

XIV PEMBUATAN CLASS OOP DAN GUI

Tujuan Instruksional Khusus :

- Mahasiswa mampu membuat program GUI Java dengan menggunakan class dan object dan bisa melakukan passing object antar object

Deskripsi Materi Perkuliahan :

Dalam bagian ini akan dijelaskan topik-topik :

- Langkah-langkah pembuatan class OOP dalam pemrograman GUI melalui contoh sekaligus juga passing object dalam OOP serta diberikan penanganan event untuk kegiatan input, proses dan output serta event penanganan kesalahan yang mungkin terjadi pada kegiatan tersebut.

Contoh 1:

Penyelesaian permasalahan mencari luas persegi panjang bisa dilakukan dengan cara yang lain yaitu dengan memecah permasalahan menjadi bagian class tersendiri (dalam suatu paket) dan bagian interface (GUI)

Langkah-langkah Penyelesaian adalah sebagai berikut :

1. Buatlah class dari PersegiPanjang model OOP yang dilengkapi dengan bagian constructor, Method input, Method proses dan Method output.
Class ini disimpan dalam paket tersendiri (misal paketKu).

Buat Method input dengan nama fungsi setNamaClass atau setNamaAtribut

Buat Method output dengan nama fungsi getNamaAtribut.

Buat Method proses yang bersesuaian dengan permasalahan.

Contoh dari class yang akan dibuat adalah sbb:

```
package paketKu;

public class PersegiClass {
    private float panjang;
    private float lebar;

    // constructor 1
    public PersegiClass(){
        panjang=0;
        lebar=0;
    }

    // constructor 2
    public PersegiClass(float panjang, float lebar){
        this.panjang=panjang;
        this.lebar=lebar;
    }

    // method Input

    public void setPersegiClass(float pjg, float lbr) {
        panjang=pjg;
        lebar=lbr;
    }

    public void setPanjang(float pjg){
        panjang=pjg;
    }

    public void setLebar(float lbr){
        lebar=lbr;
    }

    //Method Output
    public float getPanjang(){
        return panjang;
    }

    public float getLebar(){
        return lebar;
    }

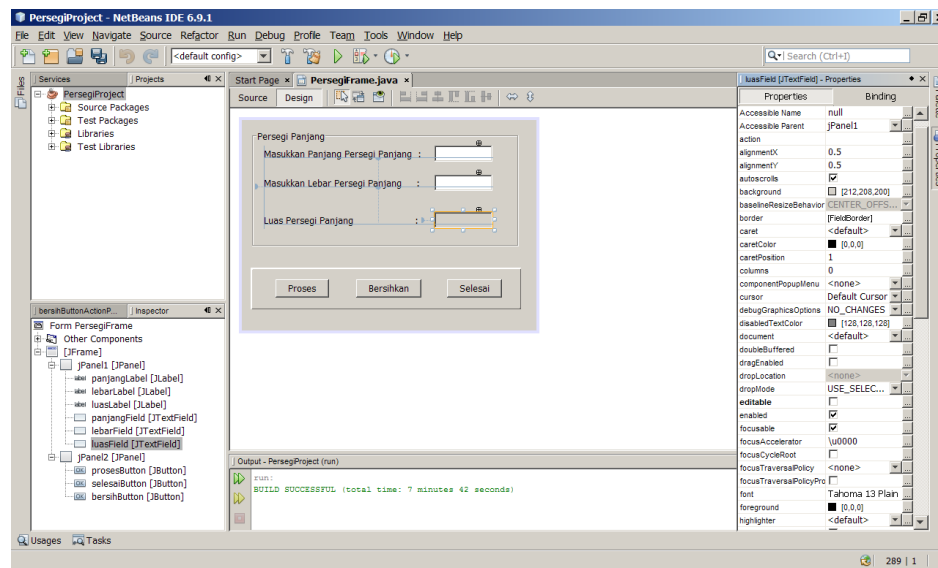
    // Method PROSES -----
    public float hitungLuas(){
        return (panjang*lebar);
    }

}
```

2. Berdasarkan class itu maka akan dibuatkan interface GUI untuk menginputkan data kedalam class, melakukan pemanggilan pemrosesan serta menampilkan hasil outputnya.

