

LAPORAN TUGAS AKHIR APLIKASI KALKULATOR BERBASIS JAVA

Diajukan untuk memenuhi salah satu tugas dari matakuliah Pemrograman III



Oleh :
Nama : Ai Titin
Npm : 1142228

**STMIK AMIK BANDUNG
BANDUNG
2012**

BAB I PENDAHULUAN

Latar Belakang

Sesuai dengan perkembangan zaman, teknologi dan komunikasi semakin berkembang, salah satunya perkembangan java API (Application Programming Interface). Kebanyakan *programmer* membuat aplikasi dengan menggunakan *Application Programming Interface*(API). Dalam API itu terdapat fungsi-fungsi/perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system calls* sesuai dengan sistem operasinya.

Application Programming Interface (API) adalah kumpulan dari pustaka class atau komponen atau library, yang sudah disediakan oleh Sistem Operasi, yang berupa kumpulan perintah yang membentuk sebuah komponen, sehingga akan membantu para programmer dalam membangun sebuah aplikasi.

Keuntungan memprogram dengan menggunakan API adalah:

- **Portabilitas.** Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter- *install* API tersebut.
- **Lebih Mudah Dimengerti.** API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal editing dan pengembangan.

System call interface ini berfungsi sebagai penghubung antara API dan *system call* yang dimengerti oleh sistem operasi. *System call interface* ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil *system calls* yang diperlukan.

Untuk membuka suatu *file* tersebut *user* menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari *user* tersebut diterjemahkan dulu oleh program menjadi perintah *open()*. Perintah *open()* ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan *user* dapat dimengerti oleh sistem operasi, maka perintah *open()* tadi diterjemahkan ke dalam bentuk *system call* oleh *system call interface*. Implementasi perintah *open()* tadi bisa bermacam-macam tergantung dari sistem operasi yang kita gunakan.

Maksud dan Tujuan

Application Programming Interface(API) adalah cara program dan user berkomunikasi. *Application Programming Interface*(API) merupakan setiap program yang dirancang untuk melakukan fungsi yang khusus atau spesifik untuk pengguna atau untuk kasus-kasus tertentu, untuk program aplikasi lainnya sehingga maksud

dan tujuan dari para *programmer* menggunakan API adalah untuk memudahkan mereka dalam membuat program aplikasi.

Batasan Masalah

Membahas sejauh mana user membuat aplikasi dengan menggunakan API (Application Programming Interface) dan pengenalan lebih dalam mengenai Application Programming Interface.

Sistematika Penulisan Laporan

Laporan tugas akhir ini dibuat dengan sistem penulisan sebagai berikut:

