

MANUAL BOOK
PROJECT LAB REKAYASA PERANGKAT LUNAK 2

Kelas : 4IA24
NPM : 51421020
Nama : Muhammad Naziev Ash'Saviq
Ketua Asisten : Robby N

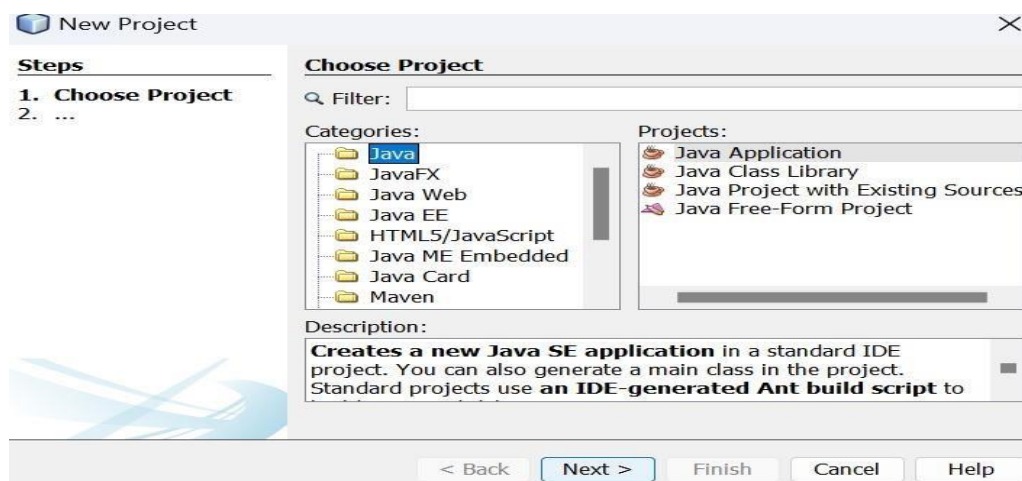


LABORATORIUM TEKNIK INFORMATIKA
UNIVERSITAS GUNADARMA
2024

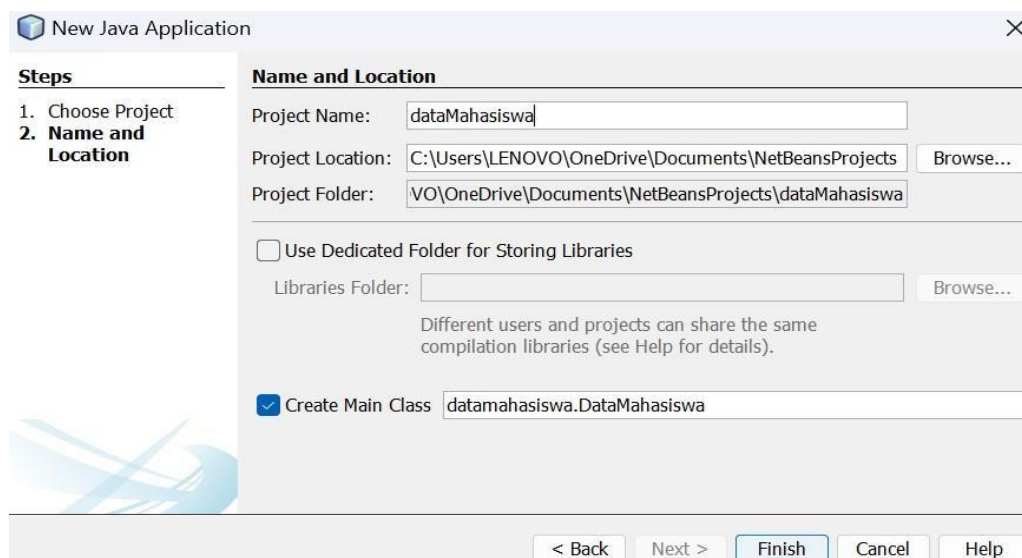
MANUAL BOOK REKAYASA PERANGKAT LUNAK 2



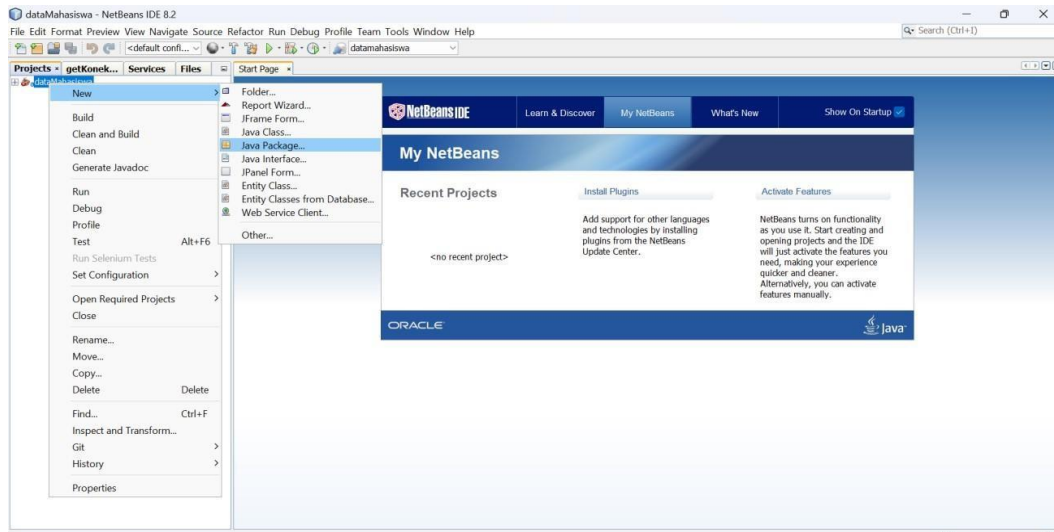