Langkah-langkahnya adalah sbb:

- a. Buatlah tampilan GUI frame sbb:



Gambar 14.1 Form Class Persegi Panjang

- b. Buatlah fungsi untuk menangani input, proses dan output

Buat variabel lokal (optional) dan buat object berdasarkan class PersegiClass tsb.

Ambil data dari form (dari textField dan kirim ke class)

Proses dan tampilkan hasil ke form lagi.

```
private void prosesHitungLuas() {  
    float panjang, lebar, luas; // variable lokal  
    PersegiClass objPersegi = new PersegiClass(); // buat sebuah objek  
  
    try {  
        // konversi dari isian textField  
        panjang = Float.parseFloat(panjangField.getText());  
        lebar = Float.parseFloat(lebarField.getText());  
  
        objPersegi.setPanjang(panjang); // input kirim ke class  
        objPersegi.setLebar(lebar);  
        luas = objPersegi.hitungLuas(); // Proses  
  
        luasField.setText(String.valueOf(luas)); // tampilkan ke GUI  
    }  
    catch (Throwable t) {
```

```

        JOptionPane.showMessageDialog(null,"Nilai bukan numerik, "
        + "silahkan diperbaiki untuk bisa diproses");
    }
}

private void bersihField(){
    panjangField.setText("");
    lebarField.setText("");
    luasField.setText("");
}

```

3. Lakukan pemanggilan fungsi diatas melalui penanganan event yang bersesuaian seperti pada button proses dan button bersih.

Fungsi nya adalah sbb:

```

private void prosesButtonActionPerformed(java.awt.event.ActionEvent evt) {
    prosesHitungLuas();
}

private void bersihButtonActionPerformed(java.awt.event.ActionEvent evt) {
    bersihField();
    panjangField.requestFocus();
}

```

Contoh 2:

Akan dibuat program untuk menangani pengiriman object (passing object).

Kasusnya adalah mencari titik tengah antara dua buah titik dalam koordinat kartesian.

Langkah-langkah yang dilakukan adalah :

1. Membuat class Koordinat (OOP) yang berisi atribut, constructor, method input, method proses dan method output.

Class ini disimpan dalam paket tersendiri (misal paketKu).

Contoh dari class yang akan dibuat adalah :

```

package paketKu;

public class Koordinat{
    // Atribut -----

    private float absis;
    private float ordinat;

    // Constructor -----
    public Koordinat(){
        absis=0;
        ordinat=0;
    }
}

```

```

        public Koordinat(float absis, float ordinat){
            this.absis=absis;
            this.ordinat=ordinat;
        }

// Method Input -----
        public void setKoordinat(float pAbsis, float pOrdinat){
            absis=pAbsis;
            ordinat=pOrdinat;
        }

        public void setAbsis(float pAbsis){
            absis=pAbsis;
        }

        public void setOrdinat(float pOrdinat){
            ordinat=pOrdinat;
        }

// Method Output -----
-
        public float getAbsis(){
            return absis;
        }

        public float getOrdinat(){
            return ordinat;
        }

// Method Proses -----
--

        public void titikTengah(Koordinat P1, Koordinat P2 ){
            absis=(P1.absis + P2.absis)/2;
            ordinat=(P1.ordinat + P2.ordinat)/2;
        }

        public void cerminX(Koordinat P1){
            absis=P1.absis;
            ordinat=-(P1.ordinat);
        }

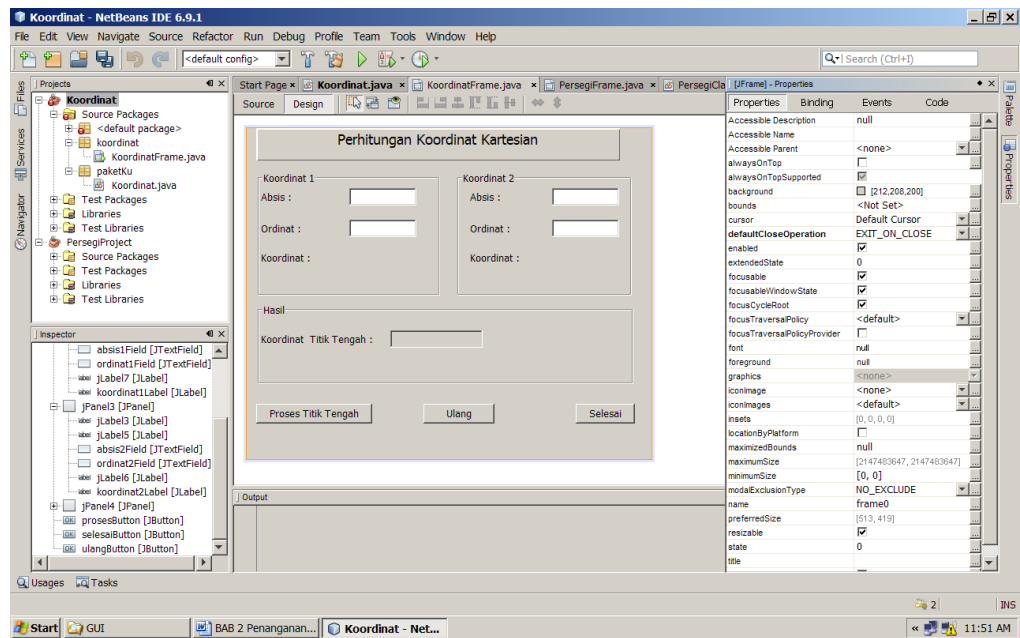
        public Koordinat titikTengah2(Koordinat P ){
            Koordinat pHasil=new Koordinat();
            pHasil.absis=(P.absis + absis)/2;
            pHasil.ordinat=(P.ordinat + ordinat)/2;
            return pHasil;
        }
    }

