



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA
DAN KOMPUTER BANI SALEH
LAPORAN PRAKTIKUM KOMPUTER**

Nama	RIZKI WAHYUDI	Ruang	LAB. PEMROGRAMAN
NIM	43A87006180117	Mata Kuliah	PEMROGRAMAN JAVA 2
Jurusan	S1/TEKNIK INFORMATIKA/ 4A/ MALAM	Praktikum	JAVA GUI
Dosen	MUHAMMAD NUR, S.kom., M.kom.	Hari / Tanggal	21 MARET 2020
Judul: MEMBUAT APLIKASI DATA MAHASISWA			

1. Tujuan Praktikum

Mempelajari sebuah dasar-dasar pemrograman berbasis JAVA GUI dengan memakai Software Apache NetBeans IDE, membuat aplikasi Data Mahasiswa dan mengenal komponen-komponen di dalam JAVA GUI.

2. Dasar Teori

1. **Graphic User Interface (GUI)** adalah pemrograman dengan bahasa Java yang dibuat menggunakan aplikasi yang berbasis GUI. Tujuannya adalah menambahkan beberapa komponen yang tidak bisa dibuat dalam basis text. Komponen – komponen tersebut bisa berupa tombol, gambar, dll. Tujuannya adalah untuk memudahkan user menggunakan program yang dibuat tersebut.
2. **Table Model (TableModelColumn)** digunakan sebagai pusat pengelolaan kolom tabel di pemrograman Java. Table column model mengimplementasikan interface TableColumnModel yang mendefinisikan banyak penanda metoda umum seperti menambah, menghapus dan memindahkan kolom, mengembalikan jumlah keseluruhan kolom atau hanya kolom yang dipilih, menentukan apakah kolom tabel dapat dipilih atau tidak, menentukan margin kolom dan sebagainya.
3. **Model-View-Controller** atau **MVC** adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data dari tampilan dan cara bagaimana memprosesnya. Dalam implementasinya kebanyakan kerangka kerja dalam aplikasi web adalah berbasis arsitektur MVC.

3. Langkah Praktikum

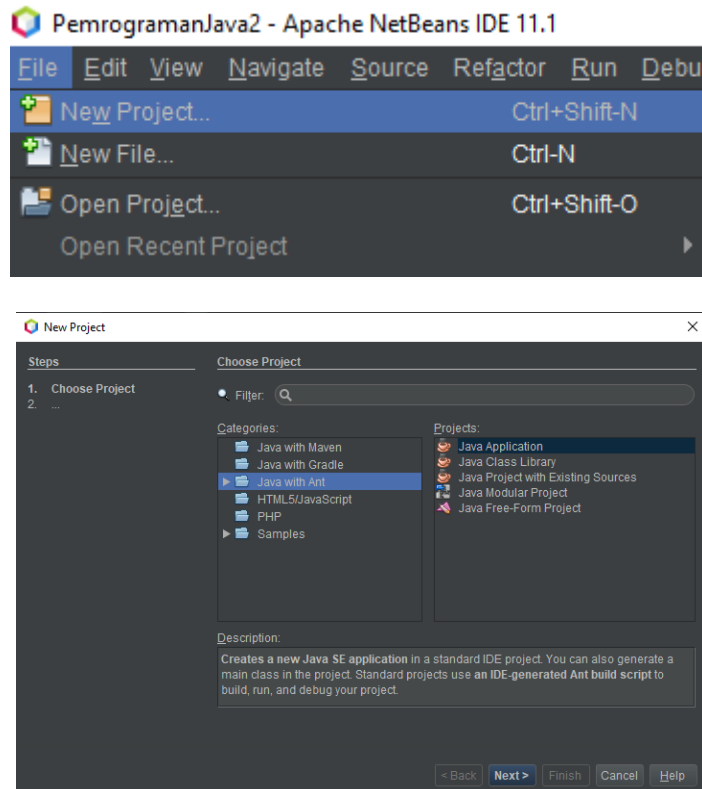
1. Membuat Project Baru

1.1 *Membuat Project*

Project adalah sebuah Aplikasi, kerjaan, atau apapun yang menghasilkan akhir atau output untuk membantu pekerjaan manusia, kali ini saya akan membuat sebuah project Java GUI menggunakan sebuah Software Apache NetBeans IDE 11.1.

