

MODUL 1**PENGENALAN JAVA, EDITOR DAN SINTAKS JAVA****A. Target Pembelajaran**

1. Menenal bahasa pemrograman JAVA.
2. Mampu membuat kelas sederhana dengan bahasa pemrograman JAVA.
3. Mengerti penggunaan komentar dalam bahasa pemrograman JAVA.
4. Mengetahui cara menggunakan Netbeans.
5. Mengetahui Penggunaan Character Literal.

B. Materi Dasar**1. Pengenalan JAVA**

Java adalah salah satu bahasa pemrograman yang bersifat open source yang merupakan produk dari Sun Microsystem. Bahasa Java adalah bahasa modern yang telah diterima masyarakat komputasi dunia. Hampir semua perusahaan perangkat lunak dan komputer besar mendukung dan mengembangkan aplikasi sistem berbasis Java.

Java adalah salah satu bahasa pemrograman Multiplatform (Bisa berjalan di berbagai macam system operasi) karena pada dasarnya java mempunyai Jre (java runtime environment) atau dapat kita artikan sebagai mesin tersendiri untuk mengeksekusi binary code hasil dari kompilasi program yang telah kita buat, berbeda dengan bahasa pemrograman vb, c++ yang memanfaatkan komponen system dalam Windows untuk mengeksekusi binary code hasil kompilasi program.

Pada Tahun 1996, Sun Microsystem secara resmi merilis versi awal Java yang kemudian terus berkembang hingga muncul JDK 1.1 (Java Development Kit versi 1.1). Perkembangan terus dilakukan hingga muncul versi baru yang disebut Java 2. Perubahan utama antara versi sebelumnya adalah adanya Swing yang merupakan teknologi Graphical User Interface (GUI) yang mampu menghasilkan aplikasi desktop yang benar-benar baik. Untuk mengembangkan aplikasi berbasis java, kita memerlukan Java Development Kit(JDK), saat ini java telah mencapai versi 2 yang dikenal dengan java 2 platform. Java 2 platform, dibagi menjadi tiga kategori yaitu:

1. Java 2 Standard Edition (J2SE)
2. Java 2 Enterprise Edition(J2EE)
3. Java 2 Micro Edition(J2ME)

2. Instalasi JDK

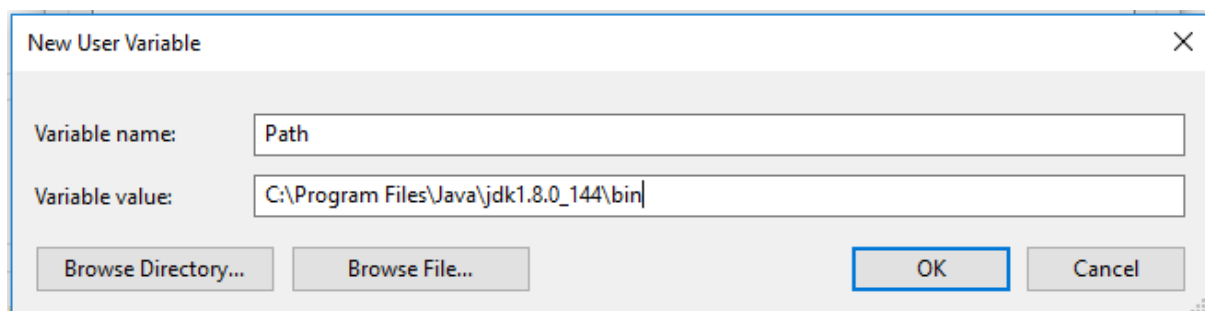
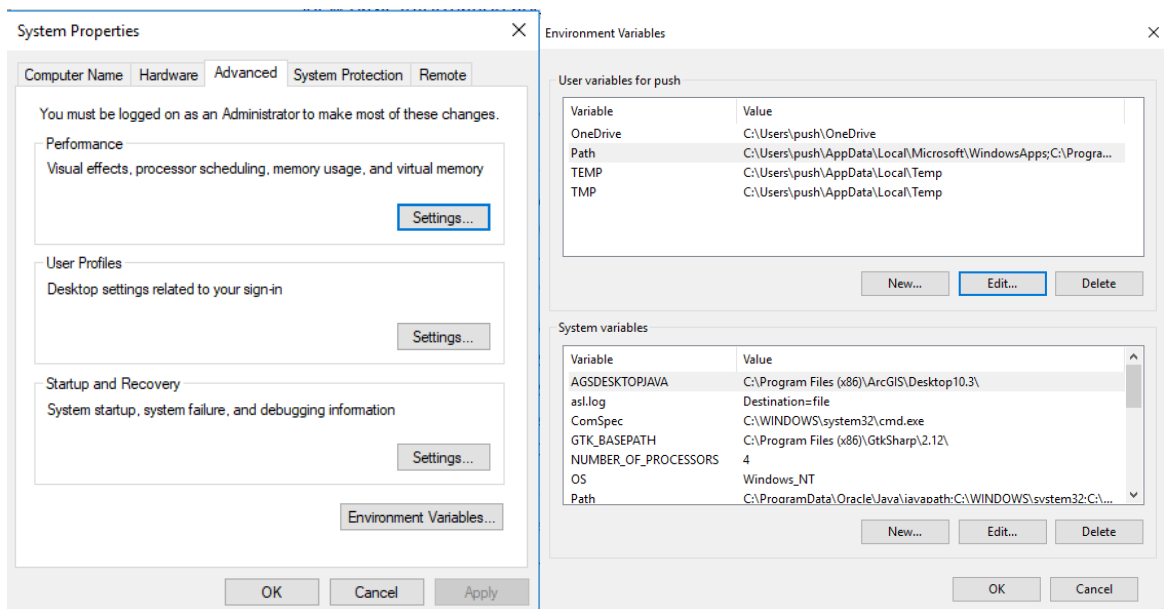
Sebelum anda membuat aplikasi dengan java maka anda harus menginstal software Bahasa pemrograman java. java dapat anda dapatkan dengan cara download di alamat link berikut <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Kemudian setelah anda mendownloadnya maka mulai dengan menginstal, pada dasarnya proses instalasi java sangat

mudah yaitu dengan cara mengklik ganda pada filemaster javakemudian ikuti prosedur yang diperintahkan.

3. Seting path

Untuk dapat dijalankan di semua lingkungan pada suatu operasi system, terlebih dahulu kita harus melakukan seting path, untuk melakukan setting path silahkan perhatikan berikut:

- Klik kanan my Computer kemudian pilih properties
- Kemudian muncul jendela seperti berikut, kemudian klik advanced
- Kemudian klik tombol environment variable



- Setelah itu akan muncul jendela environment variable, kemudian klik new pada panel System variable, dan isikan keterangan berikut :
 1. Variable name = path
 2. Variable value = C:\Program Files\Java\jdk1.8.0_144\bin

4. Editor Java

Ada banyak editor yang bisa kita gunakan untuk menuliskan soce code bahasa pemrograman

java, bahkan semua editor dimungkinkan selama editor tersebut mampu menyimpan file dengan ekstensi .java yang merupakan ekstensi dari file untuk menyimpan source code bahasa pemrograman java.

5. Program Java Pertama

Kita akan coba untuk menganalisa program Java pertama :

```
public class percobaan1
{
    /**
     * My first java program
     */
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

Baris pertama kode :

```
public class percobaan1
```

menandakan nama class yaitu Hello. Dalam Java, semua kode seharusnya didalam deklarasi class. kita melakukannya dengan menggunakan kata kunci class. Baris berikutnya yaitu yang terdiri atas kurung kurawal { menandakan awal blok kita dapat juga meletakkan kurung kurawal ini setelah baris pertama dari kode yang kita tulis. Jadi, kita dapat menulis kode kita sebagai berikut:

```
public class Hello
```

```
{
```

atau

```
public class Hello {
```

Tiga baris selanjutnya menandakan adanya komentar Java. Komentar adalah sesuatu yang digunakan untuk mendokumentasikan setiap bagian dari kode yang ditulis. Komentar bukan merupakan bagian dari program itu sendiri, tetapi digunakan untuk tujuan dokumentasi. Komentar itu sendiri dapat ditambahkan pada kode yang Anda tulis sebagai petunjuk yang dapat membantu proses pembelajaran pemrograman yang baik.

```
/**
```

```
 * My first java program
```

```
 */
```

Komentar dinyatakan dengan tanda “/*” dan “*/”. Segala sesuatu yang ada diantara tanda tersebut diabaikan oleh compiler Java, dan mereka hanya dianggap sebagai komentar sedangkan untuk membuat sebuah atau satu baris komentar dengan cara berikut ini

// ini adalah program pertamaku. Semua tulisan yang berada satu baris dengan tanda // akan diabaikan oleh compiler.

Baris selanjutnya,

```
public static void main(String[] args) {
```

ataudapat juga ditulis sebagai berikut

```
public static void main(String[] args)
{
```

mengindikasikan nama suatu method dalam class Hello yang bertindak sebagai method utama. Method utama adalah titik awal dari suatu program Java. Semua program kecuali applet yang ditulis dalam bahasa Java dimulai dengan method utama. Yakinkan untuk mengikuti kaidah penulisan tanda yang benar.

Tugas Pendahuluan

1. Sebutkan aturan penamaan class!
2. Jelaskan apa itu Java Virtual Machine?
3. Jelaskan apa itu Jre? Apa kegunaannya!
4. Apa yang dimaksud dengan Multiplatform?
5. Sebutkan editor yang bisa dipakai untuk menuliskan code java!
6. Jelaskan tahap membuat proyek aplikasi baru dan class baru di netbeans!
7. Sebutkan macam-macam contoh dari Integer Literals, Floating-Point Literals, Boolean Literals, Character Literals dan String Literals!
8. Berikan contoh class java sederhana yang dapat menghasilkan output berikut ini!
 - a. Hallo Dunia
 - b. Hallo Duke

Percobaan Dilakukan

1. Buat proyek baru di netbeans dan sebuah class lalu ketik Ulang listing program berikut ini !

```
class hallo_dunia
{
    public static void main(String args[])
    {
        system.out.println("Nama : Tuliskan Nama Anda");
        System.out.println("Gender : Tuliskan Gender Anda");
        System.out.println("Alamat : Tuliskan Alamat Anda");
        System.out.println("Tanggal Lahir : Tuliskan Tanggal lahir Anda");
    }
}
```

Instruksi yang harus dijalankan :

- Compile dan jalankan programnya!
- Jika ada kesalahan silahkan anda benarkan,

- Apa yang dapat anda jelaskan dari program sederhana tersebut?
2. Buat kode berikut ini kemudian masukkan character literals yang ada didalam table dibawahnya:

```
class percobaankedua
{
    public static void main(String args[])
    {
        System.out.print("Nama Buah : ");
        System.out.print("1. Manggis");
        System.Out.print("2. Semangka");
        System.out.print("3. Durian");
        System.out.Print("3. Salak");
    }
}
```

Kode	Nama
<code>\b</code>	<i>Backspace</i>
<code>\n</code>	<i>Line feed</i>
<code>\t</code>	<i>Tabulasi</i>

Instruksi yang harus dijalankan :

- Compile dan jalankan programnya!
- Jika ada kesalahan silahkan anda benarkan,
- Sisipkan character literals yang ada ditabel kedalam akhir string literalsnya!
- Apa yang dapat anda jelaskan dari program sederhana tersebut?

Pertanyaan Praktikum

1. Apa yang anda pahami dari praktikum yang telah anda lakukan?
2. Berikan kesimpulan yang dapat anda ambil dari praktikum ini!

MODUL 2**TIPE DATA, KEYWORD, VARIABEL, OPERATOR DAN TEKNIK KONVERSI****A. Target Praktikum**

1. Mahasiswa mengetahui berbagai macam tipe data dan penggunaannya
2. Mahasiswa mengetahui berbagai macam keyword
3. Mahasiswa mengetahui cara pembuatan variable dan penggunaannya.
4. Mahasiswa mengetahui konversi tipe data
5. Mahasiswa mampu menyelesaikan permasalahan terkait penggunaan variable dan tipe data
6. Mahasiswa mengetahui berbagai macam operator java dan penggunaannya
7. Mahasiswa mampu menyelesaikan permasalahan terkait penggunaan operator

B. Materi Dasar**1. Keyword dalam Java**

Kata kunci adalah identifier yang telah dipesan untuk didefinisikan sebelumnya oleh Java untuk tujuan tertentu. Anda tidak dapat menggunakan keyword sebagai nama variabel, class, method anda, dsb

2. Tipe Data Primitif

Bahasa pemrograman Java mendefinisikan delapan tipe data primitif. Mereka diantaranya adalah boolean (untuk bentuk logika), char (untuk bentuk tekstual), byte, short, int, long (integral), double and float (floating point).

3. Variabel

Variabel adalah item yang digunakan data untuk menyimpan pernyataan objek. Variabel memiliki **tipe**, **data** dan **nama**. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri. Nama variabel harus mengikuti aturan untuk identifier.

3.1 Deklarasi dan Inisialisasi Variabel

Untuk deklarasi variabel adalah sebagai berikut,

`<data tipe><name> [=initial value];`

Nilainya berada diantara `<>` adalah nilai yang disyaratkan, sementara nilai dalam tanda `[]` bersifat optional sesuai dengan tipe data yang dipakai untuk deklarasi variabel.

3.2 Menampilkan Data Variabel

Untuk mengeluarkan nilai dari variabel yang diinginkan, kita dapat menggunakan perintah sebagai berikut, `System.out.println()` atau `System.out.print()`.

4. Teknik Konversi Data

Konversi adalah cara untuk memberikan nilai suatu data dari satu tipe data ke tipe data yang lain.

Contoh:

