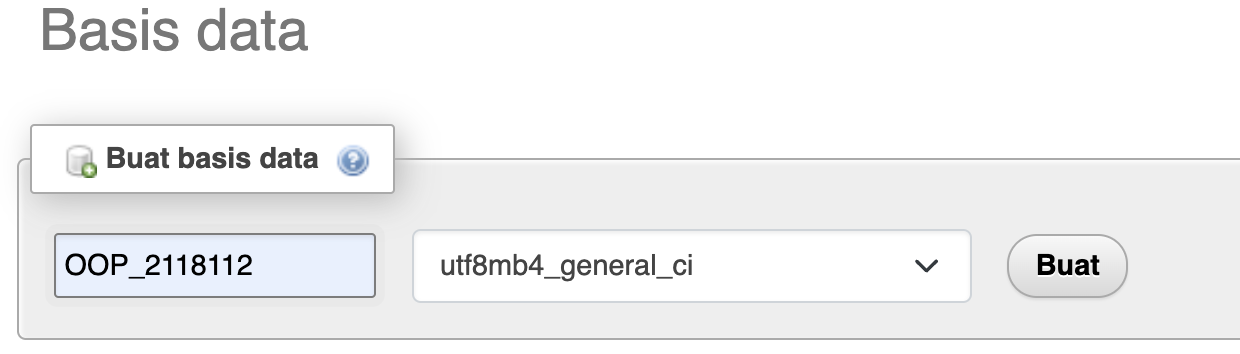
1. **Project Praktikum Import Database**
2. Buka aplikasi XAMPP lalu Start pada bagian Apache dan MySQL.



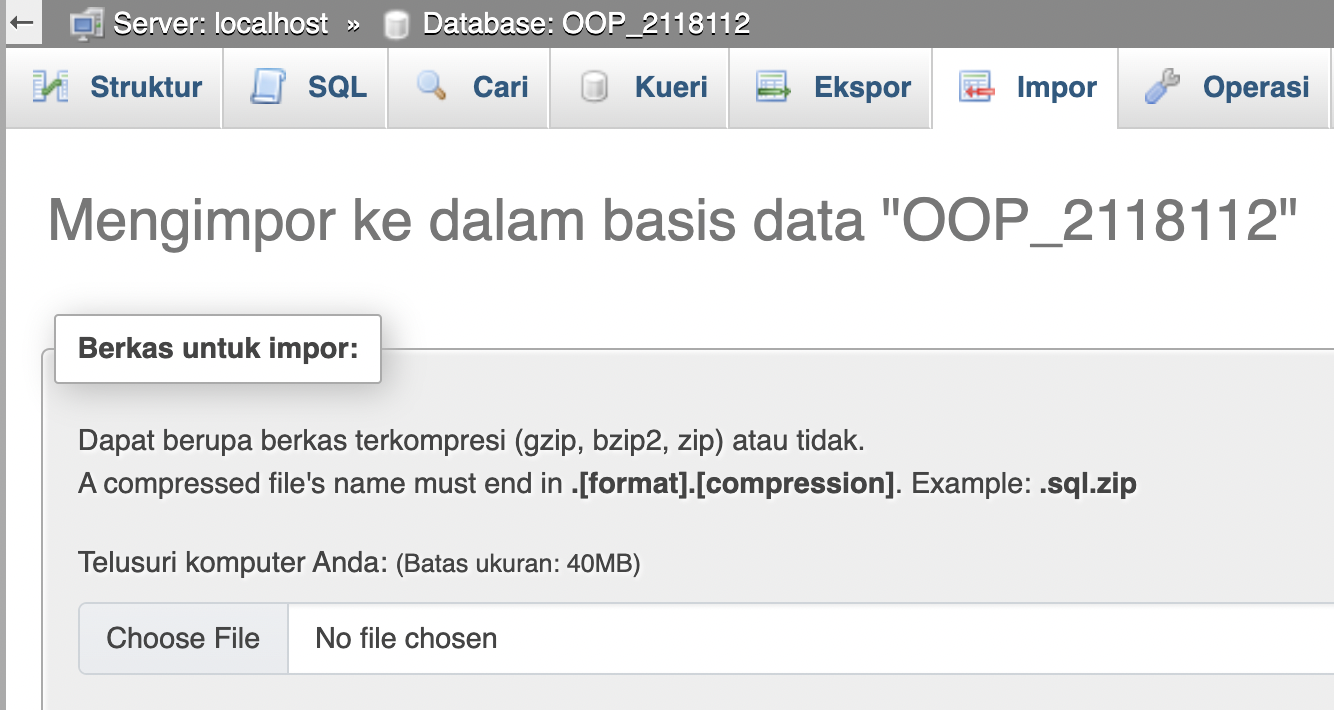
1. Buka Browser lalu ketik pada Address Bar “localhost/phpmyadmin” seperti gambar di bawah ini.



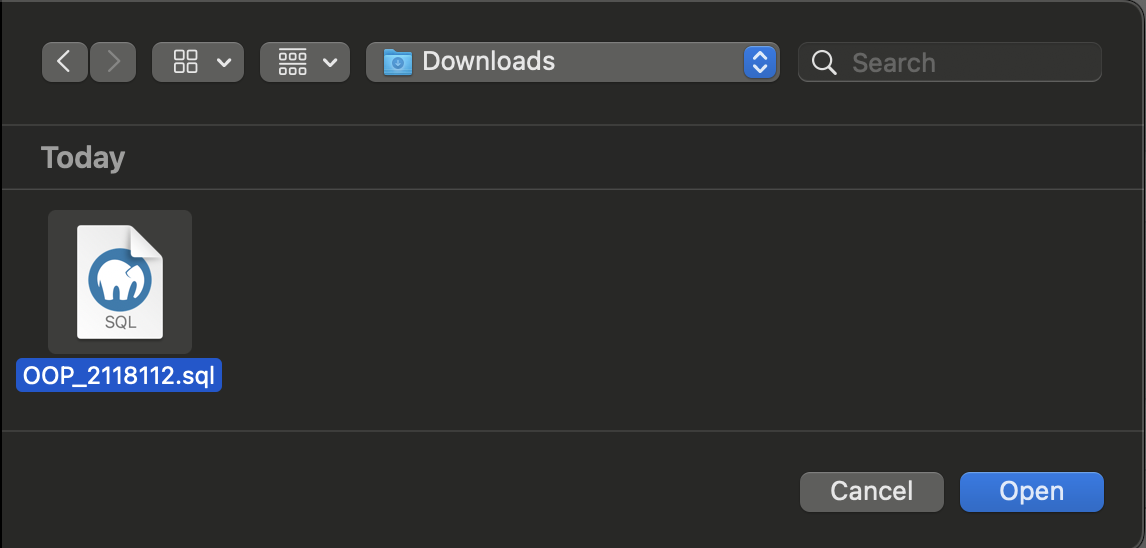
1. Klik pada bagian tab “Basis Data” lalu beri nama Database dengan sesuai OOP\_NIM lalu klik Buat.



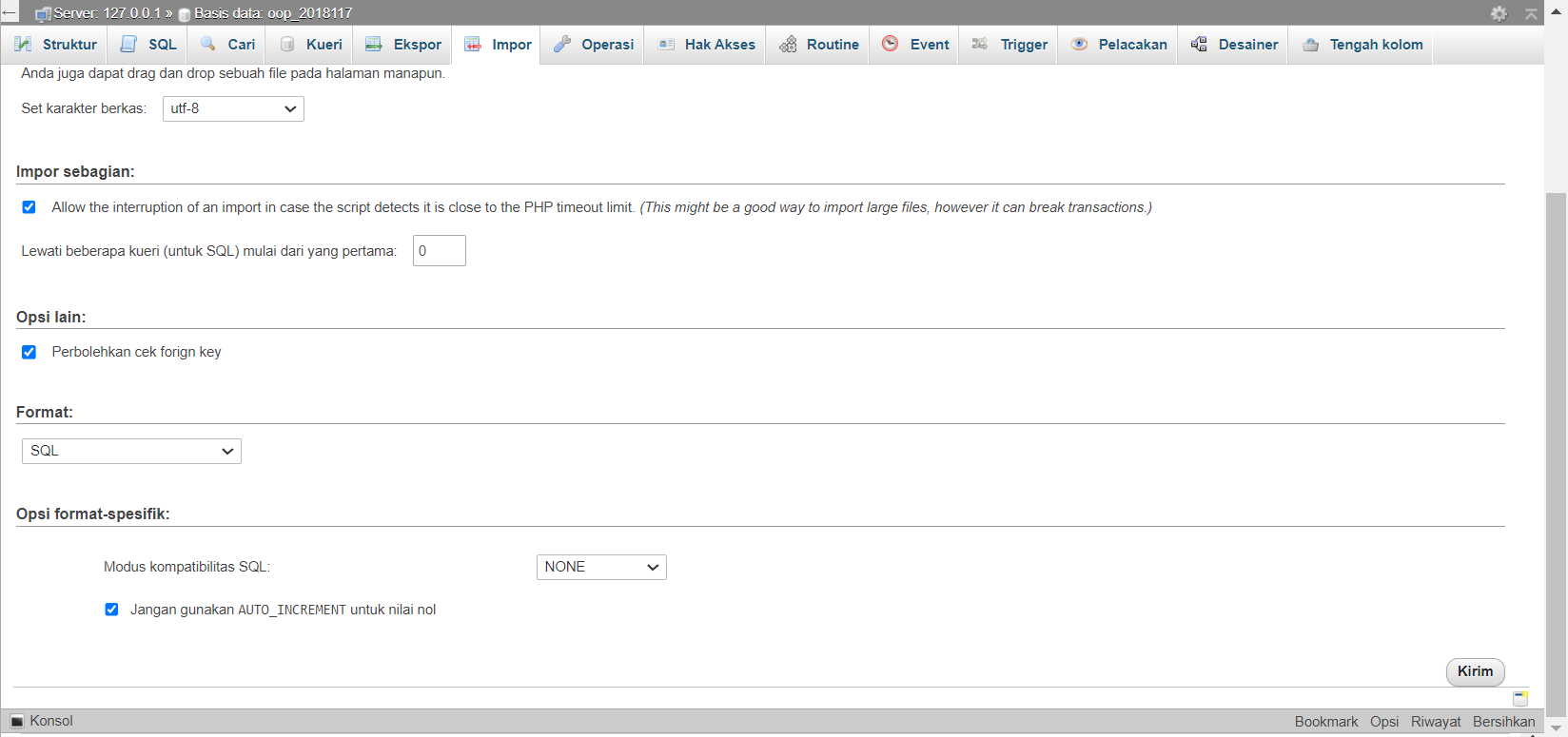
1. Setelah itu, klik database yang dibuat, pada menu bagian atas pilih impor, kemudian pilih file .sql yang sudah didownload sebelumnya.