Pada project kali ini, kita akan membuat sebuah program pada netbeans, langkah pertama yaitu Membuka NetBEans IDE 8.2.



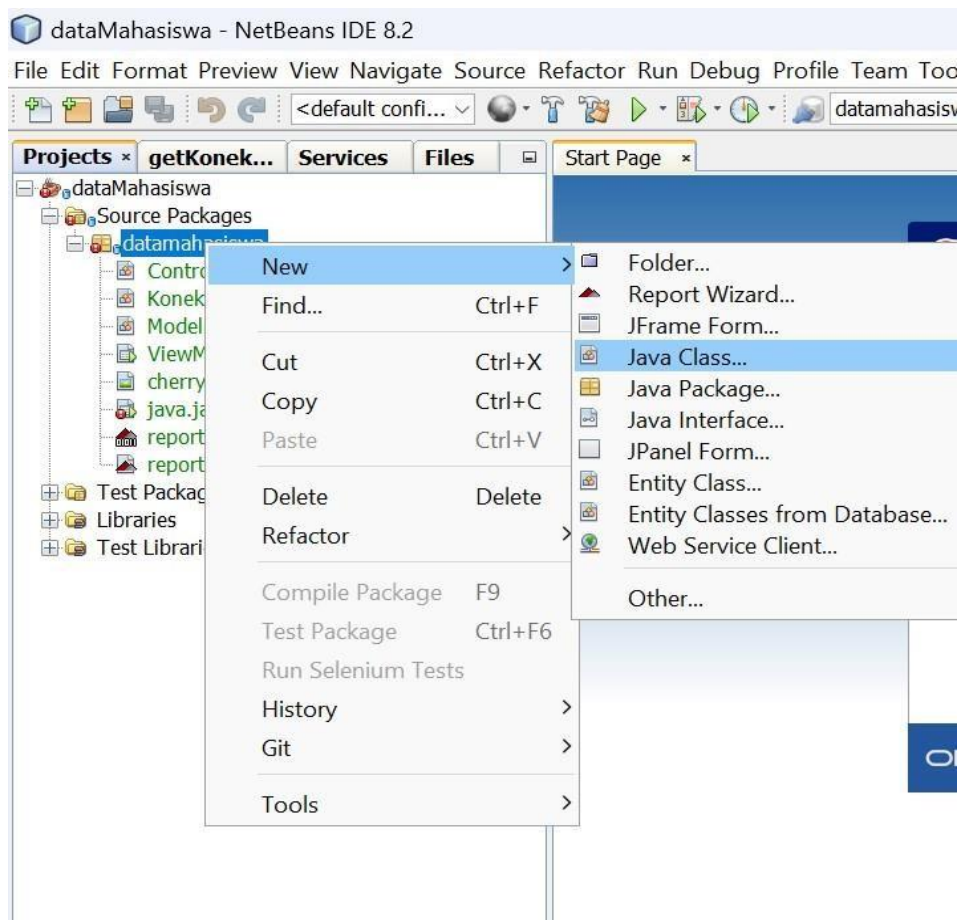
Setelahnya pada menu “Categories” Pilih java > Lalu pada menu “Projects” pilih Java Application lalu klik tombol next.



Masukkan Project Name, disini saya menggunakan nama “dataMahasiswa” lalu klik finish.



Klik kanan pada **project dataMahasiswa** > **new** > **java Package**.



