

## **PERTEMUAN 4 : GRAPHICAL USER INTERFACE (GUI)**

### **A. TUJUAN PERKULIAHAN:**

Pada pertemuan ini akan dijelaskan mengenai penggunaan komponen untuk pembuatan table dan Multiple-Document Interface (MDI) serta pengolahannya. Setelah mempelajari materi perkuliahan ini, mahasiswa mampu:

- 4.1 Menggunakan objek GUI
- 4.2 Mengidentifikasi package dan class yang digunakan untuk pembuatan menu utama (parent form) dan sub-menu (child-form)
- 4.3 Mengidentifikasi Overloading konstruktor dan cara pembuatannya
- 4.4 Menggunakan Overloading konstruktor pada program

### **B. URAIAN MATERI**

#### **Tujuan Pembelajaran 4.1:**

Penggunaan objek GUI

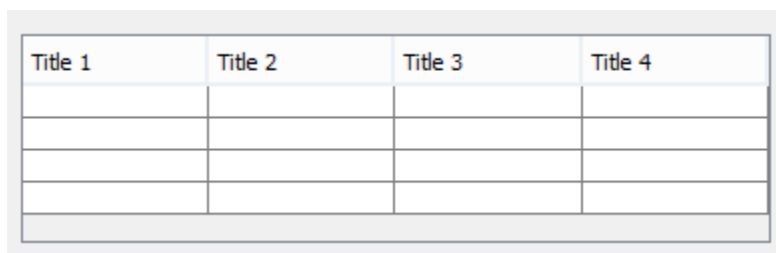
#### **Penggunaan Class JTable**

JTable adalah komponen yang digunakan untuk menampilkan data dalam bentuk table (terdiri dari baris dan kolom).

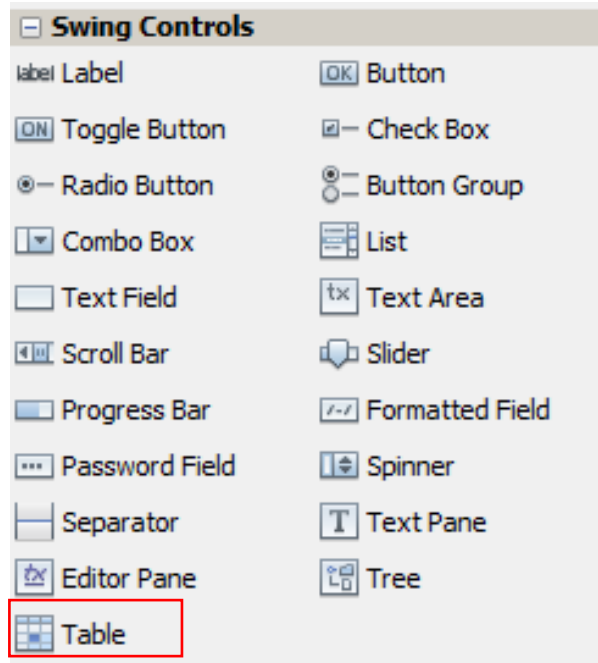
Pada penggunaan JTable ada beberapa class dan metode yang digunakan dan saat membentuk table kita bisa melakukan dengan beberapa cara.

Langkah-langkah:

- Gunakan class objek JTable, dan letakkan pada frame kemudian atur nama variable objeknya