```

2. Berdasarkan class itu maka akan dibuatkan interface secara GUI untuk menginputkan data kedalam class, melakukan pemanggilan pemrosesan serta menampilkan hasil outputnya.

Langkah-langkahnya adalah sbb:

- a. Buatlah tampilan GUI frame sbb:



Gambar 14.2 Form Perhitungan Koordinat Kartesian

b. Buatlah fungsi untuk menangani input, proses dan output

```
private void prosesTitikTengah() {
    float absis1, absis2, ordinat1, ordinat2;

    Koordinat titik1 = new Koordinat();
    Koordinat titik2 = new Koordinat();
    Koordinat tengah = new Koordinat();

    try {
        absis1 = Float.parseFloat(absis1Field.getText());
        ordinat1 = Float.parseFloat(ordinat1Field.getText());
        absis2 = Float.parseFloat(absis2Field.getText());
        ordinat2 = Float.parseFloat(ordinat2Field.getText());

        titik1.setKoordinat(absis1, ordinat1);
        titik2.setKoordinat(absis2, ordinat2);

        tengah.titikTengah(titik1, titik2);

        tengahField.setText("(" + String.valueOf(tengah.getAbsis())
            + " , " + String.valueOf(tengah.getOrdinat()) + " )");

        tengahLabel.setText("(" + String.valueOf(tengah.getAbsis())
            + " , " + String.valueOf(tengah.getOrdinat()) + " )");
    }
    catch (Throwable t) {
        JOptionPane.showMessageDialog(null, "Nilai bukan numerik, "
            + "silahkan diperbaiki untuk bisa diproses");
    }
}

private void ulang() {
    absis1Field.setText("");
    ordinat1Field.setText("");
    absis2Field.setText("");
}
```

```

        ordinat2Field.setText("");
        tengahField.setText("");
        tengahLabel.setText("");
        absis1Field.requestFocus();
    }

```

3. Lakukan pemanggilan fungsi diatas melalui penanganan event yang bersesuaian seperti pada button proses dan button ulang.

Fungsi nya adalah sbb:

```

private void prosesButtonActionPerformed(java.awt.event.ActionEvent evt) {
    prosesTitikTengah();
}

private void ulangButtonActionPerformed(java.awt.event.ActionEvent evt) {
    ulang();
}

private void selesaiButtonActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

```

Tujuan Instruksional Khusus :