1.2 *Cara membuat Project*

Cara membuat project di NetBeans sangat mudah sekali, dengan mengklik **File -> New Project -> Java -> Java Application..** atau **Ctrl+Shift+N**.

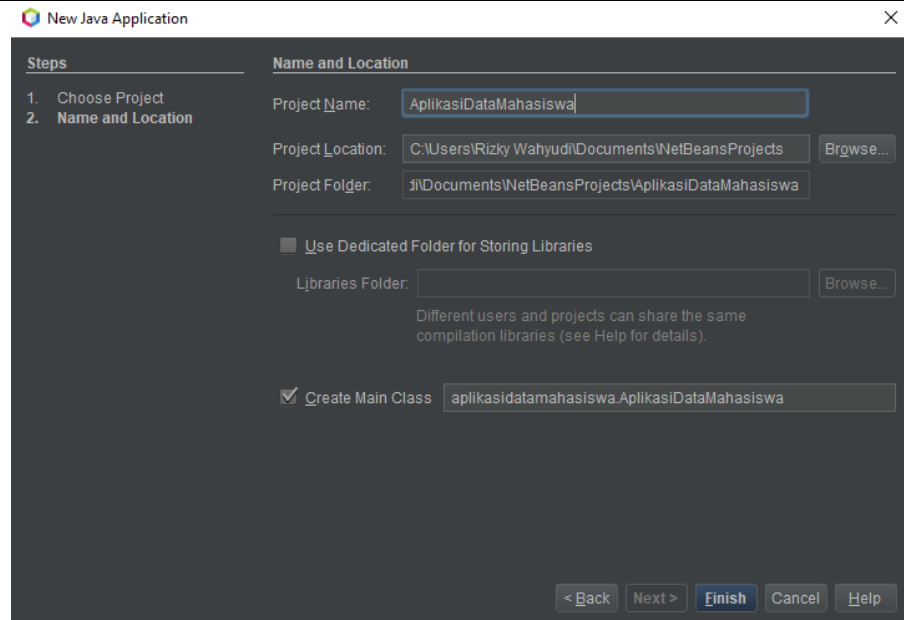


2. Memberi Nama Project

2.1 *Memberi Nama Project*

Nama Project : **Aplikasi Data Mahasiswa**

Uncheck : **Create Main Class -> klik Finish.**

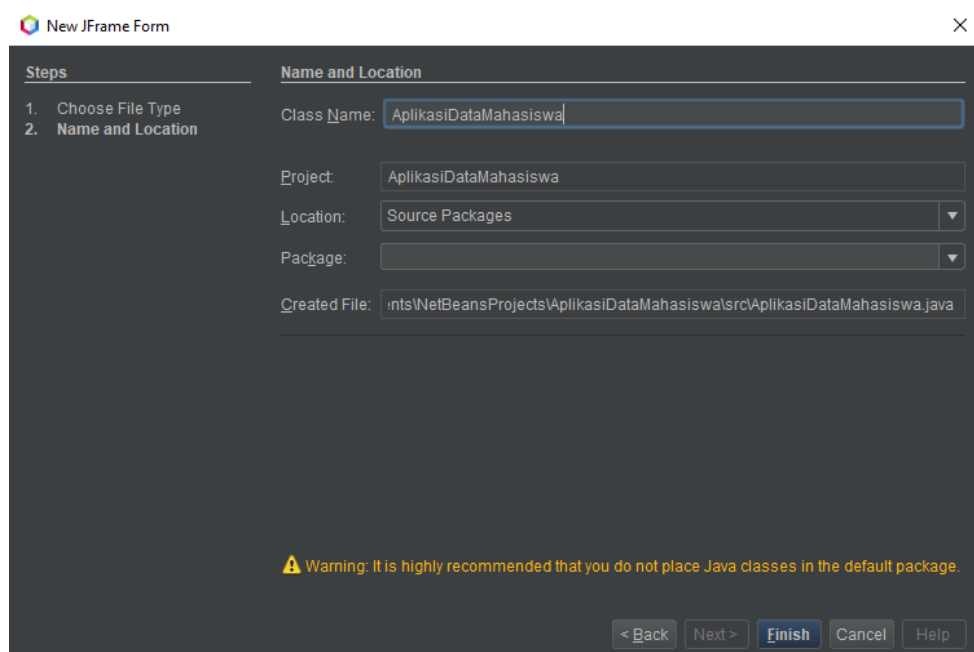
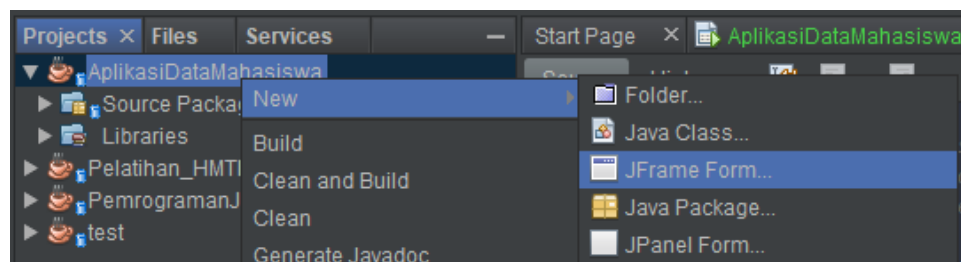