Title 1	Title 2	Title 3	Title 4

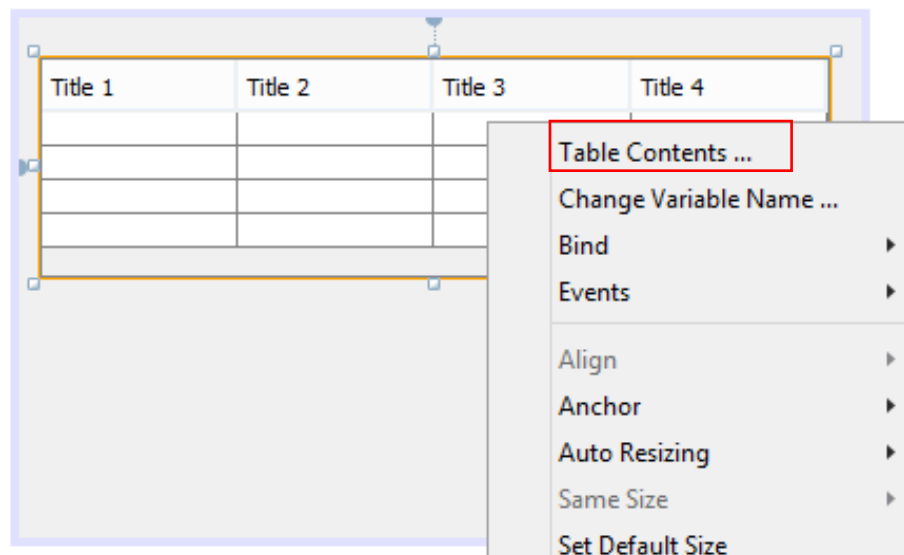


- Membentuk kolom-kolom table

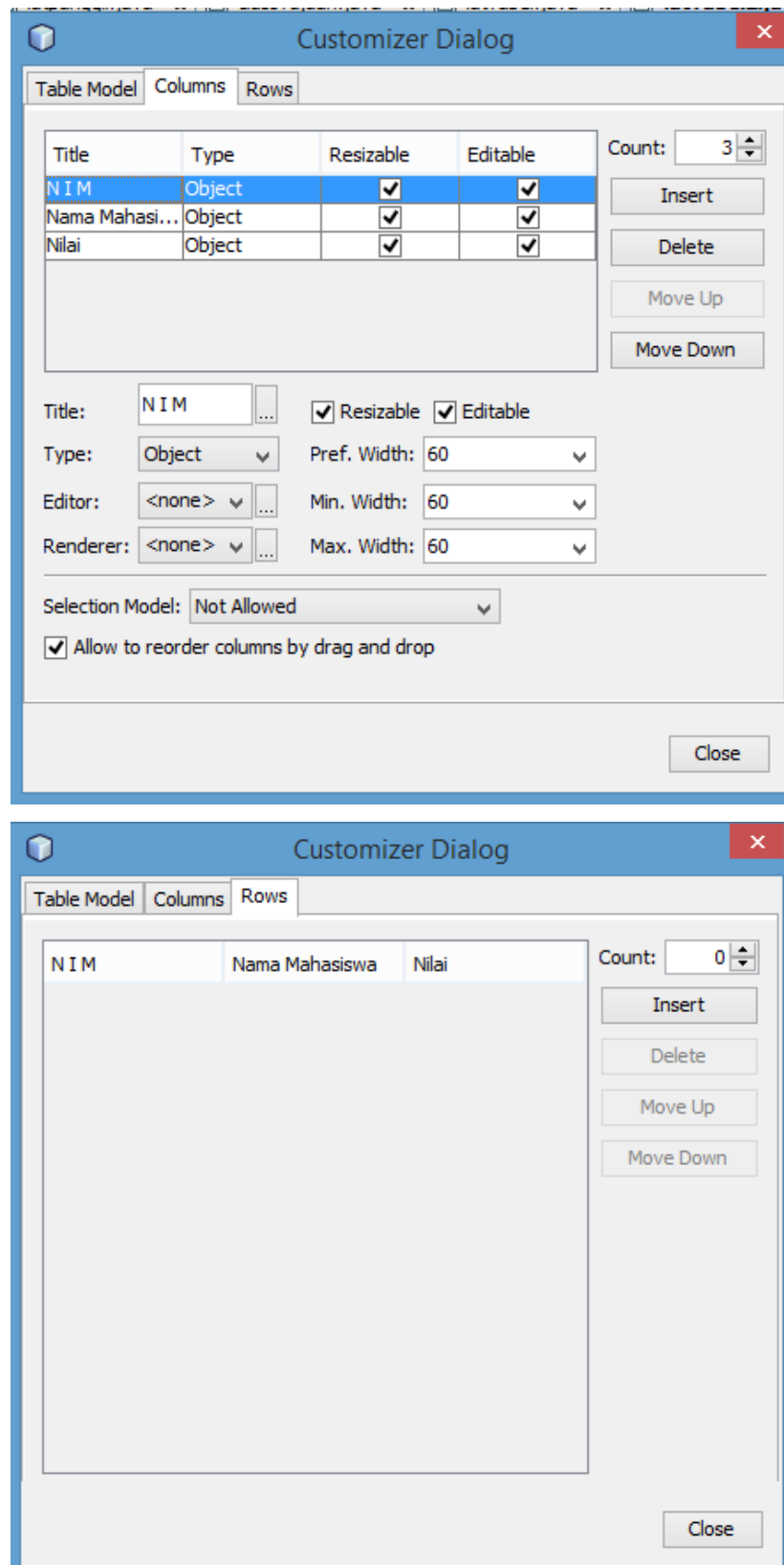
Cara I:

Membentuk kolom melalui properties

- ✓ Klik kanan pada table > Table Content



- ✓ Atur kolom dan baris



Gb. 4.1 Setting tabel

Cara 2:

melalui kode program

- Gunakan kode program untuk membentuk judul-judul kolom table

Sintaks:

