

# Dasar-dasar Pemrograman Java

**Akhmad Sofwan**

sofwan@sofwan.net

<http://www.sofwan.net>

## **Lisensi Dokumen:**

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Untuk membuat sebuah aplikasi, diperlukan bahasa pemrograman dan salah satu bahasa pemrograman populer yang terkenal tangguh adalah Java. Pada saat tulisan ini di launching di [ilmukomputer.org](http://ilmukomputer.org), Juli 2018, menurut situs [tiobe.com](http://tiobe.com), Java adalah bahasa pemrograman yang paling banyak digunakan di seluruh dunia, disusul oleh C dan C++. Java dapat digunakan untuk membuat aplikasi berbasis konsole atau text, GUI, web dan mobile device. Pada tutorial ini, penulis memperkenalkan dasar-dasar Java yang dijalankan di konsol dan juga GUI dengan menggunakan Swing. Java bersifat platform-independence, artinya, aplikasi yang dibuat dengan Java dapat dijalankan di platform atau Sistem Operasi populer seperti Windows, Linux dan Macintosh tanpa harus merubah source code aplikasi.

Pada tutorial ini akan dibahas materi java dengan konsole dan melalui GUI dengan menggunakan Swing.

Tutorial ini merupakan tutorial yang penulis gunakan dalam kuliah pemrograman Java yang penulis ajar di sebuah kampus di Tangerang.

## **Pendahuluan**

Dalam setiap pembelajaran sebuah bahasa program, selalu dimulai dengan mempelajari dasar-dasar bahasa pemrograman tersebut, terutama bagi para pemula, termasuk juga dalam mempelajari bahasa pemrograman Java.

Untuk memulai belajar pemrograman Java, kita harus mempersiapkan perangkat-perangkatnya, yaitu: komputer yang sudah terinstall Java Development Kit (JDK) dan editor atau IDE. JDK adalah sebuah aplikasi di mana program java dijalankan diatas nya, sedangkan editor, kita dapat menggunakan beberapa editor, seperti Netbeans ([www.netbeans.org](http://www.netbeans.org)) dan Eclipse ([www.eclipse.org](http://www.eclipse.org)).

## **Bab I**

### **Mengenal Java**

Java di ciptakan oleh James Gosling yang bekerja pada Sun Microsystems dan secara resmi di perkenalkan oleh Sun Microsystems pada kuartal kedua tahun 1995. Bahasa ini dengan cepat menjadi tool pilihan untuk mengembangkan aplikasi internet. Java menjadi semakin populer ketika Netscape Communication memutuskan untuk memberikan dukungan pada Java dalam produk browser nya Netscape Navigator 2.0. Langkah ini di ikuti juga oleh Microsoft yang kemudian memberikan dukungan pada Java sejak Internet Explorer 3.0.

File kode Java atau class yang di buat memiliki extension java, misal nilai.java.  
Lambang Java adalah secangkir Kopi.

#### **Apa itu Java ?**

Karakteristik Java adalah sebagai bahasa pemrograman yang “Sederhana, Berorientasi Object, Bertipe Statik, Terkompilasi, Multi-Threaded, Garbage Collected, Neutral-architecture, robust, aman dan dapat di perluas.

#### **\* Simple**

Pengembang Java banyak membuang fitur-fitur yang tidak di perlukan seperti yang di miliki oleh bahasa pemrograman tingkat tinggi lainnya. Contoh, Java tidak mendukung pointer matematis, casting tipe implisit dan header file. Java di buat dengan mengambil banyak hal dari C++ dan membuang hal lain dari C++, namun tetap mempertahankan kehandalan sebagai bahasa pemrograman. Java juga lebih mudah di pelajari di banding C++.

#### **\* Berorientasi-Object**

Seperti hal nya C++, Java menggunakan class-class untuk mengelola kode ke dalam modul-modul logika. Pada saat runtime, program menciptakan object dari class yang ada. Class Java dapat merupakan warisan dari class lain, tetapi pewarisan majemuk di mana suatu class mewariskan metode-metode dan field yang berasal lebih dari satu class tidak di perkenankan.

#### **\* Terkompilasi**

Source code Java di kompilasi oleh kompilator Java dengan perintah Javac. Hasil kompilasi berupa file kode byte (byte-code). Kemudian Interpreter membaca file kode byte dan menterjemahkannya menjadi perintah bahasa mesin yang dapat di eksekusi langsung oleh mesin yang menjalankan program Java dengan menggunakan perintah Java. Jadi dapat di katakan, bahwa Java merupakan bahasa yang di kompilasi sekaligus di interpretasi.

#### **\* Multi Threaded**

Program Java dapat berisi eksekusi multiple threads, yang memungkinkan program mengulas citra pada layar pada satu thread, sementara tetap menerima masukan dari Keyboard pada thread utama nya.

#### **\* Garbage Collected**

Program Java tidak perlu menghapus object-object yang di alokasinya di memori. Ia menjadikan pemrogram bebas dari masalah-masalah pengelolaan memori.

#### **\* Bebas Arsitektur / Architectural Neutral**

Pada dasar nya, Java tidak di rancang untuk prosesor ataupun Sisyem Operasi tertentu. Ia di rancang untuk bekerja pada berbagai arsitektur prosesor dan berbagai Sistem Operasi.

**\* Robus (Tangguh)**

Karena interpreter Java memeriksa seluruh akses System yang di lakukan program, maka program Java tidak akan membuat System menjadi Crash. Bila terjadi masalah serius, program Java membuat pengecualian (Exception). Exception ini dapat di tangani dan di kelola oleh program tanpa beresiko memacetkan System.

**\* Aman**

System Java tidak hanya memverifikasi seluruh akses ke memori, tetapi juga menjamin bahwa tidak ada virus yang membonceng pada applet yang sedang berjalan. Karena Java tidak mendukung pointer, program tidak dapat memperoleh akses ke arena system di mana ia tidak mendapatkan otorisasi.

**\* Dapat di perluas**

Program Java mendukung metode native, yakni fungsi-fungsi yang di tulis dalam bahasa lain, biasa nya C++. Dukungan terhadap metode native ini memungkinkan pemrogram menulis fungsi yang bisa di eksekusi secara lebih cepat daipada fungsi yang sama yang di tulis dalam Java.

**Java Platform**

Platform adalah lingkungan hardware atau software tempat berjalannya suatu program. Sebagai contoh adalah Linux, Windows dan Macintosh.

Java Platform memiliki 2 komponen, yaitu :

- Java Virtual Machine
- Java Application Programming Interface (API)

**Java Features**

1. Development Tools : Development tools menyediakan segala nya untuk compiling, running, monitoring, debugging dan mendokumentasikan aplikasi kita. Sebagai java developer baru, tools utama yang anda butuhkan adalah *javac* compiler, *java launcher* dan *javadoc* documentation tool.
2. Application Programming Interface (API) :API menyediakan fungsi inti Java. API menawarkan cakupan luas dari penggunaan class yang siap di gunakan di dalam aplikasi anda yang meliputi segala hal dari object dasar, ke networking dan security hingga XML dan akses Database. Core API sangat besar.
3. Deployment Technologies : JDK menyediakan mekanisme standard seperti Java Web Start Software dan Java Plug-in software untuk instalasi.
4. User Interface Toolkits : Swing dan Java2D memungkinkan anda untuk membuat GUI yang canggih.
5. Integration libraries : Integration libraries seperti Java IDL API, JDBC API dan Java RMI.

**Tool Java**

Salah satu tools Java adalah Java Developer's Kit dari Sun Microsystem. JDK dapat di download secara gratis di situs java.com atau oracle.com.

Setidak-tidak nya ada 7 program bantu (tool) yang tersedia pada JDK, seperti di tabulasikan dan di jelaskan pada tabel berikut ini :

<b>Nama Program Bantu</b>	<b>Deskripsi</b>
javac	Kompilator Java untuk mengubah file sumber ke dalam file yang dapat di eksekusi. File ini di sebut sebagai file kode byte dan memiliki ekstensi .class

java	Interpreter java untuk mengeksekusi file kelas (.class) atau file kode byte.
Jdb	Debugger java, yang bekerja seperti interpreter mengeksekusi kelas Java dan juga memiliki kemampuan untuk menghentikan eksekusi program pada titik-titik henti terpilih dan untuk menampilkan nilai-nilai variable class. Kemampuan ini penting untuk mencari kesalahan program.
Javap	Disassemble Java, menerima file kode byte dan menampilkan class, variabel dan metode yang telah di kompilasi menjadi kode byte. Anda dapat menggunakan program bantu ini untuk mendapatkan kode sumber dari kelas Java yang sudah terkompilasi.
Appletviewer	Program bantu untuk menampilkan applet Java yang berada dalam halaman web.
Javadoc	Program bantu dokumentasi otomatis yang di gunakan untuk mengkonversi kode sumber Java ke dalam file HTML.
Javah	Program bantu file header C, di gunakan untuk membangkitkan header bahasa C.