- Mahasiswa mampu membuat program GUI Java dengan menggunakan Tabel

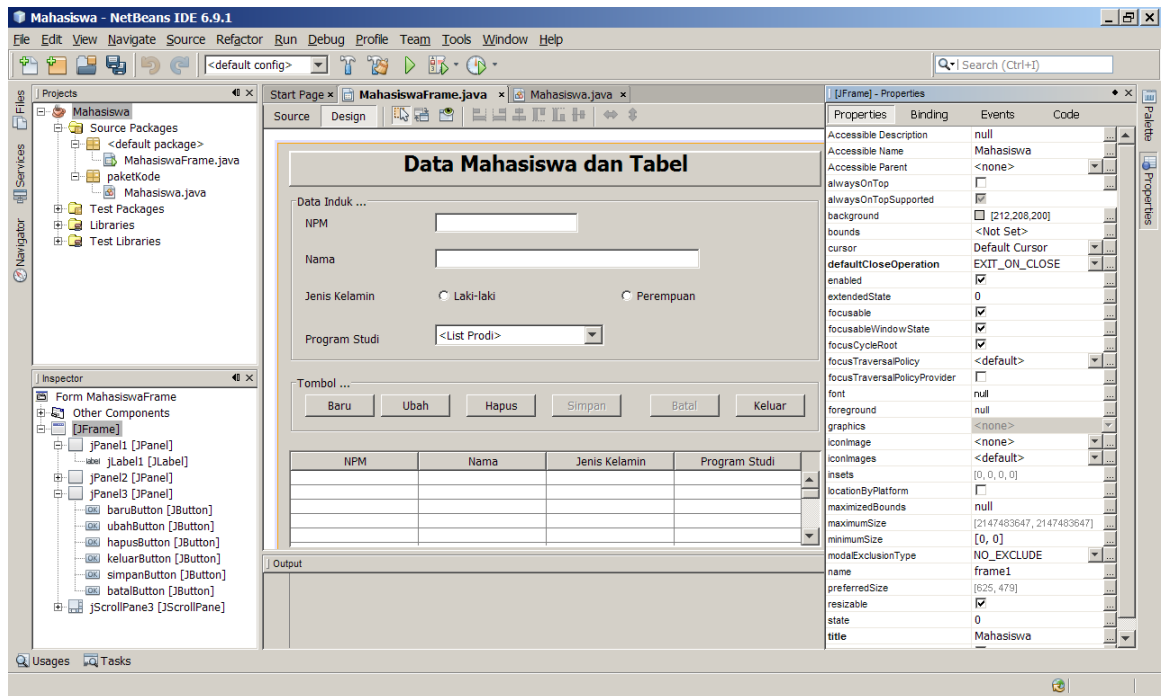
Deskripsi Materi Perkuliahan :

Dalam bagian ini akan dijelaskan topik-topik :

- Langkah-langkah pembuatan table (JTable) dalam pemrograman GUI melalui contoh

Dalam Netbeans (Java), pembuatan table bisa dilakukan dengan menggunakan komponen JTable pada bagian palette. Dalam hal ini akan diberikan suatu contoh pembuatan suatu aplikasi Pendataan Mahasiswa dengan atributnya yaitu NPM, nama, jenis kelamin dan program studi. **Semua data yang diolah akan disimpan pada JTable.** Terhadap tabel tersebut bisa dilakukan manipulasi yaitu dilakukan penambahan(insert), perubahan (edit) dan penghapusan (delete).

Tampilan dari aplikasi adalah sbb:



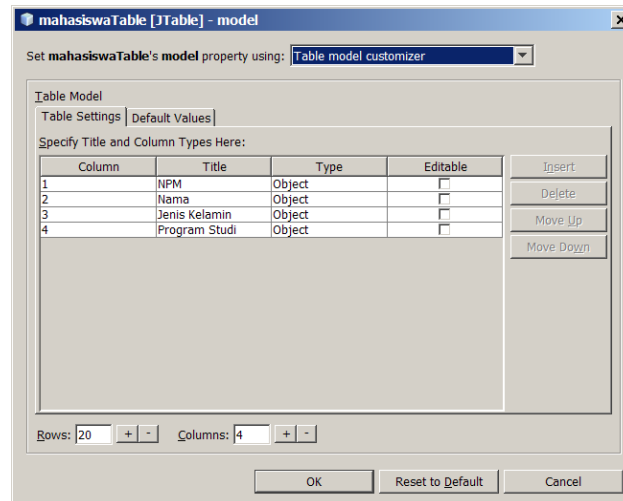
Gambar 15.1 Form Tabel Mahasiswa

Langkah-langkah pembuatan aplikasi:

1. Buatlah class OOP Mahasiswa yang berisi atribut (npm, nama, jenkel, dan prodi), constructor, method input (fungsi setAtribut), method output (fungsi getAtribut).
Catatan : method proses nantinya berupa akses database (atau yang lain).
2. Buat rancangan Form GUI seperti gambar diatas. Susun dan beri nama variable yang bersesuaian untuk setiap komponen yang dibuat, terutama komponen yang akan dilakukan perubahan dan pengolahan terhadapnya, antara lain : npmField, namaField, priaRadio, wanitaRadio, prodiComboBox, baruButton, ubahButton, hapusButton, batalButton, keluarButton, dan counterLabel.