```
int a=50;
double b=40.5;
String konfersi_a =String.valueOf(a); // konfersi dari integer ke string
String konfersi_b =String.valueOf(b); //konfersi dari double ke string
int pindah_keinteger =Integer.parseInt(konfersi_a); // konfersi dari string ke integer
double pindah_kedouble =Double.parseDouble(konfersi_b); // konfersi dari string ke double
```

Yang perlu diketahui, bahwa konfersi data dilakukan jika kita membutuhkan suatu tipe data yang sama untuk menjalankan sebuah proses, jika data tersebut sudah sama tipe datanya maka tidak perlu dilakukan proses konfersi data:

Contoh:

Penjumlahan antara **int a=2** dan **int b=4** maka hal ini bisa langsung kita lakukan dengan menjumlahkannya misal **int hasil=a+b**; namun jika ada **String a="2"**; dan **int b=4**; maka untuk menjumlahkannya terlebih dahulu harus kita lakukan proses konfersi yaitu mengubah nilai string menjadi int dengan cara **int konv_a=Integer.parseInt(a)**; baru kita lakukan proses penambahan.

5. Operator

Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti bermacam-macam prioritas yang pasti sehingga kompilernya akan tahu yang mana operator untuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan.

5.1 Operator Aritmatika

Operator aritmatika merupakan operator yang berhubungan dengan fungsi matematika.

Contoh	Nama	Hasil
a + b	Penjumlahan	Jumlah dari \$a dan \$b
a - b	Pengurangan	Selisih dari \$a dan \$b
a * b	Perkalian	Hasil kali dari \$a dan \$b
a / b	Pembagian	Pembagian dari \$a dan \$b
a % b	Modulus	Sisa dari \$a dibagi dengan \$b

Penulisan Ekspresi Aritmatika

$$\frac{a+b}{a-b} - pq = (a+b)/(a-b) - p * q$$

$$\frac{a+b/q}{ab} - q = (a+b/p)/(a*b) - q$$

5.2 Operator Relasi

Operator Relasi membandingkan dua nilai dan menentukan keterhubungan diantara nilai-nilai tersebut. Hasil keluarannya berupa nilai boolean yaitu true atau false.

Contoh	Nama	Hasil
a == b	sama dengan	benar jika a sama dengan b

$a \neq b$	tidak sama	benar jika a tidak sama dengan b
$a < b$	lebih kecil	benar jika a lebih kecil b
$a > b$	lebih besar	benar jika a lebih besar b
$a \leq b$	lebih kecil atau	benar jika a lebih kecil atau sama dengan b
$a \geq b$	sama dengan	benar jika a lebih besar atau sama dengan b

5.3 Operator Tambahan dan Pengurangan

Tabel operator Increment/decrement adalah:

Contoh	Nama	Hasil
++a	Pre-increment	Nilai a ditambah satu, kemudian operasi berjalan
a++	Post-increment	Operasi dilaksanakan, kemudian a ditambah satu
--a	Pre-decrement	Nilai a dikurang satu, kemudian operasi dilaksanakan satu
a--	Post-decrement	Operasi dilaksanakan, kemudian a dikurangi satu

5.3 Operator Logika

Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean. Terdapat enam operator logika yaitu: && (logika AND), & (boolean logika AND), || (logika OR), |(Boolean logika inclusive OR), ^ (boolean logika exclusive OR), dan ! (logika NOT).

Contoh	Nama	Hasil
a && b	AND	benar jika a dan b benar
a b	OR	benar jika a atau b benar
a ^ b	XOR	benar jika a atau b benar tetapi tidak keduanya

Operator Perbandingan/Kondisi

Operator perbandingan ini digunakan pada struktur kontrol program, seperti if, elseif, di mana dilakukan perbandingan antar dua nilai.

Contoh	Nama	Hasil
$a == b$	sama dengan	benar jika a sama dengan b
$a \neq b$	tidak sama	benar jika a tidak sama dengan b
$a < b$	lebih kecil	benar jika a lebih kecil b
$a > b$	lebih besar	benar jika a lebih besar b
$a \leq b$	lebih kecil atau	benar jika a lebih kecil atau sama dengan b
$a \geq b$	sama dengan	benar jika a lebih besar atau sama dengan b

Tugas Pendahuluan

1. Sebutkan macam-macam keyword java!
2. Lengkapi Tabel berikut ini!

Data Type	Data Type
byte	
short	
int	
long	
float	
double	
char	
boolean	

3. Diberikan tabel dibawah ini, deklarasikan variabel yang terdapat didalamnya dengan tipe data yang sesuai dan berikan nilai inisialisasi. Tampilkan hasil outputnya yaitu nama variabel dan nilainya

<i>Nama Variabel</i>	<i>Tipe Data</i>	<i>Nilai Awal</i>
number	integer	10
letter	character	a
result	boolean	true
str	String	hello

Berikut ini merupakan tampilan yang diharapkan sebagai hasil eksekusi program,

Number = 10

letter = a

result = true

str = hello

4. Berikan contoh listing program proses konfersi data, dengan arah konfersi tipe data sesuai dengan keinginan anda, minimal 5 proses konfersi!
5. Cari contoh listing kode yang menggunakan operator Perbandingan, Increment/decrement, dan Aritmatika!

Percobaan Dilakukan

1. Buatlah program untuk menghitung nilai rata2 dari tiga angka berikut ini 30, 30, 60!
2. Bangunlah sebuah program untuk menghitung gaji bersih setelah dipotong pajak 10%.

Input : gajikotor.

Proses : potongan = pajak*gajikotor,

Gajibersih = gajikotor-potongan.

Output : gajibersih.

3. Pak Soleh Ingin naik haji, kemudian ikut tabungan haji ONH, biaya haji pada waktu itu adalah 50jt, pak soleh sudah mengumpulkan tabungan dirumah sebanyak 20jt, kemudian sebagai tabungan pembuka ia masukkan tabungan ONH sebesar 10jt, jika bulan ini pak soleh baru memulai bukatabungan dan cicilan pertama dimulai bulan depan, maka berapa minimal cicilan yang harus iatabungkan jika dijadwalkan pak soleh akan naik haji 2 tahun lagi? Selesaikan program untuk menghitung nilai cicilan tersebut!
4. Selesaikan perhitungan berikut :
- a. $2^8 + 4^4$
 - b. $2^2 + 4^1 / 2^2$
 - c. Hitung nilai x jika diketahui $2x+6/6-4=2x-2/3$

Pertanyaan Praktikum

1. Apa yang anda pahami dari praktikum yang telah anda lakukan?
2. Berikan kesimpulan yang dapat anda ambil dari praktikum ini!

MODUL 3

INPUT DARI KEYBOARD DAN STRUKTUR KONTROL

A. Target Pembelajaran

1. Membuat program Java yang interaktif yang bisa mendapatkan input dari keyboard
2. Menggunakan class `BufferedReader` untuk mendapatkan input dari keyboard melalui layar console
3. Menggunakan class `JOptionPane` untuk mendapatkan input dari keyboard menggunakan GUI
4. Menggunakan struktur kontrol pemilihan (if, else, switch) yang digunakan untuk memilih blok kode yang akan dieksekusi

B. Materi Dasar

Setelah kita mempelajari konsep dasar pada Java dan menulis beberapa program sederhana. Sekarang kita akan mencoba membuat program kita lebih interaktif dengan menggunakan input dari keyboard. Pada bab ini, kita akan mempelajari dua cara memberikan input, yang pertama adalah menggunakan class `BufferedReader` dan melalui GUI (Graphical User Interface) dengan menggunakan class `JOptionPane`

1. Menggunakan `BufferedReader`

Pada bagian ini, kita akan menggunakan class `BufferedReader` yang berada di package `java.io` untuk mendapatkan input dari keyboard. Berikut ini adalah langkah-langkah yang diperlukan untuk mendapatkan input dari keyboard:

- a. Tambahkan di bagian paling atas code Anda:

```
import java.io.*;
```

- b. Tambahkan statement berikut:

```
BufferedReader dataIn = new BufferedReader(new InputStreamReader(  
System.in) );
```

- c. Deklarasikan variabel String temporary untuk mendapatkan input, dan gunakan fungsi `readLine()` untuk mendapatkan input dari keyboard. Anda harus mengetikkannya di dalam blok try-catch:

```
try {  
    String temp = dataIn.readLine();  
    catch( IOException e ){  
        System.out.println("Error in getting input");  
    }
```

Berikut ini adalah contoh source code lengkapnya:

```

1  import java.io.BufferedReader;
2  import java.io.InputStreamReader;
3  import java.io.IOException;
4  public class GetInputFromKeyboard
5  {
6      public static void main( String[] args ){
7          BufferedReader dataIn = new BufferedReader(new InputStreamReader( System.in));
8          String name = "";
9          System.out.print("Silahkan Masukkan Nama Anda:");
10         try{
11             name = dataIn.readLine();
12         }catch( IOException e ){
13             System.out.println("Error!");
14         }
15         System.out.println("Hallo " + name + "Anda Berhasil Membuatnya!!");
16     }
17 }

```

Berikutnya akan penjelasan setiap baris dari code tersebut:

- Statement,

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

```

menjelaskan bahwa kita akan menggunakan class `BufferedReader`, `InputStreamReader` dan `IOException` yang berada di `java.io` package. Java Application Programming Interface (API) berisi ratusan class yang sudah didefinisikan sebelumnya yang dapat digunakan untuk program Anda. Class-class tersebut dikumpulkan di dalam packages.

- blok di bawah ini merupakan try-catch block,

```

try{
    name = dataIn.readLine();
}catch( IOException e ){
    System.out.println("Error!");
}

```

Pada baris ini menjelaskan bahwa kemungkinan terjadi error pada pernyataan,

```
name = dataIn.readLine();
```

akan ditangkap. Kita akan membahas tentang penanganan exception pada bab selanjutnya dari pembahasan ini, tetapi untuk sekarang, Anda cukup mencatat bahwa Anda perlu menambahkan kode ini untuk menggunakan method `readLine()` dari `BufferedReader` untuk mendapatkan input dari user.