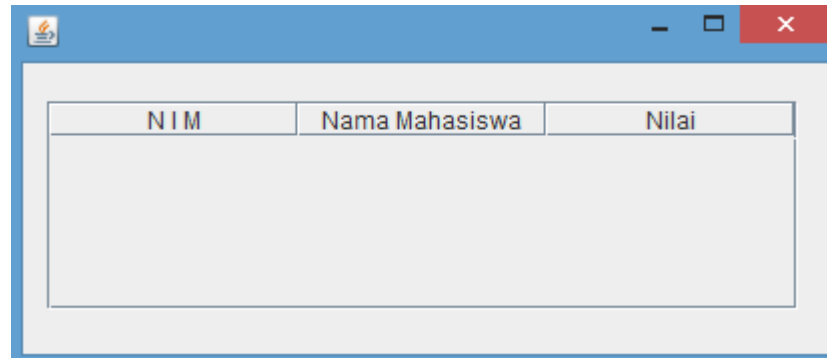
**Penggunaan jTable**

```
private DefaultTableModel nmVarmodel; → deklarasi  
var_model table  
nmVarmodel=new DefaultTableModel();  
namaVarobjTbl.setModel(nmVarmodel);  
nmVarmodel.addColumn("judul kolom1");  
nmVarmodel.addColumn("judul kolom2");  
nmVarmodel.addColumn("judul kolom3");  
nmVarmodel.addColumn("judul kolom4");  
nmVarmodel.addColumn(".....");
```

contoh:

```
package MDI01;  
  
import javax.swing.table.DefaultTableModel;  
  
/**  
 *  
 * @author SRI MULYATI  
 */  
public class latTabel extends javax.swing.JFrame {  
    private DefaultTableModel mod;  
  
    void judulTabel() {  
        mod=new DefaultTableModel();  
        tabel.setModel(mod);  
        mod.addColumn("N I M");  
        mod.addColumn("Nama Mahasiswa");  
        mod.addColumn("Nilai");  
    }  
  
    /** Creates new form latTabel */  
    public latTabel() {  
        initComponents();  
        judulTabel();  
    }  
}
```

Hasil tampilan table



NIM	Nama Mahasiswa	Nilai

### Memasukkan data ke jTable

Sintaks:

**Object[ ] nmVardata=new Object[jml. kolom];** → deklarasi var. data

*nmVardata* [0]=data1;

*nmVardata* [1]=data2;

*nmVardata* [2]=data3;

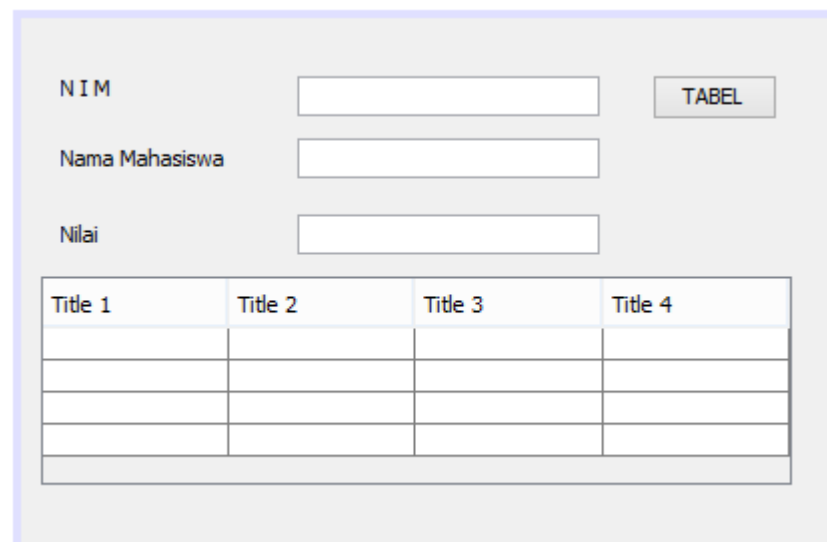
*nmVardata* [3]=data4;

.....

*nmVarmodel.addRow(nmVardata);*

**Object[ ] nmVardata={ data1, data2,  
data3, data4, ...};**

contoh:



NIM

Nama Mahasiswa

Nilai

TABEL

Title 1	Title 2	Title 3	Title 4

**Jika menggunakan cara 1:**

(tanpa membuat kode untuk judul kolom)

```

package MDI01;

import javax.swing.table.DefaultTableModel;

/**
 *
 * @author SRI MULYATI
 */
public class latTabel3 extends javax.swing.JFrame {
    private DefaultTableModel mod;

    /** Creates new form latTabel */
    public latTabel3() {
        initComponents();
        mod=(DefaultTableModel) tabel.getModel();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        Object data[] ={nimTxt.getText(),namaTxt.getText(),nilTxt.getText()};
        mod.addRow(data);
    }

```

**Jika menggunakan cara 2:**

```

package MDI01;

import javax.swing.table.DefaultTableModel;

/**
 *
 * @author SRI MULYATI
 */
public class latTabel extends javax.swing.JFrame {
    private DefaultTableModel mod;
    Object[ ] data=new Object[3];

```

```

    /** Creates new form latTabel */
    public latTabel() {
        initComponents();
        judulTabel();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        data [0]=nimTxt.getText();
        data [1]=namaTxt.getText();
        data [2]=nilTxt.getText();
        mod.addRow(data);
    }