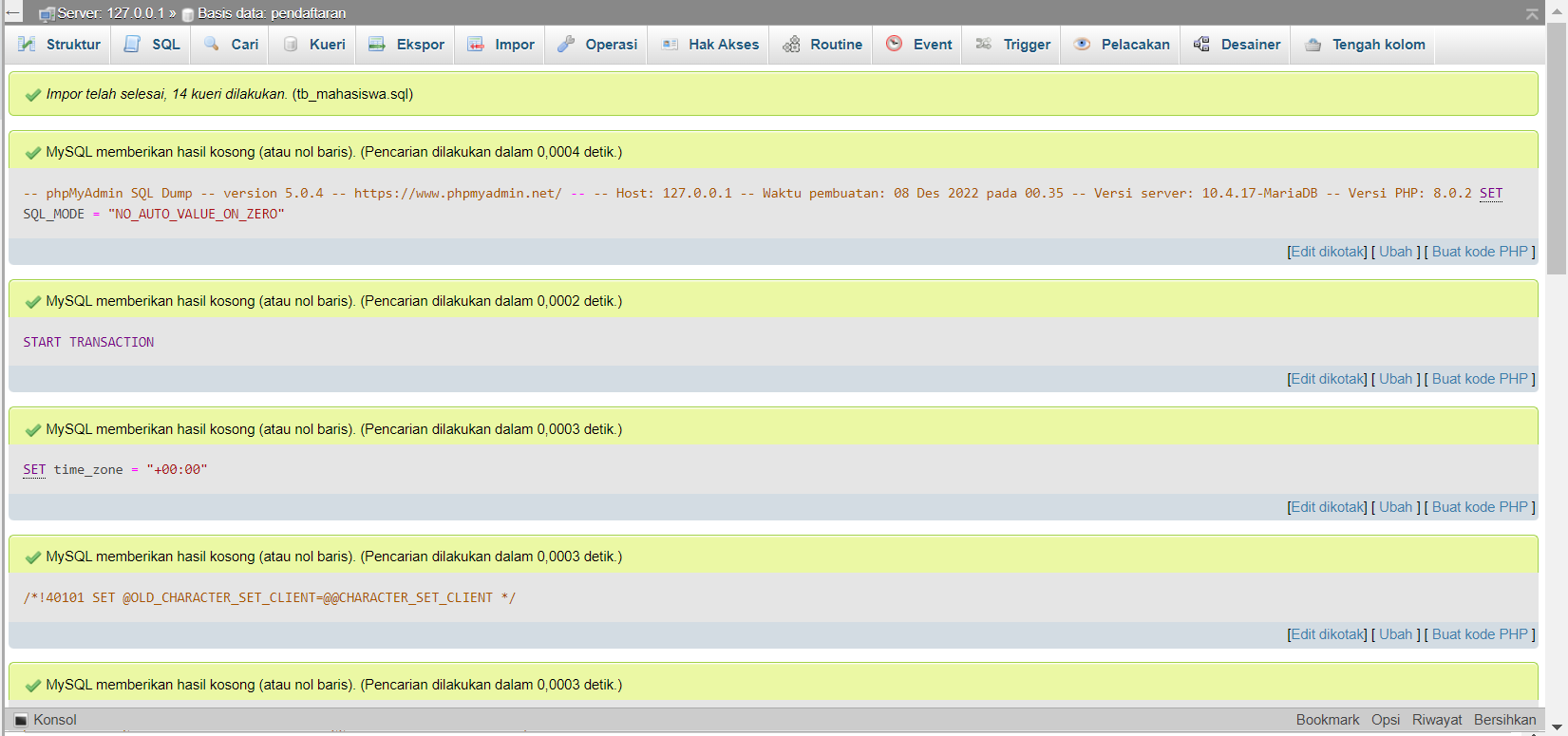
1. Browse file .sql yang sudah didownload kemudian pilih ,setelah itu klik open



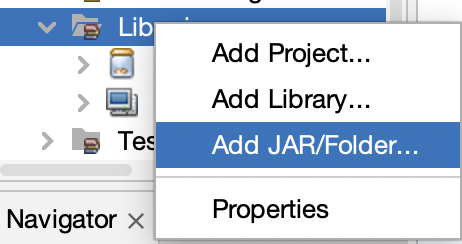
1. Setelah klik open, selanjutnya pada halaman paling bawah , klik kirim



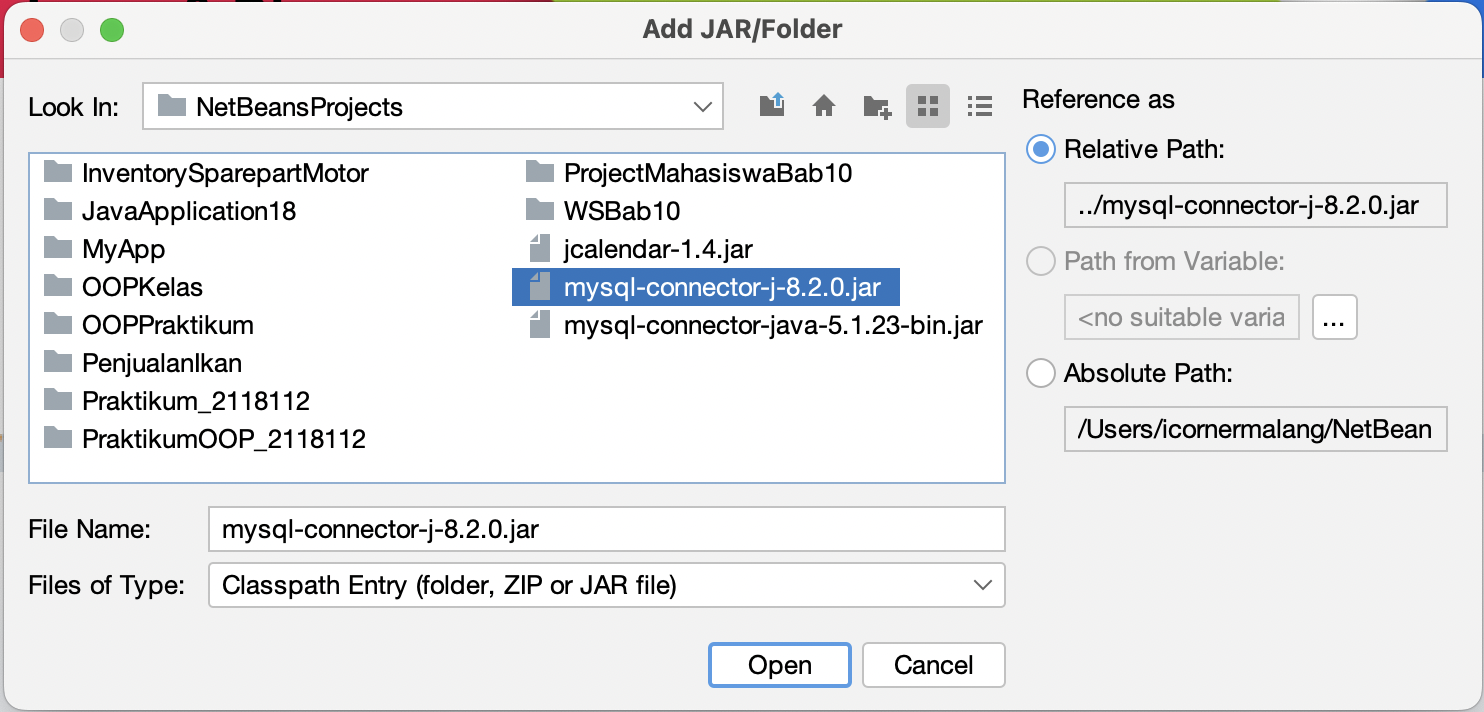
1. Jika berhasil akan tampil seperti dibawah. Lakukan hal yang sama untuk impor file lainnya.



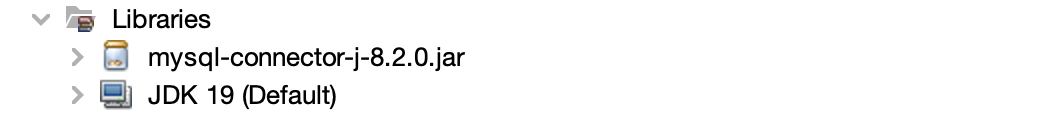
1. **Menambah Library pada netbens**
2. **NB: untuk folder mysql.jar usahakan masukkan ke dalam file Neatbeans Project**
3. Tambahkan library dengan cara klik kanan pada library, pada project yang akan dikoneksikan dengan database kemudian pilih **Add JAR/Folder**