### **Kompilasi Java**

Setelah anda menulis kode java anda, yang berekstensi .java, maka anda perlu mengcompile, dengan cara :

**javac nama\_file.java**

Misal : **javac bilangan.java**

Setelah sukses melakukan kompilasi, akan terbentuk file dengan ekstensi .class, untuk contoh dia atas, maka akan terbentuk : **javac.class**

Setelah itu, lakukan interpretasi dengan cara :

**java nama\_file**

Misal : **java bilangan**

### **Keunggulan Java**

1. **Dapat di pelajari dengan cepat (Get Started quickly).** Terutama untuk programmer yang sudah familiar dengan C atau C++.
2. **Kode lebih sedikit (Write less code).** Perbandingan dari metric program (jumlah class, jumlah method dan lainnya), membuat sebuah program dalam Java dapat lebih sedikit 4 kali di banding dengan C++.
3. **Menulis kode dengan lebih baik (Write better code).** Pemrograman Java membantu anda dalam hal mempraktekan pemrograman dengan baik dan automatic garbage collection membantu anda dalam menghindari memory leaks
4. **Membuat program lebih cepat (Develop program more quickly).** Di banding C++, Java lebih simple dan baris kode nya lebih sedikit, sehingga mengakibatkan program dapat di selesaikan secara lebih cepat.
5. **Menghindari ketergantungan platform (Avoid platform dependencies).** Anda dapat menjaga program anda portable dengan menghindari penggunaan library yang di tulis di bahasa lain.
6. **Menulis sekali, di jalankan di mana saja (Write once, run anywhere).** Program Java independent terhadap platform, karena di compile dalam JVM. Dapat di develop dan di jalankan di platform mana saja.
7. **Menyebarkan software dengan lebih mudah (Distribute software more easily).**

Dengan Java Web Start, users dapat menjalankan aplikasinya dengan hanya mengklik satu tombol mouse saja.

## Bab II Variabel dan Tipe data

Variabel adalah lokasi di memori di mana suatu nilai di simpan. Setiap variabel memiliki nama, tipe dan nilai. Sebelum dapat di gunakan, variabel harus di deklarasikan terlebih dahulu dan dapat langsung di beri nilai atau nilai di berikan setelah di deklarasikan.

Java memiliki 3 macam variabel, yaitu variabel Instance, variabel class dan variabel lokal. Variabel Instance di gunakan untuk mendefinisikan atribut dari object tertentu. Variabel Class sama dengan variabel instance, kecuali bahwa nilai nya di berikan ke seluruh instance class dan ke class itu sendiri. Variabel lokal di deklarasikan dan di pakai di dalam definisi metode. Variabel lokal di gunakan untuk menyimpan informasi yang di perlukan oleh satu metode dan variabel instance untuk menyimpan informasi yang di perlukan oleh beberapa metode dalam object. Meskipun ketiga jenis variabel di deklarasikan dengan cara yang sama, variabel class dan variabel Instance di akses dengan cara yang berbeda dari variabel lokal.

### **Pendeklarasian Variabel, Identifier dan Tipe Data**

Variabel memiliki nama, tipe data dan scope. Variabel yang di deklarasikan di dalam sebuah block atau metode di sebut variabel lokal yang musti di inisialisasi secara manual.

### **Jenis-jenis Tipe data :**

Integer : byte, short, int dan long  
Floating Point : float dan double  
Character : char  
Boolean : variabel dengan nilai true or false

### **contoh :**

```
class contoh {  
    public static void main (String args[])  
    {  
        // Deklarasi atau Inisialisasi  
        int nilai_1,nilai_2,hasil;  
        String judul;  
  
        nilai_1=10;  
        nilai_2=15;  
        Judul="Proses Penjumlahan";  
  
        // Proses  
        hasil=nilai_1+nilai_2;  
  
        // Output  
        System.out.println(judul);  
        System.out.println (nilai_1+"+"+nilai_2+"="+hasil);  
    }  
}
```

**Tipe data**

Tipe	Ukuran	Rentang Nilai
byte	8 bit	-128 sampai 127
Short	16 bit	-32.768 sampai 32.767
Int	32 bit	-2.147.483.648 sampai 2.147.483.647
Long	64 bit	-9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807

Tipe char di gunakan untuk karakter secara individual. Tipe char memiliki presisi 16 bit tak bertanda. Tipe boolean memiliki 2 nilai, yaitu true atau false.

**Pemberian Nilai ke Variabel**

Setelah variabel di deklarasikan, anda dapat memberikan nilai kepada variabel tsb, dengan cara seperti contoh :

```
bilangan_1=10;
nama="Sofwansan"
```

**Komentar**

Komentar dapat berupa :

```
/*
  Komentar di sini
  Di sini juga boleh
*/
```

atau dengan ini :

```
// Komentar nich ...
```

Kode Khusus :

Kode	Arti
\n	Baris Baru
\t	Tab
\b	Backspace
\r	Carriage return
\f	Formfeed01
\\	Backslash
\'	Tanda kutip tunggal
\"	Tanda kutip ganda
\ddd	Bilangan oktal
\xdd	Bilangan heksadesimal
\udddd	Karakter Unicode

Aritmetika

Operator	Arti	Contoh Operasi
+	Penjumlahan	4 + 5
-	Pengurangan	'10 - 8
*	Perkalian	2 * 5
/	Pembagian	05/04/10
%	Modulus (Sisa hasil bagi)	21 % 2

Pembandingan

Operator	Arti	Contoh
==	Sama dengan	X =6
!=	Tida sama dengan	X != 6
<	Kurang dari	X < 6
>	Lebih besar dari	X > 6
<=	Lebih kecil atau sama dengan dari	X <=6
>=	Lebih besar atau sama dengan dari	X >= 6

## Bab III Control

### I. Condition

#### I.1 If Bersyarat :

Merupakan kontrol untuk menjalankan suatu statement jika kondisi tertentu terpenuhi.

Bentuk umum :

```
if (kondisi)
  { statement 1 }
else
  { statement 2 }
```

atau

```
if (kondisi1)
  { statement 1 }
else if (kondisi 2)
  { statement 2 }
else
  { statement 3 }
```

Contoh :

```
class contoh1
{
  public static void main (String args[])
  {
    int a,b,c;
    a=5;
    b=7;
    c=a+b;
    if (c>10) // Jika c lebih besar dari 10
    {
      System.out.println("Hasil lebih besar dari 10");
    }
    else
    {
      System.out.println("Hasil lebih kecil dari 10");
    }
  }
}
```

```
class contoh2
{
  public static void main (String args[])
  {
    int a,b,c;
    a=5;
    b=7;
    c=a+b;
```



```
if (c>15) // Jika c lebih besar dari 5
{
    System.out.println("Hasil lebih besar dari 15");
}
else if (c > 10) // Jika c lebih besar dari 10
{
    System.out.println("Hasil lebih besar dari 10");
}
else if (c > 5) // Jika c lebih besar dari 15
{
    System.out.println("Hasil lebih kecil dari 5");
}
}
```

## **I.2 Switch atau case**

Merupakan bentuk kontrol yang mana menguji suatu kondisi yang sama. Kontrol ini menjalankan suatu statement jika kondisi tertentu terpenuhi. Kontrol ini hanya menguji variabel yang sama, sedangkan kontrol if ... end if dapat menguji variabel yang berbeda .

Bentuk Umum :

```
Switch (variabel_a)
{
    case kondisi_1:
        statement 1;
        break;
    case kondisi_2:
        statement 2;
        break;
    case kondisi_3:
        statement 3;
        break;
    default : statement 4;
}
```

### **Contoh :**

```
class contoh3
{
    public static void main (String args[])
    {
        int a,b,c;
        a=5;
        b=7;
        c=a+b;
        switch ( c )
        {
            case 4:
                System.out.println("Hasil sama dengan 4");
                break;
            case 5:
                System.out.println("Hasil sama dengan 5");
                break;
            case 6:
```

```
        case 7:
            System.out.println("Hasil sama dengan 6 atau 7");
            break;
        default :
            System.out.println("Hasil Invalid ");
    }
}
```

## **II Iterasi**

### **II.1 Loop for**

Adalah bentuk pengulangan untuk mengulang sesuatu pernyataan atau blok sampai syarat terpenuhi.

#### **Bentuk Umum :**

```
for (inisialisasi ; tes ; tambah-satu)
{
    pernyataan;
}
```

inisialisasi : menampilkan variabel untuk di gunakan dalam loop.

Tes : Kondisi yang di penuhi pada saat Loop

tambah-satu : penambahan variabel inisialisasi 1 pada loop

#### **contoh :**

```
class contoh3
{
    public static void main (String args[])
    {
        int a,b;
        b=5;
        for (a=1 ; a<=b;a=a+5)
            { System.out.println("Urutan ke :"+a); }
    }
}
```

### **II.2 Nested Loop**

Adalah loop yang berada di dalam loop yang lain :

Bentuk umum :

```
for (inisialisasi1 ; tes1 ; tambah-satu)
{
    for (inisialisasi2; tes2; tambah-satu)
    {
        statement
    }
}
```

#### **contoh :**

```
class contoh4
```

```
{
  public static void main (String args[])
  {
    int a,b;
    b=5;
    for (a=1 ; a<=b;a++)
      { for (a=1 ; a<=b;a++)
        {
          System.out.println("Urutan ke :"+a);
        }
      }
  }
}
```

### **II.3 Loop While**

Di gunakan untuk mengulang statement tertentu selama kondisi terpenuhi

Bentuk Umum :

```
while (syarat)
{
  badan_loop;
}
```

Contoh :

```
class contoh5
{
  public static void main (String args[])
  {
    int a=1,b;
    b=5;
    while (a <= b)
    {
      System.out.println("Bilangan :"+ a);
      a++;
    }
  }
}
```

### **II.4 Loop Do .. while**

Di gunakan untuk mengulang badan loop selama kondisi terpenuhi. Loop Do ... while minimal melakukan eksekusi hanya 1 kali.

**Bentuk Umum :**

```
do
{
  badanloop;
}
while (syarat);
```

**Contoh :**

```
class contoh6
{
```

```
public static void main (String args[])
{
    int a=1,b;
    b=5;
    do
    {
        System.out.println("Bilangan :"+ a);
        a++;
    } while (a<=5)
}
```

## Bab IV Control dan Array

### I.Statement Transfer

#### I.1 continue

Fungsi : Di gunakan untuk tidak mengeksekusi sebuah eksekusi di sebuah looping.