Klik kanan pada **package dataMahasiswa** > **new** > **java class**, lalu beri nama **ControllerMahasiswa.java**.

```

1 package datamahasiswa;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.SQLException;
6 import java.util.ArrayList;
7 import javax.swing.JOptionPane;
8 import javax.swing.table.DefaultTableModel;
9
10 public class ControllerMahasiswa {
11     ArrayList<ModelMahasiswa> dataArrayList;
12     DefaultTableModel labelList;
13     private Connection conn;
14
15     public ControllerMahasiswa() {
16         conn = Koneksi.getKoneksi();
17         dataArrayList = new ArrayList<>();
18         labelList = new DefaultTableModel(new String[]{"NPM", "Nama", "Tinggi", "Pindahan"}, 0);
19     }
20
21     public void insertData(String npm, String nama, int tinggi, boolean pindahan) {
22         ModelMahasiswa mhs = new ModelMahasiswa(npm, nama, tinggi, pindahan);
23         dataArrayList.add(mhs);
24         insertDataToDatabase(mhs);
25     }
26
27     public void insertDataToDatabase(ModelMahasiswa mhs) {
28         String sql = "INSERT INTO mahasiswa (npm, nama, tinggi, pindahan) VALUES (?, ?, ?, ?)";
29         try (PreparedStatement pst = conn.prepareStatement(sql)) {
30             pst.setString(1, mhs.getNPM()); // Ubah ke setInt
31             pst.setString(2, mhs.getNama());
32             pst.setInt(3, mhs.getTinggi());
33             pst.setBoolean(4, mhs.isPindahan());
34             pst.executeUpdate();
35             JOptionPane.showMessageDialog(null, "Data berhasil disimpan ke database!");
36         } catch (SQLException e) {

```

```

73
74     }
75
76
77     private void deleteDataFromDatabase(String npm) {
78         String sql = "DELETE FROM mahasiswa WHERE npm = ?";
79         try (PreparedStatement pst = conn.prepareStatement(sql)) {
80             pst.setString(1, npm);
81             pst.executeUpdate();
82             JOptionPane.showMessageDialog(null, "Data berhasil dihapus!");
83         } catch (SQLException e) {
84             JOptionPane.showMessageDialog(null, "Gagal menghapus data: " + e.getMessage());
85         }
86     }
87
88     }
89
90
91     public DefaultTableModel showData() {
92         labelList.setRowCount(0);
93         for (ModelMahasiswa mhs : dataArrayList) {
94             labelList.addRow(new Object[]{
95                 mhs.getNPM(),
96                 mhs.getNama(),
97                 mhs.getTinggi(),
98                 mhs.isPindahan() ? "Ya" : "Tidak"
99             });
100         }
101         return labelList;
102     }
103 }
104

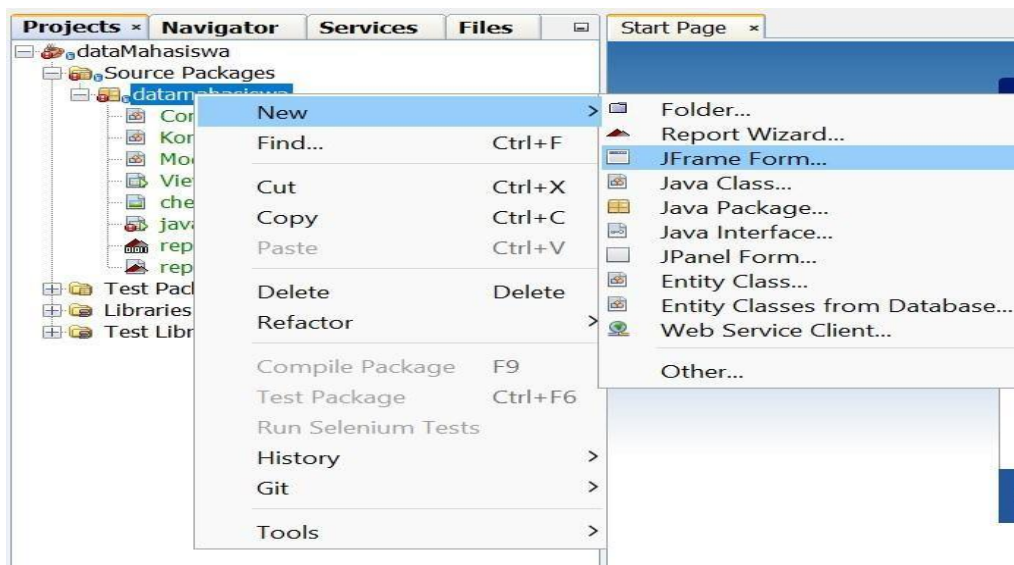
```

```

73
74
75
76
77 private void deleteDataFromDatabase(String npm) {
78     String sql = "DELETE FROM mahasiswa WHERE npm = ?";
79     try (PreparedStatement pst = conn.prepareStatement(sql)) {
80         pst.setString(1, npm);
81         pst.executeUpdate();
82         JOptionPane.showMessageDialog(null, "Data berhasil dihapus!");
83     } catch (SQLException e) {
84         JOptionPane.showMessageDialog(null, "Gagal menghapus data: " + e.getMessage());
85     }
86 }
87
88
89
90
91 public DefaultTableModel showData() {
92     labelList.setRowCount(0);
93     for (ModelMahasiswa mhs : dataArrayList) {
94         labelList.addRow(new Object[]{
95             mhs.getNPM(),
96             mhs.getNama(),
97             mhs.getTinggi(),
98             mhs.isPindahan() ? "Ya" : "Tidak"
99         });
100     }
101     return labelList;
102 }
103
104

```

Pada program diatas merupakan isi dari “ControllerMahasiswa.java”. Konstruktor ControllerMahasiswa bertujuan menginisialisasi ArrayList yang akan menampung semua objek ModelMahasiswa.