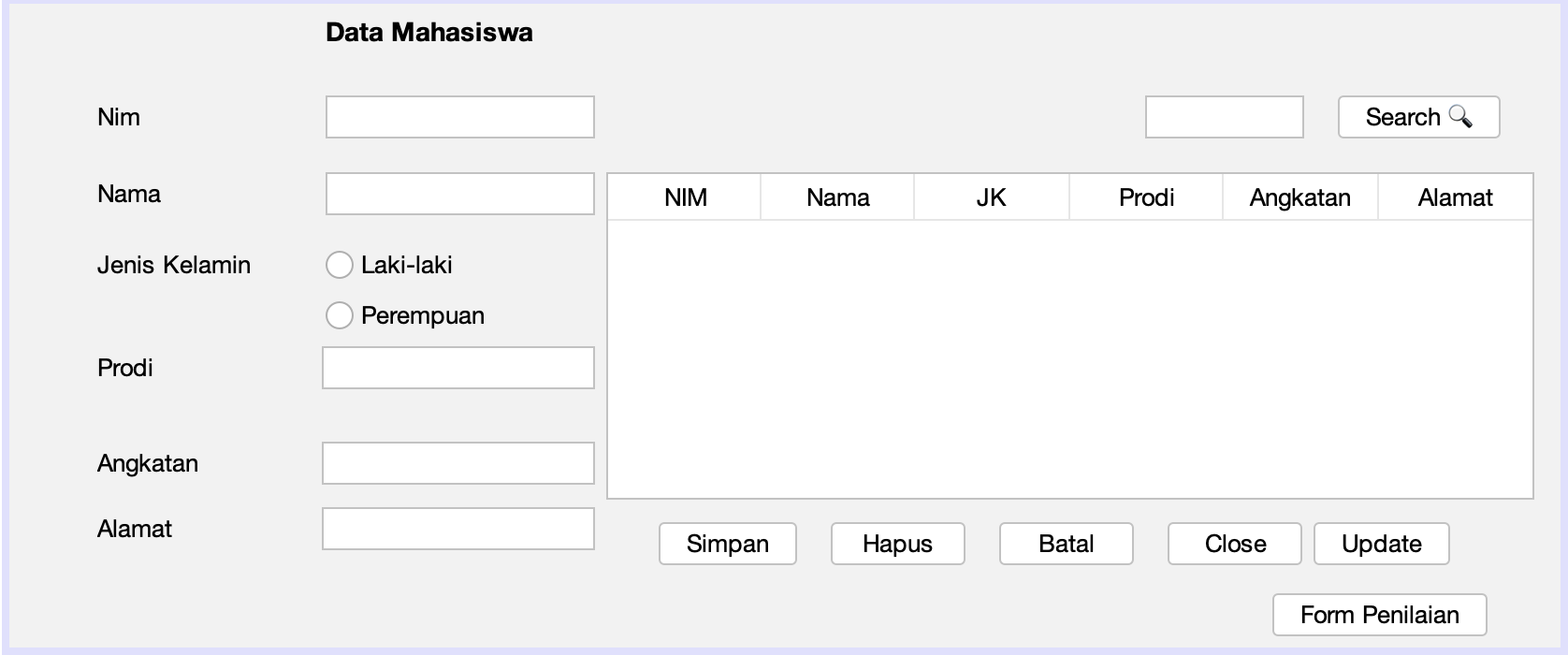
1. Kemudian cari file **mysql-connector-j-8.2.0.jar** > **klik file** tersebut kemudian **Open**



1. Jika berhasil melakukan import pada libraries akan ditampil seperti gambar dibawah



1. **Project Praktikum Mengkoneksikan GUI\_Mahasiswa dengan Database.**
2. Buka GUI\_Mahasiswa pada package “Project\_praktikum” yang telah di buat.

****

1. Kemudian pada GUI\_Mahasiswa, pilih tab Source, masukan Script di bawah ini di bagian paling atas dalam Source GUI\_Mahasiswa.

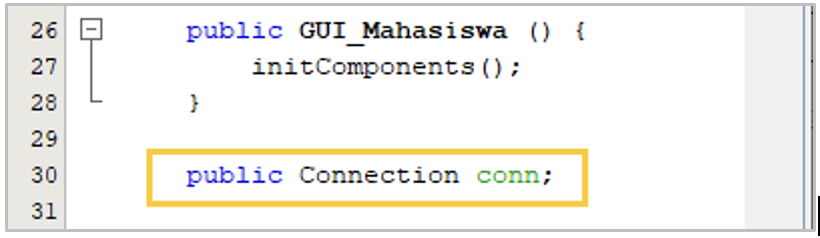
*Source Code:*

|  |
| --- |
| import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

1. Buat Variable Koneksi, Letakan script di bawah ini di bawah Konstruktor GUI\_Mahasiswa.

|  |
| --- |
| public Connection conn; |

Lokasi Peletakkan Script :



1. Membuat Method “**koneksi()**” letakan script di bawah ini di bawah pendeklarasian Variable “Conn”.

*Source code :*

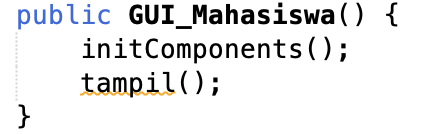
|  |
| --- |
| public void koneksi() throws SQLException {  try {  conn = null;  Class.forName("com.mysql.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost/OOP\_2118112?user=root&password=");  } catch (ClassNotFoundException ex) {  Logger.getLogger(GUI\_Mahasiswa.class.getName()).log(Level.SEVERE, null, ex);  } catch (SQLException e) {  Logger.getLogger(GUI\_Mahasiswa.class.getName()).log(Level.SEVERE, null, e);  } catch (Exception es) {  Logger.getLogger(GUI\_Mahasiswa.class.getName()).log(Level.SEVERE, null, es);  }  } |

1. Membuat Method “**tampil()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

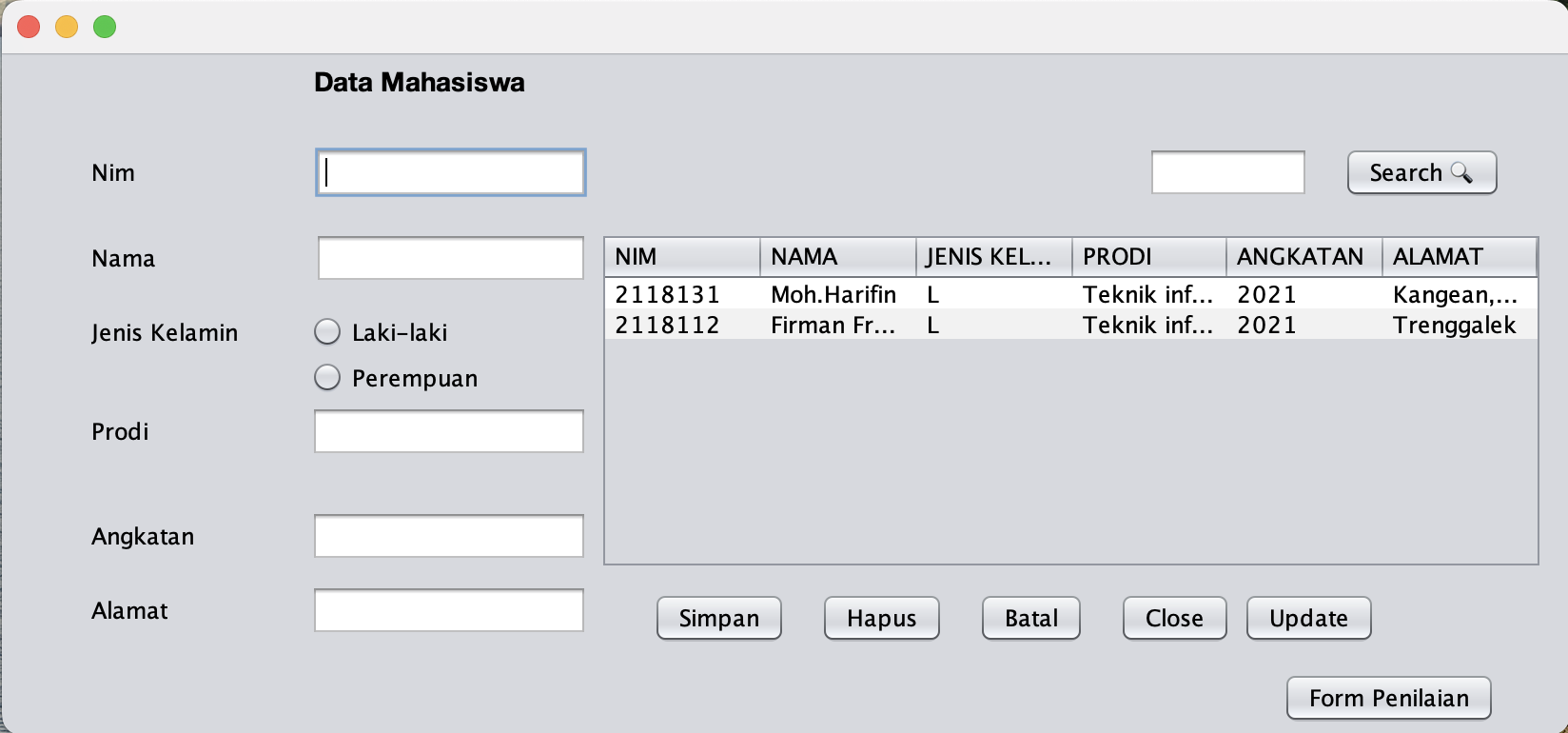
*Source Code:*

|  |
| --- |
| public void tampil() {  DefaultTableModel tabelhead = new DefaultTableModel();  tabelhead.addColumn("NIM");  tabelhead.addColumn("NAMA");  tabelhead.addColumn("JENIS KELAMIN");  tabelhead.addColumn("PRODI");  tabelhead.addColumn("ANGKATAN");  tabelhead.addColumn("ALAMAT");  try {  koneksi();  String sql = "SELECT \* FROM tb\_mahasiswa";  Statement stat = conn.createStatement();  ResultSet res = stat.executeQuery(sql);  while (res.next()) {  tabelhead.addRow(new Object[]{res.getString(2), res.getString(3), res.getString(4), res.getString(5), res.getString(6), res.getString(7),});  }  table\_data\_mahasiswa.setModel(tabelhead);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "BELUM TERKONEKSI");  }  } |

1. Panggil method “**tampil()**” pada konstruktor GUI\_Mahasiswa agar ketika program pertama kali di jalankan langsung menjalankan method tersebut.

****

1. Jalankan GUI\_Mahasiswa jika Table Langsung menampilkan isi database saat pertama kali program berjalan maka berhasil.



1. Membuat method “**refresh()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void refresh() {  new GUI\_Mahasiswa().setVisible(true);  this.setVisible(false);  } |

Method ini membuat Method CRUD akan lebih responsive di karenakan saat selesai melakukan perintah CRUD akan langsung merefresh program.