3. Menambahkan JFrame Form Pada Project

3.1 *Menambahkan dan Membuat Nama JFrame Form*

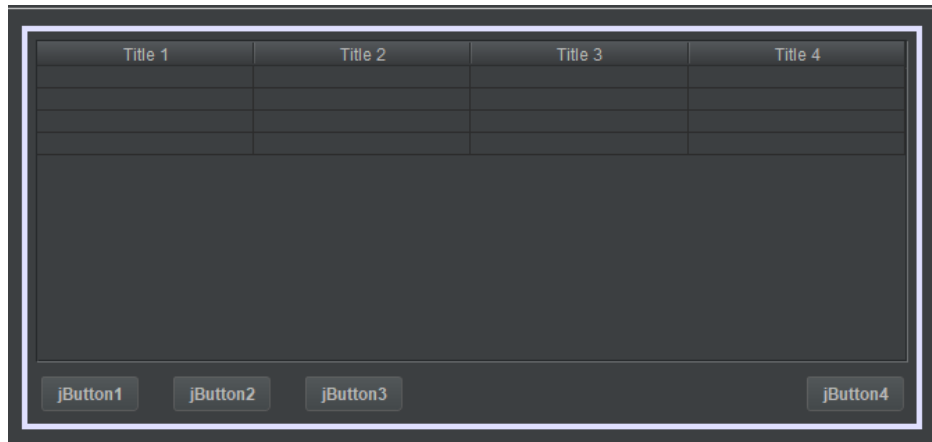
Nama Frame : **AplikasiDataMahasiswa**

Package : **GUI -> Klik Finish.**



4. Menempatkan GUI Component ke Design (Frame)

4.1 *Menambahkan Komponen JTable*



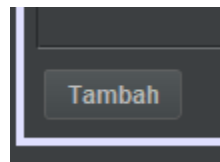
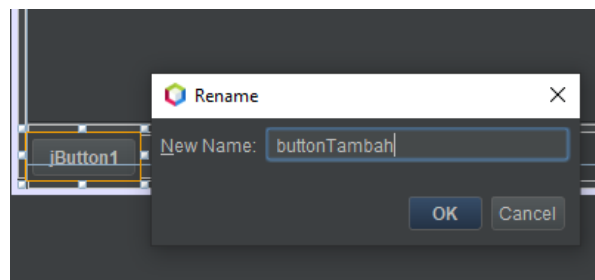
5. Mengubah Text dari GUI Component (Edit Text)