Khusus untuk table akan diberi nama yaitu mahasiswaTable.

Properties yang diatur pada table adalah pada bagian model :



Gambar 15.2 Model objek tabel

3. Akan ditangani semua event yang penting baik terhadap elemen data, terhadap table ataupun terhadap tombol button yang disusun
4. Pertama sekali akan diatur komponen-komponen yang dibolehkan diakses (enabled dan disabled) dengan menggunakan fungsi tersendiri.

```
private void setDisabledKomponen() {
    npmField.setEnabled(false);
    namaField.setEnabled(false);
    priaRadio.setEnabled(false);
    wanitaRadio.setEnabled(false);
    prodiComboBox.setEnabled(false);
//    mahasiswaTable.setEnabled(false);
}

private void setEnabledKomponen() {
    npmField.setEnabled(true);
    namaField.setEnabled(true);
    priaRadio.setEnabled(true);
    wanitaRadio.setEnabled(true);
    prodiComboBox.setEnabled(true);
//    mahasiswaTable.setEnabled(true);
}

private void bersihKomponen() {
    npmField.setText("");
    ...
    ...
}
```

5. Gunakan event dari frame yaitu `formWindowsOpened` sebagai penanganan awal terhadap aplikasi yang dibuat.

```

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    baruButton.setEnabled(true);
    ubahButton.setEnabled(true);
    hapusButton.setEnabled(true);
    simpanButton.setEnabled(false);
    batalButton.setEnabled(false);
    keluarButton.setEnabled(true);

    setDisabledKomponen();
    bersihKomponen();

    baruButton.setMnemonic('B');
    ubahButton.setMnemonic('U');
    hapusButton.setMnemonic('H');
    simpanButton.setMnemonic('S');
    batalButton.setMnemonic('t');
    keluarButton.setMnemonic('K');

    baruButton.requestFocus();    }

```

Bisa juga dengan mengganti properties setiap tombol pada bagian enabled dan bagian mnemonic

Posisi kursor aktif adalah pada tombol baruButton.

6. Cara menginput data kedalam ke JTable :

Pertama kali deklarasikan dulu variable array satu dimensi sebagai tempat penampung sementara dan sebagai perantara antara form inputan dengan JTable. Array ini digunakan untuk memasukkan atau mengambil data atribut Mahasiswa

```
private String data[] = new String [4];
```

Siapkan variable counter sebagai atribut yang akan selalu mencatat jumlah record dari jTable.

```
static int counter=0;
```

Nilai ini digunakan untuk proses penambahan data baru yang akan meletakkan record baru pada posisi counter ;

Siapkan satu atribut status yang menunjukkan nantinya apakah dilakukan penekanan tombol baru atau tombol edit untuk membedakan dimana posisi record akan disimpan ketika ada penekanan tombol simpan . Jika dilakukan proses baru (status=1) maka simpan akan meletakkan record pada posisi counter+1, tetapi jika tombol edit atau double click pada record (status=2), maka simpan akan meletakkan record pada posisi ketika edit dilakukan.

```
private int status=0;
```

Alur untuk penambahan data yaitu ditekan dulu tombol **baru**, kemudian data diisi dan selanjutnya ditekan tombol simpan.