BAB I PENDAHULUAN

BAB II LANDASAN TEORI

BAB III ANALISIS

BAB IV IMPLEMENTASI

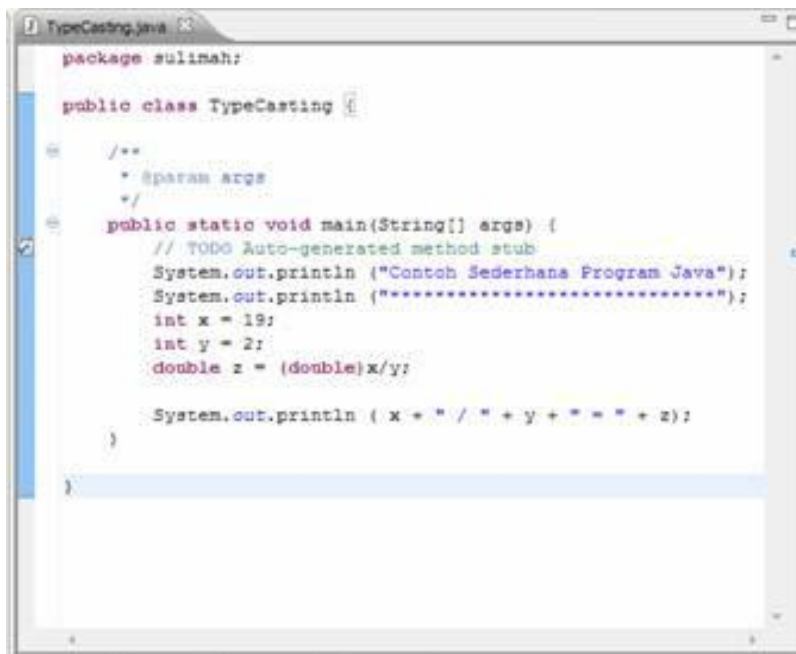
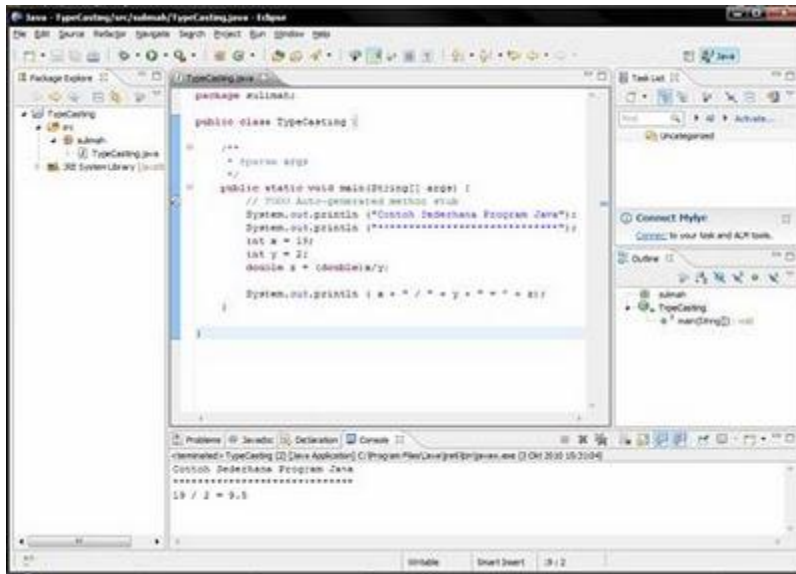
BAB II LANDASAN TEORI

JAVA Application Programming Interface (API)

Java adalah bahasa pemrograman yang dapat dijalankan di komputer atau di telepon genggam. Java juga merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek (*OOP*) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada *OOP*, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi singkat *OOP* adalah mengorganisasikan program sebagai kumpulan komponen yang disebut dengan objek. Program java tersusun dari bagian-bagian yang disebut kelas. Kelas itu sendiri terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya.

Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (*API*). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada Java *API*. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Sedangkan pengertian *API* itu sendiri ialah seperangkat fungsi standar yang disediakan oleh OS atau Bahasa. Dalam Java, *API* dimasukkan ke dalam *package-package* yang sesuai dengan fungsinya.

Beberapa fungsi java yaitu java merupakan bahasa yang sederhana. Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori.



Penjelasan Program :

System.out.print () dan System.out.println () merupakan dua instruksi dasar untuk menampilkan informasi. Secara umum untuk menampilkan informasi ke layar

digunakan perintah `System.out` () atau `System.out.println` (). Perbedaannya terletak pada posisi kursor setelah informasi tercetak di layar. Jika kita menggunakan `System.out.print` () maka posisi kursor ada di posisi akhir setelah informasi yang ditampilkan, sedangkan kita menggunakan `System.out.println` () posisi kursor ada pada baris berikutnya.

Gunakan pasangan kurung buka "(" dan kurung tutup ")" pada ekspresi matematis seperti pada program diatas. Pada `System.out.print` () atau `System.out.println` (). Beberapa informasi yang akan ditampilkan dipisahkan dengan karakter (+), karakter penjumlahan (+) berarti penggabungan string. Berbeda dengan operator aritmatika dan operator logika. Cara penulisannya harus di ikuti dengan tanda " " sehingga dapat membandingkan nilai variable. Dan jika kita membuat program seperti ini `System.out.println z=x/y;` saja tanpa menambahkan `System.out.println` maka hasilnya adalah "8" tanpa ada penjelasan 16/2. Itu disebabkan karena ekspresi tersebut merupakan informasi pertama yang akan ditampilkan.

Pada program diatas menggunakan tipe data yang diproses tipe data lain, misalnya data string angka diproses sebagai data numerik. Dalam situasi seperti ini diperlukan konversi tipe data yaitu Type-Casting. Type-Casting adalah istilah dimana suatu data 'dipaksa' menjadi tipe data lain secara sementara pada saat akan diproses.