Selanjutnya kita akan membuat file “dataMahasiswa”. Klik kanan pada **package dataMahasiswa** > **new > JFrame Form** dengan nama file ViewMahasiswa.java.

Buat desain untuk Tampilam JForm ViewMahasiswa.java seperti pada gambar dibawah.

Data Mahasiswa

NPM

Nama

Tinggi

Pindahan?

Ya

Simpan

Edit

Delete

NPM	NAMA	TINGGI	PINDAH

Langkah pembuatan:



JLabel digunakan untuk membuat Data Mahasiswa, NPM,Nama,Tinggi,Pindahan?

Data Mahasiswa

NPM

Nama

Tinggi

Pindahan?

Ya

Simpan

Edit

Delete

NPM	NAMA	TINGGI	PINDAH



JLabel digunakan untuk membuat tempat pengisian user.

Data Mahasiswa

NPM

Nama

Tinggi

Pindahan?

Ya ▼

Simpan

Edit

Delete

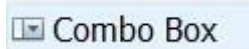
NPM	NAMA	TINGGI	PINDAH

Lalu melakukan perubahan sebagai berikut:

JTextField1 digunakan untuk melakukan input yang dilakukan oleh user, dengan klik kanan lalu **change variabel name** mengganti **JTextField1** menjadi **NPM**.

JTextField2 digunakan untuk melakukan input yang dilakukan oleh user, dengan klik kanan lalu **change variabel name** mengganti **JTextField2** menjadi **Nama**.

JTextField3 digunakan untuk melakukan input yang dilakukan oleh user, dengan klik kanan lalu **change variabel name** mengganti **JTextField3** menjadi **Tinggi**.



Menambahkan Combo Box di GUI Designer

Data Mahasiswa

NPM

Nama

Tinggi

Pindahan? Ya ▼

NPM	NAMA	TINGGI	PINDAH

Swing Controls

Label Button

JButton digunakan untuk membuat tombol.

Data Mahasiswa

NPM

Nama

Tinggi

Pindahan? Ya ▼

NPM	NAMA	TINGGI	PINDAH

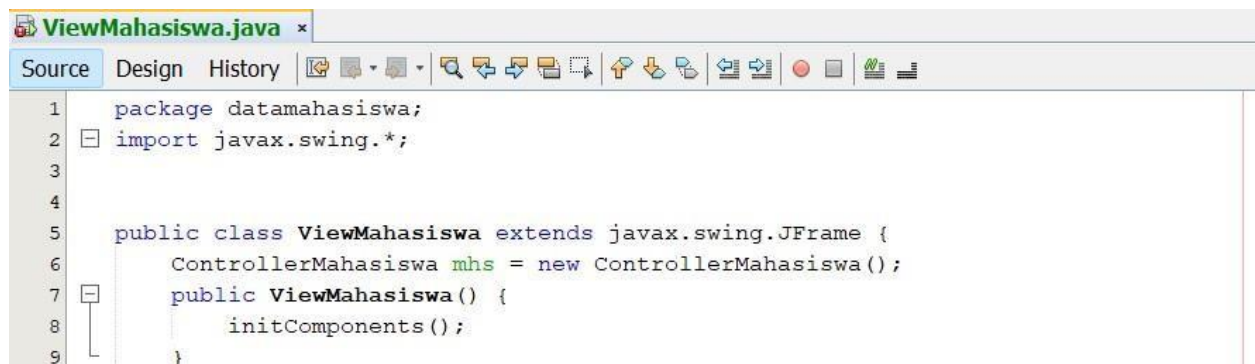
dengan klik Bawah ini lalu edit text dengan kata SUBMIT, lalu klik kanan lagi pilih event > action
> actionPerformed[JButton1actionPerformed]