Gambar 15.3 Tampilan Eksekusi Mahasiswa dan Tabel

6.1 Tombol baru digunakan untuk mengosongkan komponen isian dan mengaktifkan tombol simpan dan tombol batal.

```
private void baruButtonActionPerformed(java.awt.event.ActionEvent evt) {
    setEnabledKomponen();
    bersihKomponen();
    mahasiswaTable.setEnabled(false);

    ubahButton.setEnabled(false);
    hapusButton.setEnabled(false);
    baruButton.setEnabled(false);
    simpanButton.setEnabled(true);           // simpan dan batal aktifkan
    batalButton.setEnabled(true);
}
```

```

        npmField.requestFocus();
        status=1;
    }
    // baru

```

6.2. Gunakan tombol simpan untuk menyimpan data yang diinputkan. Pada saat tombol simpan ditekan maka akan dilakukan event berikut:

Keadaan tombol button disusun lagi (enabled/disabled)

Ambil data yang diinput dalam form dengan membuat fungsi getIsianKeArray

Periksa apakah ada data kosong / tidak

Jika tidak ada yang kosong periksa apakah status data baru atau edit

Lakukan penyimpanan berdasarkan status penyimpanan tersebut.

```

private void getIsianKeArray() {
    String jenkel="";
    if (priaRadio.isSelected())
        jenkel="Laki-laki";
    else
        jenkel="Perempuan";

    data[0]=npmField.getText();
    data[1]=namaField.getText();
    data[2]=jenkel;
    data[3]=prodiComboBox.getSelectedItem().toString();
}

```

//Catatan : Bisa juga dengan mengirim data dari form ke Class Mahasiswa

```

private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
    int barisSimpan,row;
    ubahButton.setEnabled(true);
    hapusButton.setEnabled(true);
    baruButton.setEnabled(true);
    simpanButton.setEnabled(false);
    batalButton.setEnabled(false);

    getIsianKeArray();

    if ((data[0].isEmpty()) || (data[1].isEmpty()) || (data[2].isEmpty())
||
        (data[3].isEmpty())){
        JOptionPane.showMessageDialog(null,"Masih ada nilai yang kosong, "
            + "silahkan dilengkapi");
        npmField.requestFocus();
        return;
    }
    else {
        if (status==1){
            barisSimpan=counter++;
            // baru

        }
        else {
            // status =2 ubah
            row=mahasiswaTable.getSelectedRow();
            barisSimpan=row;

```

```

    }

    for (int i=0;i<4;i++){        // i menyatakan kolom
        mahasiswaTable.setValueAt(data[i], barisSimpan, i);
    }
}
counterLabel.setText("Counter Data :"+Integer.toString(counter));
setDisabledKomponen();
bersihKomponen();
mahasiswaTable.setEnabled(true);
ubahButton.setEnabled(true);
hapusButton.setEnabled(true);
baruButton.setEnabled(true);
simpanButton.setEnabled(false);
batalButton.setEnabled(false);
}

```

Penjelasan :

```

row=mahasiswaTable.getSelectedRow();
    // ambil posisi baris aktif dari jTable yang dipilih
mahasiswaTable.setValueAt(data, brs, klm);
    // isikan ke jTable yaitu data pada posisi baris brs dan kolom klm

```

NPM	Nama	Jenis Kelamin	Program Studi
14003001	Parto Andre	Laki-laki	Manajemen Informatika

Gambar 15.4 Operasi Menambah data ke dalam Tabel

7. Cara Menampilkan Data dari Jtable kedalam Form di Netbeans:

Dibuatkan dulu fungsi tampil untuk menampilkan data dari tabel dengan posisi record aktif ke form isian.

Posisi record aktif disimpan pada variabel row

```

public void tampilRowTable() {
    int row;
    row=mahasiswaTable.getSelectedRow();

    npmField.setText(mahasiswaTable.getValueAt(row, 0).toString());
    namaField.setText(mahasiswaTable.getValueAt(row, 1).toString());
}

```

```

        if (mahasiswaTable.getValueAt(row, 2).toString().equals("Laki-laki")) {
            priaRadio.setSelected(true);
            wanitaRadio.setSelected(false);
        }
        else {
            priaRadio.setSelected(false);
            wanitaRadio.setSelected(true);
        }

        prodiComboBox.setSelectedItem(mahasiswaTable.getValueAt(row, 3).toString());
    }

```

Fungsi ini akan dipanggil melalui event penekanan mouse dua kali (double click pada record) dengan bentuk fungsi sbb:

```

private void mahasiswaTableMouseClicked(java.awt.event.MouseEvent evt) {
    try {
        if (evt.getClickCount() == 2) {
            setDisabledKomponen();
            tampilRowTable();
        }
    }
    catch (Throwable t) {
        JOptionPane.showMessageDialog(null, "Record Kosong... ");
    }
}

```

Gambar 15.5 Event double click pada table

Selain dengan double click, fungsi tampilRowTable ini akan diaktifkan melalui penekanan tombol edit / Ubah atau tombol hapus.