Int atau integer jika dipakai menggunakan tipe data bilangan bulat. Jika bilangannya pecahan maka menggunakan double. Jika karakter berarti menggunakan char, Boolean untuk tipe data logika dan yang terakhir string adalah tipe data karakter.

Variable yang terlibat dalam perhitungan ini, x dan y, bertipe int; sedangkan variable untuk menampung hasil perhitungan z bertipe double. Ada satu aturan bahwa jika variable yang terlibat semuanya bertipe ilangan bulat, hasil perhitungan pun akan bertipe bilangan bulat tidak tergantung pada tipe variable penampungnya. Dengan kata lain diambil bagian integernya saja. Konsekuensinya adalah hasil proses yang terjadi memiliki kesalahan. Jika salah satu atau semua operand yang terlibat dalam ekspresi matematis bernilai pecahan maka hasilnya juga pecahan.

Penulisan `(double) x` menyatakan bahwa dalam proses perhitungan ini nilai x diubah menjadi tipe double. Nilai aktualnya tetap dalam tipe int. perubahan ini hanya bersifat sementara pada saat proses perhitungan dilakukan. Dalam hal ini yang diubah menjadi pecahan hanya variable x, sedangkan variable y tetap.

```
package sulimah;

public class TypeCasting {
    //
    * System args
    */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println ("Contoh Sederhana Program Java");
        System.out.println ("*****");
        int x = 19;
        int y = 2;
        double z = (double)x/y;

        System.out.println ( x + " / " + y + " = " + z);
    }
}
```

```
//awal memulai program java selalu dengan class
//contoh dengan nama TypeCasting
public class TypeCasting {
    //memulai penulisan method selalu menggunakan void
    //method bernama main
    public static void main (String[] args{
        //menulis fungsi untuk mencetak
        System.out.println("Contoh Sederhana Program Java");
        System.out.println("*****");
        //memasukkan deklarasi variable
        int x = 19;
        int y = 2;
        double z = (double)x/y;

        //menulis fungsi untuk mencetak
        System.out.println (x + " / " + y + " = " + z);
    }
}
//akhir dari class TypeCasting
}
```

Dalam program java memiliki bagian-bagian struktur. Adapun bagiannya yaitu class, method dan deklarasi variable sesuai dengan contoh program dibawah, yaitu :

- Class : Class pada java terbentuk atas *atribute* (variabel/ciri-ciri) dan *behaviour* (fungsi). Class mengizinkan Anda dalam mendeklarasikan tipe data baru. Ia dijalankan sebagai *blueprint*, dimana model dari object yang Anda buat berdasarkan pada tipe data baru ini.
- Method : Sebuah method menjelaskan behaviour dari sebuah object. Method juga dikenal sebagai fungsi atau prosedur.
- Variabel : Item yang digunakan data untuk menyimpan pernyataan dari objek.

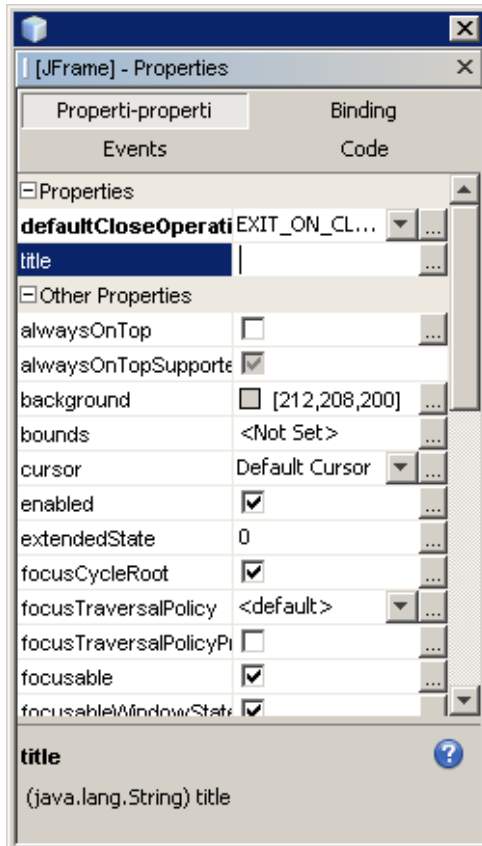
Selain itu terdapat pula bagian struktur lainnya yang ada diluar program dibawah, yaitu :