Contoh :

```
public class contin {
    public static void main (String args[])
    {
        System.out.println("Odd Numbers");
        for (int i=1;i<=10;i++)
        {
            if (i % 2 ==0)
                continue;

            // Rest of loop body skipped when i is even
            System.out.print(i + "\n");
        }
    }
}
```

Output :

```
Odd Numbers
1
3
5
7
9
```

#### Keterangan :

Pada contoh di atas, fungsi continue di gunakan jika I bernilai genap atau habis di bagi dua.

#### I.2 break

Fungsi : Di gunakan untuk menghentikan iterasi.

Contoh :

```
// Contoh penggunaan fungsi break
public class brea {
    public static void main (String args[])
```

```
{
    System.out.println("Mencetak Angka");

    for (int i=1;i<=8;i++)
    {
        System.out.println(i);
        if (i==6)
            break;
    }
}
```

Output :

Mencetak Angka  
1  
2  
3  
4  
5  
6

**Keterangan :**

Pada contoh di atas, jika i=6, maka program akan keluar dari iterasi for.

**I.3 try and catch**

Fungsi : Untuk menguji statement di block try. Jika statement error, maka akan di jalankan

statement yang berada di dalam kolom catch.

Contoh :

```
class try_catch {
    public static void main(String args[]) {
        int d, a;

        try
        { // monitor a block of code.
            d = 0;
            a = 42 / d;
            System.out.println("This will not be printed.");
        }

        catch (ArithmeticException e)
        {
            // catch divide-by-zero error
            System.out.println("Division by zero.");
        }
        System.out.println("After catch statement.");
    }
}
```

```
}
```

#### **I.4 assert**

Fungsi : Menguji sebuah statement. Jika sebuah kondisi tidak memenuhi syarat yang di terapkan, maka akan di eksekusi sebuah statement.

Contoh :

```
import java.util.*;

import java.util.Scanner;

public class asser
{
public static void main( String args[] )
{
Scanner scanner = new Scanner( System.in );

System.out.print( "Enter a number between 0 and 20: " );
int value = scanner.nextInt();

assert( value >= 0 && value <= 20 ) :
"Invalid number: " + value;
System.out.printf( "You have entered %d\n", value );
}
}
```

Untuk menjalankan fungsi assert, maka pada saat menjalankan file class nya, gunakan feature -ea.

```
Javac asser.java
java -ea asser
```

Fungsi assert tidak dapat berjalan, jika di compile di dalam Netbeans IDE.

#### Keterangan kode :

Kode di atas berfungsi untuk menginput angka dari 0 hingga 20, yang jika di input angka 21, maka program akan error.

#### **I.5 Return**

Fungsi : Mengembalikan program kepada program utama, sehingga program akan berhenti.

Contoh :

```
package latihan;
class Return1 {
public static void main(String args[])
{
```

```
        boolean t = true;
        System.out.println("Before the return.");
        if(t) return; // return to caller
        System.out.println("This won't execute.");
    }
}
```

Pada contoh di atas, program hanya akan mencetak kalimat “Before the return” dan tidak mencetak kalimat This won't execute. Karena akan berhenti di syntax return.

### 3. Array

Pengertian : Suatu data majemuk yang memiliki no.index dan nilai.

Deklarasi dalam Java :

```
int A[]=new int[5]; // Memberi nilai A dengan 5 data elemen.
```

Atau

```
int data[]={1,4,5,6,7,8,9}; // Langsung di beri nilai pada deklarasi array.
```

#### **Contoh Program :**

```
import java.util.*;

public class array {
    public static void main (String [] args)
    {
        int data[]={1,4,5,6,7,8,9}; // index array berapa ?
        // dari berapa-
        int i=0;
        for (i=0;i<=6;i++)
        {
            System.out.print(data[i]+" ");
        }
        System.out.println("");
    }
}
```

Berarti :

```
data[0]=1
data[1]=4
data[2]=5
data[3]=6
data[4]=7
data[5]=8
data[6]=9
```

```
public class array1 {
    public static void main (String args[])
    {
```



```
int A[]=new int[5];
int j=0;
for (int i=0;i<=4;i++)
{
    A[i]=j;
    j=j+2;
}

for (int k=0;k<=4;k++)
{
    System.out.println(A[k]);
}
}
```

## Bab V

### Konsep Pemrograman Berorientasi Objek

#### Object

Object adalah sebuah paket code yang berhubungan dengan state dan behaviour. Object sering di hubungkan dengan object di dalam kehidupan sehari-hari.

Di dalam dunia nyata atau dalam sebuah program, Object memiliki 2 karakteristik, yaitu state dan behaviour. State adalah keadaan dari sebuah Object, sedang behaviour adalah kelakuan dari object tersebut.

#### Contoh :

Object : Mobil

State : warna, merk, tahun pembuatan, model

Behaviour : melaju, berbelok, membunyikan klakson

Sebuah Object yang di bentuk dari class biasa di sebut instance, arti nya Object tersebut adalah wujud nyata dari sebuah class. Variabel dan method dari instance ini di sebut variabel instance dan metode instance. Setiap instance menyimpan variabel nya sendiri-sendiri, jadi nilai variabel untuk tiap instance bisa berbeda.

Membuat Class, Object, property dan method :

```
public class mobil
{
    public static void main (String [] args)
    {
        // Buat object berbeda
        mobilku mobil1=new mobilku();
        mobilku mobil2=new mobilku();

        mobil1.ubahban("Buruk");
        mobil1.ubahrem("Buruk");
        mobil1.penumpang(4);
        mobil1.cetak();

        mobil2.ubahban("Prima");
        mobil2.ubahrem("Prima");
        mobil2.penumpang(5);
        mobil2.cetak();
    }
}

class mobilku
{
    String rem="Baik";
    String ban="Baik";
    int penumpang=0;
```

```
void ubahban(String vban)
{
    ban=vban;
}

void ubahrem(String vrem)
{
    rem=vrem;
}

void penumpang(int a)
{
    penumpang=a;
}

void cetak()
{
    System.out.println("ban :"+ban+"Rem :"+rem);
    System.out.println("Jumlah Penumpang :"+penumpang);
}
}
```

Pada source code di atas, mobil dan mobilku adalah class. String rem, String ban, int penumpang adalah property. Ubahban, ubahrem, penumpang dan cetak adalah method. Mobil1 dan mobil2 adalah instance, yang seringkali juga dapat di sebut object

### **Class**

Class adalah sebuah prototype yang mana, Object di buat. Class juga sebuah structure yang mendefinisikan data dan metode untuk bekerja dalam data. Ketika kita menulis Java programming, semua data program di bungkus di dalam class, apakah itu class yang di buat sendiri atau class dari Java Platform API Libraries. Class juga dapat di representasikan di dalam kehidupan sehari-hari.

### **Class yang di buat sendiri :**

```
public class Bicycle {

    // the Bicycle class has three fields
    public int cadence;
    public int gear;
    public int speed;

    // the Bicycle class has one constructor
    public Bicycle(int startCadence, int startSpeed, int startGear) {
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;
    }

    // the Bicycle class has four methods
    public void setCadence(int newValue) {
        cadence = newValue;
    }
}
```

```
public void setGear(int newValue) {
    gear = newValue;
}

public void applyBrake(int decrement) {
    speed -= decrement;
}

public void speedUp(int increment) {
    speed += increment;
}
```

### **Method**

Method adalah sama dengan function di dalam pemrograman procedural. Method berfungsi untuk mengelompokkan event yang di lakukan di dalam program. Hanya saja dalam OOP, method di akses via objek yang di buat.

### **Inheritance / Pewarisan**

Inheritance menyediakan sebuah mekanisme yang powerful dan natural untuk mengatur program yang kita buat. Inheritance berarti sebuah class mewarisi state dan behaviour dari kelas lain. Class yang mewarisi state dan behaviour di sebut super class sedangkan class yang di warisi state dan behaviour di sebut sub class.

Untuk menginherit properties atau method dari dari sebuah class, di gunakanlah keyword **extends**.

Formula :

```
class Super {
.....
}

class Sub extends Super
{
.....
.....
}
```

### **Contoh Inheritance :**

Di dalam contoh di bawah ini, anda akan meng observasi 2 class dengan nama **Calculation** dan **My\_Calculation**. Menggunakan keyword extends, My\_Calculation menginherit method addition() dan Subtraction() dari class Calculation.

```
class Calculation {
    int z;
    public void addition(int x, int y){
        z=x+y;
        System.out.println("The sum of the given numbers:"+z);
    }

    public void Substraction(int x,int y){
        z=x-y;
    }
}
```

```
        System.out.println("The difference between the given
numbers:"+z);
    }
}

public class My_Calculation extends Calculation{

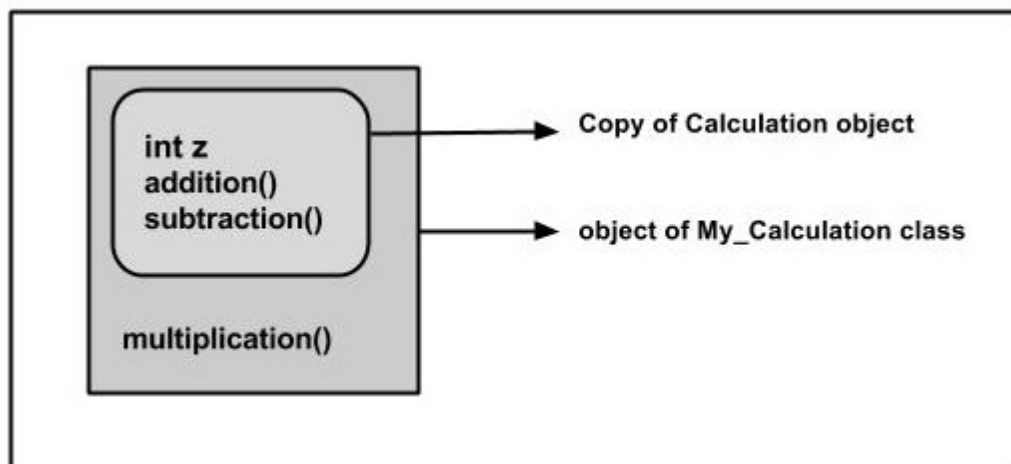
    public void multiplication(int x, int y){
        z=x*y;
        System.out.println("The product of the given numbers:"+z);
    }

    public static void main(String args[]){
        int a=20, b=10;
        My_Calculation demo = new My_Calculation();
        demo.addition(a, b);
        demo.Substraction(a, b);
        demo.multiplication(a, b);
    }
}
```