```

#### Tujuan Pembelajaran 4.2:

Penulisan source code sesuai event yang dipilih

#### Membuat menu utama (parent-form)

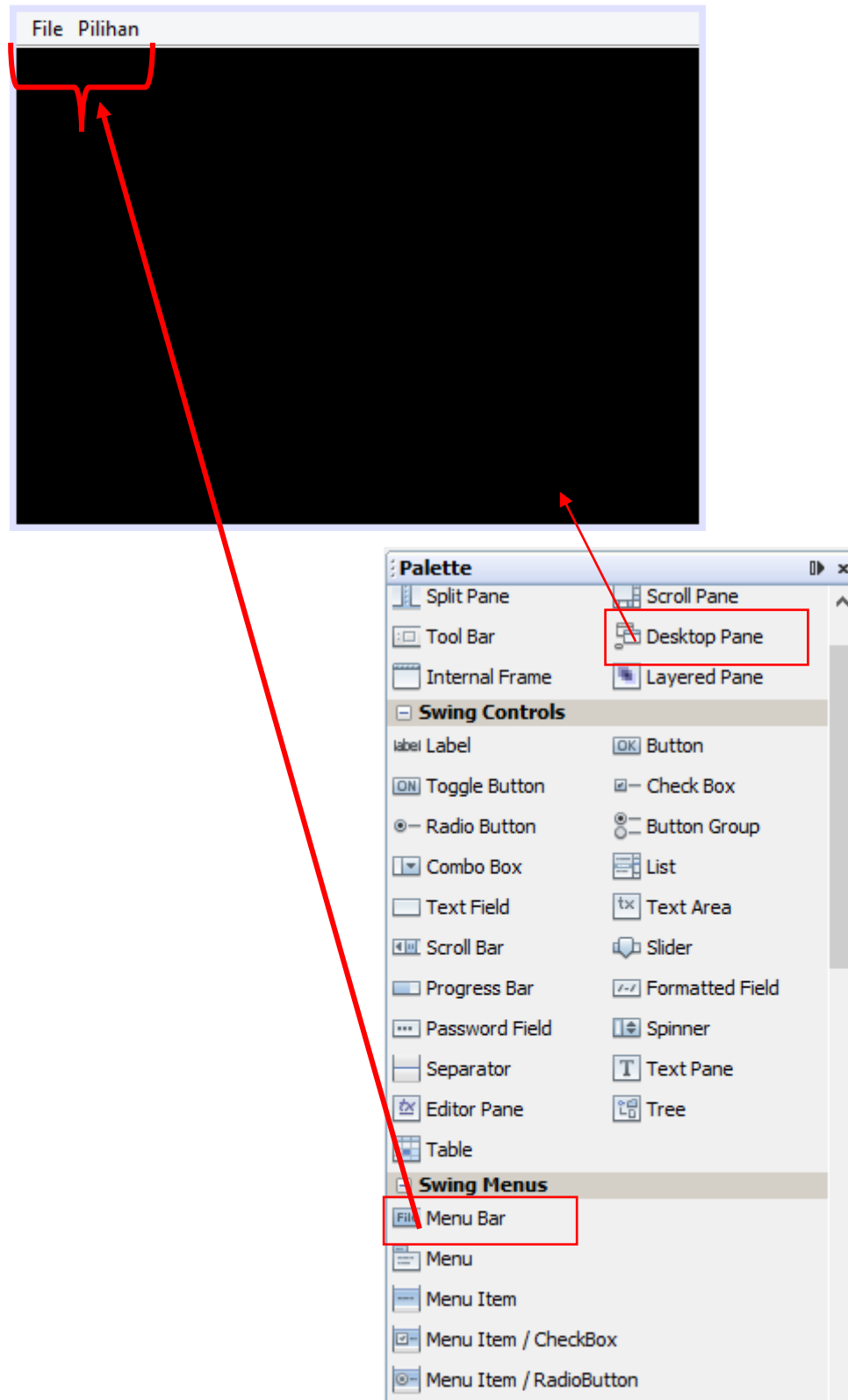
Cara membuat frame untuk menu utama sama dengan seperti biasa membuat desain interface sebelumnya.

##### a. Buat proyek baru

1. Dari menu, pilih menu 'File' > 'NewProject'. Akan keluar kotak dialog New Project.
2. pilih Java>Java Application'.
3. Tentukan nama dan lokasi project.
4. Hilangkan tanda cek pada pilihan 'Create Main Class'. Klik 'Finish'.

##### b. Menambahkan Main Menu (Parent Form/MDI Form)

- tambahkan frame : 'New' > **JFrameForm**
- Tentukan nama class dan nama package
- Tambahkan menu bar (Menu-menu pada menu bar)
- Pada Frame : tambahkan Desktop Panel (JDesktopPane)