1. Membuat method “**insert()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

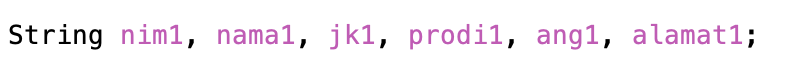
|  |
| --- |
| public void insert() {  String Nim = txtNim.getText();  String Nama = txtNama.getText();  String jk;  if (radiobtnLaki.isSelected()) {  jk = "L";  } else {  jk = "P";  }  String Prodi = txtProdi.getText();  String Ang = txtAngkatan.getText();  String alamat = txtAlamat.getText();  try {  koneksi();  Statement statement = conn.createStatement();  statement.executeUpdate("INSERT INTO tb\_mahasiswa (nim, nama,jk, prodi, th\_angkatan,alamat)"  + "VALUES('" + Nim + "','" + Nama + "','" + jk + "','" + Prodi + "','" + Ang + "','" + alamat + "')");  statement.close();  JOptionPane.showMessageDialog(null, "Berhasil Memasukan Data Mahasiswa!" + "\n" + alamat);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Terjadi Kesalahan Input!");  }  refresh();  } |

1. Panggil Method “**Insert()**” pada button “**btnSimpan**”.

*Source Code:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  insert();  } |

1. Membuat variable baru yang akan digunakan pada method itempilih dan update. Letakkan dibawah class GUI\_Mahasiswa.

****

1. Membuat method “**update()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void update() {  String Nim = txtNim.getText();  String Nama = txtNama.getText();  String jk;  if (radiobtnLaki.isSelected()) {  jk = "L";  } else {  jk = "P";  }  String Prodi = txtProdi.getText();  String Ang = txtAngkatan.getText();  String alamat = txtAlamat.getText();  String Nimlama = nim1;  try {  Statement statement = conn.createStatement();  statement.executeUpdate("UPDATE tb\_mahasiswa SET nim='" + Nim + "'," + "nama='" + Nama + "',"  + "jk='" + jk + "'" + ",prodi='" + Prodi + "',alamat='" + alamat + "',th\_angkatan='"  + Ang + "' WHERE nim = '" + Nimlama + "'");  statement.close();  conn.close();  JOptionPane.showMessageDialog(null, "Update Data Mahasiswa Berhasil!");  } catch (Exception e) {  System.out.println("Error : " + e);  }  refresh();  } |

1. Panggil Method “**update()**” pada Button “**btnUbah**”

*Source Code:*

|  |
| --- |
| private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  update();  } |

1. Membuat method “**delete()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void delete() {  int ok = JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ?", "Konfirmasi", JOptionPane.YES\_NO\_OPTION);  if (ok == 0) {  try {  String sql = "DELETE FROM tb\_mahasiswa WHERE nim='" + txtNim.getText() + "'";  java.sql.PreparedStatement stmt = conn.prepareStatement(sql);  stmt.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di hapus");  batal();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Data gagal di hapus");  }  }  refresh();  } |

1. Panggil Method “**delete()**” pada button “**btnHapus**”.

*Source Code:*

|  |
| --- |
| private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  delete();  } |

1. Membuat method “**cari()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void cari() {  try {  try ( Statement statement = conn.createStatement()) {  String sql = "SELECT \* FROM tb\_mahasiswa WHERE `nim` LIKE '%" + txtSearch.getText() + "%'";  ResultSet rs = statement.executeQuery(sql); //menampilkan data dari sql query  if (rs.next()) // .next() = scanner method  {  txtNim.setText(rs.getString(2));  txtNama.setText(rs.getString(3));  String jk = rs.getString(4);  if (jk.equalsIgnoreCase("L")) {  radiobtnLaki.setSelected(true);  } else {  radiobtnPerempuan.setSelected(true);  }  txtProdi.setText(rs.getString(4));  txtAngkatan.setText(rs.getString(5));  txtAlamat.setText(rs.getString(6));  } else {  JOptionPane.showMessageDialog(null, "Data yang Anda cari tidak ada");  }  }  } catch (Exception ex) {  System.out.println("Error." + ex);  }  } |

1. Panggil Method “**cari()**” pada button “**btnCari**”.

*Source Code:*

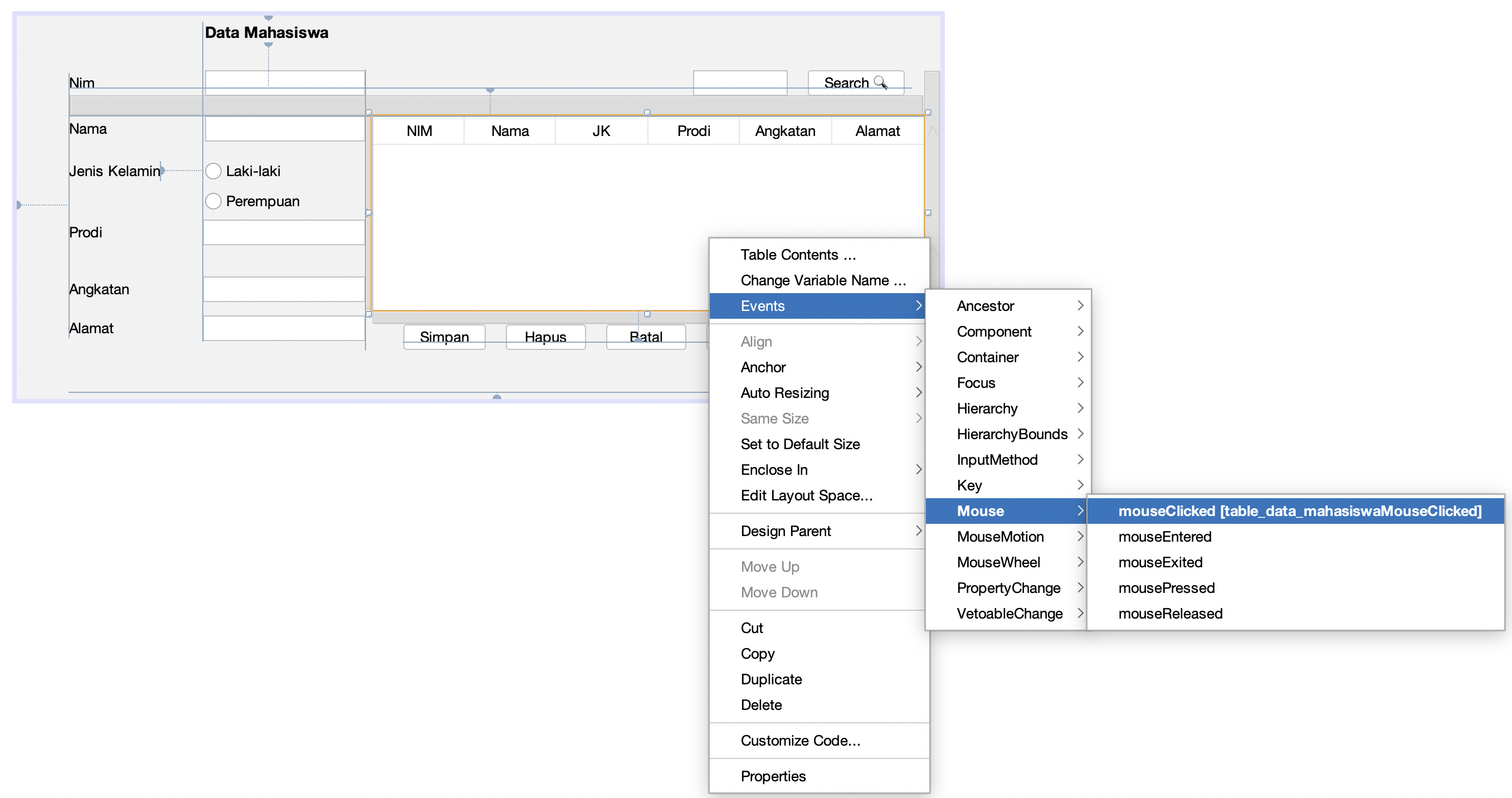
|  |
| --- |
| private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  cari();  } |

1. Membuat method “**itempilih()”**.

*Source code:*

|  |
| --- |
| public void itempilih() {  txtNim.setText(nim1);  txtNama.setText(nama1);  txtProdi.setText(prodi1);  txtAngkatan.setText(ang1);  txtAlamat.setText(alamat1);  if (jk1.equalsIgnoreCase("L")) {  radiobtnLaki.setSelected(true);  } else {  radiobtnPerempuan.setSelected(true);  }  } |

1. Menambahkan events mouseClicked pada jTabel yang tersedia di GUI\_Mahasiswa dengan cara pilih design GUI\_Mahasiswa, kemudian klik kanan pada jTabel, pilih Mouse, kemudian pilih mouseClicked seperti pada gambar berikut.

****

1. Setelah itu tambahkan scipt dibawah ini pada tabel.

*Source code :*

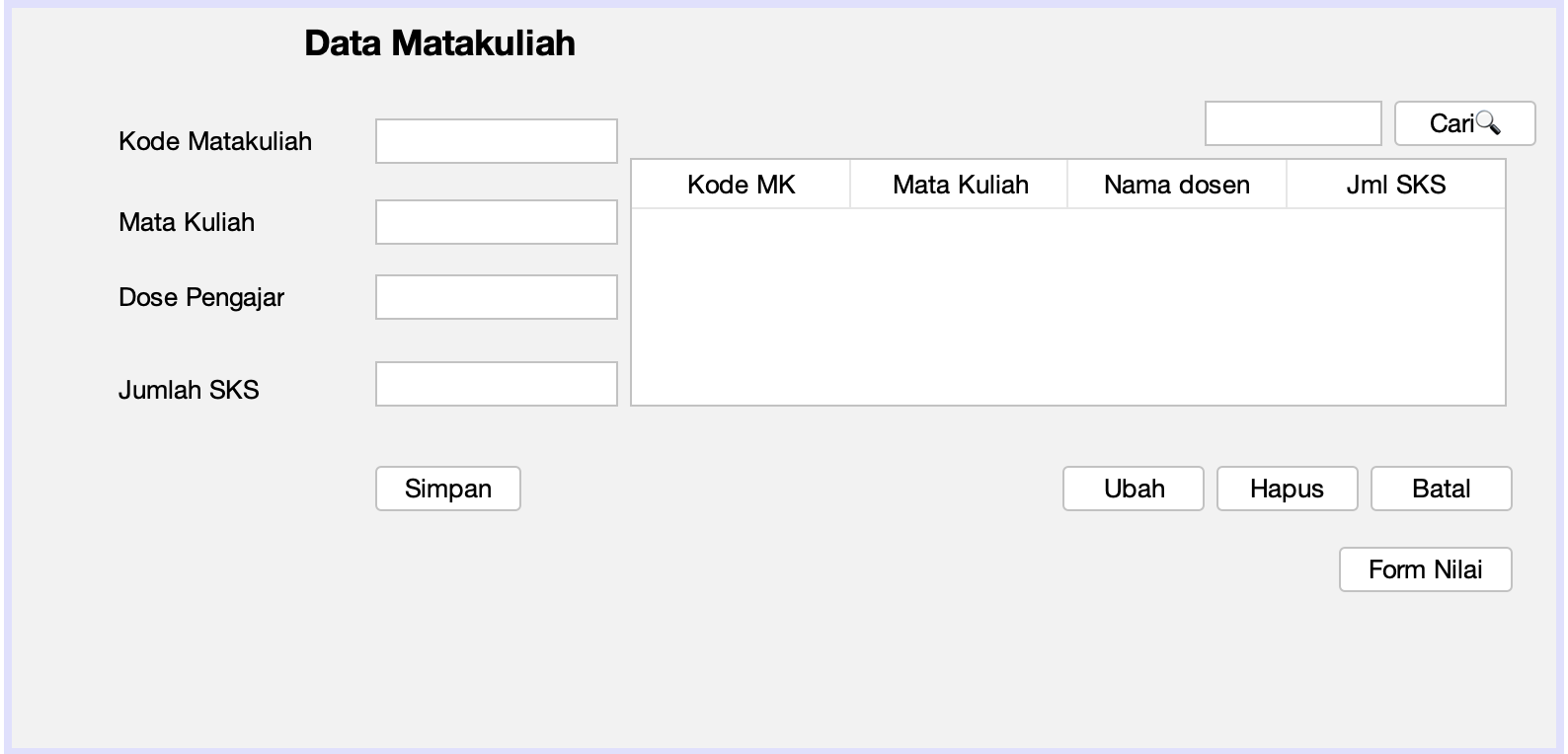
|  |
| --- |
| int tabel = table\_data\_mahasiswa.getSelectedRow();  nim1 = table\_data\_mahasiswa.getValueAt(tabel, 0).toString();  nama1 = table\_data\_mahasiswa.getValueAt(tabel, 1).toString();  jk1 = table\_data\_mahasiswa.getValueAt(tabel, 2).toString();  prodi1 = table\_data\_mahasiswa.getValueAt(tabel, 3).toString();  ang1 = table\_data\_mahasiswa.getValueAt(tabel, 4).toString();  alamat1 = table\_data\_mahasiswa.getValueAt(tabel, 5).toString();  itempilih(); |