**Output :**

The sum of the given numbers : 30  
The difference between the given numbers : 10  
The product of the given numbers : 200

Source code di atas menunjukkan class Calculation adalah superclass sedangkan class My\_Calculation adalah class subclass. Pada syntax public class My\_Calculation extends Calculation, di artikan, bahwa : class My\_Calculation meng extend / mewarisi properties dan method yang di miliki oleh class Calculation, sehingga instance demo dengan dengan reference dan object My\_Calculation dapat menggunakan method addition dan Subtraction yang ada di dalam class Calculation. Instance demo juga mewarisi properties selain method dari super class. Inheritance pada source code di atas, dapat di gambarkan pada gambar berikut :



**Keyword super**

Keyword **super** sama dengan keyword **this** dalam hal berikut :  
- Di gunakan untuk membedakan member superclass dari anggota subclass, jika mereka

memiliki nama yang sama.

- Di gunakan untuk menginvoke / memohon constructor superclass dari subclass.

### **Membedakan nama**

Jika sebuah class mewarisi properties dari class yang lain dan jika anggota superclass memiliki nama yang sama dengan sub class, untuk membedakan variabel-variabel ini, kita menggunakan keyword super seperti di bawah ini :

```
super.variable  
super.method();
```

### **Contoh kode :**

Nama File : Sub\_class.java

```
class Super_class{  
    int num=20;  
  
    //display method of superclass  
    public void display(){  
        System.out.println("This is the display method of superclass");  
    }  
}  
  
public class Sub_class extends Super_class {  
    int num=10;  
  
    //display method of sub class  
    public void display(){  
        System.out.println("This is the display method of subclass");  
    }  
  
    public void my_method(){  
  
        //Instantiating subclass  
        Sub_class sub=new Sub_class();  
  
        //Invoking the display() method of sub class  
        sub.display();  
  
        //Invoking the display() method of superclass  
        super.display();  
  
        //printing the value of variable num of subclass  
        System.out.println("value of the variable named num in sub  
class:"+  
        sub.num);  
  
        //printing the value of variable num of superclass  
        System.out.println("value of the variable named num in super  
class:"+  
        super.num);  
    }  
  
    public static void main(String args[]){  
        Sub_class obj = new Sub_class();  
        obj.my_method();  
    }  
}
```

```
    }}  
Output :  
This is the display method of subclass  
This is the display method of superclass  
value of the variable named num in sub class:10  
value of the variable named num in super class:20
```

### **Invoking superclass constructor**

Jika sebuah class mewarisi properties dari class yang lain, subclass secara otomatis mengakuisisi constructor default dari super class. Namun jika anda ingin memanggil sebuah constructor yang memiliki parameter dari super class, anda perlu menggunakan keyword super sebagaimana yang di tunjukkan sebagai berikut :

super (values)

Contoh kode :

Nama file : Subclass.java

```
class Superclass{  
  
    int age;  
  
    Superclass(int age)  
    {  
        this.age=age;  
    }  
  
    public void getAge(){  
        System.out.println("The value of the variable named age in super  
class is: " +age);  
    }  
}  
  
public class Subclass extends Superclass {  
    Subclass(int age){  
        super(age);  
    }  
  
    public static void main(String argd[]){  
        Subclass s= new Subclass(24);  
        s.getAge();  
    }  
}
```

```
// Output :  
The value of the variable named age in super class is : 24
```

Pada kode di atas, terdapat class Superclass dengan konstruktor Superclass(int age). Konstruktor menggunakan properties age dari konstruktor super class dan ia harus menggunakan keyword super(age) .

## Bab VI

### Konsep Object Oriented dan Applet

#### Object Oriented

##### 1. Overriding

- Overriding atau pengalihan adalah pengalihan penggunaan method antara super class dan sub class, dengan pemanggilan object yang berbeda dari referensinya saat pembuatan sebuah instance. Method yang dapat di override, namanya harus sama dan dimiliki oleh super class dan sub class.

##### Contoh kode :

```
class Animal {
    public void move(){
        System.out.println("Animals can move");
    }
}

class Dog extends Animal{
    public void move(){
        System.out.println("Dogs can walk and run");
    }
}

public class TestDog{
    public static void main(String args[]){
        Animal a = new Animal(); // Animal reference and object
        Animal b = new Dog(); // Animal reference but Dog object

        a.move();// runs the method in Animal class

        b.move();//Runs the method in Dog class
    }
}
```

##### Output :

```
Animals can move
Dogs can walk and run
```

Pada source code di atas, walaupun b memiliki reference Animal, namun ia menjalankan method move pada class Dog. Hal ini disebabkan b memiliki object Dog. Penjelasan hal ini adalah sbb : Saat compile time, pengecekan berdasarkan tipe reference, namun saat runtime, JVM mendapatkan tipe object dan menjalankan method yang dimiliki oleh tipe object tersebut.

Hal-hal yang harus diperhatikan di dalam overriding, di antaranya adalah sebagai berikut :

- Sebuah method yang di declare final tidak dapat di override.
- Jika sebuah method tidak dapat di wariskan, maka method tersebut juga tidak dapat di override.
- Sebuah method yang di deklarasikan static tidak dapat di override namun dapat di re-declared.
- Constructor tidak dapat di override
- Sebuah subclass di dalam package yang sama, sebagai instance dari super class dapat mengoverride method super class yang lain, yang tidak di deklarasikan private atau final.



Menggunakan keyword **super** .

Ketika menginvoke / memohon sebuah versi superclass sebuah method yang di override, keyword super di gunakan .

```
class Animal{    public void move(){
    System.out.println("Animals can move");
    }
}

class Dog extends Animal{

    public void move(){
        super.move(); // invokes the super class method
        System.out.println("Dogs can walk and run");
    }
}

public class TestDog{
    public static void main(String args[]){

        Animal b = new Dog(); // Animal reference but Dog object
        b.move(); //Runs the method in Dog class
    }
}
```

**Output :**

Animals can move  
Dogs can walk and run

**2. Polymorphism**

Polymorphism adalah kemampuan sebuah object untuk memiliki banyak bentuk. Penggunaan paling umum dari polymorphism di dalam OOP adalah ketika parent class reference di gunakan untuk me refer object class child.

Object Java apapun yang memiliki lebih dari satu IS-A dapat menjadi polymorphic. Dalam Java, semua Object Java adalah Polymorphic karena semua object memiliki lebih dari 1 IS-A untuk masing-masing tipe dan object class nya.

Satu-satu nya cara untuk mengakses sebuah object adalah dengan mengakses sebuah object melalui sebuah reference variable. Sebuah variabel reference hanya dapat satu tipe saja. Satu kali di declare, tipe variable reference tidak dapat di ubah. Sebuah variable reference dapat di deklarasikan sebagai sebuah tipe class atau interface.

```
public interface Vegetarian {}
public class Animal {}
public class Deer extends Animal implements Vegetarian {}
```

Pada kode di atas, Deer adalah Polymorphic, karena memiliki inheritance multiple. Kalimat-kalimat berikut adalah benar untuk contoh di atas, yakni :

- A Deer IS-A animal
- A Deer IS-A Vegetarian
- A Deer IS-A Deer
- A Deer IS-A Object

Ketika anda menerapkan variable reference ke reference object Deer, deklarasi berikut adalah legal :

```
Deer d = New Deer();  
Animal a = d;  
Vegetarian v = d;  
Object o = d;
```

Semua variable reference d,a,v,o menunjuk ke Object Deer yang sama.

