

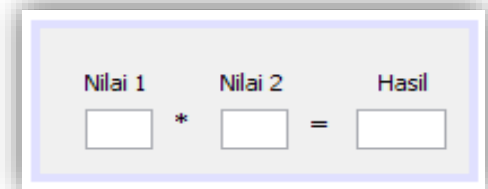
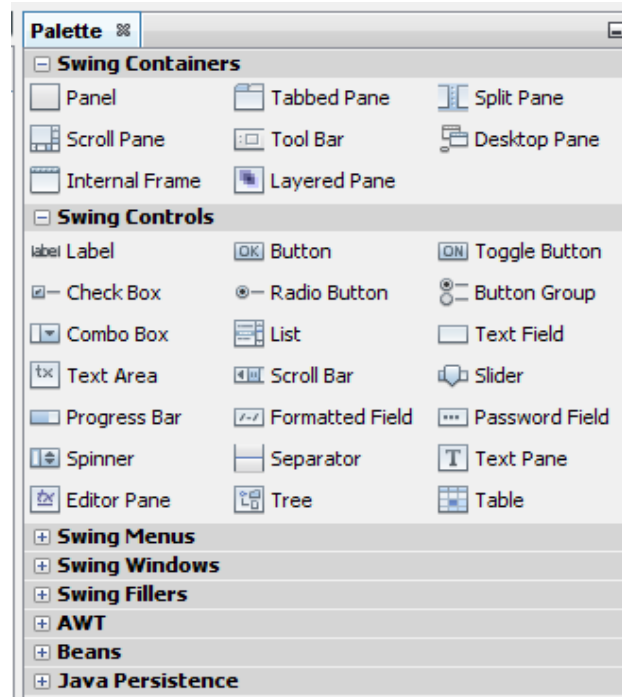
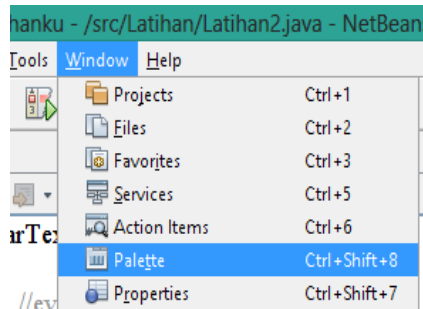
PERTEMUAN 09

Membuat Form dengan JFrame Form dan Object Swing

Contoh Penerapan

Netbeans memiliki taskpane palette dimana didalamnya terdapat objek-objek yang kita butuhkan untuk pembuatan form Java. Untuk menampilkan palette di Netbeans dapat memilih menu window → Palette atau dapat menekan Ctrl+Shift+8.

Contoh JFrame Form



Penjelasan Listing Program

Ada dua jenis peng-konversi-an dalam java. `Integer.parseInt(tNilai1.getText())`, artinya nilai `tNilai1` diambil dan dikonversi ke tipe data `Integer`, hal ini dapat juga disebut sebagai persiapan kalkulasi. Sedangkan `tHasil.setText(Integer.toString(c))`, memiliki makna bahwa `TextField tHasil` akan diberikan nilai yang diambil dari variable `c` (variable penampung) yang memiliki tipe data `Integer` dan dengan method **`Integer.toString`** nilai tersebut diubah mejadi `String` agar dapat diterima oleh `TextField tHasil`.

Penjelasan Listing Program(lanjutan)

“int a” merupakan pendeklarasian variable secara langsung di dalam event. Pengetikkan tipe data dalam membuat program java memiliki beberapa perbedaan. Sebagai contoh, untuk deklarasi variable tipe data integer menjadi “int a”, namun pada saat proses konversi tipe data, integer diketik “Integer” dengan huruf kapital “I”. Terkecuali untuk tipe data String, pengetikannya harus “String” bukan “string”, baik itu dalam pendeklarasian variable maupun konversi tipe data.

Membuat Form dan Implementasi OOP

Pengenalan Class

Java merupakan salah satu bahasa pemrograman berbasis object dan memiliki beberapa karakteristik seperti yang sudah dibahas di bab sebelumnya. Pada saat anda membuat form, dengan sendirinya main class utama akan terbentuk. Dengan begitu kita hanya perlu membuat sub class-nya untuk menerapkan bahasa pemrograman yang berbasis object. Biasanya untuk program java yang kompleks, programmer membutuhkan beberapa sub class. Sub class tersebut dapat membantu programmer dalam mengelompokkan proses yang akan terjadi di main class.

Pengenalan Class(lanjutan)

```
import javax.swing.*;
```

Import package

```
public class NilaiRata extends javax.swing.JFrame {
```

```
Ratarata rata = new Ratarata();
```

```
/**
```

```
 * Creates new form NilaiRata
```

```
 */
```

```
public NilaiRata() {
```

```
    initComponents();
```

```
    nonAktif();
```

```
}
```

Instansiasi object dari
class Ratarata

Method nonAktif() dijalankan
saat form di run

```
void Bersih() {
```

```
    tNilai1.setText("");
```

```
    tNilai2.setText("");
```

```
    tRata.setText("");
```

```
}
```

Method
tambahan
dalam form

```
void Aktif() {
```

```
    tNilai1.setEnabled(true);
```

```
    tNilai2.setEnabled(true);
```

```
    tRata.setEnabled(true);
```

```
}
```

Contoh source code pada form yang menerapkan Obejct Oriented Programming. salah satu ciri dari OOP adalah adanya instansiasi object, dimana method yang terdapat di subclass dapat digunakan di form.

Satu main class bisa saja membutuhkan dua atau lebih sub class dalam pembuatan dan prosesnya. Berikut adalah contoh dimana main class dan subclass dibuat menjadi satu kesatuan program. Langkah-langkah:

1. Membuat JFrame Form (main class) dan beri nama NilaiRata.java. Program sederhana ini akan menghitung nilai rata-rata dari nilai yang diinput.



Jpanel → di panel Swing Container

Properties :

Border→title border→title
:Pilih tombol

Properties object pada form:

Object	Edit Text	Variable Name
JLabel1	Nilai 1	-
JLabel2	Nilai 2	-
JLabel3	Rata-rata	-
TextField1	-	tNilai1
TextField2	-	tNilai2
TextField3	-	tRata
JBUTTON1	Hitung	bHitung
JBUTTON2	Bersih	bBersih
JBUTTON3	Tutup	bTutup