- pernyataan,

```
name = dataIn.readLine();
```

method diatas memanggil `dataIn.readLine()`, mendapatkan input dari user dan memberikan sebuah nilai `String`. Nilai ini akan disimpan ke dalam variabel `name`, yang akan kita gunakan pada statement akhir untuk menyambut user,

```
System.out.println("Hello " + name + "!!");
```

2. JOptionPane

Cara lain untuk mendapatkan input dari user adalah dengan menggunakan class `JOptionPane` yang didapatkan dari `javax.swing` package. `JOptionPane` memudahkan memunculkan dialog box standar yang memberikan kepada user sebuah nilai atau menginformasikan sesuatu. Berikut Contoh Kode yang digunakan:

```
import javax.swing.JOptionPane;

public class GetInputFromKeyboard2 {
    public static void main( String[] args ){
        String name = "";
        name = JOptionPane.showInputDialog("Please enter your name");
        String msg = "Hello " + name + "!";
        JOptionPane.showMessageDialog(null, msg);
    }
}
```

Berikut penjelasannya

- Statement pertama,

```
import javax.swing.JOptionPane;
```

Menjelaskan bahwa kita mengimpor class JOptionPane dari package javax.swing. Bisa juga ditulis seperti,

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

- Pernyataan,

```
name = JOptionPane.showInputDialog("Please enter your name");
```

membuat sebuah input dialog JOptionPane, yang akan menampilkan dialog dengan sebuah pesan, sebuah textfield dan tombol OK seperti pada gambar. Hasil dari dialog tersebut adalah String dan disimpan ke dalam variabel name. Sekarang kita membuat pesan selamat datang, yang akan disimpan ke dalam variabel msg,

```
String msg = "Hello " + name + "!";
```

Baris selanjutnya adalah menampilkan sebuah dialog yang berisi sebuah pesan dan tombol OK,

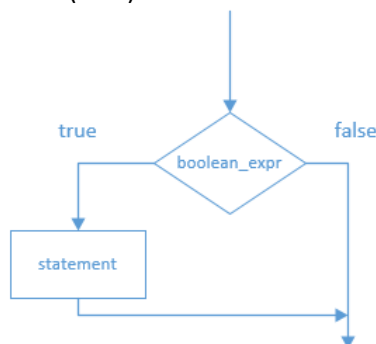
```
JOptionPane.showMessageDialog(null, msg);
```

3. Struktur Kontrol

Pada bagian ini, kita akan mempelajari tentang struktur kontrol dimana kita dapat mengubah cara eksekusi pada pernyataan yang dibuat di program kita. Struktur kontrol adalah pernyataan dari Java yang memungkinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

3.1 Statement if

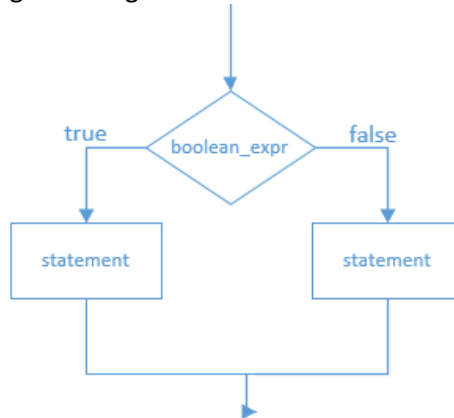
Pernyataan if akan menentukan sebuah pernyataan (atau blok kode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar (true).



Gambar 1. Flowchart statement If

3.2 Statement if-else

Pernyataan if-else digunakan apabila kita ingin mengeksekusi beberapa pernyataan dengan kondisi true dan pernyataan yang lain dengan kondisi false.



Gambar 2. Flowchart statement If-else

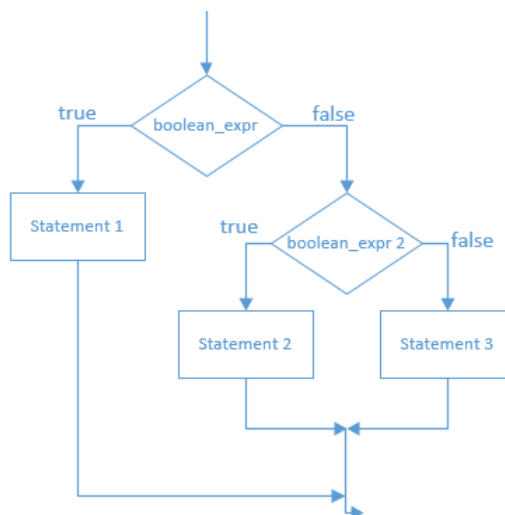
3.3 Statement if-else-if

Pernyataan pada bagian kondisi else dari blok if-else dapat menjadi struktur if-else yang lain. Kondisi struktur seperti ini memungkinkan kita untuk membuat seleksi persyaratan yang lebih kompleks.

Bentuk statement if-else if,

```

if( boolean_expression1 )
    statement1;
else if( boolean_expression2 )
    statement2;
else
    statement3
    
```



Gambar 3. Flowchart statement If-else-If

3.4 Statement switch

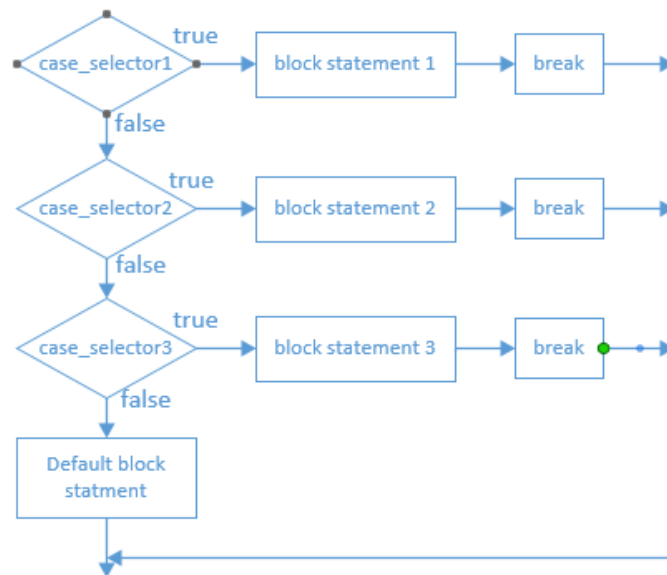
Caralainuntuk membuat cabang adalah dengan menggunakan kata kunci switch. Switchmengkonstruksikan cabang untuk beberapa kondisi dari nilai. Bentuk statement switch,

```

switch( switch_expression )
{
    case case_selector1: statement1;
    break;
    case case_selector2: statement2;
    break;
    
```

```
default: statement3;
break;
}
```

switch_expression adalah ekspresi integer atau karakter dan case_selector1, case_selector2 dan seterusnya adalah konstanta unik dari nilai integer atau karakter.



Gambar 4. Flowchart statement switch

Tugas Pendahuluan

1. Cari contoh sederhana inputan keyboard dengan menggunakan BufferedReader dengan menggunakan 2 kali proses inputan, Kemudian jelaskan !
2. Cari contoh sederhana inputan keyboard dengan menggunakan JOptionPane dengan menggunakan 2 inputan textfield, kemudian jelaskan !
3. Cari contoh sederhana program Struktur Kontrol dengan menggunakan struktur control berikut ini:
 - If
 - If-else
 - If-else-if
 - If-else-else-if dan
 - switch- case
4. Buat program untuk menghitung nilai dari penjumlahan 2 bilangan a=3 dan b=4 dimana kedua bilangan tersebut diinputkan dengan :
 - BufferedReader
 - JOptionPane

Percobaan Dilakukan

1. Buat aplikasi dengan menggunakan bufferedreader yang menanyakan input bilangan sebanyak 3 kali. Output yang diharapkan :
 Masukkan Nilai Pertama = 20
 Masukkan Nilai Kedua = 12
 Masukkan Nilai ke dua = 3
 Nilai-nilai yang dimasukkan = 20, 12, 3
2. Dari Soal nomor 1 tambahkan perintah untuk mencari nilai terbesar dan terkecil dari angka – angka yang telah dimasukkan tadi dengan tambahan output yang diharapkan sebagai berikut,

Nilai Terbesar Adalah = 20

Nilai Terkecil Adalah = 3\

3. Sebuah perusahaan memberikan komisi kepada salesmannya dengan ketentuan sebagai berikut:
- Bila seorang salesman dapat menjual barang hingga Rp 2.000.000 maka dia akan mendapat uang jasa sebesar Rp. 100.000 ditambah dengan uang komisi sebesar 10% dari pendapatan hari itu.
 - Bila seorang salesman dapat menjual barang di atas Rp. 2.000.000 hingga 5.000.000 maka dia akan mendapat uang jasa sebesar Rp 200.000 ditambah dengan uang komisi sebesar 15% dari pendapatan hari itu.
 - Bila seorang salesman dapat menjual barang di atas Rp. 5.000.000 maka dia akan mendapat uang jasa sebesar Rp. 300.000 ditambah dengan uang komisi sebesar 20% dari pendapatanhari itu.

Bangunlah program untuk menghitung pendapatan seorang salesman hari itu. Nilai hasilpenjualandimasukkan melalui JOptionPane.

4. Pada suatu ujian didapatkan nilai hasil ujian sebgai berikut:

Nama	Nilai	Huruf
Adi	70	
Budi	65	
Caca	90	
Deny	75	

Buat listing program untuk memberikan nilai huruf pada masing-masing mahasiswa pada tabletersebut jika kaidah pemberian nilai huruf seperti berikut ini:

36-45 Nilai D

46-55 Nilai C

56-65 Nilai C+

66-75 Nilai B

76-85 Nilai B+

86-100 Nilai A

Kemudian munculkan message yang mempunyai nilai terendah dan yang mempunyai nilai terkecil!

Pertanyaan Praktikum

1. Apa yang anda pahami dari praktikum yang telah anda lakukan?
2. Berikan kesimpulan yang dapat anda ambil dari praktikum ini!

MODUL 4

PERULANGAN/LOOPING

A. Target Pembelajaran

1. Mahasiswa mengetahui struktur dasar statement perulangan(looping)
2. Mahasiswa mengetahui struktur dasar statement perulangan(looping)
3. Mahasiswa mengetahui cara pengoperasian statement perulangan
4. Mahasiswa mampu menyelesaikan permasalahan terkait penggunaan statement perulangan

B. Materi Dasar

Pengulangan adalah berupa pernyataan dari Java yang memungkinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macam jenis dari struktur pengulangan yaitu while, do-while, dan for-loops.

1. while loop

Pernyataan while loop adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok. Bentuk pernyataan while,

```
while( boolean_expression )
{
    statement1;
    statement2;
}
```

Pernyataan di dalam while loop akan dieksekusi berulang-ulang selama kondisi boolean_expression bernilai benar (true). Contoh, pada kode dibawah ini,

```
int i = 4;
while ( i > 0 ){
    System.out.print(i);
    i--;
}
```

Contoh diatas akan mencetak angka 4321 pada layar.

2. do-while loop

Do-while loop mirip dengan while-loop. Pernyataan di dalam do-while loop akan dieksekusi beberapa kali selama kondisi bernilai benar(true). Perbedaan antara while dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali. Perbedaan antara while dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali. Bentuk pernyataan do-while,

```
do{
    statement1;
    statement2;
}while( boolean_expression );
```

Pernyataan di dalam do-while loop akan dieksekusi pertama kali, dan akan dievaluasi kondisi dari boolean_expression. Jika nilai pada boolean_expression tersebut bernilai true, pernyataan di dalam do-while loop akan dieksekusi lagi. Berikut ini beberapa contoh do-while loop:

```
int x = 0;
```



```
do
{
System.out.println(x);
x++;
}while (x<10);
```

Contoh ini akan memberikan output 0123456789 pada layar.

3. for loop

Pernyataan for loop memiliki kondisi hampir mirip seperti struktur pengulangan sebelumnya yaitu melakukan pengulangan untuk mengeksekusi kode yang sama sebanyak jumlah yang telah ditentukan. Bentuk dari for loop,

```
for (InitializationExpression; LoopCondition; StepExpression){
statement1;
statement2;
}
```

- **InitializationExpression** – inialisasi dari variabel loop.
- **LoopCondition** - membandingkan variabel loop pada nilai batas tertentu
- **StepExpression** - melakukan update pada variabel loop.

Berikut ini adalah contoh dari for loop,

```
for( int i = 0; i < 10; i++ ){
System.out.print(i);
}
```

Pada contoh ini pernyataan i=0 merupakan inisialisasi dari variabel. Selanjutnya, kondisi i<10 diperiksa. Jika kondisi bernilai true, pernyataan di dalam for loop dieksekusi. Kemudian, ekspresi i++ dieksekusi, lalu akan kembali pada bagian pemeriksaan terhadap kondisi i<10 lagi. Kondisi ini akan dilakukan berulang-ulang sampai mencapai nilai yang salah (false).

4. Percabangan dalam Perulangan

Pernyataan percabangan memungkinkan kita untuk mengatur aliran eksekusi program. Java memberikan tiga bentuk pernyataan percabangan: break, continue dan return.

4.2 Pernyataan break

Pernyataan break memiliki dua bentuk: tidak berlabel (unlabeled) dan berlabel (labeled).

4.2.1 Pernyataan break tidak berlabel

Pernyataan break tidak berlabel (unlabeled) digunakan untuk menghentikan jalannya pernyataan switch. Selain itu pernyataan break unlabeled juga bias digunakan untuk menghentikan pernyataan pernyataan for, while atau do-while loop.

```
for( int i=0; i< 5; i++ )
{
if( i==3)
{
System.out.println(i);
break;
}
}
```

Dari pernyataan diatas didapatkan jika i berposisi pada nilai maka perulangan akan dihentikan. Dan akan melakukan pencetakan angka i, yaitu angka 3.