### **Virtual Method**

Di dalam source code di bawah ini, di tunjukkan bagaimana behaviour dari method yang di override di dalam Java mengizinkan anda untuk mengambil manfaat dari polymorphism ketika mendesain class anda.

```
/* File name : Employee.java */  
public class Employee  
{  
    private String name;  
    private String address;  
    private int number;  
    public Employee(String name, String address, int number)  
    {  
        System.out.println("Constructing an Employee");  
        this.name = name;  
        this.address = address;  
        this.number = number;  
    }  
    public void mailCheck()  
    {  
        System.out.println("Mailing a check to " + this.name  
        + " " + this.address);  
    }  
    public String toString()  
    {  
        return name + " " + address + " " + number;  
    }  
    public String getName()  
    {  
        return name;  
    }  
    public String getAddress()  
    {  
        return address;  
    }  
    public void setAddress(String newAddress)  
    {  
        address = newAddress;  
    }  
    public int getNumber()  
    {  
        return number;  
    }  
}
```

Sekarang kita mengextend class employee sebagai berikut :

```
public class Salary extends Employee

{
    private double salary; //Annual salary
    public Salary(String name, String address, int number, double
        salary)
    {
        super(name, address, number);
        setSalary(salary);
    }

    public void mailCheck()
    {
        System.out.println("Within mailCheck of Salary class ");
        System.out.println("Mailing check to " + getName()
            + " with salary " + salary);
    }

    public double getSalary()
    {
        return salary;
    }

    public void setSalary(double newSalary)
    {
        if(newSalary >= 0.0)
        {
            salary = newSalary;
        }
    }

    public double computePay()
    {
        System.out.println("Computing salary pay for " + getName());
        return salary/52;
    }
}

/* File name : VirtualDemo.java */

public class VirtualDemo{
    public static void main(String [] args)
    {
        Salary s = new Salary("Mohd Mohtashim", "Ambehta, UP", 3,
3600.00);
        Employee e = new Salary("John Adams", "Boston, MA", 2, 2400.00);
        System.out.println("Call mailCheck using Salary reference --");
        s.mailCheck();
        System.out.println("\n Call mailCheck using Employee
reference--");
        e.mailCheck();
    }
}
```

### **Output :**

Constructing an Employee

Constructing an Employee

Call mailCheck using Salary reference --

Within mailCheck of Salary class

ailing check to Mohd Mohtashim with salary 3600.0

Call mailCheck using Employee reference--

Within mailCheck of Salary class

ailing check to John Adams with salary 2400.0

Di sini kita menginstantiate 2 object salary. Satu menggunakan Salary reference s dan yang lainnya menggunakan Employee reference e. Pada invoking s.mailCheck(), compiler melihat mailCheck() dalam class salary saat compile time dan JVM menginvoke mailCheck() dalam class salaray saat run time.

Pada Invoking e.mailCheck(), compiler melihat method mailCheck pada class Employee, karena e di nyatakan sebagai :

Employee e = new Salary("John Adams", "Boston, MA", 2, 2400.00);

Sehingga e memiliki reference Employee.

### **4. Applet**

Applet adalah sebuah program Java yang berjalan di sebuah Web Browser. Beberapa Perbedaan antara Applet dengan Aplikasi Java Stand alone adalah :

1. Sebuah Applet adalah sebuah class Java yang meng extends java.applet. Class applet Applet tidak membutuhkan method main().
2. Applet di desain untuk di embed ke dalam html
3. JVM di butuhkan untuk menjalankan Applet. JVM dapat sebagai plug-in web browser atau berjalan di lingkungan yang berbeda.

Empat Methode di dalam class Applet , yaitu :

4. init : Method ini untuk menginisialisasi apapun.
5. start : Method ini otomatis di panggil setelah browser memanggil method init. Juga di sebut kapan saja user mengembalikan halaman berisi applet setelah selesai di proses lain.
6. Stop : Method ini secara otomatis di panggil ketika user mengembalikan halaman yang mana applet berada. Di panggil secara berulang di dalam applet yang sama.
7. Destroy :Method ini di panggil sekali sebelum browser melakukan unloads pada applet.

Contoh :

**Membuat Applet Hello World**

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends Applet
{
    public void paint (Graphics g)
    {
        g. drawString("Hello World",25,50);
    }
}
```

Pada contoh di atas, menggunakan class import java.applet dan java.awt. 2 class import di atas lah yang penting untuk membuat applet.

**Insert Applet ke dalam html**

Applet dapat di gunakan jika di insert ke dalam html. Pada kode applet di atas, di insert ke dalam html dengan cara sbb :

```
<html>
<head>
  <title>Hello in Java Applet</title>
</head>
<body>
  <applet code="hello.class" width="320" height="120">
    If your browser was Java-enabled, a "Hello World" message would appear here
  </applet>
</body>
</html>
```

## **Bab VII**

### **JFC DAN SWING**

#### **VII.1 Pendahuluan**

JFC, singkatan dari Java Foundation Classes, yang merupakan sekumpulan feature untuk membuat GUI (Graphical User Interface) dan menambahkan fungsi grafis dan interaktif ke sebuah aplikasi Java.

Swing adalah inti dari Java GUI Widget toolkit dan merupakan bagian dari JFC. Swing di buat untuk menyediakan kumpulan komponen GUI yang lebih canggih dari sebelum nya, AWT.

Feature-feature dari JFC adalah sbb :

##### **1.Swing GUI Component**

Mencakup semua komponen, dari button ke tables

##### **2.Pluggable Look and Feel Support**

Aplikasi Look and Feel Swing dapat di terapkan di Aplikasi Java

##### **3. Accessibility API**

Mengaktifkan teknologi pembantu, seperti screen readers dan Braille Displays, untuk mendapatkan informasi dari User Interface.

##### **4. Java 2D API**

Membuat Developer dapat dengan mudah menggunakan Grafik 2D berkualitas tinggi, text, dan images dalam Aplikasi Java dan Applet.

##### **6. Internazionalization**

Berguna untuk menjadikan sebuah aplikasi Java dapat mengenali berbagai bahasa dan karakter huruf, seperti bahasa Jepang, China dan Korea.

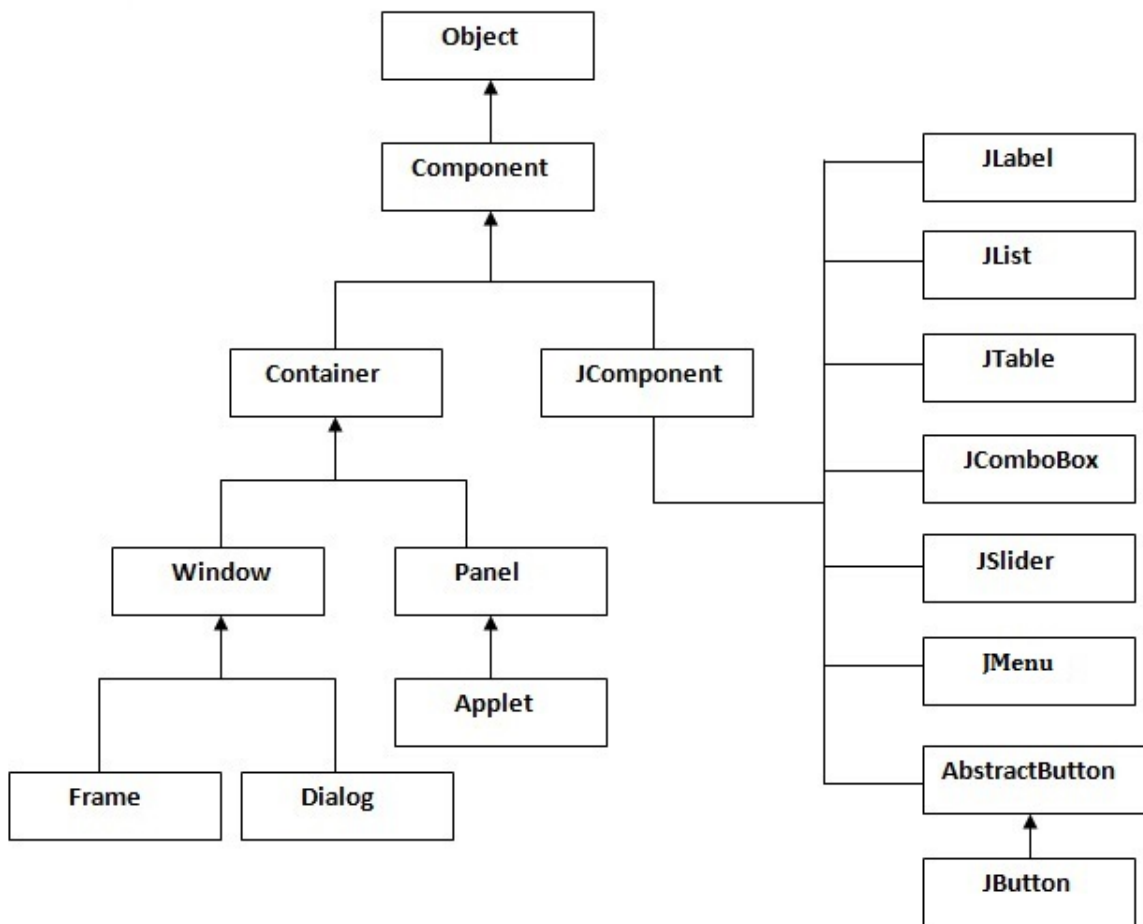
#### **VII.2 Swing Packages**

Swing menyediakan sejumlah public packages, yaitu :

8. javax.accessibility
9. javax.swing
10. javax.swing.border
11. javax.swing.colorchooser
12. javax.swing.event
13. javax.swing.filechooser
14. javax.swing.plaf
15. javax.swing.plaf.basic
16. javax.swing.plaf.metal
17. javax.swing.plaf.multi
18. javax.swing.plaf.synth
19. javax.swing.table
20. javax.swing.text
21. javax.swing.text.html
22. javax.swing.text.html.parser
23. javax.swing.text.rtf
24. javax.swing.tree
25. javax.swing.undo

Kebanyakan packages di atas adalah hanya sebuah subset API kecil. Minimal di perlukan sebuah Package untuk membuat GUI, yaitu : javax.swing

Hierarki dari API Java Swing adalah sebagai berikut:



**Gambar 1: Hierarki Java Swing**

Method-method dari class komponen yang sering dipergunakan di dalam Java Swing adalah sebagai berikut:

Method	Deskripsi
Public void add (Component c)	Menambahkan komponen di atas komponen yang lain
Public void setSize(int width, int height)	Mengatur ukuran dari komponen
Public void setLayout(Layout Manager m)	Mengatur layout manager dari komponen
Public void setVisible(boolean b)	Mengatur visibility dari komponen. Defaultnya adalah false.

**Contoh Java Swing**

- Ada 2 cara untuk membuat sebuah frame, yakni:
- Dengan membuat object dari class Frame (association)
  - Dengan menambahkan frame class (inheritance)

### Contoh pembuatan Java Swing

Pada contoh berikut, kita akan membuat kode Java Swing dengan membuat tombol button yang diletakkan di atas object JFrame di dalam method main(). Terdapat 3 cara pembuatan program di atas, yaitu:

#### Nama File: swing1