Ket : Source code diatas bertujuan menampung data baris yang dipilih “getSelectedRow” kedalam sebuat variable bertipe data int. Variable nim1 bertujuan untuk menampung value dari row yang terpilih dari index ke 0. Selanjutnya “**itempilih()”** bertujuan untuk memanggil method dari itempilih.

1. Tambahkan Script berikut pada button form penilaian

|  |
| --- |
| new GUI\_Penilaian().setVisible(true); |

1. **Project Praktikum Mengkoneksikan GUI\_DataMatkul dengan Database**
2. Buka GUI\_DataMatkul pada package “Project\_praktikum” yang telah di buat.



1. Kemudian pada GUI\_ Matkul, pilih tab Source, masukan Script di bawah ini di bagian paling atas dalam Source GUI\_ DataMatkul.

*Source Code:*

|  |
| --- |
| import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

1. Buat Variable Koneksi, Letakan script di bawah ini di bawah Konstruktor GUI\_Matkul.

|  |
| --- |
| public Connection conn; |

1. Tambahkan Variable baru seperti dibawah ini

|  |
| --- |
| String kode1, mk1, dosen1, jmlsks1; |

1. Membuat Method “**koneksi()**” letakan script di bawah ini di bawah pendeklarasian Variable “Conn”.

*Souce Code:*

|  |
| --- |
| public void koneksi() throws SQLException {  try {  conn = null;  Class.forName("com.mysql.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost/OOP\_2118112?user=root&password=");  } catch (ClassNotFoundException ex) {  Logger.getLogger(GUI\_DataMatkul.class.getName()).log(Level.SEVERE, null, ex);  } catch (SQLException e) {  Logger.getLogger(GUI\_DataMatkul.class.getName()).log(Level.SEVERE, null, e);  } catch (Exception es) {  Logger.getLogger(GUI\_DataMatkul.class.getName()).log(Level.SEVERE, null, es);  }  } |

1. Membuat Method “**tampil()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

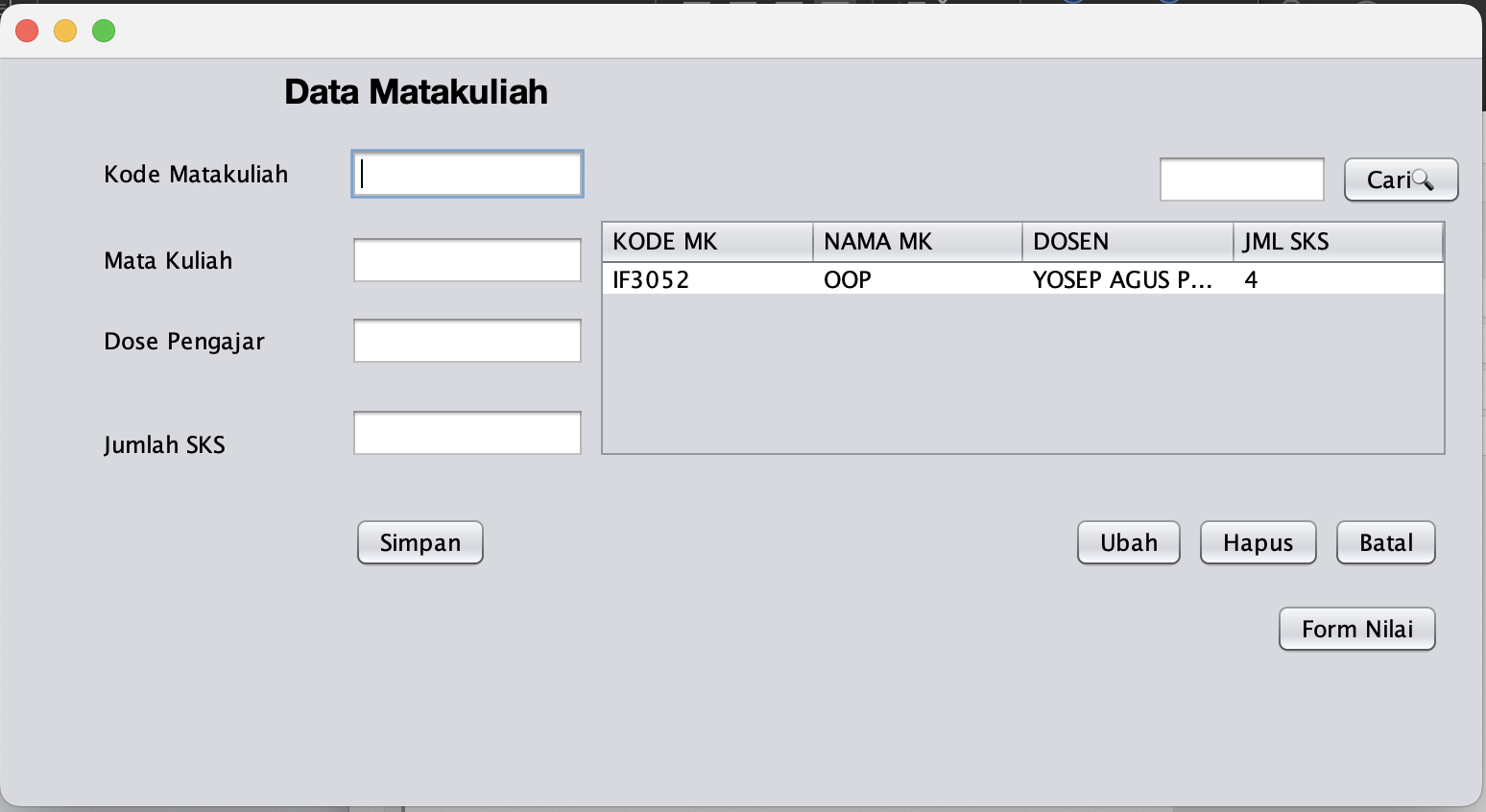
*Source Code:*

|  |
| --- |
| public void tampil() {  DefaultTableModel tabelhead = new DefaultTableModel();  tabelhead.addColumn("KODE MK");  tabelhead.addColumn("NAMA MK");  tabelhead.addColumn("DOSEN");  tabelhead.addColumn("JML SKS");  try {  koneksi();  String sql = "SELECT \* FROM tb\_matkul";  Statement stat = conn.createStatement();  ResultSet res = stat.executeQuery(sql);  while (res.next()) {  tabelhead.addRow(new Object[]{res.getString(2), res.getString(3), res.getString(4), res.getString(5),});  }  tabel\_data.setModel(tabelhead);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "BELUM TERKONEKSI");  }  } |

1. Panggil method “**tampil()**” pada konstruktor GUI\_ Matkul agar ketika program pertama kali di jalankan langsung menjalankan method tersebut.

|  |
| --- |
| public GUI\_DataMatkul() {  initComponents();  tampil();  } |

1. Jalankan GUI\_Matkul jika Table Langsung menampilkan isi database saat pertama kali program berjalan maka berhasil.

****

1. Membuat method “**refresh()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void refresh() {  new GUI\_DataMatkul().setVisible(true);  this.setVisible(false);  } |

Method ini membuat Method CRUD akan lebih responsive di karenakan saat selesai melakukan perintah CRUD akan langsung merefresh program.