- Object : Sebuah entiti yang memiliki keadaan, *behaviour* dan identitas yang tugasnya dirumuskan dalam suatu lingkup masalah dengan baik. Inilah instance sebenarnya dari sebuah class. Ini juga dikenal sebagai *instance*. *Instance* dibuat sewaktu Anda meng-*instantiate* class menggunakan kata kunci *new*.
- Atribut : Atribut menunjuk pada elemen data dari sebuah object. Atribut menyimpan informasi tentang object. Dikenal juga sebagai member data, variabel instance, properti atau sebuah field data.
- Konstruktor : Sebuah tipe khusus dari method yang digunakan untuk membuat dan menginisialisasi sebuah object baru. Ingat bahwa konstruktor bukan member (yaitu atribut, method atau inner class dari sebuah object).

Class JFrame()

Merupakan class yang membentuk sebuah frame (form).

Properties



defaultCloseOperation:

memfungsikan tombol2 yang bisa digunakan untuk keluar dari frame in (menutup frame ini).

title:

men-set caption (title bar) dengan sebuah string.

alwaysOnTop:

menjadikan frame ini selalu aktif dimuka jendela lain.

background:

mengubah warna belakang frame.

bounds:

mengatur posisi kiri atas (x,y) serta lebar dan tinggi dari frame ini.

cursor:

mengubah bentuk kursor (mouse) pada saat berada di frame ini.

name:

mengubah nama frame.

resizable:

menjadikan frame dapat berubah ukuran lebar maupun tinggi.

Class JLabel

Class ini dipergunakan untuk menambahkan suatu String pada JFrame, dipergunakan untuk keterangan atau label sebuah komponen.

Beberapa properties pada class ini sama dengan class JFrame, tetapi ada beberapa yang berbeda.

Properties

background:

mengubah warna belakang.

displayedMnemonic:

men-set sebuah karakter menjadi hot-key (alt-karakter) pada saat menggunakan keyboard.

font:

menggunakan font tertentu.

icon:

mengubah bentuk kursor (mouse)

pada saat berada pada class ini.

labelFor:

pada saat menggunakan hot key, memindahkan kursor ke sebuah komponen tertentu.

tooltiptext:

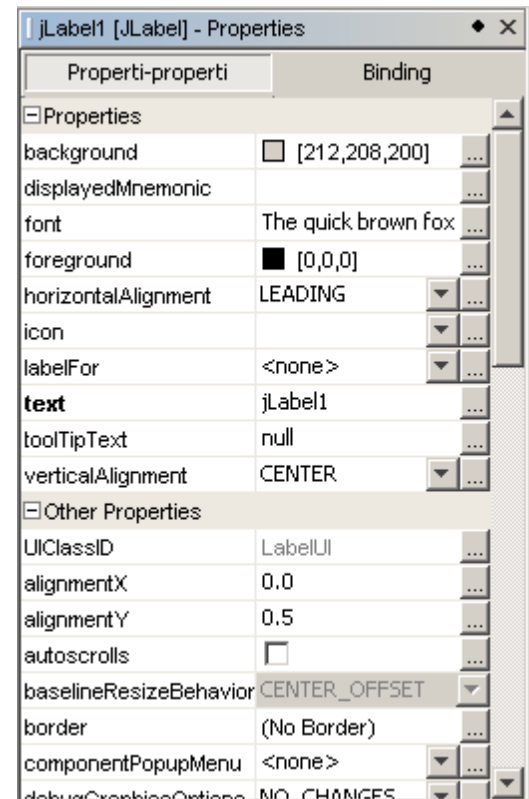
menampilkan string bantuan (hint) pada saat pointer berada pada komponen ini.

verticalAlignment:

posisi string secara vertikal terhadap tinggi (height) class ini.

border:

menggunakan border (kotak luar) tertentu.



Class JTextField

Class ini digunakan untuk memasukan sebuah data (entri data) dimana datanya akan (selalu) ber-tipe string.