NPM	NAMA	TINGGI	PINDAH

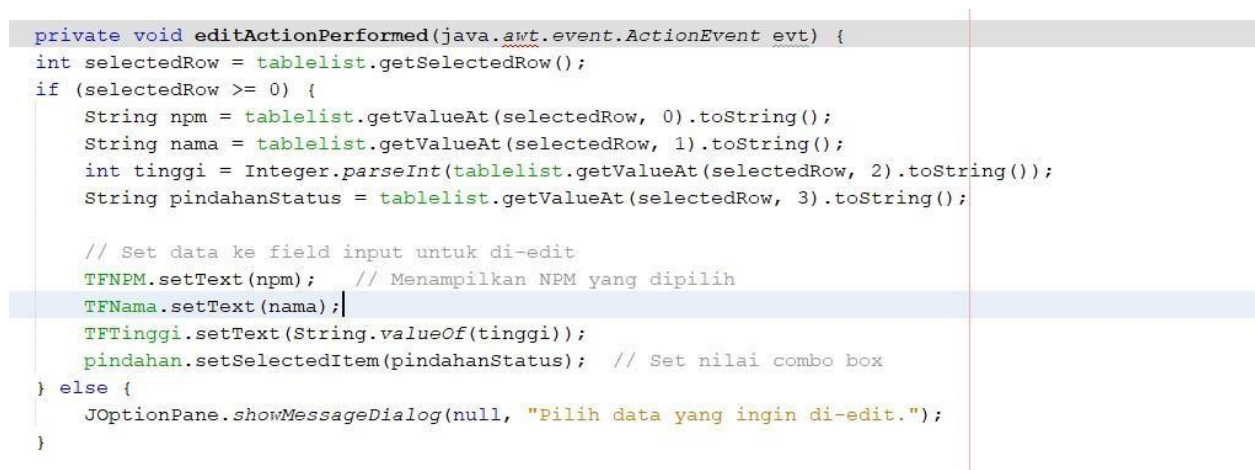
Menu Palette menggunakan item dari swing controls bernama JTable, dengan klik kanan lalu change variabel name mengganti JTable1 menjadi tbdata, klik kanan lagi lalu table content untuk pengaturan kolom pada tabel .



Lalu pilih menu source untuk menuliskan sintak program agar program berjalan.



Source Jform viewkaryawan.java berisi kodingan yang akan mengatur JTextField1, JTextField2, JTextField3, JTextField4, Combo Box, JButtonActionPerformed Mahasiswa" dan menggunakan kerangka dasar NetBeans untuk membuat tampilan berbasis Swing.



Validasi Data: Pastikan data di tabel memiliki tipe yang sesuai agar tidak terjadi kesalahan konversi (misalnya, kolom tinggi harus benar-benar angka).

Combo Box: Pastikan data di combo box pindahan telah diisi dengan nilai-nilai yang valid (misalnya: "Ya", "Tidak") sehingga nilai dapat dipilih secara otomatis.

Tabel: Model tabel (DefaultTableModel) harus sudah diatur untuk menyimpan data dalam format yang sesuai.

Validasi input: Menambahkan validasi untuk memastikan bahwa nilai-nilai yang diberikan valid (misalnya, NPM tidak kosong, tinggi tidak negatif). Handling error: Menambahkan penanganan error untuk kasus di mana mahasiswa tidak ditemukan atau terjadi kesalahan saat memperbarui database. Efisiensi: Jika dataArrayList sangat besar, pertimbangkan untuk menggunakan struktur data yang lebih efisien seperti HashMap untuk mencari mahasiswa berdasarkan NPM.

```
54 private void updateDataToDatabase(ModelMahasiswa mhs) {
55     String sql = "UPDATE mahasiswa SET nama = ?, tinggi = ?, pindahan = ? WHERE npm = ?";
56     try (PreparedStatement pst = conn.prepareStatement(sql)) {
57         pst.setString(1, mhs.getNama());
58         pst.setInt(2, mhs.getTinggi());
59         pst.setBoolean(3, mhs.isPindahan());
60         pst.setString(4, mhs.getNPM());
61         pst.executeUpdate();
62         JOptionPane.showMessageDialog(null, "Data berhasil diperbarui!");
63     } catch (SQLException e) {
64         JOptionPane.showMessageDialog(null, "Gagal memperbarui data: " + e.getMessage());
65     }
66 }
```

Logging: Menambahkan log untuk mencatat setiap operasi update, termasuk waktu, data yang diperbarui, dan hasil operasi. Validasi: Melakukan validasi terhadap nilai-nilai yang akan diperbarui untuk memastikan konsistensi data. Transaksi: Membungkus operasi update dalam transaksi untuk menjaga konsistensi data jika terjadi kesalahan.

```
69 public void deleteData(String npm) {
70     // Hapus data dari dataArrayList
71     dataArrayList.removeIf(mhs -> mhs.getNPM().equals(npm));
72     deleteDataFromDatabase(npm);
73 }
74
75
76
77 private void deleteDataFromDatabase(String npm) {
78     String sql = "DELETE FROM mahasiswa WHERE npm = ?";
79     try (PreparedStatement pst = conn.prepareStatement(sql)) {
80         pst.setString(1, npm);
81         pst.executeUpdate();
82         JOptionPane.showMessageDialog(null, "Data berhasil dihapus!");
83     } catch (SQLException e) {
84         JOptionPane.showMessageDialog(null, "Gagal menghapus data: " + e.getMessage());
85     }
86 }
87
88 }
```