1. Membuat method “**insert()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void insert() {  String Kode = txtKdMatakuliah.getText();  String MK = txtMatakuliah.getText();  String Dosen = txtDosenPengajar.getText();  String jmlsks = txtJmlSks.getText();  try {  koneksi();  Statement statement = conn.createStatement();  statement.executeUpdate("INSERT INTO tb\_matkul(kode\_mk, matakuliah, dosenpengajar,jmlsks)"  + "VALUES('" + Kode + "','" + MK + "','" + Dosen + "','" + jmlsks + "')");  statement.close();  JOptionPane.showMessageDialog(null, "Berhasil Memasukan Data Matakuliah!");  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Terjadi Kesalahan Input!");  }  refresh();  } |

1. Panggil Method “**Insert()**” pada button “**btnSimpan**”.

*Source Code:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  insert();  } |

1. Membuat method “**update()**” Letakkan script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void update() {  String Kode = txtKdMatakuliah.getText();  String MK = txtMatakuliah.getText();  String Dosen = txtDosenPengajar.getText();  String jmlsks = txtJmlSks.getText();  String KdMkLama = kode1;  try {  Statement statement = conn.createStatement();  statement.executeUpdate("UPDATE tb\_matkul SET kode\_mk='" + Kode + "'," + "matakuliah='" + MK + "',"  + "dosenpengajar='" + Dosen + "'" + ",jmlsks='" + jmlsks + "'WHERE kode\_mk = '" + KdMkLama + "'");  statement.close();  conn.close();  JOptionPane.showMessageDialog(null, "Update Data MataKuliah!");  } catch (Exception e) {  System.out.println("Error : " + e);  }  refresh();  } |

1. Panggil Method “**update()**” pada Button “**btnUbah**”

*Source Code:*

|  |
| --- |
| private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  update();  } |

1. Membuat method “**delete()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void delete() {  int ok = JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ?", "Konfirmasi", JOptionPane.YES\_NO\_OPTION);  if (ok == 0) {  try {  String sql = "DELETE FROM tb\_matkul WHERE kode\_mk='" + txtKdMatakuliah.getText() + "'";  java.sql.PreparedStatement stmt = conn.prepareStatement(sql);  stmt.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di hapus");  batal();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Data gagal di hapus");  }  }  refresh();  } |

1. Panggil Method “**delete()**” pada button “**btnHapus**”.

*Source Code:*

|  |
| --- |
| private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  delete();  } |

1. Membuat method “**cari()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void cari() {  try {  try ( Statement statement = conn.createStatement()) {  String sql = "SELECT \* FROM tb\_matkul WHERE `kode\_mk` LIKE '%" + txtKdMatakuliah.getText() + "%'";  ResultSet rs = statement.executeQuery(sql); //menampilkan data dari sql query  if (rs.next()) // .next() = scanner method  {  txtKdMatakuliah.setText(rs.getString(2));  txtMatakuliah.setText(rs.getString(3));  txtDosenPengajar.setText(rs.getString(4));  txtJmlSks.setText(rs.getString(5));  } else {  JOptionPane.showMessageDialog(null, "Data yang Anda cari tidak ada");  }  }  } catch (Exception ex) {  System.out.println("Error." + ex);  }  } |

1. Panggil Method “**cari()**” pada button “**btnCari**”.

*Source Code:*

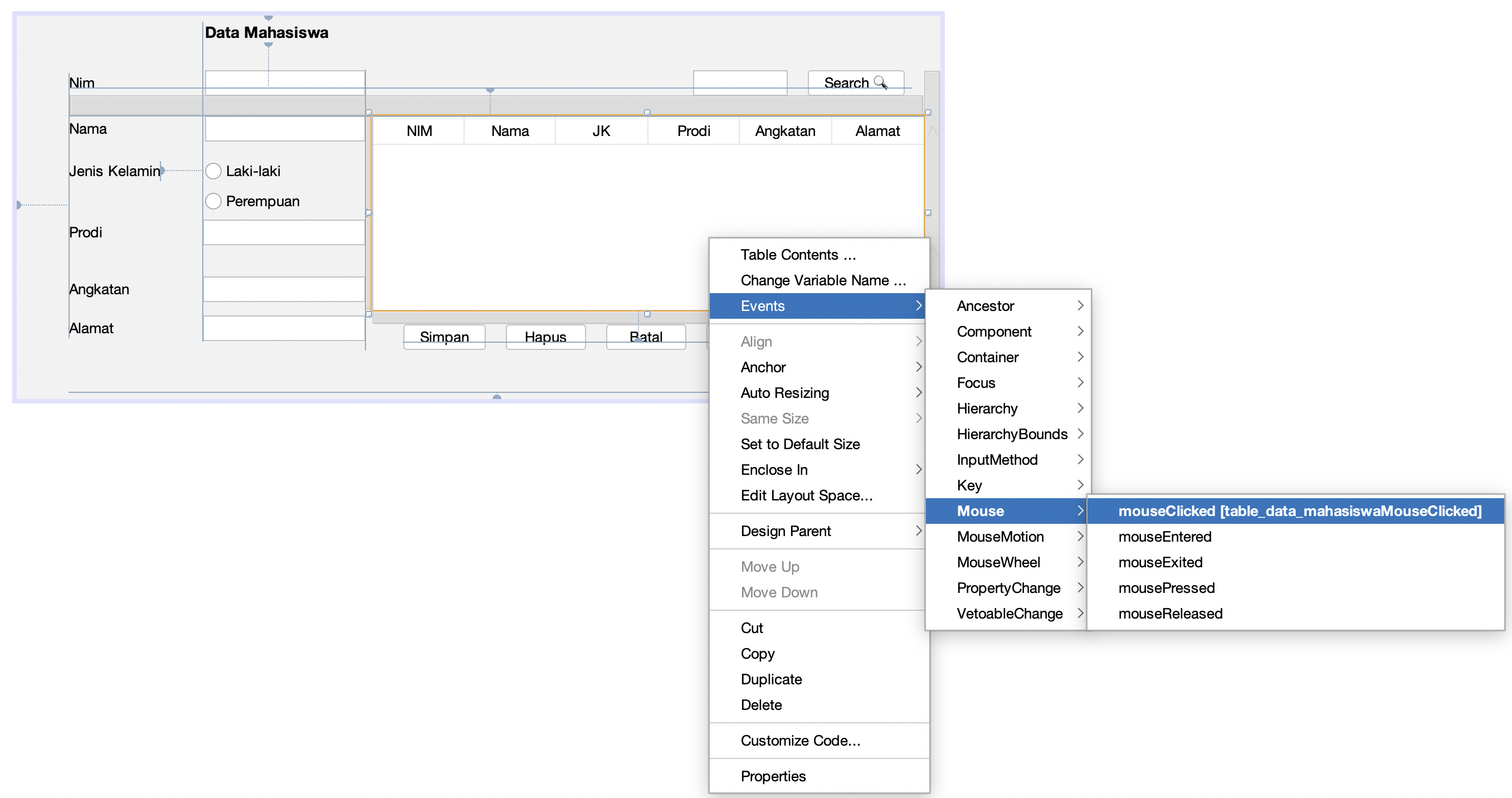
|  |
| --- |
| private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  cari();  } |

1. Buatlah Method “**itempilih()**”.

*Source code:*

|  |
| --- |
| public void itempilih() {  txtKdMatakuliah.setText(kode1);  txtMatakuliah.setText(mk1);  txtDosenPengajar.setText(dosen1);  txtJmlSks.setText(jmlsks1);  } |

1. Menambahkan events mouseClicked pada jTabel yang tersedia di GUI\_Matkul dengan cara pilih design GUI\_Matkul, kemudian klik kanan pada jTabel, pilih Mouse, kemudian pilih mouseClicked.

****

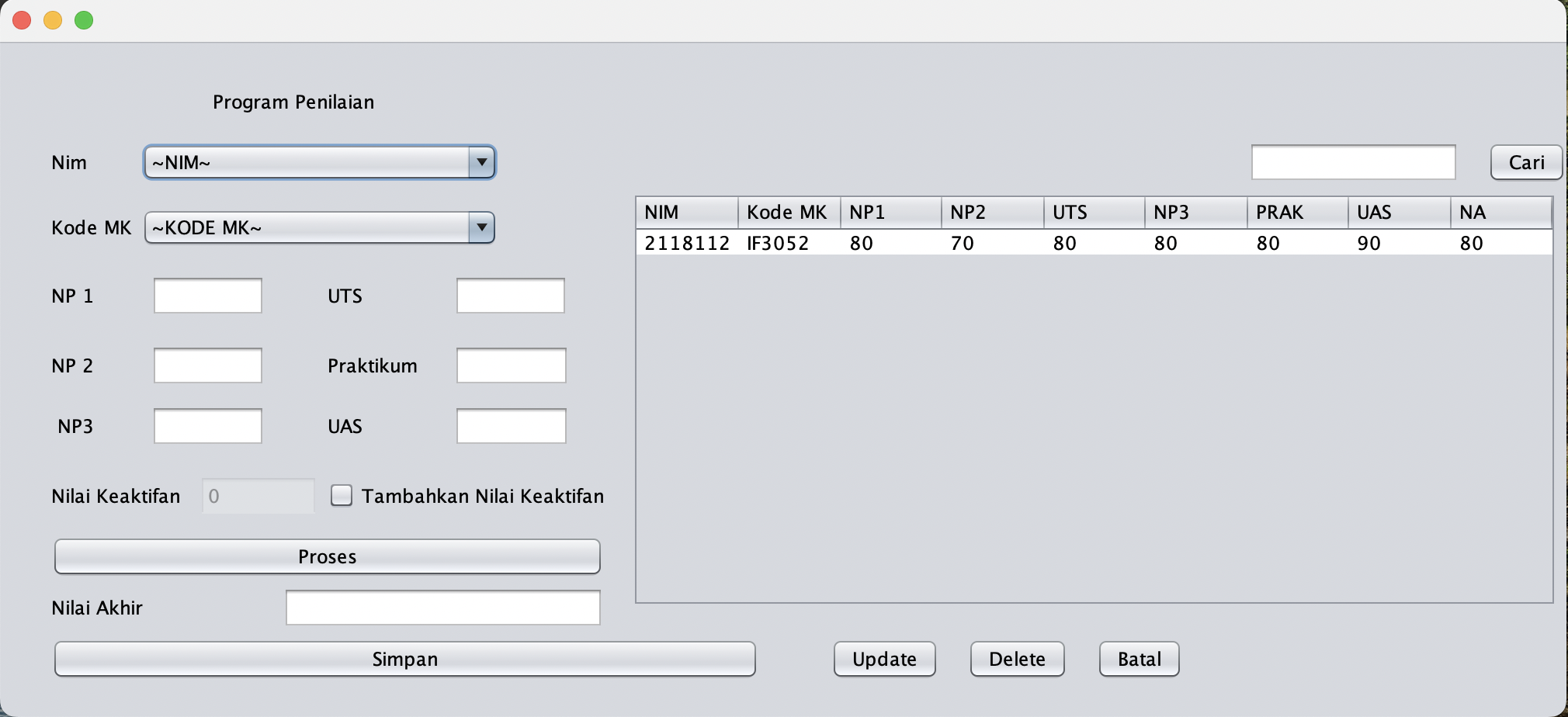
1. Tambahkan script dibawah ini.

|  |
| --- |
| int tabel = tabel\_data.getSelectedRow();  kode1 = tabel\_data.getValueAt(tabel, 0).toString();  mk1 = tabel\_data.getValueAt(tabel, 1).toString();  dosen1 = tabel\_data.getValueAt(tabel, 2).toString();  jmlsks1 = tabel\_data.getValueAt(tabel, 3).toString();  itempilih(); |

1. Tambahkan script pada button Form nilai di GUI\_Matkul

|  |
| --- |
| private void btnNilaiActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  new GUI\_Penilaian().setVisible(true);  } |

1. **Project Praktikum Mengkoneksikan GUI\_Nilai dengan Database.**
2. Buka GUI\_Nilai pada package “Penilaian” yang telah di buat.