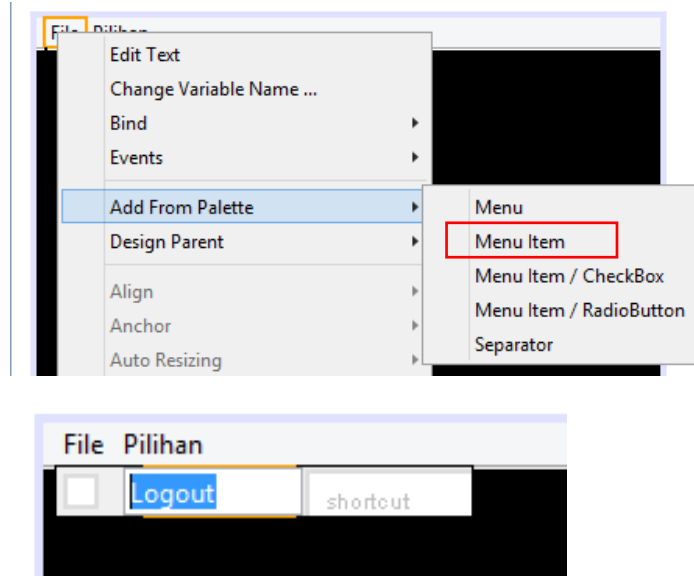




## Menambahkan menu pilihan

Caranya:

- Klik kanan pada menu bar >



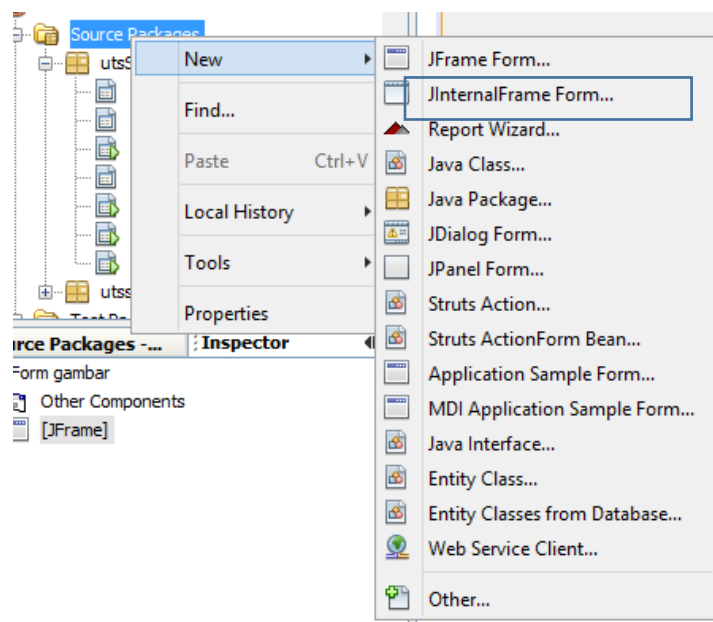
- **Membuat sub-frame (child-form)**

1. Buat frame Internal : klik kanan Source Package > **New >**

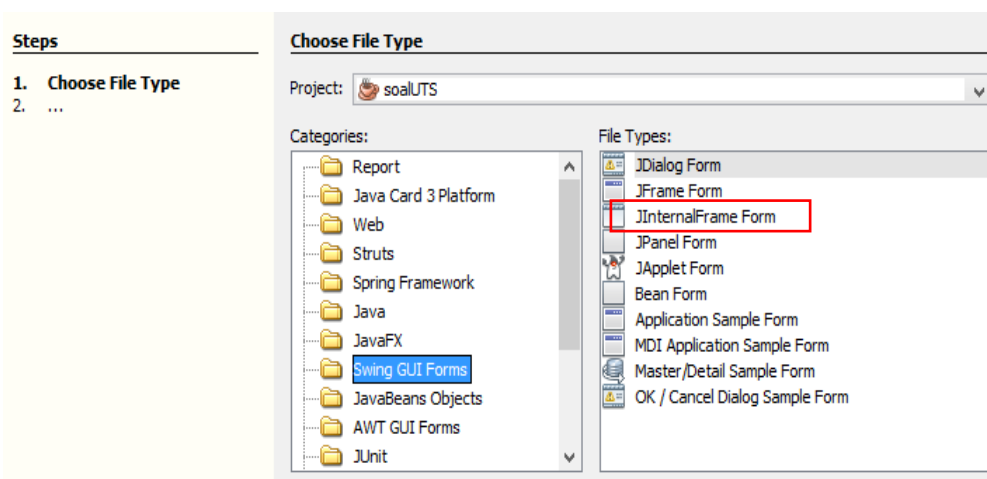
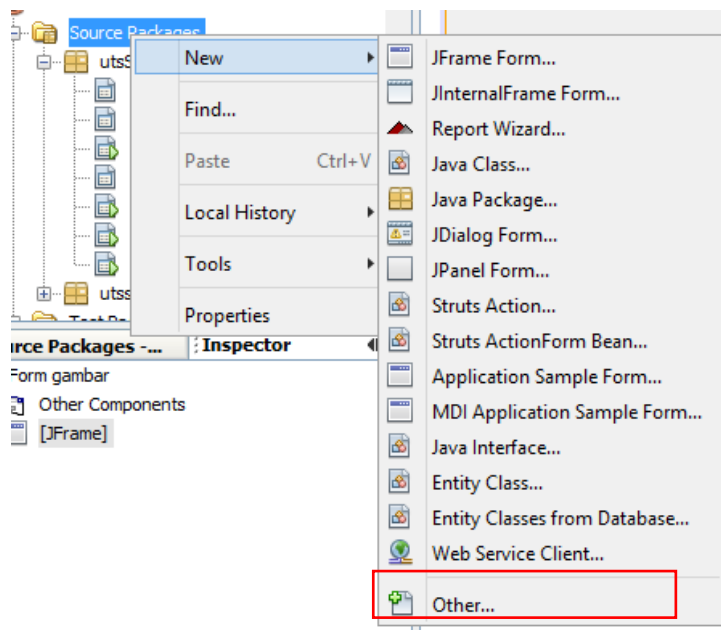
### **JInternalFrame Form**