```
import javax.swing.*;
public class swing1 {
    public static void main (String args[])
    {
        JFrame f = new JFrame(); // Creating instance of
        JFrame
        JButton b = new JButton("Click");
        b.setBounds(130, 100, 100, 40); //x axis,y
        axis,width,height
        f.add(b); // adding button in JFrame
        f.setSize(400,500); // 400 width and 500 height
        f.setLayout(null); // using no layout managers
        f.setVisible(true); // making the frame visible
    }
}
```

Kita dapat juga menulis semua kode untuk membentuk JFrame, JButton dan method untuk memanggilnya di dalam konstruktor java.

#### Nama File: swing1a

```
import javax.swing.*;

public class swing1a {
    JFrame f;
    swing1a()
    {
        f=new JFrame(); // Creating instance of JFrame
        JButton b=new JButton("Click"); // Creating
        instance of
        JButton
        b.setBounds(130,100,100,40);
        f.add(b); //adding button in JFrame
        f.setSize(400,500); // 400 width and 500 height
        f.setLayout(null); // using no layout managers
        f.setVisible(true); //making the frame visible
    }

    public static void main (String args[])
    {
        new swing1a();
    }
}
```



```
}  
}
```

Pada cara ke 3, kita dapat juga meng inherit class JFrame, sehingga tidak perlu membuat instance class JFrame.

### **Nama File: swing1b**

```
import javax.swing.*;  
public class swing1b extends JFrame { // Inhereting  
    JFrame  
    JFrame f;  
    swing1b()  
    {  
        JButton b=new JButton("Click"); // Create Button  
        b.setBounds(130,100,100,40);  
        add(b); // adding button on frame  
        setSize(400,500);  
        setLayout(null);  
        setVisible(true);  
    }  
  
    public static void main(String args[])  
    {  
        new swing1b();  
    }  
}
```

Output dari ketiga file di atas adalah sama, yaitu:



**Gambar 2: Tampilan program**

Setiap User Interface memiliki 3 hal berikut, yaitu:

1. UI elements : Adalah tampilan layar yang di lihat dan akan berinteraksi dengan user. Beberapa elemen adalah seperti berikut: JLabel, JButton, Jcolorchooser, dll.
2. Layout : Adalah cara bagaimana UI element di atur peletakkannya di atas layar.
3. Behaviour : Ini adalah event yang akan terjadi, ketika user berinteraksi dengan UI element.

Setiap SWING control mewariskan properti dari hierarki class komponen berikut:

#### **1. Component**

Component adalah class base abstract untuk kontrol user-interface non menu di Swing. Component mewakili sebuah object dengan representasi grafik.

#### **2. Container**

Container adalah sebuah component yang dapat berisi komponen Swing lainnya.

#### **3. Jcomponent**

Jcomponent adalah base class untuk semua Swing UI component. Untuk menggunakan component

Swing yang mewarisi dari Jcomponent, component harus berada di dalam sebuah hierarchy containment yang root nya adalah container top-level Swing.

### **VII.3 UI Elemen Swing adalah:**

1. JLabel
2. JButton
3. JColor Chooser
4. JcheckBox
5. JRadioButton
6. Jlist
7. JcomboBox
8. JtextField
9. JpasswordField
10. JtextArea
11. ImageIcon
12. JScrollbar
13. JoptionPane
14. JfileChooser
15. JProgressBar
16. JSlider
17. JSpinner

### **VII.4 Membuat Menu Login**

Menu Login yang akan kita buat terdiri dari beberapa komponen, yaitu, Label “Username” dan “Password”, Text Field “Username” dan “Password” dan tombol “OK”. Berikut adalah kodenya:

**package** baru;

```
import javax.swing.*;
public class swing2 {
    swing2()
    {
        JFrame myframe=new JFrame();
        JLabel label1=new JLabel();
```

```
JLabel label2=new JLabel();
JLabel lbllogin=new JLabel();
JTextField txtusername=new JTextField();
JPasswordField txtpassword=new JPasswordField();
JButton btnOk=new JButton("OK");

lbllogin.setText("LOGIN");
lbllogin.setBounds(170,20,100,30);
label1.setText("Username");
label1.setBounds(70, 60,100,30);
label2.setText("Password");
label2.setBounds(70,60,100,80);
txtusername.setBounds(150, 65, 150, 20);
txtpassword.setBounds(150, 90, 150, 20);
btnOk.setBounds(70, 130, 60, 40);

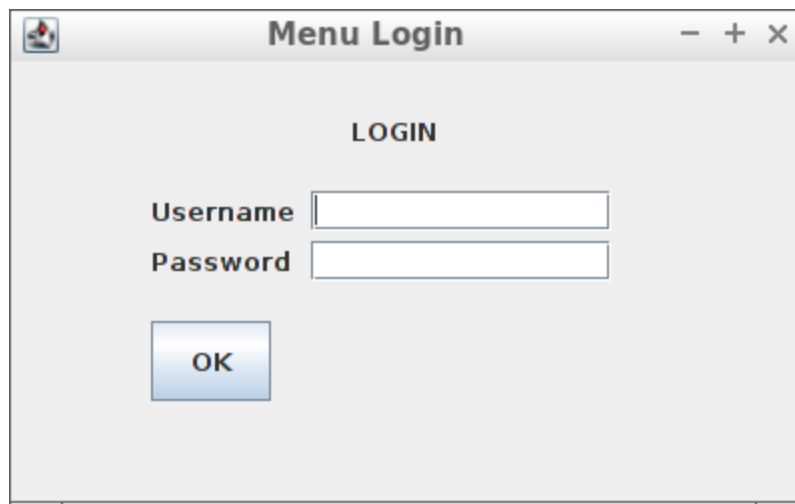
myframe.add(label1);
myframe.add(label2);
myframe.add(txtusername);
myframe.add(txtpassword);
myframe.add(lbllogin);
myframe.add(btnOk);

myframe.setTitle("Menu Login");
myframe.setSize(400, 250);
myframe.setLayout(null);
myframe.setVisible(true);

myframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public static void main (String args[])
{
    new swing2();
}
}
```

Pada kode di atas, pertama kali kita membuat instance dari masing-masing komponen, yakni instance untuk Label, Text field dan Button. Lalu kita menggunakan metode dari class komponen untuk memberi text dan memberi koordinat, height dan width untuk di letakkan di atas frame. Selanjutnya, komponen-komponen di letakkan di atas frame dengan menggunakan metode "add". Konstruktors "swing2". Output dari kode di atas adalah:



**Gambar 3: Menu Login**

## Bab VIII

### Swing Elements, Event Handling dan Event Class

#### IX.1 Event Handling

##### IX.1.1 Event

Event adalah perubahan di dalam sebuah komponen, seperti penekanan tombol keyboard, click mouse atau hover mouse. Java menggunakan Model Event Delegasi untuk menangani event. Model ini mendefinisikan mekanisme standard untuk membuat dan menangani event.

Model Event Delegasi memiliki kunci partisipasi yang bernama:

\* Source – Source adalah sebuah object yang mana event terjadi. Source bertugas untuk menyediakan informasi dari event yang terjadi ke handlernya. Java menyediakannya sebagai class untuk object sumber.

\* Listener – Dikenal juga sebagai event handler. Listener bertugas untuk membuat respon ke event. Listener juga merupakan object. Saat event di terima, listener mengolah event dan kemudian mengembalikannya.

Contoh langkah-langkah di dalam Event handling:

- User mengclick button dan event terjadi.
- Object berada pada class event yang dibentuk otomatis dan informasi tentang sumber dan event yang terbentuk di dalam object yang sama.
- Event object di forward ke method dari class listener registered.
- Method saat ini di eksekusi dan return.

##### IX.1.2 Contoh Event Handling

**package** baru;

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

public class SwingControlDemo
{
    // Deklarasi komponen //
    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    // Buat constructor //
    public SwingControlDemo()
    {
        // Panggil method prepareGUI
        prepareGUI();
    }

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

```
{
    SwingControlDemo swingControlDemo = new
SwingControlDemo();
    swingControlDemo.showEventDemo();
}

private void prepareGUI()
{
    // Set frame dengan instance mainFrame //
    mainFrame=new JFrame("Java Swing Examples");
    mainFrame.setSize(400,400);
    mainFrame.setLayout(new GridLayout(3,1));

    headerLabel=new JLabel("",JLabel.CENTER);
    statusLabel=new JLabel("",JLabel.CENTER);

    statusLabel.setSize(350,100);
    mainFrame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent
windowEvent)
        {
            System.exit(0);
        }
    });
    controlPanel=new JPanel();
    controlPanel.setLayout(new FlowLayout());

    mainFrame.add(headerLabel);
    mainFrame.add(controlPanel);
    mainFrame.add(statusLabel);
    mainFrame.setVisible(true);
}

private void showEventDemo()
{
    // Set property komponen
    headerLabel.setText("Control in Action: Button");

    JButton okButton=new JButton("OK");
    JButton submitButton=new JButton("Submit");
    JButton cancelButton=new JButton("Cancel");

    okButton.setActionCommand("OK");
    submitButton.setActionCommand("Submit");
    cancelButton.setActionCommand("Cancel");
}
```