5.1 *Mengubah Nama Text*

- JTable => menjadi **Kosong**
- JButton1 => menjadi **Tambah**
- JButton2 => menjadi **Ubah**
- JButton3 => menjadi **Hapus**
- JButton4 => menjadi **Keluar**

5.2 *Cara Mengubah*

Klik Field Text yang ingin diubah -> klik kanan, pilih Edit Text -> kursor akan fokus ke bagian yang akan diganti text nya.



Ulangi sampai sesuai dengan ketentuan dan kebutuhan.

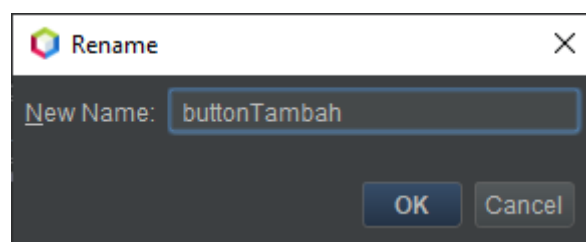
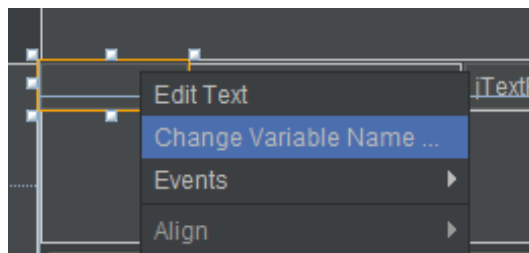
6. Mengubah Nama Variable dari Setiap GUI Component (Change Variable Name)

6.1 Mengubah Nama Variable Setiap Komponen

JTable => menjadi **tabelMahasiswa**
JButton1 => menjadi **buttonTambah**
JButton2 => menjadi **buttonUbah**
JButton3 => menjadi **buttonHapus**
JButton4 => menjadi **buttonKeluar**

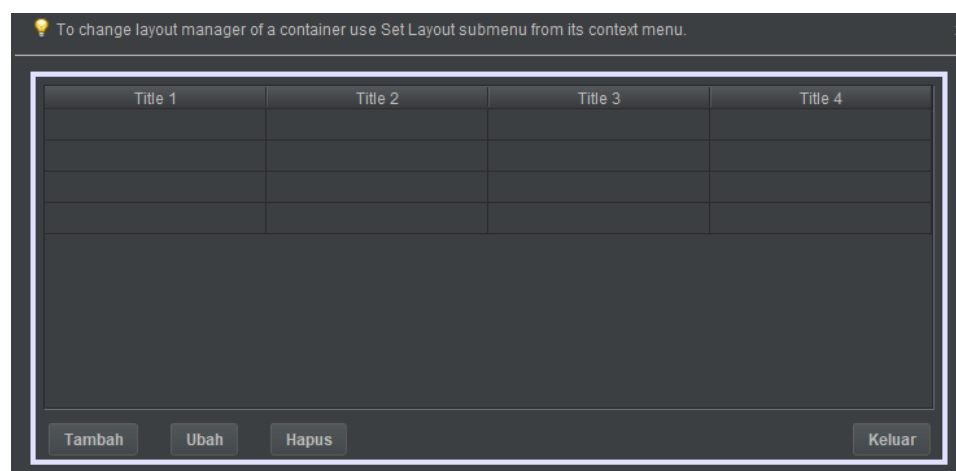
6.2 Cara Merubahnya

Klik Field Text yang ingin diubah Variable Namanya-> klik kanan, pilih Change Variable Name -> akan ada popup informasi untuk merubah nama -> tuliskan nama -> klik ok.