Tentukan nama class

2. Desain interface sesuai keperluan



Jika pilihan **JInternalFrame Form** tidak ada, maka:



### Cara memanggil/menampilkan frame

1. Memanggil antar Frame (JFrame Form)

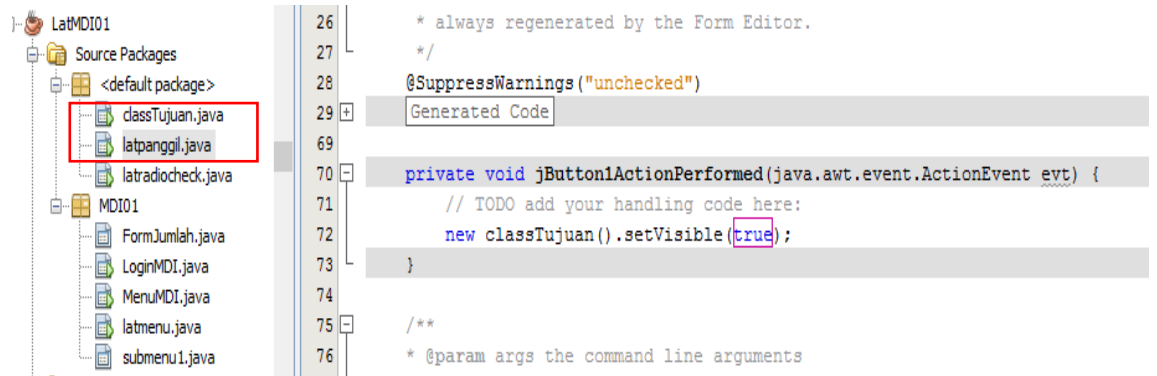
```
new namaClassFrame().setVisible(true);
```

atau

```
namaClassFrame namaVar=new namaClassFrame ();  
namaVar.setVisible(true);
```

Nama class yang  
akan ditampilkan

contoh:



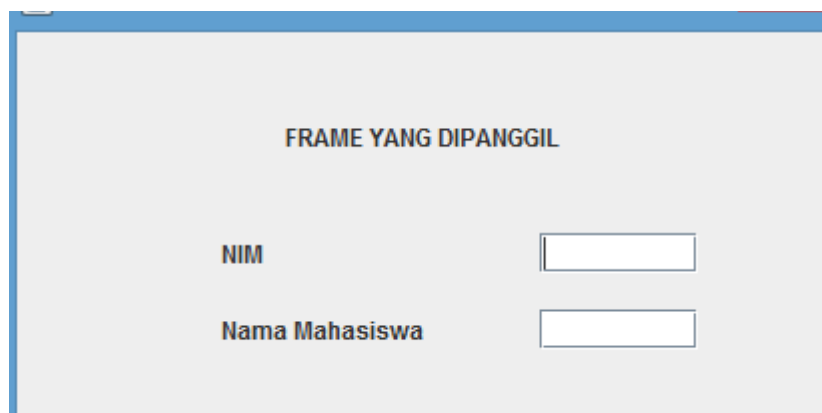
Atau

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // new classTujuan().setVisible(true);