4.2.2 Pernyataan break berlabel

Bentuk label dari pernyataan break akan menghentikan pernyataan di luarnya, dimanasebelumnya harus diberikan label yang sudah di spesifikasikan pada program pada pernyataanbreak. Ketika sebuah nilai ditemukan, break akan menghentikan pernyataan yang diberi labelsearchLabel yang terletak di luar pernyataan for loop.

```
searchLabel:
for( int i=0; i<6; i++ )
{
if(i==5 )
{
System.out.println(i);
break searchLabel;
}
}
```

Pernyataan break menghentikan pernyataan yangdiberi label; dan tidak menjalankan aliran kontrol apapun pada label.Aliran control pada label akandiberikan secara otomatis pada pernyataanyang terletak dibawahlabel. Jadi jika pernyataan yang tidak dibawahnya maka tidak akan mengenali label yang telah dituliskan.

4.3 Pernyataan Continue

Pernyataan continue memiliki dua bentuk: berlabel dan tidakberlabel. Andadapat menggunakan pernyataan continue untuk melanjutkan pengulangan yang sedang dijalankan oleh pernyataan for,while, atau do- while loop.

4.3.1 Pernyataan continue tidak berlabel

Bentuk pernyataan continue tidak berlabel (unlabeled) akan melewati bagian pernyataan setelah pernyataan ini dituliskan dan memeriksa eksepresi logika(boolean) yang mengkontrol pengulangan. Jikaekspresilogika(boolean) masih bernilai true,maka pengulangan tetap dilanjutkan.Padadasarnya pernyataan ini akanmelanjutkan bagian pengulangan pada pernyataan loop.

```
for( int i=0; i<6; i++ )
{
if(i==3 )
{
continue;
}
System.out.println(i);
}
```

4.3.2 Pernyataan continue berlabel

Bentuk pernyataan continue berlabel (labeled) akan melanjutkan pengulangan yang sedang terjadidan dilanjuti ke pengulangan berikutnya dari pernyataan pengulangan yang diberi label (tanda).

```
search:
for( int i=0; i<6; i++ )
{
if(i==3 )
{
continue search;
}
}
```

```
System.out.println(i);  
}
```

4.4 Pernyataan Return

Pernyataan `return` digunakan untuk keluar dari sebuah method. Pernyataan `return` memiliki dua bentuk: memberikan sebuah nilai, dan tidak memberikan nilai. Untuk memberikan sebuah nilai, cukup berikan nilai (atau ekspresi yang menghasilkan sebuah nilai) sesudah kata `return`. Contohnya,

```
return ++count; atau return "Hello";
```

Tipe data dari nilai yang diberikan harus sama dengan tipe dari method yang dibuat. Ketika sebuah method `void` dideklarkan, gunakan bentuk `return` yang tidak memberikan nilai. Contohnya,

```
return;
```

Kita akan membahas lebih lanjut tentang pernyataan `return` anda akan mempelajari lebih lanjut ketika mempelajari pemrograman berorientasi objek.

Tugas Pendahuluan

1. Ada tiga jenis perulangan di java yaitu `while`, `do-while` dan `for`, masing masing perulangan tersebut memiliki karakter atau perilaku berbeda, coba jelaskan karakter masing masing perulangan tersebut!
2. Buatlah sebuah program perulangan sederhana dengan output angka 1-10 dengan menggunakan perulangan `while`, `do-while` dan `for`. Kemudian lanjutkan hingga program yang anda buat bisa menampilkan total 10 angka dari perulangan tersebut!

Percobaan Dilakukan

1. Ketik kembali kode sederhana berikut ini, simpan dengan nama file sesuai dengan nama kelasnya:

```
public class cobaFor {  
    public static void main( String[] args )  
    {  
        int jml=0;  
        for (int i=1;i<=10;i++) {  
            jml+=i;  
        }  
        System.out.print ("Jumlah Semua Bilangan : "+jml);  
    }  
}
```

```
public class CobaFor1 {  
    public static void main(String args[])  
    {  
        for(double jari=1.0;jari<=2.0;jari+=0.2){  
            System.out.println ("Radius =" +jari+ " Luas= " +jari*Math.PI*jari);  
        }  
    }  
}
```

```
public class CobaWhile {
    public static void main(String args[]) {
        double r = 0;
        while (r < 0.99d) {
            r = Math.random();
            System.out.println(r);
        }
    }
}
```

```
public class CobaDoWhile {
    public static void main(String args[]) {
        double r=0;
        do{
            r=Math.random();
            System.out.println(r);
        }
        while (r < 0.99d);
    }
}
```

```
public class Faktorial{
    public static void main(String args[]) {
        long batas=4;
        long faktorial=1;
        for(int i=0;i<=batas;i++){
            faktorial=1;
            for(int faktor=2; faktor<=i; faktor++)
                faktorial*=faktor;
            System.out.println(i+"!"+ " adalah = "+faktorial);
        }
    }
}
```

Instruksi yang harus dijalankan:

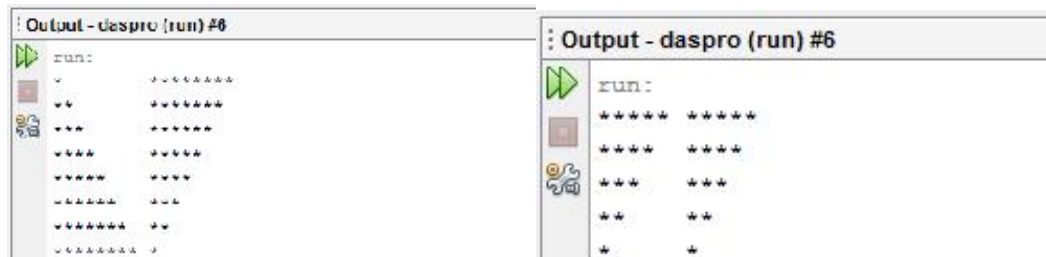
- Ketik kembali kode sederhana diatas, Compile dan jalankan programnya!
 - Jelaskan hasilnya
2. Dari program sederhana diatas, anda bangun sebuah program yang mampu menghasilkan output hasil perhitungan berikut (Tidak boleh menggunakan fungsi Math.pow):
 - a. $4^3 + 5^2$
 - b. $5! + 4!$
 - c. $(4^2 + 5^1) / 5 + 4^2$
 3. Ambil sebuah angka sebagai input dari user, dan outputnya berupa kata yang sesuai dengan angka. Angka yang dimasukkan antara 1-10. Jika user memasukkan nilai yang tidak sesuai berikan output "Invalid number".
 - a. Gunakan statement if-else untuk menyelesaikan
 - b. Gunakan statement switch untuk menyelesaikan
 4. Buat Program untuk mencari nilai bilangan ganjil dan genap sekaligus hitung jumlahnya dengan menggunakan operator sisa bagi dari n angka yang dimasukkan lewat:
 - a. Nim Genap: Konsole

- b. Nim Ganjil: JOptionPane
5. Ketik kode sederhana berikut ini, dan lanjutkan instruksi yang ada dibawahnya!

```
import javax.swing.*;
public class soal4 {
    public static void main( String[] args){
        String angka=JOptionPane.showInputDialog("Masukkan Angka Yang Anda Inginkan : ");
        int n=Integer.parseInt(angka);
        for(int i=1;i<=n;i++){
            for(int j=1;j<=i;j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Instruksi yang harus dijalankan :

- a. Ketik kembali kode sederhana diatas, Compile dan jalankanprogramnya!
- b. Kembangkan program tersebut hingga mampu menghasilkan output seperti gambar berikut ini:



Catatan: Jumlah bintang tergantung dari n angka yang diinputkan!

MODUL 5**ARRAY/LARIK (1 Dimensi)****A. Target Pembelajaran**

1. Mahasiswa mengetahui struktur dasar Array
2. Mahasiswa mengetahui cara pengoperasian Array
3. Mahasiswa mampu menyelesaikan permasalahan terkait penggunaan Array

B. Materi Dasar

Dalam bab ini, kita akan mendiskusikan mengenai array dalam Java. Pertama, kita akan mendefinisikan apa yang dimaksud dengan array, kemudian kita juga akan mendiskusikan bagaimana mendeklarasikannya dan menggunakannya dalam Java. Array adalah suatu type data yang mampu diisi dengan lebih dari satu nilai. Dan untuk pengaksesan nilai pada array ini digunakan indeks atau berdasarkan letak nilai tersebut pada array.

- Array satu dimensi

Yaitu array yang hanya mempunyai 1 baris yang didalamnya terdapat data yang mempunyai type sama. Atau dapat diartikan sejumlah data yang ditampung oleh suatu variable yang mempunyai type yang sama dalam satu baris dan satu kolom.

1. Pendeklarasian Array

Array harus dideklarasikan seperti layaknya sebuah variabel. Pada saat mendeklarasikan array, anda harus membuat sebuah daftar dari tipe data, yang diikuti oleh sepasang tanda kurung [], lalu diikuti oleh nama identifier-nya. Sebagai contoh,

```
int []usia;
```

atau Anda dapat menempatkan sepasang tanda kurung [] sesudah nama identifier. Sebagai contoh,

```
int usia[];
```

Setelah pendeklarasian array, kita harus membuat array dan menentukan berapa panjangnya dengan sebuah konstruktor. Proses ini di Java disebut sebagai instantiation (istilah dalam Java yang berarti membuat).

Sebagai catatan bahwa ukuran dari array tidak dapat diubah setelah anda menginisialisasinya. Sebagai contoh,

```
int ages[]; //deklarasi
ages = new int[100]; //instantiate obyek
```

Selain menggunakan sebuah pernyataan new untuk meng-instantiate array, Anda juga dapat mendeklarasikan, membangun, kemudian memberikan sebuah nilai pada array sekaligus dalam sebuah pernyataan. Sebagai contoh,

```
boolean results[] = { true, false, true, false };
double []grades = {100, 90, 80, 75};
String days[] = { "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"};
```

2. Pengaksesan elemen array

Untuk mengakses sebuah elemen dalam array, atau mengakses sebagian dari array, Anda harus menggunakan indeks atau subscript.

“Nilai indeks selalu dalam tipe integer, dimulai dari angka nol dan dilanjutkan ke angka berikutnya sampai akhir array. Sebagai catatan bahwa indeks didalam array dimulai dari 0 sampai dengan (ukuranArray-1).”

Berikut ini adalah contoh kode untuk mencetak seluruh elemen didalam array. Dalam contoh ini digunakanlah pernyataan for loop, sehingga kode kita menjadi lebih pendek.

```
public class ArraySimple
{
    public static void main( String[] args )
    {
        int[] ages = new int[100];
        for( int i=0; i<100; i++ )
        {
            System.out.print( ages[i] );
        }
    }
}
```

3. Panjang Array

Untuk mengetahui berapa banyak elemen didalam sebuah array, Anda dapat menggunakan atribut length dari array. Atribut ini akan mengembalikan ukuran dari array itu sendiri. Sebagai contoh, NamaArray.length Pada contoh sebelumnya, kita dapat menuliskannya kembali seperti berikut ini,

```
public class ArraySimple
{
    public static void main( String[] args )
    {
        int[] ages = new int[100];
        for( int i=0; i<ages.length; i++ )
        {
            System.out.print( ages[i] );
        }
    }
}
```

Tugas Pendahuluan

1. Apa sebenarnya array 1 dimensi itu?
2. Bagaimana cara mengetahui jumlah elemet array 1 dimensi?
3. Bagaimana cara mengambil nilai element pada array 1 dimensi?
4. Dari listing kode dibawah ini, tambahkan inputan JOptionPane Sehingga ketika anda memasukkansalah satu nilai element kedalam inputan tersebut maka muncul posisi indexnya!