7. Merapikan Tampilan Frame Program

Merapikan tampilan Frame adalah membuang Frame sisa yang tidak digunakan sebagai fungsi Appearance atau keindahannya bagi user, kita hanya menggunakan pallet yang penting dari program kita saja.



8. Buat Code untuk Event Handling Penampilan Data

8.1 *Membuat Variable dalam Mahasiswa.java*

8.1.1 Source Code

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package PertemuanKe4.AplikasiGUI3;

/**
 *
 * @author Rizky Wahyudi
 */
public class Mahasiswa {

    //pembuatan variable
    private String nim;
    private String nama;
    private String jurusan;
    private String alamat;
    private String email;
    private String telepon;

    public Mahasiswa() {
    }

    //pembuatan constructor
    public Mahasiswa(String nim, String nama, String jurusan, String alamat
, String email, String telepon) {
        this.nim = nim;
        this.nama = nama;
        this.jurusan = jurusan;
        this.alamat = alamat;
        this.email = email;
        this.telepon = telepon;
    }

    //pembuatan getter dan setter untuk variable yang sudah dibuat
    public String getNim() {
        return nim;
    }

    public void setNim(String nim) {
        this.nim = nim;
    }
}
```

```
public String getNama() {  
    return nama;  
}  
  
public void setNama(String nama) {  
    this.nama = nama;  
}  
  
public String getJurusan() {  
    return jurusan;  
}  
  
public void setJurusan(String jurusan) {  
    this.jurusan = jurusan;  
}  
  
public String getAlamat() {  
    return alamat;  
}  
  
public void setAlamat(String alamat) {  
    this.alamat = alamat;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getTelepon() {  
    return telepon;  
}  
  
public void setTelepon(String telepon) {  
    this.telepon = telepon;  
}  
  
//pembuatan object untuk tampil data  
public Object getValue(int index) {  
    switch (index) {  
        case 0:  
            return nim;  
        case 1:  
            return nama;  
        case 2:  
            return jurusan;  
    }  
}
```

```

        case 3:
            return alamat;
        case 4:
            return email;
        case 5:
            return telepon;
        default:
            return null;
    }
}
}

```

8.2 Membuat Model di MahasiswaTableModel.java

8.2.1 Source Code

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package PertemuanKe4.AplikasiGUI3;

import java.util.List;
import javax.swing.table.AbstractTableModel;

/**
 *
 * @author Rizky Wahyudi
 */
public class MahasiswaTableModel extends AbstractTableModel{
    //pembuatan generic list
    private List<Mahasiswa> list;

    //pembuatan constructor tanpa parameter
    public MahasiswaTableModel() {
    }
    //pembuatan constructor dengan parameter
    public MahasiswaTableModel(List<Mahasiswa> list) {
        this.list = list;
    }

    @Override
    public int getRowCount() {
        //pembuatan size baris atau data pada list
        return list.size();
    }
}

```



```

@Override
public int getColumnCount() {
    //mengikuti entiti pada table
    return 6;
}

@Override
public Object getValueAt(int rowIndex, int columnIndex) {
    //memanggil object pada baris dan kolom
    return list.get(rowIndex).getValue(columnIndex);
}

private String[] columns = {"NIM", "Nama", "Jurusan", "Alamat", "Email", "Telepon"};
@Override
public String getColumnName(int index) {
    return columns[index];
}
}

```

8.3 Membuat Event Handling AplikasiDataMahasiswa.java

8.3.1 Source Code

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package PertemuanKe4.AplikasiGUI3;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

/**
 *
 * @author Rizky Wahyudi
 */
public class AplikasiDataMahasiswa extends javax.swing.JFrame {
    //membuat object dari mahasiswa
    private MahasiswaTableModel model;
    //membuat list untuk menampung data
    private List<Mahasiswa> listData;
    /**
     * Creates new form AplikasiDataMahasiswa
     */
}

```