Menghapus dari List: Menghapus data mahasiswa dari dalam list di aplikasi. Membuat perintah SQL:

Menyusun perintah DELETE untuk menghapus data mahasiswa dari database berdasarkan NPM. Membuat PreparedStatement: Membuat objek PreparedStatement untuk mengeksekusi query secara terstruktur dan aman. Mengisi parameter: Mengisi placeholder pada query dengan nilai NPM yang akan dihapus. Mengeksekusi query: Melakukan operasi delete pada database. Menampilkan pesan: Memberikan feedback kepada pengguna mengenai hasil operasi delete.

```
private void deleteActionPerformed(java.awt.event.ActionEvent evt) {  
    int selectedRow = tablelist.getSelectedRow();  
    if (selectedRow >= 0) {  
        String npm = tablelist.getValueAt(selectedRow, 0).toString();  
        mhs.deleteData(npm); // Hapus data dari controller  
        tablelist.setModel(mhs.showData()); // Refresh tabel  
    } else {  
        JOptionPane.showMessageDialog(null, "Pilih data yang ingin dihapus.");  
    }  
}
```

Kode di atas merupakan bagian dari sebuah program Java yang berfungsi untuk menghapus data mahasiswa dari suatu sistem. Bagian kode ini kemungkinan besar merupakan handler untuk sebuah event, seperti ketika pengguna mengklik tombol "Hapus" pada sebuah GUI.

Data Mahasiswa

NPM:

Nama:

Tinggi:

Pindahan?:

NPM	Nama	Tinggi	Pindahan
514214444	Sulthan Raziek	123	Ya

```

String npm = TFNPM.getText();
String nama = TFNama.getText();
int tinggi = Integer.parseInt(TFTinggi.getText());
boolean pindahanStatus = pindahan.getSelectedIndex().toString().equals("Ya");

// Cek apakah data dengan NPM ini sudah ada
boolean isExisting = false;
for (ModelMahasiswa mhs : mhs.dataArrayList) {
    if (mhs.getNPM().equals(npm)) {
        isExisting = true;
        break;
    }
}

if (isExisting) {
    // Jika sudah ada, lakukan update
    mhs.updateData(npm, nama, tinggi, pindahanStatus);
} else {
    // Jika belum ada, lakukan insert
    mhs.insertData(npm, nama, tinggi, pindahanStatus);
}

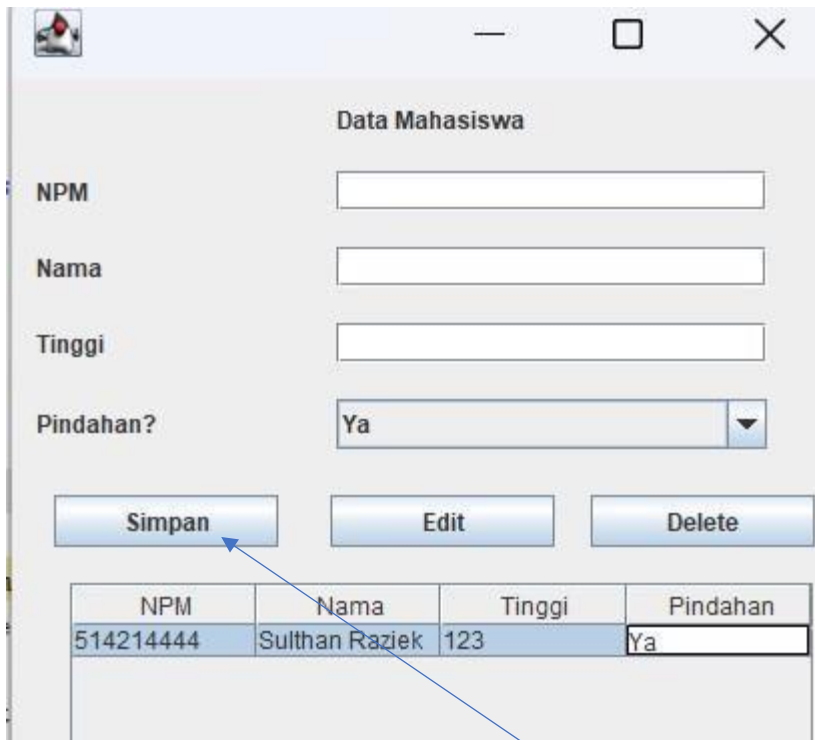
// Setelah simpan, refresh tabel
tablelist.setModel(mhs.showData());
// Reset field input setelah simpan
TFNPM.setText("");
TFNama.setText("");
TFTinggi.setText("");
pindahan.setSelectedIndex(0);

```

Validasi input: Menambahkan validasi untuk memastikan data yang dimasukkan oleh pengguna valid (misalnya, NPM harus unik, tinggi harus berupa angka, dll.). Pesan konfirmasi: Menampilkan pesan konfirmasi kepada pengguna sebelum melakukan perubahan data. Penanganan kesalahan: Menambahkan penanganan kesalahan untuk kasus-kasus seperti koneksi database gagal atau kesalahan saat melakukan operasi database. Penggunaan library database: Menggunakan library database yang lebih canggih untuk mempermudah operasi database dan meningkatkan keamanan.