```
// Panggil method addActionListener, untuk
menjalankan aksi
okButton.addActionListener(new ngeclick());
submitButton.addActionListener(new ngeclick());
cancelButton.addActionListener(new ngeclick());

controlPanel.add(okButton);
controlPanel.add(submitButton);
controlPanel.add(cancelButton);
}

// class ngeclick yang mengandung method actionPerformed
private class ngeclick implements ActionListener {
    public void actionPerformed(ActionEvent e)
    {
        // Pilihan masing button dengan kalimat berbeda
        yang
        dihasilkan //

        String command=e.getActionCommand();
        if (command.equals("OK"))
        {
            statusLabel.setText("OK Button
clicked");
        }
        else if (command.equals("Submit"))
        {
            statusLabel.setText("Submit Button
clicked");
        }
        else
        {
            statusLabel.setText("Cancel Button
clicked");
        }
    }
}
}
```



Gambar IX.1: Contoh Event Handling

Contoh Event Handling di atas, dibuat 3 button, yaitu: Button “OK”, Button “Submit” dan button “Cancel”. Jika Button “OK” di tekan, maka akan muncul kalimat “OK Button clicked”. Jika Button “Submit” ditekan, maka akan muncul tulisan: “Submit Button clicked”. Jika Button “Cancel” ditekan, maka akan muncul tulisan “Cancel Button clicked”. Program di atas menggunakan beberapa komponen, yaitu: JLabel dan JButton. Program memiliki class “ngeclick” yang melakukan implement atas ActionListener.

## **IX.2 Class EventObject**

Semua event di constructed dengan sebuah reference ke object. Class ini di definisikan di dalam java.util package.

### **IX.2.1 Class declaration**

Berikut adalah deklarasi untuk java.util.EventObject class:

```
public class EventObject
    extends Object
        implements serializable
```

## **IX.3 Class Swing Event**

Berikut adalah daftar class event yang sering digunakan:



No	Control & Description
1	<b>AWTEvent</b> Class event root untuk semua event SWING.
2	<b>ActionEvent</b> ActionEvent di bentuk ketika tombol button di click atau item sebuah list melakukan dobel click.
3	<b>InputEvent</b> class InputEvent adalah class root event untuk semua komponen.
4	<b>KeyEvent</b> Saat memasukkan karakter, event Key dibuat.
5	<b>MouseEvent</b> Event ini mengindikasikan aksi mouse terjadi di dalam sebuah komponen.
6	<b>WindowEvent</b> Object class ini mewakili perubahan dalam state sebuah window
7	<b>AdjusmentEvent</b> Object class ini mewakili adjustment event yang dibuat oleh object yang dapat di adjust
8	<b>ComponenEvent</b> Object class ini mewakili perubahan di dalam state sebuah window
9	<b>ContainerEvent</b> Object class ini mewakili perubahan di dalam state sebuah window
10	<b>MouseMotionEvent</b> Object class ini mewakili perubahan di dalam state sebuah window
11	<b>PaintEvent</b> Object class ini mewakili perubahan di dalam state sebuah window

#### **IX.4 Swing – Event Listeners**

Event listener mewakili interface, bertanggung jawab untuk menangani event. Java menyediakan berbagai macam class Event listener, namun di sini akan ditampilkan class yang sering digunakan.

##### **IX.4.1 EventListener interface**

Class ini didefinisikan di dalam java.util package.

##### **IX.4.2 Class declaration**

Berikut adalah deklarasi untuk interface java.util.EventListener:

```
public interface EventListener
```

##### **IX.4.3 Interface Swing EventListener:**

Berikut adalah daftar event listener yang sering digunakan.

NO	Control dan Description
1	<b>ActionListener</b> Interface ini digunakan untuk menerima action events
2	<b>ComponentListener</b> Interface ini digunakan untuk menerima component events
3	<b>ItemListener</b>

	Interface ini digunakan untuk menerima item events
4	<b>KeyListener</b> Interface ini digunakan untuk menerima key events
5	<b>MouseListener</b> Interface ini digunakan untuk menerima mouse events
6	<b>WindowListener</b> Interface ini digunakan untuk menerima window events
7	<b>AdjustmentListener</b> Interface ini digunakan untuk menerima adjustment events
8	<b>ContainerListener</b> Interface ini digunakan untuk menerima container events
9	<b>MouseMotionListener</b> Interface ini digunakan untuk menerima mouse motion events
10	<b>FocusListener</b> Interface ini digunakan untuk menerima focus events

**Contoh Soal:**

Buatlah program penjumlahan 2 bilangan di dalam GUI.

**Jawaban:**

**package** baru;

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

```
public class kalkulator {  
    public double xnilai,xnilai2;  
    public JLabel lblHasil=new JLabel();  
    public JTextField txtnilai=new JTextField();  
    public JTextField txtnilai2=new JTextField();
```

```
kalkulator()  
{
```

```
    // Buat instance komponen Swing //  
    JFrame myframe=new JFrame();  
    JLabel label1=new JLabel();  
    JLabel label2=new JLabel();  
    JLabel label3=new JLabel();  
    JLabel lblheader=new JLabel();  
    JButton btnJumlah=new JButton("Jumlah");  
  
    // Beri value label dan atur letak komponen //  
    lblheader.setText("PENJUMLAHAN");
```

```
lblheader.setBounds(170,20,100,30);
label1.setText("Nilai 1");
label1.setBounds(70, 60,100,30);
label2.setText("Nilai 2");
label2.setBounds(70,60,100,80);
label3.setBounds(70,60,100,130);
lblHasil.setBounds(150,60,150,130);
txtnilai1.setBounds(150, 65, 150, 20);
txtnilai2.setBounds(150, 90, 150, 20);
btnJumlah.setBounds(70, 160, 100, 40);

// Mengaktifkan action //
btnJumlah.setActionCommand("Jumlahkan");
btnJumlah.addActionListener(new hitung());

myframe.add(label1);
myframe.add(label2);
myframe.add(label3);
myframe.add(lblHasil);
myframe.add(txtnilai1);
myframe.add(txtnilai2);
myframe.add(lblheader);
myframe.add(btnJumlah);

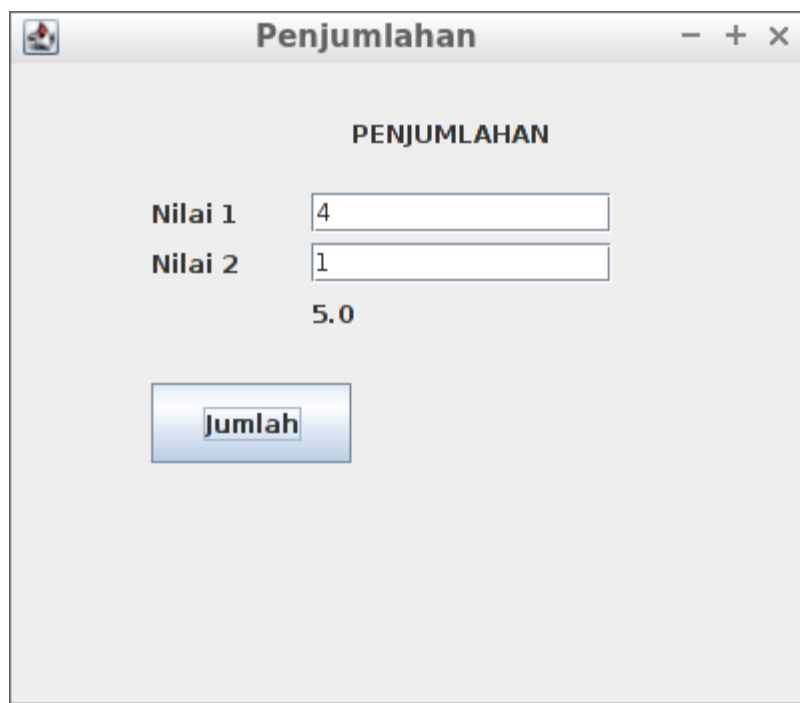
myframe.setTitle("Penjumlahan");
myframe.setSize(400, 350);
myframe.setLayout(null);
myframe.setVisible(true);

myframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

private class hitung implements ActionListener {
    public void actionPerformed(ActionEvent e)
    {
        double xhasil;
        String proses=e.getActionCommand();
        if (proses.equals("Jumlahkan"))
        {
            xnilai1=Double.parseDouble(txtnilai1.getText());
            xnilai2=Double.parseDouble(txtnilai2.getText());
            xhasil=xnilai1+xnilai2;
            lblHasil.setText(""+xhasil);
        }
    }
}
```

```
    }  
}  
  
public static void main (String args[])  
{  
    new kalkulator();  
}  
}
```

Tampilan aplikasi adalah:



Layar aplikasi kalkulator

Aplikasi kalkulator, hanya melakukan proses penjumlahan 2 bilangan. Aplikasi menggunakan komponen JLabel, JTextField dan JButton. Aplikasi mengimplementasikan ActionListener, sebagai pengatur event dari button “Jumlah”, yang melakukan proses perhitungan dan penjumlahan atas 2 angka yang diinput.

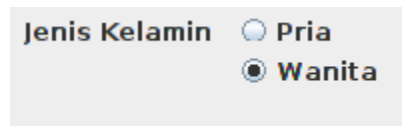
## Bab IX

### Penggunaan Radio Button, Check Box, Text Area dan Combo Box dalam swing

Beberapa komponen yang sering digunakan di dalam Swing selain text area dan button adalah Radio Button, Check Box, Select, Text area dan List. Di bawah ini, akan diterangkan komponen-komponen tersebut.

#### X.1 Radio Button

Radio button digunakan untuk memilih 1 di antara beberapa pilihan, seperti memilih jenis kelamin dan kelas dari seorang siswa. Swing menggunakan komponen JtextArea untuk membuat Textarea.



Gambar 1: Contoh Radio Button

Di dalam kode, contoh pembuatan Radio Button adalah:

```
JRadioButton xradiobutton_pria=new JRadioButton("Pria");  
JRadioButton xradiobutton_wanita=new  
JRadioButton("Wanita");
```

Text Pria dan Wanita pada contoh di atas, dapat juga di beri dengan menggunakan metode setText. Kode lengkapnya adalah sebagai berikut:

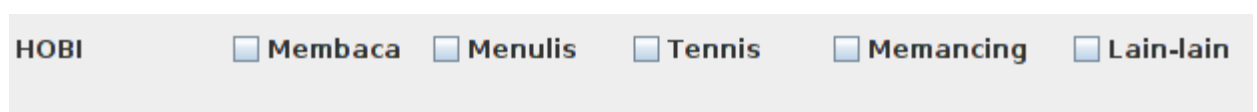
```
xradiobutton_pria.setText("Pria");
```

Pada contoh di atas, terdapat 2 pilihan untuk jenis kelamin, yaitu pria dan wanita. Untuk hanya dapat memilih salah satu saja dari 2 pilihan, maka Radio Button perlu di grouping, dengan perintah.