```

public AplikasiDataMahasiswa() {
    initComponents();
    //memposisikan table di tengah
    setLocationRelativeTo(null);
    //pengambilan data secara manual
    initData();
    //memasukkan data ke model
    model = new MahasiswaTableModel(listData);
    //memasukkan data dari model ke table mahasiswa
    tableMahasiswa.setModel(model);
}

//pembuatan fungsi initData();
private void initData(){
    //pemanggilan listData
    listData = new ArrayList<>();
    //pembuatan class mahasiswa dan pengisian data ke mahasiswa
    Mahasiswa mahasiswa = new Mahasiswa
        ("43A87006180117", "Rizki Wahyudi", "Teknik Informatika",
        "Kota Bekasi", "rizukiwahyudio@gmail.com", "08521134548
1");

    //memasukkan data ke listData mahasiswa
    listData.add(mahasiswa);

    //pembuatan data ke dua
    mahasiswa = new Mahasiswa
        ("43A87006180118", "ateng", "Teknik Informatika",
        "Kota Bogor", "ateng@gmail.com", "08971063039");
    listData.add(mahasiswa);

    //pembuatan data ke tiga
    mahasiswa = new Mahasiswa
        ("43A87006180119", "maemunah", "Teknik Informatika",
        "Kota Cimahi", "mae@gmail.com", "085788783566");
    listData.add(mahasiswa);

    //pembuatan data ke empat
    mahasiswa = new Mahasiswa
        ("43A87006180120", "bahrudin", "Teknik Informatika",
        "Kota Bandung", "bah@gmail.com", "08587651234");
    listData.add(mahasiswa);

    //pembuatan data ke lima
    mahasiswa = new Mahasiswa
        ("43A87006180121", "zulkhadafi", "Teknik Informatika",
        "Kota Makasar", "zul@gmail.com", "08229889900");
    listData.add(mahasiswa);
}
/**
 * This method is called from within the constructor to initialize the

```

```

form.
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">

    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        tableMahasiswa = new javax.swing.JTable();
        buttonTambah = new javax.swing.JButton();
        buttonUbah = new javax.swing.JButton();
        buttonHapus = new javax.swing.JButton();
        buttonKeluar = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        setTitle("Aplikasi Data Mahasiswa");

        tableMahasiswa.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        tableMahasiswa.setRowHeight(24);
        tableMahasiswa.setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
        tableMahasiswa.getTableHeader().setResizingAllowed(false);
        tableMahasiswa.getTableHeader().setReorderingAllowed(false);
        jScrollPane1.setViewportView(tableMahasiswa);

        buttonTambah.setText("Tambah");

        buttonUbah.setText("Ubah");

        buttonHapus.setText("Hapus");

        buttonKeluar.setText("Keluar");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    
```

```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 687, Short.MAX_VALUE)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(buttonTambah)
                            .addGap(18, 18, 18)
                            .addComponent(buttonUbah)
                            .addGap(18, 18, 18)
                            .addComponent(buttonHapus)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(buttonKeluar)))
                    .addContainerGap())
                );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 250, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(buttonTambah)
                        .addComponent(buttonUbah)
                        .addComponent(buttonHapus)
                        .addComponent(buttonKeluar)))
                );