Properties:

background:
mengubah warna belakang.

editable:
string didalamnya dapat diubah / tidak dapat diubah.

font:
menggunakan font tertentu.

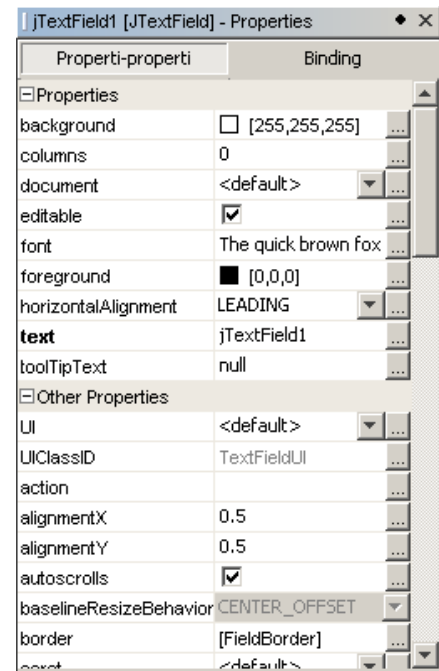
foreground:
warna belakang class ini.

verticalAlignment:
posisi string secara vertikal terhadap tinggi (height) class ini.

text:
isi / nilai dari class ini.

tooltiptext:
menampilkan string bantuan (hint) pada saat pointer berada pada komponen ini.

border:
menggunakan border (kotak luar) tertentu.



Class JButton

Class ini akan membentuk sebuah komponen button (command button), yang digunakan untuk memproses.

Properties

background:

mengubah warna belakang.

font:

menggunakan font tertentu.

foreground:

warna belakang class ini.

Icon:

Bentuk pointer pada saat berada di komponen ini.

Mnemonic:

men-set sebuah karakter menjadi hot-key (alt-karakter) pada saat menggunakan keyboard.

Text:

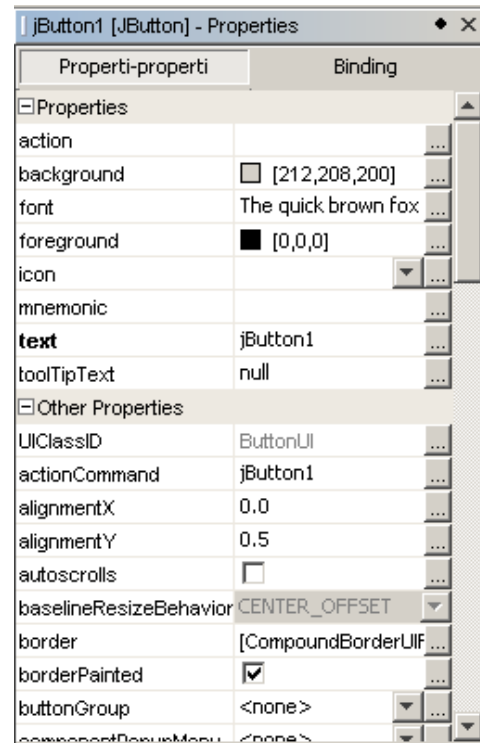
Value dari pada tombol.

tooltipText:

menampilkan string bantuan (hint) pada saat pointer berada pada komponen ini.

border:

menggunakan border (kotak luar) tertentu.



BAB III ANALISIS

Dalam pembuatan aplikasi kalkulator berjava ini, penulis menggunakan beberapa *tools* dan *hardware* sebagai pendukung dalam pembuatannya aplikasi ini. *Tools* yang digunakan adalah :

Hardawe

HARDISK	232,88 GB
RAM	2.00 GB
PROCESSOR	Intel(R) Atom(TM) CPU Z520 @1.33GHz 1.33GHz
VGA	763 MB

Tools

Dalam pembuatan aplikasi ini menggunakan **Netbeans**.

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan platform NetBeans.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak modular yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (*Java archive*) yang memuat kelas-kelas Java untuk berinteraksi dengan *NetBeans Open API* dan file manifestasi yang mengidentifikasinya sebagai modul.