    classTujuan tampil= new classTujuan();
    tampil.setVisible(true);
}

```



2. Memanggil sub-form (JInternalFrame Form) dan ditampilkan ke menu utama (JFrame Form) di DesktopPane

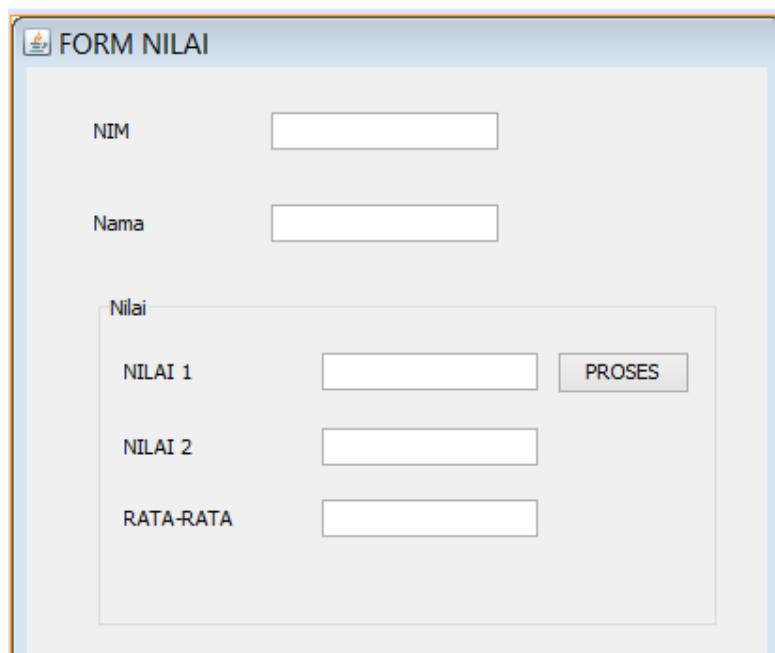
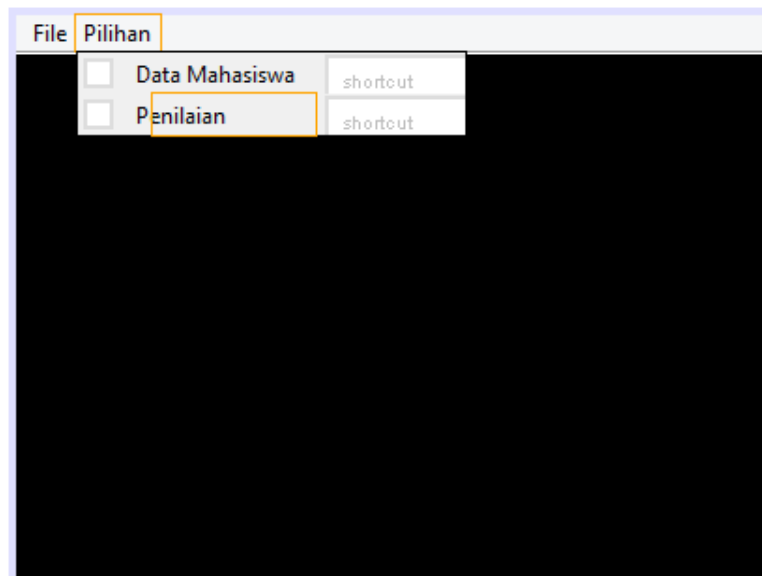
Penulisan:

```
Namaclass nmVar = new namaclass();
```

```
nmVarDeskPane.add (nmVar);
```

```
nmVar.setVisible (true);
```

**CONTOH:** menampilkan class penilaian di menu utama



```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    penilaian sub=new penilaian();
    desk.add(sub);
    sub.setVisible(true);
}
```

### Tujuan Pembelajaran 4.3:

Mengidentifikasi Overloading konstruktor dan cara pembuatannya

### Metode Konstruktor

Metode konstruktor atau biasa disebut konstruktor adalah metode yang dapat digunakan untuk memberikan nilai awal saat objek dibuat/diciptakan. Metode ini akan dipanggil secara otomatis saat class dijalankan (Run).

### Overloading

Overloading terhadap konstruktor merupakan suatu mekanisme pembuatan konstruktor yang memiliki bentuk lebih dari satu.

*Overloading konstruktor* mengijinkan sebuah *method konstruktor* dengan nama yang sama namun memiliki parameter yang berbeda sehingga mempunyai implementasi dan *return value* yang berbeda pula. Daripada memberikan nama yang berbeda pada setiap pembuatan *method konstruktor*, *overloading konstruktor* dapat digunakan pada operasi yang sama namun berbeda dalam implementasinya.

### Pembuatan Overloading Konstruktor

#### Overloading Konstruktor

Overloading konstruktor dapat digunakan untuk pengiriman data antar frame .

Langkah-langkah:

#### I. Mempersiapkan data yang dikirim (class pengirim data)

.....

`nmClasstujuan nmVar = new nmClasstujuan (par1, par2, ...);` → nilai yang akan dikirim

.....

## II. Menampilkan data yang dikirim (frame tujuan)

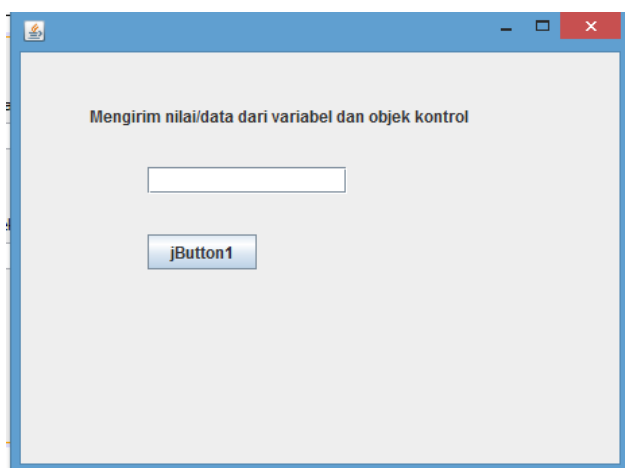
Membuat overloading konstruktor