8. Cara mengubah (edit) data dalam JTable :

Pengubahan data dilakukan setelah ada data yang disorot/ ditampilkan. Dengan melakukan penekanan tombol ubahButton, maka data yang telah ditampilkan bisa diubah/diperbaiki.

Proses yang dilakukan adalah menyusun tombol-tombol yang aktif atau tidak aktif (enabled/disabled). Jika data belum ada yang dipilih maka akan ditampilkan pesan error untuk menyorot data yang akan diedit terlebih dahulu.

```
private void ubahButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        tampilRowTable();  
        setEnabledKomponen();  
  
        npmField.setEnabled(false);           //npm tdk bisa di edit  
        namaField.requestFocus();  
  
        mahasiswaTable.setEnabled(false);  
  
        getIsianKeArray();  
        ubahButton.setEnabled(false);  
        hapusButton.setEnabled(false);  
        baruButton.setEnabled(false);  
        simpanButton.setEnabled(true);  
        batalButton.setEnabled(true);  
        status=2;                               // edit  
    }  
    catch (Throwable t){  
        JOptionPane.showMessageDialog(null,"Sorot dulu datanya, "  
            + "silahkan dipilih yg diedit");  
  
        hapusButton.setEnabled(true);  
        baruButton.setEnabled(true);  
        simpanButton.setEnabled(false);  
        batalButton.setEnabled(false);  
    }  
}
```


Data Mahasiswa dan Tabel

Data Induk ...

NPM

Nama

Jenis Kelamin ☒ Laki-laki ☐ Perempuan

Program Studi

Tombol ...

NPM	Nama	Jenis Kelamin	Program Studi
14003001	Parto Andre	Laki-laki	Manajemen Informatika
140002004	Azis Sule	Laki-laki	Manajemen Informatika

Gambar 15.6 Edit Data dalam tabel

Perhatikan bahwa key dari mahasiswa (yaitu NPM) tidak bisa diedit atau dibuat disabled.

Setelah dilakukan pengeditan maka bisa dilakukan proses penyimpanannya (dengan menekan tombol Simpan) atau membatalkan perubahan yang dilakukan (dengan menekan tombol Batal).

Data Mahasiswa dan Tabel

Data Induk ...

NPM

Nama

Jenis Kelamin ☒ Laki-laki ☐ Perempuan

Program Studi

Tombol ...

NPM	Nama	Jenis Kelamin	Program Studi
14003001	Parto Andre Taulani	Laki-laki	Teknik Informatika
140002004	Azis Sule	Laki-laki	Manajemen Informatika

Gambar 15.7 Hasil perubahan edit data

```
private void batalButtonActionPerformed(java.awt.event.ActionEvent evt) {
    ubahButton.setEnabled(true);
}
```

```

        hapusButton.setEnabled(true);
        baruButton.setEnabled(true);
        simpanButton.setEnabled(false);
        batalButton.setEnabled(false);
        bersihKomponen();
        setDisabledKomponen();
        mahasiswaTable.setEnabled(true);
        status=1;
    }

```

9. Cara menghapus (delete) data dalam JTable :

```

private void hapusButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        ubahButton.setEnabled(true);
        hapusButton.setEnabled(true);
        baruButton.setEnabled(true);
        simpanButton.setEnabled(false);
        batalButton.setEnabled(true);
        tampil();
        int tanya=JOptionPane.showConfirmDialog(null, "Yakin dihapus?",
            "Hapus",1);
//atau    int tanya=JOptionPane.showConfirmDialog(null,"Yakin dihapus ?");
        if (tanya==0){
//            mahasiswaTable.remove(row);
//            mahasiswaTable.validate();

            // PENGHAPUSAN DENGAN DATA BASE

            counter--;
        }
    } catch (Throwable t){
        ubahButton.setEnabled(true);
        hapusButton.setEnabled(true);
        baruButton.setEnabled(true);
        simpanButton.setEnabled(false);
        batalButton.setEnabled(true);
        JOptionPane.showMessageDialog(null,"Sorot dulu datanya, "
            + "silahkan dipilih yg dihapus");
    }
}

```