Platform NetBeans

Platform NetBeans adalah framework yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi desktop. Ketika aplikasi berbasis platform NetBeans dijalankan, kelas Main dari platform dieksekusi. Modul-modul yang tersedia ditempatkan di sebuah *registry* di dalam memori, dan tugas *startup*

Platform NetBeans menawarkan layanan-layanan yang umum bagi aplikasi desktop, memungkinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur yang disediakan oleh platform NetBeans:

- Manajemen antarmuka (misal: menu & toolbar)
- Manajemen pengaturan pengguna
- Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
- Manajemen jendela
- *Wizard framework* (mendukung dialog langkah demi langkah)

BAB IV IMPLEMENTASI

Implementasi Percobaan 2

Bentuk frame-nya (Percobaan 2)



Class	Properties	Values
JLabel1	Text	Bilangan ke1
JLabel2	Text	Bilangan ke2
JLabel3	Text	Hasil Jumlah
JTextField1	Text	15
JTextField2	Text	25
JTextField3	Text	40
JButton1	Text	Selesai
JButton2	Text	Reset
JButton3	Text	Hitung

Class Hitung

```
1  /*
2  | * To change this template, choose Tools | Templates
3  | * and open the template in the editor.
4  | */
5
6  /**
7  | *
8  | * @author Ai Titin
9  | */
10 public class Hitung {
11
12 |   int a;
13 |   int b;
14 |   // constructor
15 |   public Hitung() {
16 |   }
17 |   public Hitung(int a, int b) {
18 |     this.a = a;
19 |     this.b = b;
20 |   }
21 |   // method
22 |   void setA(int a) {
23 |     this.a = a;
24 |   }
25 |   void setB(int b) {
26 |     this.b = b;
27 |   }
28 |   int getA() {
29 |     return a;
30 |   }
31 |   int getB() {
32 |     return b;
33 |   }
34 |   int getHasilJumlah() {
35 |     return a + b;
36 |   }
37 |   int getHasilKurang() {
38 |     return a - b;
39 |   }
40 |   int getHasilKali() {
41 |     return a * b;
42 |   }
43 |   double getHasilBagi() {
44 |     return a / b;
45 |   }
46 | }
```

Kode Programnya :

Hitung Hasil Jumlah

```

145 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
146     // TODO add your handling code here:
147     int bil1 = Integer.parseInt(jTextField5.getText());
148     int bil2 = Integer.parseInt(jTextField2.getText());
149     int hasil = bil1 + bil2 ;
150     jTextField3.setText(hasil+"");
151 }

```

Reset

```

138 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
139     // TODO add your handling code here:
140     jTextField5.setText("");
141     jTextField2.setText("");
142     jTextField4.setText("");
143 }