```
public nmClasstujuan (tipedata nmPar1, tipedata nmPar2, ...){  
    initComponents();  
    .....(nmPar); → menampilkan nilai parameter  
    this.nmPar = nmPar; → karena terdapat objek control yg sama  
    ..... (this.nmPar.metode()); → parameter objek control  
}
```

Menggunakan overloading konstruktor pada program

Contoh:

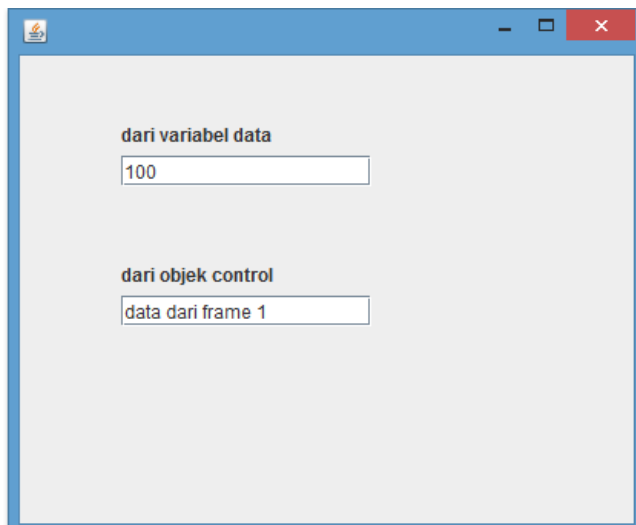
```
public class frame1 extends javax.swing.JFrame {  
    private void  
    jButton1ActionPerformed(java.awt.event.ActionEvent  
    evt) {  
        String nil="100";  
        frame3 obj=new frame3(jTextField1,nil);  
        obj.setVisible(true);  
    }  
}
```



## **Class penerima dan menampilkan data**

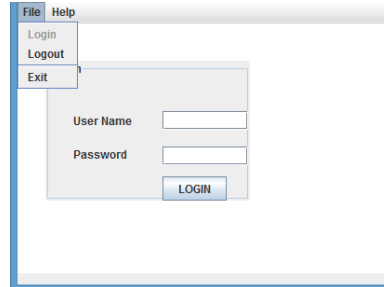
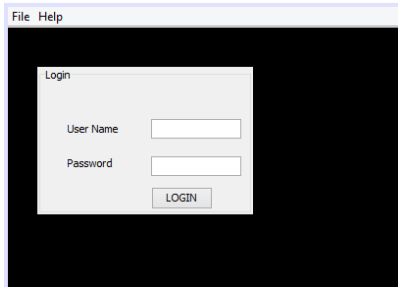
```
public class frame3 extends javax.swing.JFrame {  
    JTextField parameter;  
    /** Creates new form frame3 */  
    public frame3() {  
        initComponents();  
    }  
}
```

```
    public frame3(JTextField parameter, String par2){  
        initComponents();  
        data1.setText(par2);  
        this.parameter=parameter;  
  
        jTextField1.setText(this.parameter.getText());  
    }  
}
```



### C. LATIHAN SOAL

Buat desain dan pengolahannya untuk membuat aplikasi dengan ketentuan seperti di bawah ini!



- Tampilan awal program (run file): langsung tampil login pada menu utama dan menu login tidak aktif.
- Jika login berhasil akan tampil frame pemesanan di menu utama

Pilihan **Paket 1**: nasi putih, nasi goreng, nasi capcay

Pilihan **paket 2**: ayam penyet, ayam bakar, ayam goreng

Pilihan **paket 3**: aqua, jus alpukat, jus jeruk

Harga ditentukan sendiri!

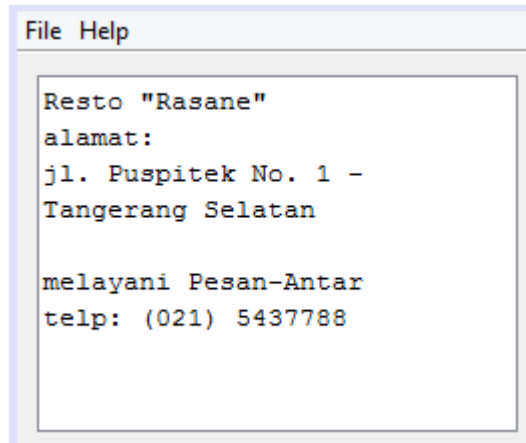
- Jika selesai pesan dan menekan tombol bayar akan tampil form pembayaran

Pilih **logout**: akan tampil login seperti menu awal dan logout tidak aktif

Pilih **exit**: akan keluar dari program



Pilih **Help**: akan muncul informasi alamat dan cara pesan



#### D. DAFTAR PUSTAKA

Budiharto, W. (2004). *Pemrograman Web Menggunakan J2EE*. Jakarta:

Elxmedia Komputindo.

JENI, T. P. (2007). JENI 1-6.

Wijono, S. H., Suharto, B. H., & Wijono, M. S. (2006). *Pemrograman Java Servlet dan JSP dengan Netbeans*. Yogyakarta: C.V ANDI OFFSET.