****

1. Kemudian pada GUI\_ Nilai, pilih tab Source, masukan Script di bawah ini di bagian paling atas dalam Source GUI\_ Nilai.

*Source Code:*

|  |
| --- |
| import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.PreparedStatement;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

1. Tambahkan script berikut dibawah class GUI\_Nilai

|  |
| --- |
| String nim1, kd\_mk1, np1, np2, np3, UTS, UAS, PRAK, keaktifan, NA; |

1. Buat Variable Koneksi, Letakan script di bawah ini di bawah Konstruktor GUI\_Nilai.

|  |
| --- |
| public Connection conn; |

1. Tambahkan method “**batal()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

|  |
| --- |
| public void batal() {  txtNP1.setText("");  txtNP2.setText("");  txtNP3.setText("");  txtPraktikum.setText("");  txtUas.setText("");  txtUts.setText("");  txtKeaktifan.setText("");  } |

1. Membuat Method “**koneksi()**” letakan script di bawah ini di bawah pendeklarasian Variable “Conn”.

*Souce Code:*

|  |
| --- |
| public void koneksi() throws SQLException {  try {  conn = null;  Class.forName("com.mysql.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost/OOP\_nim?user=root&password=");  } catch (ClassNotFoundException ex) {  Logger.getLogger(Gui\_Penilaian.class.getName()).log(Level.SEVERE, null, ex);  } catch (SQLException e) {  Logger.getLogger(Gui\_Penilaian.class.getName()).log(Level.SEVERE, null, e);  } catch (Exception es) {  Logger.getLogger(Gui\_Penilaian.class.getName()).log(Level.SEVERE, null, es);  }  } |

1. Membuat Method “**tampil()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

*Source Code:*

|  |
| --- |
| public void tampil() {  DefaultTableModel tabelhead = new DefaultTableModel();  tabelhead.addColumn("NIM");  tabelhead.addColumn("Kode MK");  tabelhead.addColumn("NP1");  tabelhead.addColumn("NP2");  tabelhead.addColumn("UTS");  tabelhead.addColumn("NP3");  tabelhead.addColumn("PRAK");  tabelhead.addColumn("UAS");  tabelhead.addColumn("NA");  try {  koneksi();  String sql = "SELECT \* FROM tb\_nilai";  Statement stat = conn.createStatement();  ResultSet res = stat.executeQuery(sql);  while (res.next()) {  tabelhead.addRow(new Object[]{res.getString(2), res.getString(3), res.getString(4), res.getString(5), res.getString(6), res.getString(7), res.getString(8), res.getString(9), res.getString(10),});  }  tabel\_data.setModel(tabelhead);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "BELUM TERKONEKSI");  }  } |

1. Membuat Method “**tampil\_mhs()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

|  |
| --- |
| public void tampil\_mhs() {  try {  koneksi();  String sql = "SELECT nim FROM tb\_mahasiswa order by nim asc";  Statement stt = conn.createStatement();  ResultSet res = stt.executeQuery(sql);  while (res.next()) {  Object[] ob = new Object[3];  ob[0] = res.getString(1);  cmbNim.addItem((String) ob[0]);  }  res.close();  stt.close();  } catch (Exception e) {  System.out.println(e.getMessage());  }  } |

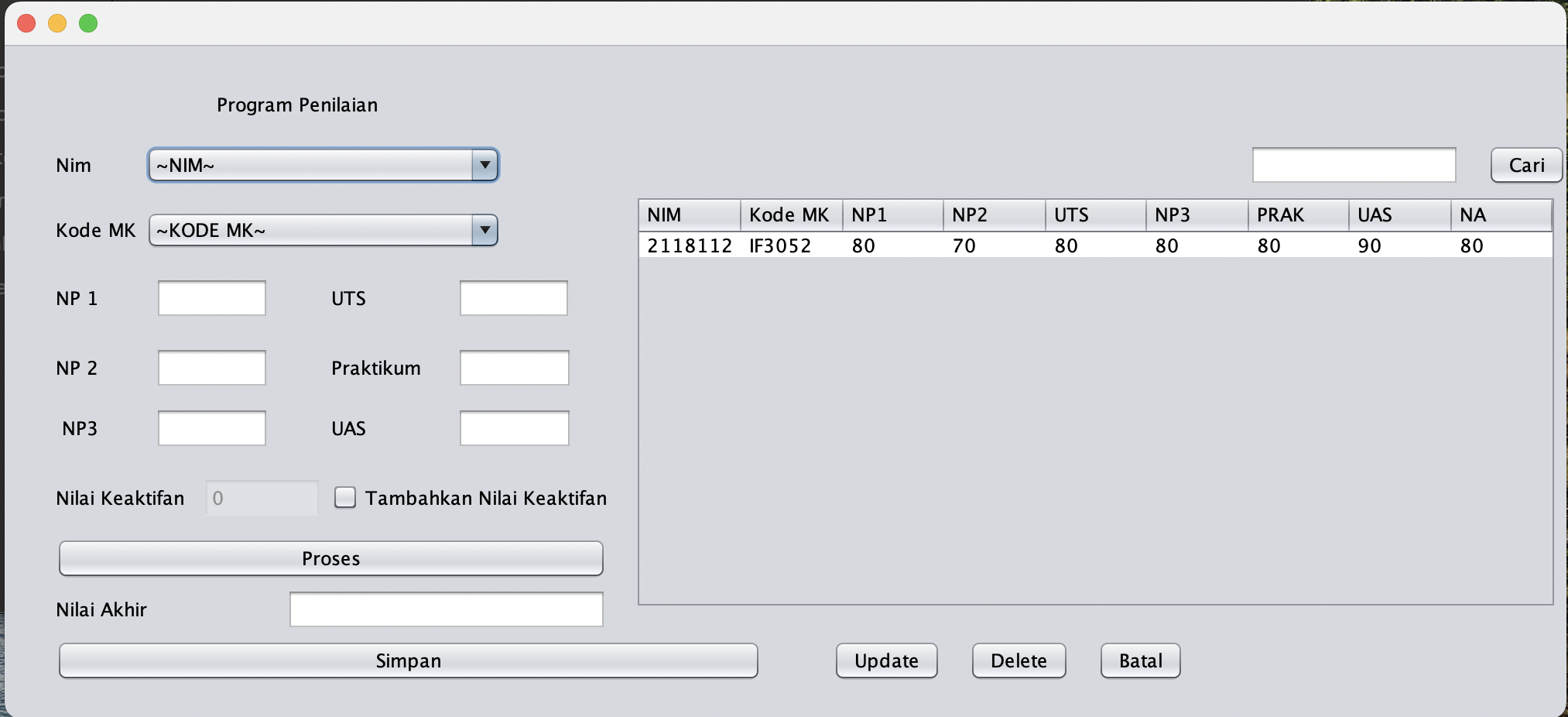
1. Membuat Method “**tampil\_mk()**” letakan script di bawah ini di bawah Method sebelumya dan seterusnya pada method lain.

|  |
| --- |
| public void tampil\_mk() {  try {  koneksi();  String sql = "SELECT kode\_mk FROM tb\_matkul order by kode\_mk asc";  Statement stt = conn.createStatement();  ResultSet res = stt.executeQuery(sql);  while (res.next()) {  Object[] ob = new Object[3];  ob[0] = res.getString(1);  cmbKodeMk.addItem((String) ob[0]);  }  res.close();  stt.close();  } catch (Exception e) {  System.out.println(e.getMessage());  }  } |

1. Panggil method “**tampil(), tampil\_mhs(), tampil\_mk()**” pada konstruktor GUI\_Nilai agar ketika program pertama kali di jalankan langsung menjalankan method tersebut.

|  |
| --- |
| txtKeaktifan.setEnabled(false);  tampil();  tampil\_mhs();  tampil\_mk();  txtKeaktifan.setText(Integer.toString(0)); |

1. Jalankan GUI\_Nilai jika Table Langsung menampilkan isi database saat pertama kali program berjalan maka berhasil.

****

1. Membuat method “**NilaiAkhir()**” Letakkan script di bawah ini dibawah method sebelumnya.

Source code :

|  |
| --- |
| public double NilaiAkhir() {  double NP1, NP2, NP3, NilaiPrak, UTS, UAS, nilaiProses, hasil, nilaiKeaktifan;  NP1 = Integer.parseInt(txtNP1.getText());  NP2 = Integer.parseInt(txtNP1.getText());  NP3 = Integer.parseInt(txtNP1.getText());  NilaiPrak = Integer.parseInt(txtNP1.getText());  UTS = Integer.parseInt(txtNP1.getText());  UAS = Integer.parseInt(txtNP1.getText());  nilaiKeaktifan = Integer.parseInt(txtKeaktifan.getText());  nilaiProses = (NP1 \* 0.1) + (NP2 \* 0.2) + (NP3 \* 0.1) + (UTS \* 0.2) + (NilaiPrak \* 0.4);  hasil = (nilaiKeaktifan \* 0.1) + nilaiProses;  txtNA.setText(Double.toString(hasil));  return hasil;  } |

1. Panggil Method “**NilaiAkhir()**” pada Button “**Proses**”

*Source Code:*

|  |
| --- |
| private void btnProsesActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  NilaiAkhir();  } |

1. Membuat method “**refresh()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void refresh() {  new GUI\_Penilaian().setVisible(true);  this.setVisible(false);  } |

Method ini membuat Method CRUD akan lebih responsive di karenakan saat selesai melakukan perintah CRUD akan langsung merefresh program.