```

public class NewClass {
    public static void main( String[] args )
    {
        int data[]={4,6,4,2,8,4,2,11};
    }
}

```

5. Cetak iterasi / tampilkan data array pada kode dibawah dengan menggunakan perulangan while,do-while dan for (Wajib ketiganya).

Percobaan Dilakukan

1. Ketik kembali kode sederhana berikut ini :

Array Statis

```

public class Array {
    public static void main(String args[]){
        int[] data={20,10,50,30,10};
        System.out.print ("Isi data anda adalah : ");
        for(int i=0;i<data.length;i++){
            System.out.print(data[i]+" ");
        }
        System.out.println();
    }
}

```

Array Dinamis dengan Bilangan Random

```

public class Ratarata {
    public static void main(String args[]){
        double[] contoh = new double[30];
        for (int i=0; i<30; i++)
            contoh[i]=100.0 * Math.random();
        double ratarata =0.0;
        for (int i=0; i<30; i++)
            ratarata += contoh[i];
        ratarata /= 30;
        System.out.println("Mean 30 bilangan random : "+ratarata);
    }
}

```

Array Dinamis Dengan Elemen Dimasukkan

```

import javax.swing.*;

public class NewClass2{
    public static void main( String[] args){
        int data[]=new int[10];
        for(int a=0;a<data.length;a++){
            data[a]=JOptionPane.showInputDialog("Masukan Element Index ke "+a);
            System.out.print("Index Ke"+a+" adalah "+data[a]);
        }
    }
}

```

Instruksi yang harus dijalankan :

1. Ketik kembali kode sederhana diatas, Compile dan jalankan programnya!
2. Lanjutkan program sederhana (Contoh Ketiga)tersebut sehingga mampu menghasilkan out put hasil perhitungan berikut:
 - a. Mencari nilai maksimal dan minimal serta tampilkan posisi indeks dari nilai tersebut!
 - b. Mencari nilai rata rata array tersebut!
 - c. Tambahkan sebuah perintah pencarian posisi index dari nilai element yang di inputkan melalui JOptionPane, Kemudian tukar nilainya dengan angka baru dengan JOtionPane, Dan munculkan kembali element arraynya!
3. Buat pogram untuk membangkitkan variabel array dengan menggunakan **JOptionpane** (Nim: Genap) dan input console (Nim : Ganjil) dengan rule sebagai berikut:
 - a. Masukkan jumlah element array 1 dimensi
 - b. memasukkan nilai element sesuai dengan jumlah elementnya.
 - c. Tampilkan nilai terbesar dan terkecil elemet yang telah di inputkan! Serta posisi indexnya!
 - d. Jumlahkan semua data array sederhana tersebut, dan cari nilai rata2 nya !
 - e. Jumlahkan semua element yang punya indeks ganjil (Praktikan dengan Nim Genap), dan jumlahkan semua element yang berindeks genap (Untuk Praktikan dengan Nim Ganjil)
 - f. Cek hasil penjumlahan dari langkah E, apakah ia bernilai ganjil atau genap. Tampilkan hasilnya.
 - g. Urutkan Element Array yang telah anda buat!
4. Apa yang bisa anda jelaskan dari listing program diatas!

MODUL 6

ARRAY/LARIK (Multidimensi)

A. Target Pembelajaran

1. Mahasiswa mengetahui struktur dasar Array Multidimensi
2. Mahasiswa mengetahui cara pengoperasian Array Multidimensi
3. Mahasiswa mampu menyelesaikan permasalahan terkait penggunaan Array Multidimensi

B. Materi Dasar

Array multidimensi diimplementasikan sebagai array yang terletak di dalam array. Array multidimensi dideklarasikan dengan menambahkan jumlah tanda kurung setelah nama array. Sebagai contoh,

```
// Elemen 512 x 128 dari integer array
int[][] twoD = new int[512][128];
```

```
// karakter array 8 x 16 x 24
char[][][] threeD = new char[8][16][24];
```

```
// String array 4 baris x 2 kolom
String[][] dogs = {{ "terry", "brown" }, { "Kristin", "white" }, { "toby",
"gray"}, { "fido",
"black" } };
```

Untuk mengakses sebuah elemen didalam array multidimensi, sama saja dengan mengakses array satudimensi. Misalnya saja, untuk mengakses element pertama dari baris pertama didalamarray dogs, kita akan menulis,

```
System.out.print( c[0][0] );
```

Kode diatas akan mencetak String "terry" di layar

Contoh:

```
public class Contoh2d {
    int[][] A={{3,7,5},{2,8,6}};
    public void cetak() {
        System.out.println(" Matrik A: ");
        for(int i=0;i<A.length;i++){
            for(int j=0;j<A[0].length;j++){
                System.out.print(A[i][j]+" ");
            }
            System.out.println();
        }
    }
    public static void main(String args[]){
        Contoh2d jm=new Contoh2d();
        jm.cetak();
    }
}
```

Tugas Pendahuluan

1. Apa array multidimensi?
2. Array mutidimensi mempunyai baris dan kolom, bagaimana cara mengetahui jumlah element multidimensi?
3. Bagaimana cara mengambil nilai element pada array multidimensi?
4. Berdasarkan program dibawah ini, tambahkan inputan JOptionPane Sehingga ketika anda memasukkan salah satu nilai element kedalam inputan tersebut maka muncul posisi indexnya!

```
public class NewClass {  
    public static void main( String[] args ){  
        int data2[][]={{4,6,4,7,8,3,2,10},{4,6,4,2,8,8,2,10}};  
    }  
}
```

5. Cetak iterasi / tampilkan data array pada kode dibawah dengan menggunakan perulangan while, do-while dan for (Wajib ketiganya) Gunakan nested loop.
6. Tambahkan kode untuk menghitung nilai rata2 array tersebut!

Percobaan Dilakukan

1. Ketik kembali kode sederhana berikut ini:

```
public class Array2D {  
    public static void main(String args[]){  
        int[][] tabel={{1,2,3,4},{5,6,7,8}};  
        System.out.println("jumlah baris = "+tabel.length);  
        System.out.println("jumlah kolom = "+tabel[0].length);  
        System.out.println();  
        for (int i=0; i<tabel.length; i++){  
            for (int j=0; j<tabel[0].length; j++){  
                System.out.print(tabel[i][j]+" ");  
                System.out.println();  
            }  
        }  
    }  
}
```

```
public class NewClass {  
    public static void main( String[] args ){  
        int data2[][]={{4,6,4,2,8,4,2,10},{4,6,4,2,8,4,2,10}};  
        for(int a=0;a<data2.length;a++){  
            for(int b=0;b<data2[0].length;b++){  
                System.out.print(data2[a][b]+" ");  
            }  
            System.out.print("\n");  
        }  
    }  
}
```

Instruksi yang harus dijalankan:

1. Ketik kembali kode sederhana diatas, Compile dan jalankan programnya!
2. Lanjutkan program sederhana tersebut (Contoh kedua) sehingga mampu menghasilkan out put hasil perhitungan berikut:

- a. Cetak iterasi / tampilkan data array pada kode dibawah dengan menggunakan perulangan while dan do-while (Wajib keduanya) Gunakan nested loop.
 - b. Cari nilai rata rata array tersebut!
 - c. Tambahkan sebuah perintah pencarian posisi index dari element array yang di inputkan melalui JOptionPane pada kedua array tersebut. Kemudian tukar nilainya dengan angka baru dengan JOptionPane! Kemudian munculkan hasilnya kembali arraynya!
 - d. Tambahkan kode yang bias menjumlahkan semua element yang mempunyai index kolom ganjil (Nim: Ganjil) dan index kolom genap (Nim: Genap).
3. String data [][] = {"ABDUL", "085646668991", "Kediri"}, {"KUSNO", "085646668992", "Trenggalek"}, {"PONIRAN", "085646668999", "Bojonegoro"};

Dari data array tersebut, buat program yang bisa menghasilkan output dengan desain seperti berikut ini:

NAMA	ALAMAT	TELEPON
ABDUL	Kediri	085646668991
KUSNO	Trenggalek	085646668992
PONIRAN	Bojonegoro	085646668999

4. Ada dua buah matrik berikut ini:
- | | | | |
|---|---|----|----|
| 2 | 9 | 5 | 17 |
| 1 | 5 | 10 | 4 |
- a. Carilah posisi index dari angka 9 dari matrik pertama, dimana nilai dimasukkan dengan menggunakan JOptionPane!
 - b. Jumlahkan dua matrik sederhana tersebut!

MODUL 7

EXCEPTION HANDLING DAN METHOD

A. Target Pembelajaran

1. Mendefinisikan exception
2. Menangani exception menggunakan blok try-catch-finally
3. Memahami Tentang Method dan penggunaannya didalam pemrograman
4. Mampu mengaplikasikan method dalam pembuatan program.

B. Materi Dasar

1. ExceptionHandling

Dalam bagian ini, kita akan mempelajari teknik yang dipakai dalam Java dalam menangani kondisi yang tidak bisaa dalam menjalankan operasi normal dalam program. Teknik ini dinamakan exception handling.

- Event yang terjadi ketika suatu kesalahan terjadi saat program dijalankan.
- Sering digunakan untuk akses sumberdaya non memori seperti akses file dan koneksi database

1.1 Apa itu Exception?

Exception adalah sebuah peristiwa yang menjalankan alur proses normal pada program. Peristiwa inibisaanya berupa kesalahan(error) dari beberapa bentuk. Ini disebabkan program kita berakhir tidaknormal.

1.2 Menangani Exception

Untuk menangani exception dalam Java, kita gunakan blok try-catch-finally. Apa yang kitalakukandalam program kita adalah kita menempatkan pernyataan yang mungkin menghasilkan exceptiondalam blok ini. Bentuk umum dari blok try-catch-finally adalah,

```
Try{
    //tuliskan pernyataan yang dapat mengakibatkan exception
    //dalam blok ini
} catch( <exceptionType1><varName1> ){
    //tuliskan aksi apa dari program Anda yang dijalankan jika ada
    //exception tipe tertentu terjadi
} catch( <exceptionTypen><varNamen> ){
    //tuliskan aksi apa dari program Anda yang dijalankan jika ada
    //exception tipe tertentu terjadi
} finally{
    //tambahkan kode terakhir di sini
}
```

Exception dilemparkan selama eksekusi dari blok try dapat ditangkap dan ditangani dalam blok catch. Kode dalam blok finally selalu di-eksekusi. Berikut ini adalah aspek kunci tentang sintak dari konstruksi try-catch-finally:

- Notasi blok bersifat perintah
- Setiap blok try, terdapat satu atau lebih blok catch, tetapi hanya satu blok finally.
- Blok catch dan blok finally harus selalu muncul dalam konjungsi dengan blok try, dan di atas urutan
- Blok try harus diikuti oleh paling sedikit satu blok catch ATAU satu blok finally, ataukeduanya.
- Setiap blok catch mendefinisikan sebuah penanganan exception. Header dari blok catch harus membawa satu argumen, dimana exception pada blok tersebut akan ditangani.

1.3 Contoh Exception

a. ArithmeticException

```
public class cobaException {  
    public static void main(String args[]){  
        int bil = 10;  
        System.out.println(bil/0);  
    }  
}
```

b. Struktur Blok Try Catch

```
try{  
    kode program yang mungkin menghasilkan exception  
}catch(Exception xx) {  
} catch(Exception xx){  
}
```

2. Method/Function

2.1 Konsep Dasar

- Fungsi adalah bagian dari kode program yang mempunyai tugas spesifik.
- Fungsi akan dipanggil ke program utama atau fungsi yang lain bila dibutuhkan.
- Sebuah fungsi bisaanya digunakan untuk menangani suatu proses tertentu yang ada di dalam sebuah program.

2.2 Bentuk penulisan fungsi/method

Kita bisa menuliskan fungsi atau method dengan beberapa cara, dan method juga mempunyai beberapa macam tipe yaitu:

- void
- string
- double
- integer

Dari beberapa tipe method diatas semua harus mempunyai nilai kembalian kecuali method yang bertipe void, dalam praktikum ini kita hanya akan membahas method yang bertipe void dan untuk method yang lainnya akan anda pelajari di matakuliah pemrograman berorientasi objek. Berikut ini cara untuk menuliskan fungsi/method:

1. Method yang tidak mempunyai parameter

```
Void hitung()  
{  
    // Statement atau perintah yang dikerjakan  
}
```

2. method yang mempunyai parameter

```
Void hitung2(int a,int b) // Method yang mempunyai parameter  
{  
    // Statement atau perintah yang dikerjakan  
}
```

2.3 Contoh Aplikasi

a. Program yang tidak menggunakan fungsi/method

```
public class faktorial
{
    public static void main(String args[]){
        int nilai=1;
        int data=4;
        if(data==0||data==1){
            nilai=1;
        }
        else{
            for(int i=1; i<=data; i++){
                nilai*=i;
            }
        }
        System.out.println(data+" ! = "+nilai);
    }
}
```

b. Program yang menggunakan fungsi/method