Selanjutnya Run Project F6



Data Mahasiswa

NPM

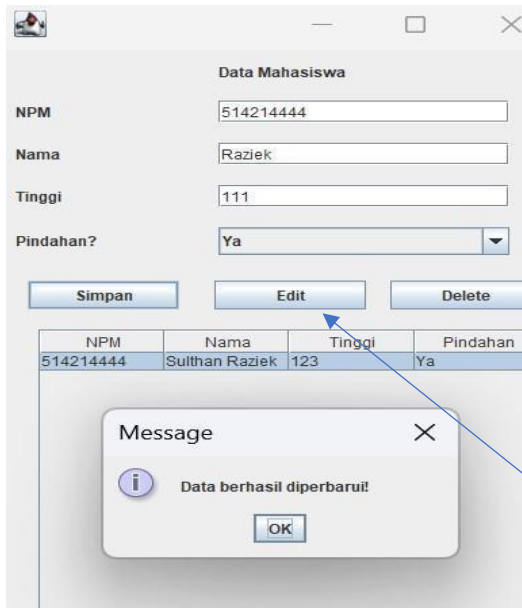
Nama

Tinggi

Pindahan?

NPM	Nama	Tinggi	Pindahan
514214444	Sulthan Raziek	123	Ya

Isi Data mahasiswa gambar di Atas lalu Simpan



Data Mahasiswa

NPM


Nama

Tinggi

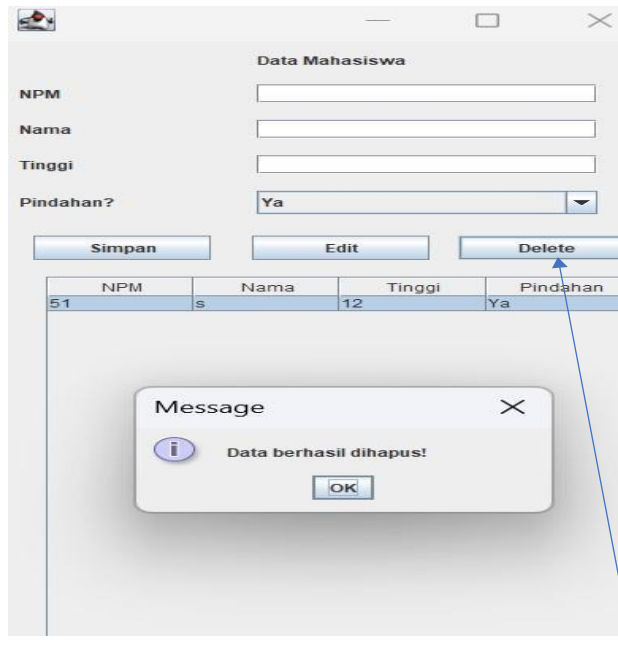
Pindahan?

NPM	Nama	Tinggi	Pindahan
514214444	Sulthan Raziek	123	Ya

Message

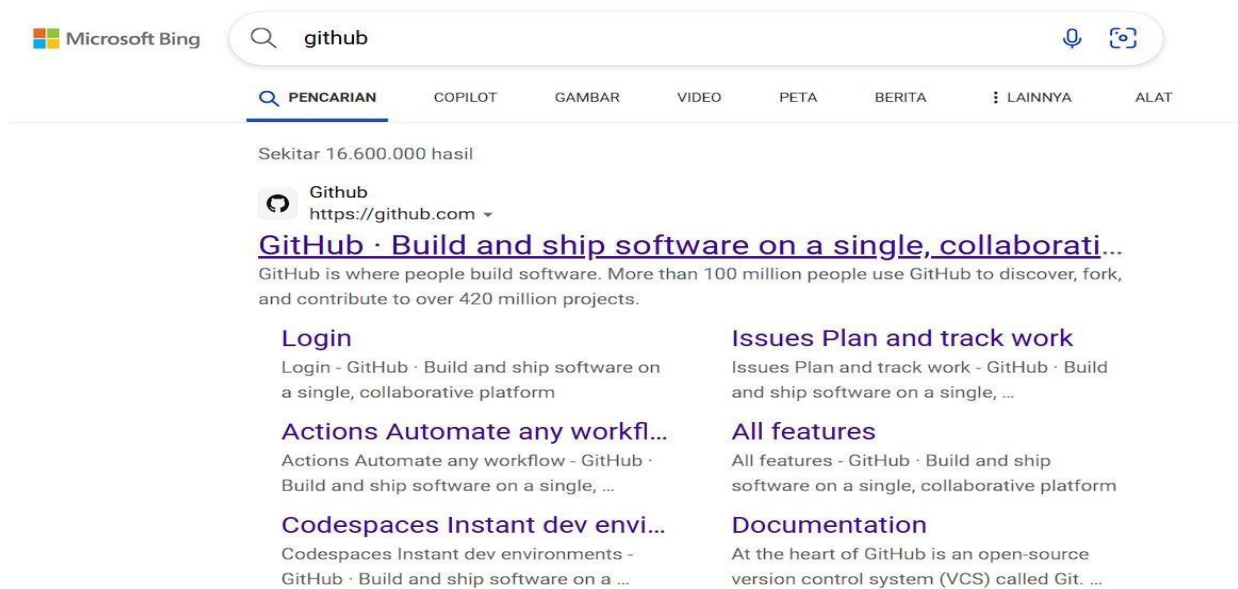
 Data berhasil diperbarui!