1. Membuat method “**insert()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void insert() {  String Nim = (String) cmbNim.getSelectedItem();  String KodeMK = (String) cmbKodeMk.getSelectedItem();  String NP1 = txtNP1.getText();  String NP2 = txtNP2.getText();  String UTS = txtUts.getText();  String NP3 = txtNP3.getText();  String PRAK = txtPraktikum.getText();  String UAS = txtUas.getText();  String NA = txtNA.getText();  try {  koneksi();  Statement statement = conn.createStatement();  statement.executeUpdate("INSERT INTO tb\_nilai(Nim, kd\_mk, NP1, NP2, UTS,NP3,prak,UAS,NA)"  + "VALUES('" + Nim + "','" + KodeMK + "','" + NP1 + "','" + NP2 + "','" + UTS + "','" + NP3 + "',"  + "'" + PRAK + "','" + UAS + "','" + NA + "')");  statement.close();  JOptionPane.showMessageDialog(null, "Berhasil Memasukan Data Nilai!");  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Terjadi Kesalahan Input!");  }  refresh();  } |

1. Panggil Method “**Insert()**” pada button “**btnSimpan**”.

*Source Code:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  insert();  } |

1. Membuat method “**update()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void update() {  String Nim = (String) cmbNim.getSelectedItem();  String KodeMK = (String) cmbKodeMk.getSelectedItem();  String NP1 = txtNP1.getText();  String NP2 = txtNP2.getText();  String UTS = txtUts.getText();  String NP3 = txtNP3.getText();  String PRAK = txtPraktikum.getText();  String UAS = txtUas.getText();  String NA = txtNA.getText();  String nim\_lama = nim1;  String kode\_lama = kd\_mk1;  try {  Statement statement = conn.createStatement();  statement.executeUpdate("UPDATE tb\_nilai SET Nim='" + Nim + "'," + "kd\_mk='" + KodeMK + "'"  + ",NP1='" + NP1 + "',NP2='" + NP2 + "',UTS='" + UTS + "',NP3='" + NP3 + "',prak='" + PRAK + "',UAS='" + UAS + "',NA='" + NA + "' WHERE Nim ='" + nim\_lama + "' AND kd\_mk='" + kode\_lama + "'");  statement.close();  conn.close();  JOptionPane.showMessageDialog(null, "Update Data Nilai!");  } catch (Exception e) {  System.out.println("Error : " + e);  }  refresh();  } |

1. Panggil Method “**update()**” pada Button “**btnUbah**”

*Source Code:*

|  |
| --- |
| private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  update();  } |

1. Membuat method “**delete()**” Letakana script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void delete() {  int ok = JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ?", "Konfirmasi", JOptionPane.YES\_NO\_OPTION);  if (ok == 0) {  try {  String sql = "DELETE FROM tb\_nilai WHERE Nim='" + cmbNim.getSelectedItem() + "' AND kd\_mk='" + cmbKodeMk.getSelectedItem() + "'";  PreparedStatement stmt = conn.prepareStatement(sql);  stmt.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di hapus");  batal();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Data gagal di hapus");  }  }  refresh();  } |

1. Panggil Method “**delete()**” pada button “**btnHapus**”.

*Source Code:*

|  |
| --- |
| private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  delete();  } |

1. Membuat method “**cari()**” Letakkan script di bawah ini di bawah method sebelumnya.

*Source Code:*

|  |
| --- |
| public void cari() {  try {  try ( Statement statement = conn.createStatement()) {  String sql = "SELECT \* FROM tb\_nilai WHERE `Nim` LIKE '%" + txtCari.getText() + "%'";  ResultSet rs = statement.executeQuery(sql);  //menampilkan data dari sql query  if (rs.next()) // .next() = scanner method  {  cmbNim.setSelectedItem(rs.getString(2));  cmbKodeMk.setSelectedItem(rs.getString(3));  txtNP1.setText(rs.getString(4));  txtNP2.setText(rs.getString(5));  txtUts.setText(rs.getString(6));  txtNP3.setText(rs.getString(7));  txtPraktikum.setText(rs.getString(8));  txtUas.setText(rs.getString(9));  txtNA.setText(rs.getString(10));  } else {  JOptionPane.showMessageDialog(null, "Data yang Anda cari tidak ada");  }  }  } catch (Exception ex) {  System.out.println("Error." + ex);  }  } |

1. Panggil Method “**cari()**” pada button “**btnCari**”.

*Source Code:*

|  |
| --- |
| private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  cari();  } |

1. Membuat method “**itempilih()**” Letakkan script di bawah ini di bawah method sebelumnya.

|  |
| --- |
| void itempilih() {  cmbNim.setSelectedItem(nim1);  cmbKodeMk.setSelectedItem(kd\_mk1);  txtNP1.setText(np1);  txtNP2.setText(np2);  txtNP3.setText(np3);  txtPraktikum.setText(PRAK);  txtUas.setText(UAS);  txtUts.setText(UTS);  txtKeaktifan.setText(keaktifan);  txtKeaktifan.setText(Integer.toString(0));  txtNA.setText(NA);  } |

1. Tambahkan script pada cheacklist.

|  |
| --- |
| if (cmKeaktifan.isSelected()) {  txtKeaktifan.setEnabled(true);  } else {  txtKeaktifan.setEnabled(false);  } |

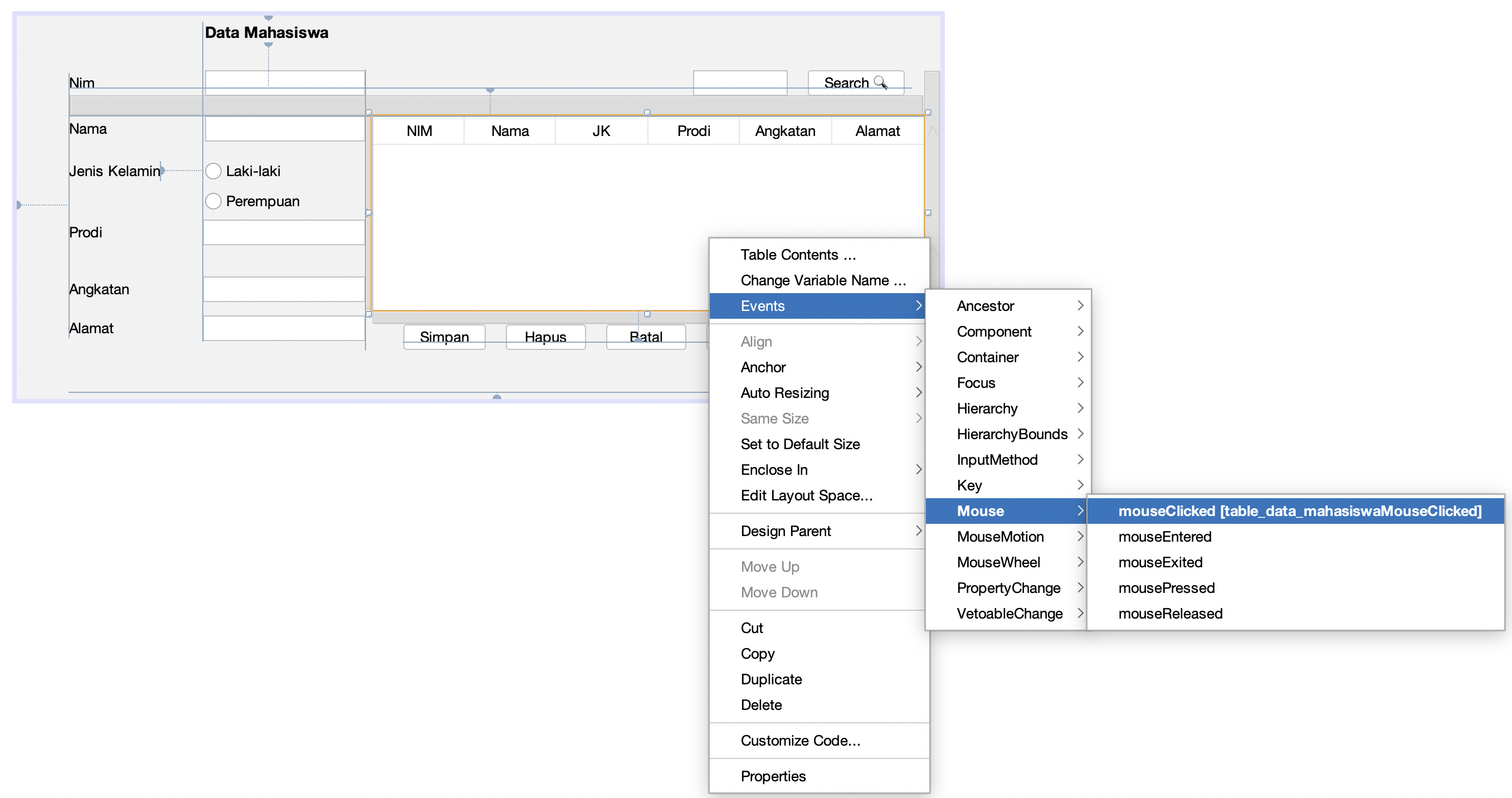
1. Tambahkan script pada button NIM.

|  |
| --- |
| new GUI\_Mahasiswa().setVisible(true); |

1. Tambahkan script pada button MK.

|  |
| --- |
| new GUI\_DataMatkul().setVisible(true); |

1. Menambahkan events mouseClicked pada jTabel yang tersedia di GUI\_Nilai dengan cara pilih design GUI\_Nilai, kemudian klik kanan pada jTabel, pilih Mouse, kemudian pilih mouseClicked.

****

1. Tambahkan script dibawah ini.

|  |
| --- |
| int tabel = table\_data.getSelectedRow(); //0  nim1 = table\_data.getValueAt(tabel, 0).toString();  kd\_mk1 = table\_data.getValueAt(tabel, 1).toString();  np1 = table\_data.getValueAt(tabel, 2).toString();  np2 = table\_data.getValueAt(tabel, 3).toString();  UTS = table\_data.getValueAt(tabel, 4).toString();  np3 = table\_data.getValueAt(tabel, 5).toString();  PRAK = table\_data.getValueAt(tabel, 6).toString();  UAS = table\_data.getValueAt(tabel, 7).toString();  NA = table\_data.getValueAt(tabel, 8).toString();  itempilih(); |

**Tugas Rumah:**

1. Menerapkan CRUD pada 3 GUI yang sudah di buat.
2. Setelah penerapan implementasi database ke 3 GUI selesai, silahkan upload project ke github(Bagi yang kemarin projectmya belum di push), bagi yang kemarin projectnya sudah di push tinggal di update ulang di github. Nantinya link project Github akan dikumpulkan dan di nilai dan kami akan masukan ke dalam Nilai Project. Jadi jangan sampai tidak mengumpulkan.

<https://drive.google.com/file/d/1J6WaNsFkd_0lbGmWxKvXDc2tENWaO1Nz/view?usp=sharing>