```
public class faktorial5
{
    public void test(int data){ // Fungsi atau method
        int nilai=1;
        if(data==0||data==1){
            nilai=1;
        }
        else{
            for(int i=1; i<=data; i++){
                nilai*=i;
            }
        }
        System.out.println(data+"! = "+nilai);
    }

    public static void main(String args[]){
        faktorial5 f5=new faktorial5();
        f5.test(4);
        f5.test(5);
    }
}
```

Dapat anda perhatikan bahwa program "a" yang tidak menggunakan fungsi, maka proses yang berjalan hanya sekali dan angka yang dihitung nilai faktorialnya cenderung statis. Berbeda dengan program "b" yang menggunakan fungsi, proses yang berjalan bisa lebih satu kali, yaitu kita bisa melakukan pemanggilan fungsi tersebut secara berulang dengan nilai angka yang parameter faktorialnya dapat diubah dengan lebih dinamis.

Tugas Pendahuluan

1. Apa yang anda ketahui tentang exception handling? Dan apa fungsinya?
2. Jelaskan bentuk umum dari exception handling! Dan sebutkan syarat yang harus dipenuhi untuk membuat sebuah exception handling.
3. Tuliskan sebuah contoh program exception handling sederhana! Dengan menggunakan 3 blok yaitu try-catch-finally.

4. Apa sebenarnya method itu? Method ada yang dinamakan dengan method rekursif, apa methodrekursif itu? Jelaskan!
5. Ada berapa macam tipe method di java? Sebutkan dan berikan contoh! Maksimal 5 baris kode untuk contoh setiap tipe method! Serta panggil method tersebut dalam main programnya!
6. Beri contoh sebuah method rekursif!
7. Java adalah bahasa pemrograman bersifat objek oriented (OOP) karena semua variabel, class, fungsi/method diakses dalam bentuk objek, dan dalam setiap objek tersebut terdapat akses modifier yang nilai defaultnya adalah public. Apa akses modifier itu? Sebutkan macam akses modifier, jelaskan serta berikan contohnya!

Percobaan Dilakukan

1. Ketik kembali kode sederhana berikut ini:

Method Tanpa Nilai Kembalian (Prosedur)

```
public class segiempat {  
    public void segiempat(int panjang, int lebar){  
        double luas;  
        double keliling;  
        luas=panjang*lebar;  
        System.out.println("-----");  
        System.out.println("Panjang Segi Empat = "+panjang);  
        System.out.println("Lebar Segi Empat = "+lebar);  
        System.out.println("Luas Segi Empat = "+luas);  
    }  
    public static void main(String args[]){  
        segiempat se=new segiempat();  
        se.segiempat(10,5);  
    }  
}
```

Input Keyboard Dengan Pemanggilan Method dengan nilai kembalian (Fungsi)

```
import java.io.*;  
public class inputconsole {  
    public static String readString(){  
        BufferedReader bfr=new BufferedReader(new InputStreamReader(System.in));  
        String string=" ";  
        try{  
            string=bfr.readLine();  
        }catch(IOException ex){  
            System.out.println(ex);  
        }return string;  
    }  
    public static int readInt(){  
        return Integer.parseInt (readString());  
    }  
    public static double readDouble(){  
        return Double.parseDouble(readString());  
    }  
}
```



```

public class bacaKonsol2 {
    public static void main(String args[]){
        System.out.print("data1 = ");
        inputconsole ic=new inputconsole();
        int data1= ic.readInt();
        System.out.print("data2 = ");
        int data2= ic.readInt();
        int data3=data1+data2;
        System.out.println("data1 + data2 = "+data3);
    }
}

```

Paduan Antara Fungsi Dan Prosedur dalam 1 kelas

```

public class metodx {
    // Fungsi
    public double luas_lingkaran(int diameter) {
        int jari2=diameter/2;
        double luas=Math.PI * Math.pow(jari2,2);
        return luas;
    }
    //Prosedur
    public void hitungLuasLingkaran(int jari2){
        double luas=Math.PI * Math.pow(jari2,2);
        System.out.println(luas);
    }
    public static void main(String[]args) {
        metodx mt=new metodx();
        mt. hitungLuasLingkaran(10);
        System.out.print(mt.luas_lingkaran(20));
        System.exit(0);
    }
}

```



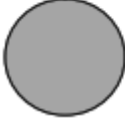
Instruksi yang harus dijalankan:

- Ketik kembali kode sederhana diatas, Compile dan jalankan programnya!
- Buat method baru yang menangani proses menghitung volume bangun yang mempunyailingkaran misal tabung, kerucut, bola. Nilai diameter dimasukkan lewat console!

2. Buat sebuah fungsi yang bisa menghitung nilai:

$$y = \sum_{i=1}^n (x + 2i)$$

3. Kali ini bantu Budi menyelesaikan permasalahannya dengan membuat sebuah program sederhana untuk menghitung luas bangun tersebut **jika nilai dari variable yang digunakan diinputkan menggunakan JOptionPane dan perhitungan masing masing bangun ditangani oleh sebuah method. Gunakan method dengan parameter!**

Bangun	Rumus
	Panjang * Lebar Panjang :20 , Lebar :30.
	$\frac{1}{2} * \text{Alas} * \text{Tinggi}$ Alas : 20 , Tinggi : 10
	$\text{Phi} * r * r$ Jari-jari : 10

MODUL 8**DASAR GUI (Graphical User Interface)****A. Target Pembelajaran**

1. Memahami persamaan dan perbedaan antara AWT dan Swing
2. Perbedaan antara komponen dan kontainer.
3. Mendesain aplikasi GUI menggunakan AWT.
4. Mendesain aplikasi GUI menggunakan Swing.
5. Menjelaskan tentang flow layout, border layout, dan grid layout dalam komponen GUI
6. Membuat tampilan yang kompleks dalam mendesain aplikasi GUI.

B. Materi Dasar

Tanpa mempelajari tentang graphical user interface (GUI) API, Anda masih tetap bisa membuat suatu program. Tetapi, program Anda akan kelihatan tidak menarik dan tidak nyaman digunakan bagi para user. Memiliki GUI yang baik dapat memberi efek pada penggunaan aplikasi. Java menyediakan banyak tool seperti Abstract Windowing Toolkit dan Swing untuk mengembangkan aplikasi GUI yang interaktif.

AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java dan applet. Tidak seperti beberapa komponen AWT yang menggunakan native code, keseluruhan Swing ditulis menggunakan bahasa pemrograman Java. Swing menyediakan implementasi platformindependent dimana aplikasi yang dikembangkan dengan platform yang berbeda dapat memiliki tampilan yang sama. Begitu juga dengan AWT menjamin tampilan look and feel pada aplikasi yang dijalankan pada dua mesin yang berbeda menjadi terlihat sama. Swing API dibangun dari beberapa API yang mengimplementasikan beberapa jenis bagian dari AWT. Kesimpulannya, komponen AWT dapat digunakan bersama dengan komponen Swing.

1. Beberapa komponen AWT

Berikut ini adalah daftar dari kontrol AWT. Kontrol adalah komponen seperti tombol atau textfield yang memungkinkan user untuk berinteraksi dengan aplikasi GUI. Berikut ini semua subclass dari class Components.

Label	Button	Choice
TextField	Checkbox	List
TextArea	CheckboxGroup	Scrollbar

Berikut adalah aplikasi membuat sebuah frame dengan kontrol yang telah dimasukkan di dalamnya.

```
import java.awt.*;

public class FrameWControls extends Frame{
    public static void main(String args[]){
        FrameWControls fwc = new FrameWControls();
        fwc.setLayout(new FlowLayout()); //akan dibahas pembahasan b
        fwc.setSize(600, 600);
        fwc.add(new Button("Test Me!"));
        fwc.add(new Label("Label"));
        fwc.add(new TextField());
        CheckboxGroup cbg = new CheckboxGroup();
        fwc.add(new Checkbox("chk1", cbg, true));
        fwc.add(new Checkbox("chk2", cbg, false));
        fwc.add(new Checkbox("chk3", cbg, false));
        fwc.setVisible(true);
    }
}
```

2. Layout Manager

Posisi dan ukuran suatu komponen ditentukan oleh layout manager. Layout manager mengatur tampilan dari komponen di dalam kontainer. Berikut ini beberapa layout manager yang terdapat di dalam Java.

1. FlowLayout
2. BorderLayout
3. GridLayout

Layout manager dapat diatur menggunakan method `setLayout` dari class `Container`. Method inidapat ditulis sebagai berikut.

```
setLayout(LayoutManager mgr)
```

2.1 FlowLayout

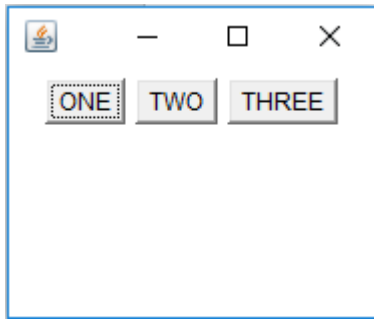
Cara meletakkan komponen dari FlowLayout Manager dimulai dari kiri ke kanandan dari atas ke bawah, dimulai dari pojok kiri atas. Seperti pada saat Anda mengetikmenggunakan editor kata pada umumnya.

Berikut adalah contoh penggunaan FlowLayout:

```
import java.awt.*;

class FlowLayoutDemo extends Frame {
    public static void main(String args[]) {
        FlowLayoutDemo fld = new FlowLayoutDemo();
        fld.setLayout(new FlowLayout());
        fld.add(new Button("ONE"));
        fld.add(new Button("TWO"));
        fld.add(new Button("THREE"));
        fld.setSize(100, 100);
        fld.setVisible(true);
    }
}
```

Berikut ini tampilan yang dihasilkan:



2.2 BorderLayout

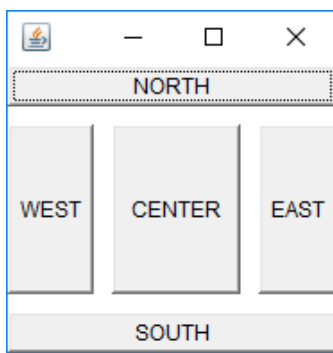
BorderLayout membagi kontainer menjadi lima bagian diantaranya utara, selatan, timur, barat, dan tengah. Setiap komponen dimasukkan membentuk jalur horizontal sedangkan region timur dan barat membentuk jalur vertikal. Dan region tengah berada pada perpotongan jalur horizontal dan vertikal. Tampilan ini adalah bersifat default untuk object Window, termasuk object dari subclass Window yaitu tipe Frame dan Dialog.

Berikut adalah contoh penggunaan BorderLayout:

```
import java.awt.*;

class BorderLayoutDemo extends Frame {
    public static void main(String args[]) {
        BorderLayoutDemo bld = new BorderLayoutDemo();
        bld.setLayout(new BorderLayout(10, 10)); //may remove
        bld.add(new Button("NORTH"), BorderLayout.NORTH);
        bld.add(new Button("SOUTH"), BorderLayout.SOUTH);
        bld.add(new Button("EAST"), BorderLayout.EAST);
        bld.add(new Button("WEST"), BorderLayout.WEST);
        bld.add(new Button("CENTER"), BorderLayout.CENTER);
        bld.setSize(200, 200);
        bld.setVisible(true);
    }
}
```

Berikut ini tampilan yang dihasilkan:



2.3 Null Layout/Tanpa Layout

Jika Anda memilih untuk tidak menggunakan layout manager, Anda dapat mengisi null sebagai argumen untuk method ini. Tetapi selanjutnya, Anda akan mengatur posisi elemen secara manual dengan menggunakan method `setBounds` dari class `Components`.