```

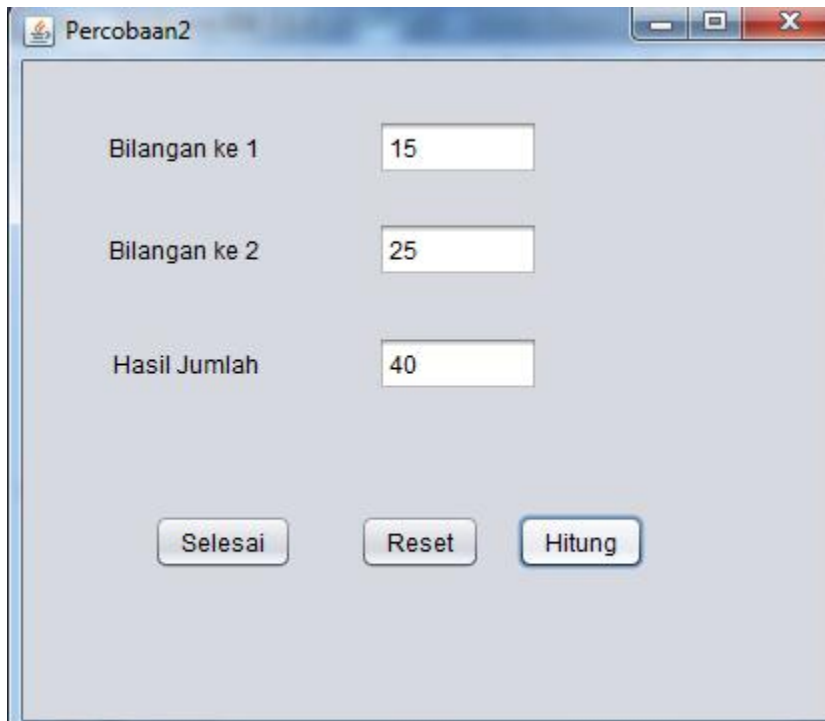
Keluaran

```

153 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
154     // TODO add your handling code here:
155     System.exit(0);
156 }

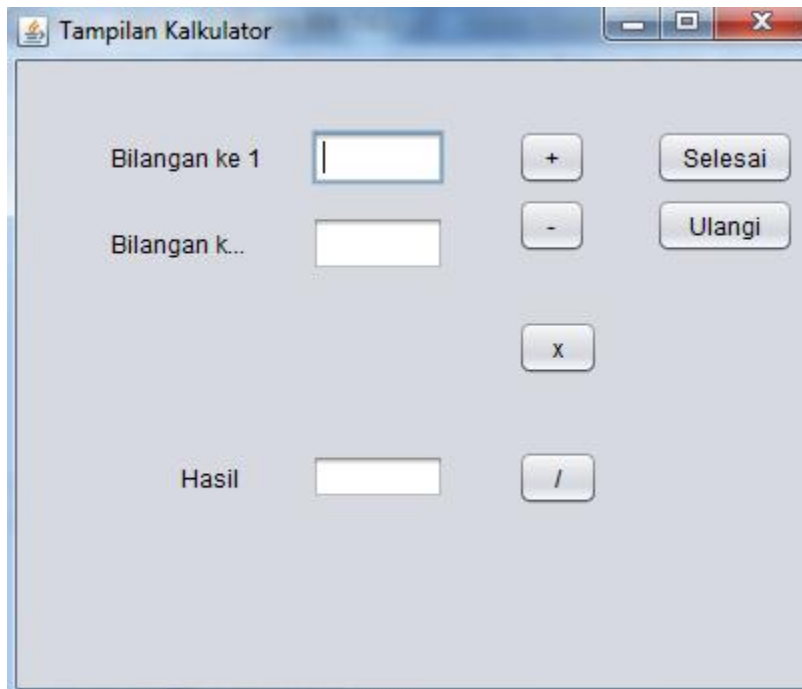
```

Hasil Running setelah menggunakan Rumus :



Tugas Praktikum

Bentuk frame-nya (Latihan 1)



Class	Properties	Values
JLabel1	Text	Bilangan ke1
JLabel2	Text	Bilangan ke2
JLabel3	Text	Hasil
JTextField1	Text	BlandText
JTextField2	Text	BlandText
JTextField3	Text	BlandText
JButton1	Text	+
JButton2	Text	-
JButton3	Text	*
JButton4	Text	/
JButton5	Text	Selesai
JButton6	Text	Ulangi

Kode Programnya :

Penjumlahan

```

178
179 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
180     // TODO add your handling code here:
181     int bil1 = Integer.parseInt(jTextField1.getText());
182     int bil2 = Integer.parseInt(jTextField2.getText());
183     int hasil = bil1 + bil2 ;
184     jTextField3.setText(hasil+"");

```

Pengurangan

```

203 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
204     // TODO add your handling code here:
205     int bil1 = Integer.parseInt(jTextField1.getText());
206     int bil2 = Integer.parseInt(jTextField2.getText());
207     int hasil = bil1 - bil2 ;
208     jTextField3.setText(hasil+"");
209 }

```

Perkalian

```

195 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
196     // TODO add your handling code here:
197     int bil1 = Integer.parseInt(jTextField1.getText());
198     int bil2 = Integer.parseInt(jTextField2.getText());
199     int hasil = bil1 * bil2 ;
200     jTextField3.setText(hasil+"");
201 }

```

Pembagian

```

211 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
212     // TODO add your handling code here:
213     int bil1 = Integer.parseInt(jTextField1.getText());
214     int bil2 = Integer.parseInt(jTextField2.getText());
215     int hasil = bil1 / bil2 ;
216     jTextField3.setText(hasil+"");
217 }