Ketika ingin mengubah Data Mahasiswa click bagian Edit



Jika ingin menghapus data Mahasiswa click Delete .

Selanjutnya jika ingin mengimport folder/program yang telah dibuat seperti sebelumnya kedalam github. Untuk langkah-langkahnya seperti berikut/



Cari di dalam web browser, pada menu search bar ketikan github.com > lalu pilih menu login untuk mengakses fitur yang disediakan pada platform tersebut.

Start a new repository for sulthanRaziek

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name *

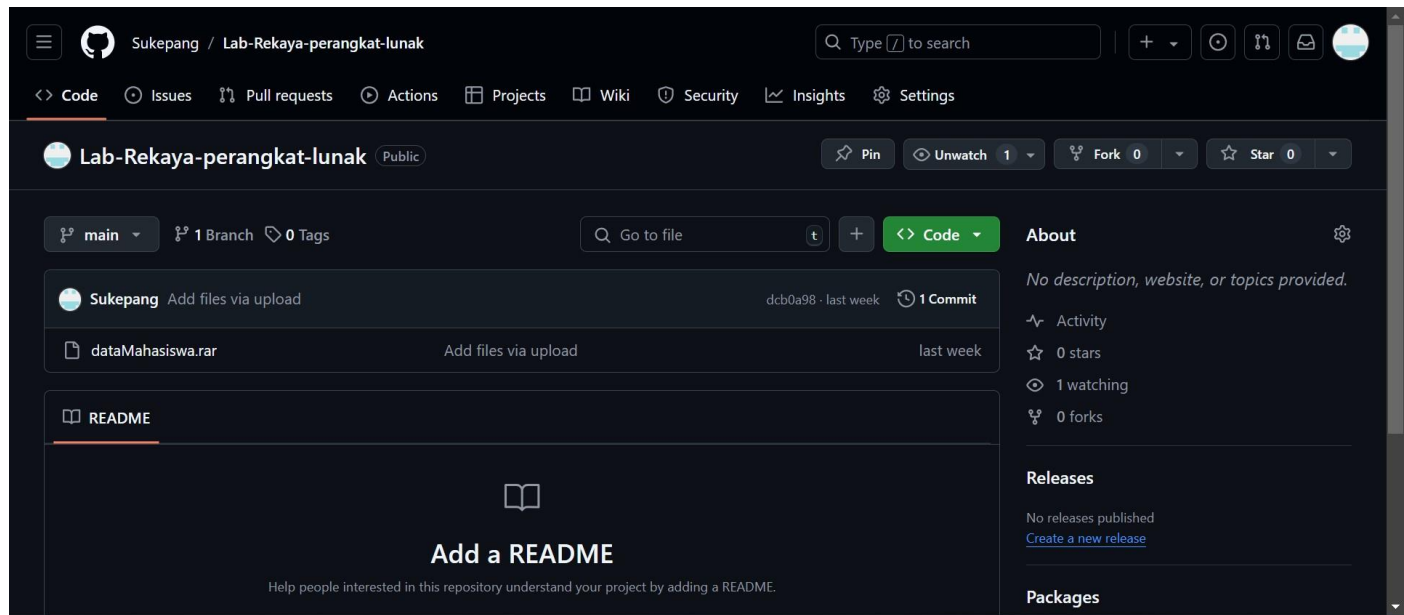
name your new repository...

☒ **Public**
Anyone on the internet can see this repository

☐ **Private**
You choose who can see and commit to this repository

Create a new repository

Lalu pilih/buat repository untuk tempat menaruh file/folder/program yang ingin disimpan di dalam github tersebut. Setelah itu silahkan Upload files dataMahasiswa di NetBeans file



SELESAI.

Link Github : <https://github.com/Sukepang/Lab-Rekaya-perangkat-lunak>