```
setBounds(int x, int y, int width, int height)
```

Method ini mengatur posisi berdasarkan pada argumen x dan y, dan ukuran berdasarkan argumen width dan height. Berikut adalah contoh penggunaan NullLayout:

```
import javax.swing.*;
import java.awt.*;

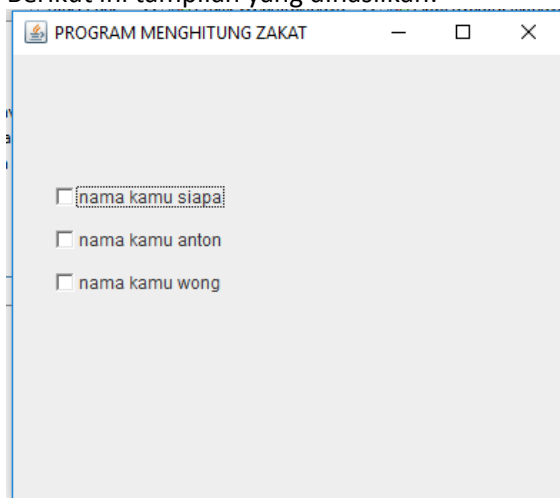
class cekBok extends JFrame {
    Checkbox unta = new Checkbox("nama kamu siapa");
    Checkbox kuda = new Checkbox("nama kamu anton");
    Checkbox kuda2 = new Checkbox("nama kamu wong");
    CheckboxGroup checkboxGroup = new CheckboxGroup();

    cekBok() {
        super("PROGRAM MENGHITUNG ZAKAT");
        setLocation(200,100);
        setSize(400,350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    void objek() {
        getContentPane().setLayout(null);
        getContentPane().add(untar);
        getContentPane().add(kuda);
        getContentPane().add(kuda2);
        unta.setBounds(30, 90, 150, 20);
        kuda.setBounds(30, 120, 150, 20);
        kuda2.setBounds(30, 150, 150, 20);
        setVisible(true);
    }

    public static void main(String args[]) {
        cekBok ZZ=new cekBok();
        ZZ.objek();
    }
}
```

Berikut ini tampilan yang dihasilkan:



3. Komponen Swing

Seperti pada package AWT, package dari Swing menyediakan banyak class untuk membuat aplikasi GUI. Package tersebut dapat ditemukan di javax.swing. Perbedaan utama antara keduanya Java. Kesimpulannya, program GUI ditulis menggunakan banyak class dari package Swing yang mempunyai tampilan look and feel yang sama meski dijalankan pada platform yang berbeda. Lebih dari itu, Swing menyediakan komponen yang lebih menarik seperti color chooser dan option pane.

Nama dari komponen GUI milik Swing hampir sama persis dengan komponen GUI milik AWT. Perbedaan jelas terdapat pada penamaan komponen. Pada dasarnya, nama komponen Swing sama dengan komponen dalam AWT tetapi dengan tambahan huruf J pada prefixnya. Sebagai contoh, satu komponen dalam AWT adalah `button class`. Sedangkan pada Swing, nama komponen tersebut menjadi `Jbutton class`. Berikut adalah daftar dari komponen Swing.

<i>Komponen Swing</i>	<i>Penjelasan</i>
JComponent	class induk untuk semua komponen Swing, tidak termasuk top-level kontainer
JButton	Tombol "push". Berhubungan dengan class <code>button</code> dalam package AWT
JCheckBox	Item yang dapat dipilih atau tidak oleh pengguna. Berhubungan dengan class <code>checkbox</code> dalam package AWT
JFileChooser	Mengijinkan pengguna untuk memilih sebuah file. Berhubungan dengan class <code>filechooser</code> dalam package AWT
JTextField	Mengijinkan untuk mengedit text satu baris. Berhubungan dengan class <code>textfield</code> dalam package AWT.
JFrame	Turunan dan Berhubungan dengan class <code>frame</code> dalam package AWT tetapi keduanya sedikit tidak cocok dalam kaitannya dengan menambahkan komponen pada kontainer. Perlu mendapatkan content pane yang terbaru sebelum menambah sebuah komponen.
JPanel	Turunan Jcomponent. Class Container sederhana tetapi bukan top-level. Berhubungan dengan class <code>panel</code> dalam package AWT.
JApplet	Turunan dan Berhubungan dengan class <code>Applet</code> dalam package AWT. Juga sedikit tidak cocok dengan class <code>applet</code> dalam kaitannya dengan menambahkan komponen pada container
JOptionPane	Turunan Jcomponent. Disediakan untuk mempermudah menampilkan pop-up kotak dialog.
JDialog	Turunan dan Berhubungan dengan class <code>dialog</code> dalam package AWT. Biasanya digunakan untuk menginformasikan sesuatu kepada pengguna atau prompt pengguna untuk input.
JColorChooser	Turunan Jcomponent. Memungkinkan pengguna untuk memilih warna yang diinginkan.

Berikut ini contoh sederhana menggunakan komponen swing:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

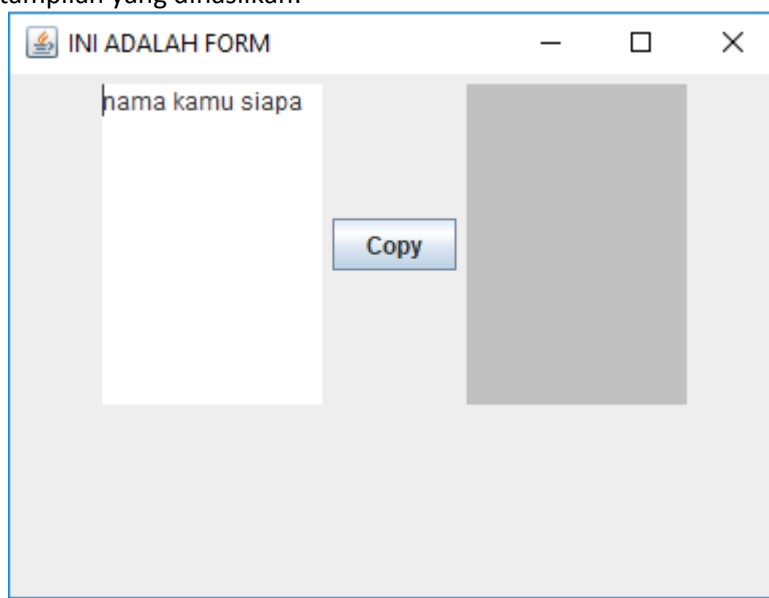
public class fil extends JFrame {
    JTextArea nama=new JTextArea(10,10);
    JButton bt=new JButton("Copy");
    JTextArea txnama=new JTextArea(10,10);

    fil(){
        super("INI ADALAH FORM");
        setLocation(200,300);
        setSize(400,300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    void tampilan(){
        getContentPane().add(nama);
        nama.append("nama kamu siapa");
        getContentPane().add(bt);
        getContentPane().add(txnama);
        txnama.setBackground(Color.LIGHT_GRAY);
        bt.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e){
                txnama.append(nama.getSelectedText());
            }
        });
        getContentPane().setLayout(new FlowLayout());
        setVisible(true);
    }

    public static void main(String args[]){
        fil f=new fil();
        f.tampilan();
    }
}
```

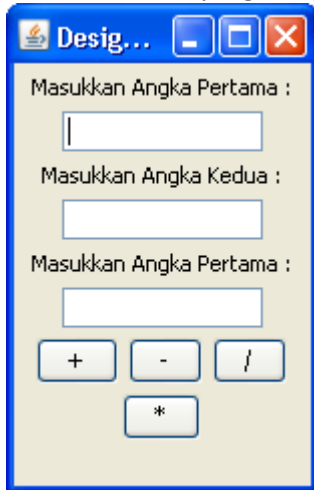
Berikut ini tampilan yang dihasilkan:



Perlu diperhatikan pada package java.awt masih saja diimpor karena layout manager yang digunakan terdapat pada package tersebut. Juga, memberi judul pada frame dan mengepack komponen di dalam frame dapat juga dilakukan untuk frame AWT.

Tugas Pendahuluan

1. Jelaskan apa yang anda pahami tentang GUI! komponen pendukung GUI dijava ditangani oleh awt dan swing, apa beda awt dan swing?
2. Apa yang anda ketahui tentang layout manager? Sebutkan dan jelaskan beberapa macam layout manager java!
3. Anda buat kode program yang bisa menampilkan output berikut ini:



Percobaan Dilakukan

1. Ketik Kode berikut ini!

GUI Dengan FlowLayout

```
import javax.swing.*;
import java.awt.*;

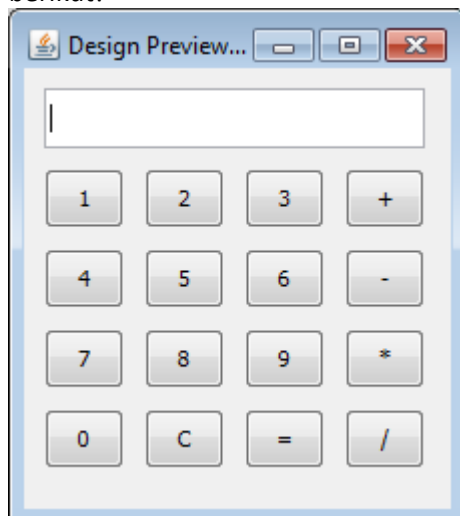
class CobaGUI_1 extends JFrame{
    Button cb1 = new Button ("Pilih A");
    Button cb2 = new Button("Pilih B");
    CobaGUI_1(){
        setTitle("Program GUI Pertamaku");
        setLocation(200,100);
        setSize(200,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    void objek(){
        getContentPane().setLayout(new FlowLayout());
        getContentPane().setBackground(Color.GREEN);
        cb1.setBackground(Color.MAGENTA);
        getContentPane().add(cb1);
        getContentPane().add(cb2);
        setVisible(true);
    }
    public static void main(String args[]){
        CobaGUI_1 ZZ=new CobaGUI_1();
        ZZ.objek();
    }
}
```

GUI Dengan NULL Layout (Tanpa Layout)

```
import javax.swing.*;
import java.awt.*;

class CobaGUI extends JFrame{
    Checkbox cb1 = new Checkbox("Pilih A");
    Checkbox cb2 = new Checkbox("Pilih B");
    Checkbox cb3 = new Checkbox("Pilih C");
    CobaGUI () {
        setTitle("Program GUI Pertamaku");
        setLocation(200,100);
        setSize(400,350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    void objek() {
        getContentPane().setLayout(null);
        getContentPane().setBackground(Color.GREEN);
        getContentPane().add(cb1);
        getContentPane().add(cb2);
        getContentPane().add(cb3);
        cb1.setBounds(30,90,150,20);
        cb2.setBounds(30,120,150,20);
        cb3.setBounds(30,150,150,20);
        setVisible(true);
    }
    public static void main(String args[]){
        CobaGUI ZZ=new CobaGUI ();
        ZZ.objek();
    }
}
```

2. Dengan menggunakan null layout Buat program yang bisa menampilkan output sebagai berikut!



3. Tambahkan warna pada komponen tombol yang telah anda buat tersebut!

MODUL 9**EVENT HANDLING/ACTION PROGRAM****A. Target Pembelajaran**

1. Menerangkan komponen-komponen delegation event model
2. Mengerti bagaimana delegation event model bekerja
3. Menciptakan aplikasi GUI yang berinteraksi dengan user
4. Mendiskusikan manfaat dari class-class adapter
5. Mendiskusikan keuntungan-keuntungan dari menggunakan inner dan anonymous

B. Materi Dasar

Pada bab ini, Anda akan belajar bagaimana mengendalikan events triggered ketika user berinteraksi dengan aplikasi GUI Anda. Setelah menyelesaikan modul ini, Anda akan dapat mengembangkan aplikasi GUI yang dapat merespon interaksi user.

1. Event Model

Delegasi event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Untuk memahami model, pertama-tama mari kita pelajari melalui tiga komponen utamanya.

- **Event Source**
Event source mengacu pada komponen GUI yang meng-generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.
- **Event Listener/Handler**
Event listener menerima berita dari event-event dan proses-proses interaksi user. Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.
- **Event Object**
Ketika sebuah event terjadi (misal, ketika user berinteraksi dengan komponen GUI), sebuah object event diciptakan. Object berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah terjadi, seperti ketika mouse telah diklik. Ada beberapa class event untuk kategori yang berbeda dari user action. Sebuah event object mempunyai tipe data mengenai salah satu dari class ini.

2. Event Listeners

Event Listeners	Deskripsi	Method
ActionListener	Bereaksi atas perubahan mouse atau keyboard.	public void actionPerformed(ActionEvent e)
MouseListener	Bereaksi atas pergerakan mouse.	<ul style="list-style-type: none">- public void mouseClicked(MouseEvent e)- public void mouseEntered(MouseEvent e)- public void mouseExited(MouseEvent e)- public void mousePressed(MouseEvent e)- public void mouseReleased(MouseEvent e)
MouseMotionListener	Interface Mouse Motion Listener mendukung MouseListener. Menyediakan method-method yang akan memantau pergerakan mouse, seperti drag dan pemindahan mouse.	<ul style="list-style-type: none">- public void mouseDragged(MouseEvent e)- public void mouseMoved(MouseEvent e)

Event listeners adalah class yang mengimplementasikan interfaces <Type>Listener. Tabel diatas menunjukkan beberapa listener interfaces yang biasanya digunakan. Pada praktikum ini kita hanya akan membahas Event ActionListener yaitu penanganan reaksi ketika ada aksi dari mouse ketika di klik dan ketika keyboard ditekan, sedangkan untuk event lainnya silahkan kembangkan sesuai kreatifitas anda.