```

Ulangi

```

228 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
229     // TODO add your handling code here:
230     jTextField1.setText("");
231     jTextField2.setText("");
232     jTextField3.setText("");
233 }

```

Keluaran

```

223 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
224     // TODO add your handling code here:
225     System.exit(0);
226 }

```

Hasil Running setelah menggunakan Rumus :



Bentuk frame-nya (Latihan 2)



Kode Programnya:

Rumus Luas dan Keliling Segitiga

```

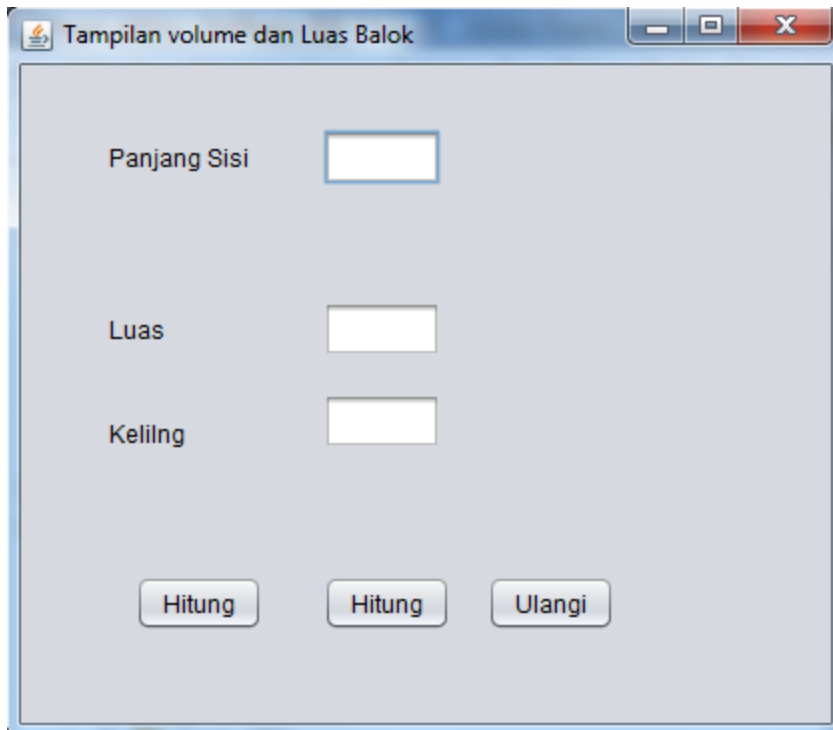
191 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
192     // TODO add your handling code here:
193     int alas=Integer.parseInt(jTextField1.getText());
194     int tinggi=Integer.parseInt(jTextField2.getText());
195     int luas = (alas/2)*tinggi;
196     double s=Math.sqrt(((alas/2)*(alas/2))+(tinggi*tinggi));
197     double keliling=s+s+alas;
198     jTextField3.setText(luas+"");
199     jTextField4.setText(keliling+"");
200 }

```

Hasil Running setelah menggunakan Rumus :

The screenshot shows a Java Swing window titled "Tampilan Luas Keliling Segitiga". The window contains four rows of input fields and buttons. The first row has "Alas" with a text field containing "6" and a "Selesai" button. The second row has "Tinggi" with a text field containing "4" and a "Ulangi" button. The third row has "Luas" with a text field containing "12" and a "Hitung" button. The fourth row has "Keliling" with a text field containing "16.0".

Bentuk frame-nya (Latihan 3)



Kode Programnya:

Luas Kubus

```
146 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
147     // TODO add your handling code here:  
148     int bill=Integer.parseInt(jTextField1.getText());  
149     int luas=6*bill*bill;  
150     jTextField2.setText(luas + " ");  
151 }  
152
```

Volume Kubus

```
161 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
162     // TODO add your handling code here:  
163     int bill=Integer.parseInt(jTextField1.getText());  
164     int keliling=bill*bill*bill;  
165     jTextField3.setText(keliling + " ");  
166 }  
167
```

Hasil Running setelah menggunakan Rumus :

Tampilan volume dan Luas Balok

Panjang Sisi	<input type="text" value="5"/>
Luas	<input type="text" value="150"/>
Keliling	<input type="text" value="125"/>