        pack();
    } // </editor-fold>

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {

```

```

        new AplikasiDataMahasiswa().setVisible(true);
    }
});
}

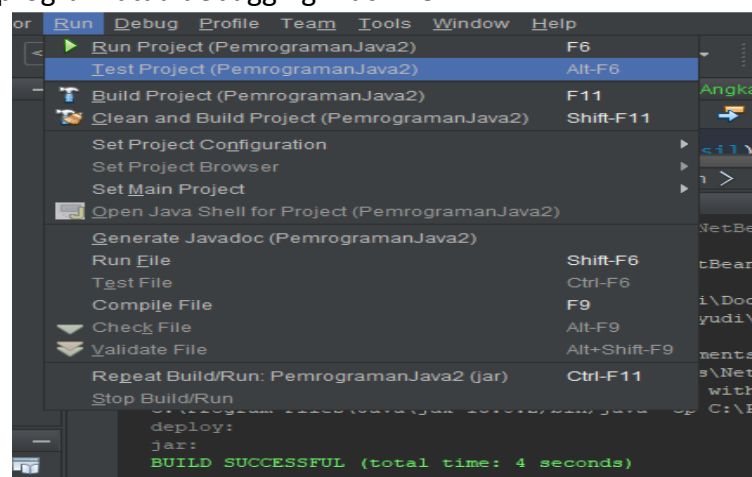
// Variables declaration - do not modify
private javax.swing.JButton buttonHapus;
private javax.swing.JButton buttonKeluar;
private javax.swing.JButton buttonTambah;
private javax.swing.JButton buttonUbah;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tableMahasiswa;
// End of variables declaration
}

```

4. Hasil Praktikum

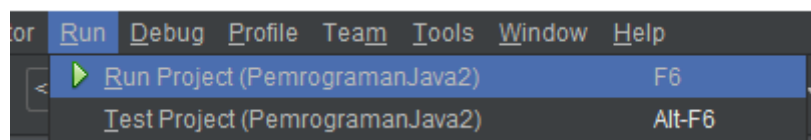
1. Mengkompilasi (Build, F11) Project

Build tools yang saya ketahui adalah membuild atau mengkompilasi sebuah Project apakah layak untuk di running, biasanya kita kenal sebagai proses debugging, ialah mencari struktur atau source logic yang salah, dan dianggap error oleh program atau debugging machine.



2. Menjalankan (Run, F6) Project

Running atau Run File Project biasanya kita kenal sebagai menjalankan sebuah project yang sudah dilakukan sebelumnya Build atau debugging, jika tidak ada yang abnormal oleh mesin, maka project yang kita buat bisa dijalankan.



3. Menampilkan Hasil Running Aplikasi



The screenshot shows a web application window titled "Aplikasi Data Mahasiswa". It contains a table with 6 columns: NIM, Nama, Jurusan, Alamat, Email, and Telepon. The table lists 5 students. Below the table is a large empty rectangular area. At the bottom of the window are four buttons: "Tambah", "Ubah", "Hapus", and "Keluar".

NIM	Nama	Jurusan	Alamat	Email	Telepon
43A87006180117	Rizki Wahyudi	Teknik Informatika	Kota Bekasi	rizukiwahyudio@g...	085211345481
43A87006180118	ateng	Teknik Informatika	Kota Bogor	ateng@gmail.com	08971063039
43A87006180119	maemunah	Teknik Informatika	Kota Cimahi	mae@gmail.com	085788783566
43A87006180120	bahrudin	Teknik Informatika	Kota Bandung	bah@gmail.com	08587651234
43A87006180121	zulkhadafi	Teknik Informatika	Kota Makasar	zul@gmail.com	08229889900

Tambah Ubah Hapus Keluar

5. Kesimpulan

Model MVC (Model-View-Controller) Sudah banyak dipakai berbagai platform Bahasa, dikarenakan sangat mudahnya konsep ini. Konsep ini juga banyak berguna untuk mempermudah suatu pekerjaan programmer, karena sudah dibuatkan skema atau alur penempatan suatu logic pemrograman, jadi kita tidak perlu kesusahan untuk mencari logic pemrograman yang ingin kita gunakan.

1. **Model**, Model mewakili struktur data. Model merupakan bagian yang bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisir data (biasanya dari basis data). Tugas yang ia lakukan meliputi memasukkan data ke basis data, pembaruan data, menghapus data, dan lain-lain. Model menjalankan tugasnya berdasarkan instruksi dari controller.
2. **View**, View merupakan bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. **Controller**, Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.