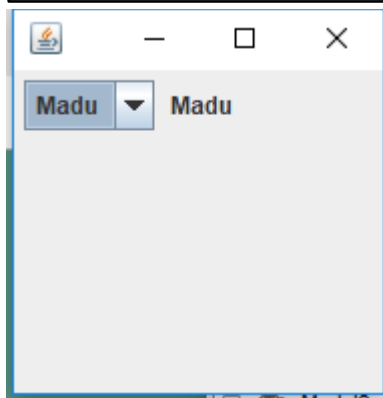
2.1 ActionListener Pada Mouse

Berikut contoh Aplikasi dengan action listener

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

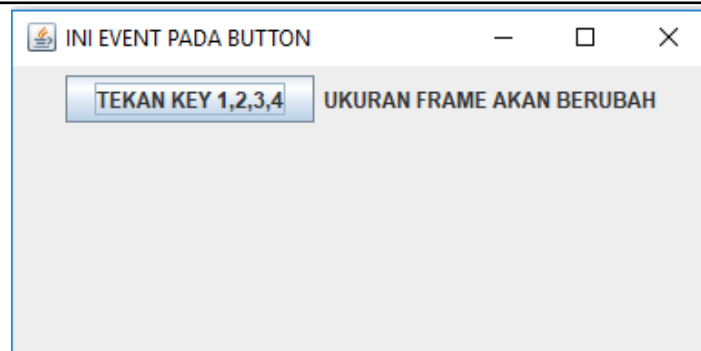
public class combo {

    public static void main(String[] args) {
        JFrame frame = new JFrame();
        final JLabel lb = new JLabel("");
        final String[] items = {" roti ", " Madu "};
        final JComboBox comboBox = new JComboBox(items);
        JPanel pn = new JPanel();
        frame.add(pn, BorderLayout.WEST);
        pn.add(comboBox, BorderLayout.CENTER);
        pn.add(lb, BorderLayout.NORTH);
        comboBox.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                String mn = (String) comboBox.getSelectedItem();
                lb.setText(mn);
            }
        });
        frame.setSize(200, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```



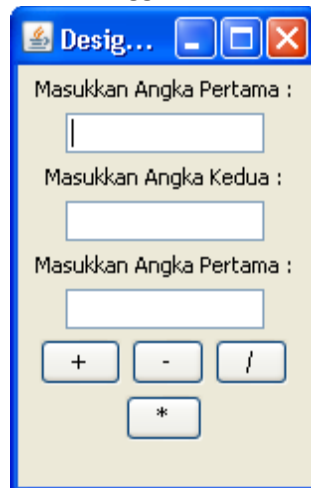
2.2 ActionListener Pada Keyboard

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*; //Package untuk penanganan event (aksi reaksi)
class Event_Button_Key extends JFrame {
    JButton m = new JButton("TEKAN KEY 1,2,3,4");
    JLabel tulisan = new JLabel("UKURAN FRAME AKAN BERUBAH");
    Event_Button_Key() {
        setTitle("INI EVENT PADA BUTTON");
        setSize(400, 200);
        setLocation(200, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    void komponen() {
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(m);
        getContentPane().add(tulisan);
        setVisible(true);
    }
    void reaksi() {
        m.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                if (e.getKeyCode() == e.VK_1) {
                    setSize(300, 500);
                }
                if (e.getKeyCode() == e.VK_2) {
                    setSize(200, 100);
                }
                if (e.getKeyCode() == e.VK_3) {
                    setSize(100, 10);
                }
                if (e.getKeyCode() == e.VK_4) {
                    setSize(900, 200);
                }
            }
        });
    }
    public static void main(String[] args) {
        Event_Button_Key eb = new Event_Button_Key();
        eb.komponen();
        eb.reaksi();
    }
}
```



Tugas Pendahuluan

1. Jelaskan apa yang dimaksud delegation event model? Dan sebutkan macamnya, serta jelaskan!
2. Apa yang anda ketahui tentang event listener? Ada berapa macam? jelaskan, serta sebutkan method methodnya!
3. Berikut ini langkah-langkah yang Anda butuhkan untuk membuat aplikasi GUI dengan event handling.
 - a. Buatlah sebuah class yang menguraikan dan membuat suatu tampilan dari aplikasi GUI.
 - b. Buatlah sebuah class yang menerapkan interface listener yang sesuai. Class ini boleh mengacu pada class yang sama seperti pada langkah awal.
 - c. Dalam menerapkan class, gunakan semua method-method dengan interface listener yang sesuai. Uraikan masing-masing method bagaimana Anda ingin mengendalikan event-event. Anda dapat memberikan implementasi kosong untuk method yang tidak ingin Anda gunakan.
 - d. Daftarkan object listener, instansiate dari class listener pada langkah 2, dengan source component menggunakan method `add<Type>Listener`.



Terapkan beberapa point yang telah disebutkan untuk menyelesaikan GUI berupa kalkulatorsederhana sesuai dengan gambar diatas! Gunakan mouse listener clicked

Percobaan Dilakukan

1. Ketik Kode berikut ini!

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Coba_Aksi_Reaksi extends JFrame {
    JTextArea nama = new JTextArea(10, 10);
    JButton bt = new JButton("Copy");
    JTextArea txnama = new JTextArea(10, 10);
    Coba_Aksi_Reaksi() {
        super("Coba Event Handling");
        setLocation(200, 300);
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    void tampilan() {
        getContentPane().add(nama);
        nama.append("nama kamu siapa");
        getContentPane().add(bt);
        getContentPane().add(txnama);
        txnama.setBackground(Color.LIGHT_GRAY);
        getContentPane().setLayout(new FlowLayout());
        setVisible(true);
    }
    void aksi_reaksi() {
        bt.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                txnama.append(nama.getSelectedText());
            }
        });
    }
    public static void main(String args[]) {
        Coba_Aksi_Reaksi f = new Coba_Aksi_Reaksi();
        f.tampilan();
        f.aksi_reaksi();
    }
}
```

2. Tambah event handling dari tugas praktikum sebelumnya (Nomor 2)! Gunakan mouse listener! Target aplikasi tersebut mampu menghitung layaknya sebuah kakulator sederhana!

MODUL 10**DASAR PEMROGRAMAN DATABASE DENGAN JAVA****A. Target Pembelajaran**

1. Mahasiswa mengetahui struktur dasar pemrograman database.
2. Mahasiswa mengetahui cara pengoperasian simpan, edit, cari dan hapus suatu data dalam database.
3. Mahasiswa mampu menyelesaikan permasalahan menggunakan pemrograman database.

B. Materi Dasar**1. Data Base**

Data Base merupakan sebuah cara dimana kita bias menyimpan data secara permanent kedalam media penyimpanan. Agar dapat terkoneksi ke MySQL dari Java maka diperlukan driver JDBC untuk MySQL. Silakan download di situs web mysql.com, diperoleh file **mysql-connector-java-5.1.8.tar** (terbaru saat ini). Kemudian ekstrak file tersebut. Salah satu hasil ekstraksinya adalah file **mysqlconnector-java-5.1.8-bin.jar** dan inilah driver JDBC MySQL yang dimaksud. Salin file ini ke direktori **C:\Program Files\Java\jre6\lib\ext** atau ke folder lain sesuai dengan lokasi instalasi JRE.

2. Perintah SQL

Structured Query Language (SQL) telah diakui sebagai bahasa standart untuk mengakses dan manipulasi database. Pada pembahasan kali ini kita akan tunjukkan beberapa intruksi SQL dan dibedakan menjadi 2 macam perintah yaitu:

2.1 DML (Data manipulation Language)

Perintah DML adalah perintah query yang digunakan untuk memanipulasi data pada suatu database. Berikut perintah yang ada pada DML,

a. Insert

```
insert into NamaTabel
values('"+field_1+"', '"+Field_2+"', '"+...+"', '"+Field_ke_n+"');
```

b. Select

Pernyataan Select digunakan untuk mengambil data dari table dalam database sesuai dengan kriteria yang ditentukan. Berikut ini adalah contoh perintah Select:

```
SELECT * FROM NamaTabel ; Mengambil seluruh data tanpa terkecuali.
```

```
SELECT field_1, field_2 FROM NamaTabel ; Mengambil data dari table Data field Kodesaja
```

c. Update

```
update NamaTabel set field_1='"+ value_1+"', field_2='"+
value_2+"', field_3='"+ value_3+"' where field_x='"+value_4+"';
```

d. Delete

```
delete from NamaTabel "+ "where Field_x='"+value_x+'";
```

2.2 DDL (Data Definition Language)

Perintah DDL adalah perintah query yang digunakan untuk membangun atau merencanakan database.

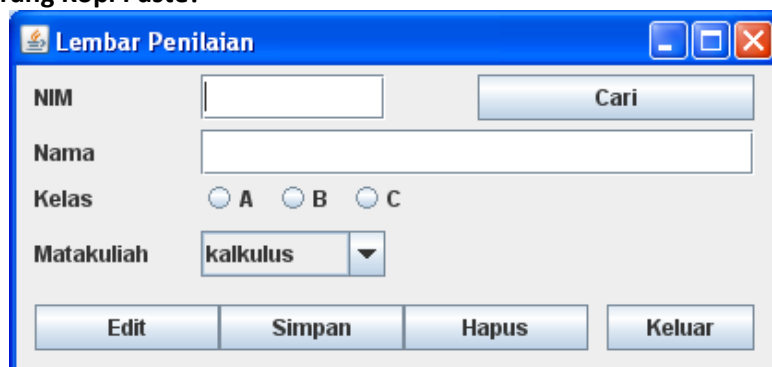
Berikut perintah yang ada pada DDL,

- a. Create
- b. Alter
- c. Drop

Dalam praktikum ini kita hanya akan mempelajari proses DML(Data Manipulation Language) yaitu proses yang nantinya kita gunakan untuk memanipulasi record yang terdapat dalam table dari database yang telah kita buat.

Tugas Pendahuluan

1. Apa yang anda ketahui tentang database?
2. Ketika membangun aplikasi berbasis database dengan java, anda membutuhkan suatu library tambahan yang dinamakan JDBC API (Application Program Interface). Apa yang anda ketahui tentang JDBC API? Silahkan sebutkan tugas dan kelebihanannya!
3. Untuk memberikan perintah kepada sebuah database anda harus menguasai bahasa SQL (Structure Query Language) yang merupakan bahasa standart yang dimiliki oleh database. SQL dibedakan menjadi 2 macam yaitu DML yang akan kita bahas garis besarnya di praktikum ini dan DDL yang akan anda pelajari di matakuliah desain database. Sebutkan perintah yang dimiliki DML (Khusus proses Simpan dan Hapus) dan berikan contoh nya.
4. Sebutkan tahap menambahkan library Mysql connector di proyek aplikasi java yang telah anda buat dengan menggunakan Netbeans. Diasumsikan nama projectnya adalah nim anda masing masing!
5. Buat program yang bisa menampilkan tampilan berikut ini, lengkap dengan Event handlingnya! **(Kode program akan dipakai ketika praktikum, jadi wajib anda selesaikan dan dibawa ketika praktikum)** Dilarang Kopi Paste!



Catatan : Untuk asisten, mohon tugas no5 dan 6 dilakukan koreksi kode programnya dan dicek databasenya.

6. Buat database dan desain tabel yang sesuai dengan tampilan tersebut, nama tabel adalah nama_nim, contoh : puspa_10650038

Percobaan Dilakukan

1. Koneksikan tugas pendahuluan nomer 5 dan 6 sehingga semua fungsi yang ada bias berjalan!
2. Dari handout materi praktikum dasar dasar pemrograman, silahkan anda membentuk sebuah kelompok untuk TUGAS AKHIR mata kuliah ini dengan ketentuan maksimal jumlah mahasiswa adalah 4, dan minimal 3 orang.
 - Buat proposal rencana pembuatan program sederhana dengan mengimplementasikan materi praktikum dari awal hingga akhir. (Hingga Database)
 - Proposal dibuat 1 kelompok 1 proposal.
 - Konsultasikan dengan Aslab masing-masing tentang judul program yang akan diajukan (untuk menentukan kelayakan)
 - Jika judul sudah di setujui, kerjakan program tersebut sebagai TUGAS AKHIR matakuliah (Nilai UAS), dan akan di presentasikan di waktu UAS.
 - Jangan lupa untuk membuat dan mengumpulkan Laporan Akhir Program (PrintOut) + SourceCode (CD).