```
ButtonGroup group_jen_kel=new ButtonGroup();  
group_jen_kel.add(xradiobutton_pria);  
group_jen_kel.add(xradiobutton_wanita);
```

#### X.2 CheckBox

CheckBox dipergunakan untuk memilih sejumlah 1 atau lebih pilihan dari sejumlah pilihan yang disediakan. Swing menggunakan komponen JCheckBox untuk menggunakan CheckBox.



Gambar 2: Contoh CheckBox

Di dalam kode, contoh pembuatan Check Box adalah:

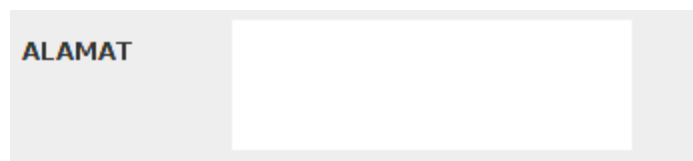
```
JCheckBox xmembaca=new JCheckBox("Membaca");
```

Teks "Membaca" pada contoh di atas, dapat juga diberikan dengan menggunakan metode `setText`. Kode lengkapnya adalah sebagai berikut:

```
xmembaca.setText("Membaca");
```

### X.3 TextArea

Textarea digunakan untuk memasukkan teks yang cukup panjang, seperti alamat. Swing menggunakan `JTextArea` untuk membuat Textarea.



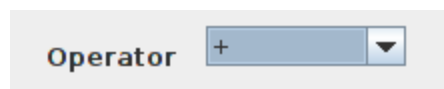
Gambar 3: Contoh Textarea

Di dalam kode, contoh pembuatan `TextArea` adalah:

```
JTextArea xtextarea_alamat=new JTextArea();
```

### X.4 Combo Box

Combo Box digunakan untuk memilih salah satu pilihan dari beberapa pilihan, seperti memilih tipe operator dalam penjumlahan 2 nilai, apakah kedua nilai tersebut harus di tambah, kurang, bagi atau di kali.



Gambar 4: Contoh Combo Box

Swing menggunakan `JComboBox` untuk membuat Combo Box nya. Pada contoh di atas, yaitu Combo Box operator Matematika, kode untuk membuatnya adalah:

```
String[] operatornya = {"+", "-", "/", "x"};  
Joperator=new JComboBox(operatornya);
```

### X.5 Komponen dalam contoh aplikasi

Di bawah ini adalah contoh aplikasi sederhana yang mengandung komponen Radio Button, Check Box dan Textarea.

Source code:

```
package baru;
```

```
// Load library
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class komponen
{
    // Buat instance komponen-komponen
    public JButton xok=new JButton("Keluar");
    public JFrame xframe=new JFrame();
    public JFrame xframebaru=new JFrame();
    public JLabel labelbaru=new JLabel();
    public JTextField xtextfield_nim=new JTextField();
    public JTextField xtextfield_nama=new JTextField();
    public JTextArea xtextarea_alamat=new JTextArea();
    public JRadioButton xradiobutton_pria=new
JRadioButton("Pria");
    public JRadioButton xradiobutton_wanita=new
JRadioButton("Wanita");
    public JCheckBox xmembaca=new JCheckBox("Membaca");
    public JCheckBox xmenulis=new JCheckBox("Menulis");
    public JCheckBox xtennis=new JCheckBox("Tennis");
    public JCheckBox xmemancing=new JCheckBox("Memancing");
    public JCheckBox xlain_lain=new JCheckBox("Lain-lain");
    public JComboBox jurusannya;

    // Buat instance komponen-komponen juga
    public JLabel xlabel_judul=new JLabel("BUKU TAMU");
    public JLabel xlabel_nim=new JLabel("NIM");
    public JLabel xlabel_nama=new JLabel("NAMA");
    public JLabel xlabel_alamat=new JLabel("ALAMAT");
    public JLabel xlabel_jurusan=new JLabel("JURUSAN");
    public JLabel xlabel_jenis_kelamin=new JLabel("JENIS
KELAMIN");
    public JLabel xlabel_hobi=new JLabel("HOBI");

    //konstruktor komponen

    komponen()
    {
        // Array untuk pemilihan jurusan di combobox
        String[] jurusan = {"Teknik Informatika","Sistem
        Informasi","Sistem Komputer","Komputerisasi
        Akuntansi"};
    }
}
```

```
// Mengatur posisi komponen di frame
xlabel_judul.setBounds(230,10, 100, 30);
xlabel_nim.setBounds(15,40, 100, 30);
xlabel_nama.setBounds(15,70, 100, 30);
xlabel_alamat.setBounds(15,100, 100, 30);
xlabel_jurusan.setBounds(15,170, 100, 30);
xlabel_jenis_kelamin.setBounds(15,210, 100, 30);
xlabel_hobi.setBounds(15, 260, 100, 30);

xtextfield_nim.setBounds(120, 40, 200, 25);
xtextfield_nama.setBounds(120, 70, 200, 25);
xtextarea_alamat.setBounds(120, 100, 200, 65);

xradiobutton_pria.setBounds(120, 210, 100, 30);
xradiobutton_wanita.setBounds(120, 230, 100, 30);
xmembaca.setBounds(120, 260, 100, 30);
xmenulis.setBounds(240, 260, 100, 30);
xtennis.setBounds(340, 260, 100, 30);
xmemancing.setBounds(120, 290, 120, 30);
xlain_lain.setBounds(240, 290, 100, 30);
xok.setBounds(20, 340, 100, 30);

// Grouping Radio Button agar hanya dapat dipilih 1
// pilihan
// saja
ButtonGroup group_jen_kel=new ButtonGroup();
group_jen_kel.add(xradiobutton_pria);
group_jen_kel.add(xradiobutton_wanita);
xradiobutton_pria.setActionCommand("pria");
jurusannya=new JComboBox(jurusan);
jurusannya.setBounds(120, 175, 200, 30);

// Add action listener komponen untuk actionnya
xradiobutton_pria.addActionListener(new tampil());
xradiobutton_wanita.setActionCommand("wanita");
xradiobutton_wanita.addActionListener(new tampil());

xok.setActionCommand("keluar");
xok.addActionListener(new tampil());

// Masukkan komponen ke frame

xframe.add(xlabel_judul);
xframe.add(xlabel_nim);
xframe.add(xlabel_nama);
xframe.add(xlabel_alamat);
xframe.add(xlabel_jurusan);
```



```
xframe.add(xlabel_jenis_kelamin);
xframe.add(xlabel_hobi);

xframe.add(xtextfield_nim);
xframe.add(xtextfield_nama);
xframe.add(xtextarea_alamat);
xframe.add(xradiobutton_pria);
xframe.add(xradiobutton_wanita);
xframe.add(xmembaca);
xframe.add(xtennis);
xframe.add(xmenulis);
xframe.add(xmemancing);
xframe.add(xlain_lain);
xframe.add(xok);
xframe.add(jurusannya);

// Atur properti frame
xframe.setTitle("Komponen-komponen");
xframe.setSize(500,450);
xframe.setLayout(null);
xframe.setVisible(true);
xframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

// Class tampil implement ActionListener
private class tampil implements ActionListener {
    public void actionPerformed(ActionEvent e)
    {
        String xnama,xnim;
        String proses=e.getActionCommand();
        if (proses.equals("pria"))
        {
            JOptionPane.showMessageDialog(null,"Pria
Dipilih","Pesan",JOptionPane.INFORMATION_MESSAGE);
        }
        else if (proses.equals("wanita"))
        {
            JOptionPane.showMessageDialog(null,"Wanita
Dipilih","Pesan",JOptionPane.INFORMATION_MESSAGE);
        }

        if (proses.equals("keluar"))
        {
            xframe.dispatchEvent(new WindowEvent(xframe,
WindowEvent.WINDOW_CLOSING));
        }
    }
}
```

```
}  
}  
  
public static void main (String args[])  
{  
    new komponen();  
}  
}
```

The screenshot shows a Java Swing window titled "Komponen-komponen" with standard window controls (minimize, maximize, close). The window contains a form titled "BUKU TAMU" with the following fields and controls:

- NIM:** A single-line text input field.
- NAMA:** A single-line text input field.
- ALAMAT:** A multi-line text area.
- JURUSAN:** A dropdown menu currently displaying "Teknik Informatika".
- JENIS KELAMIN:** Two radio buttons labeled "Pria" and "Wanita".
- HOBI:** Five checkboxes labeled "Membaca", "Menulis", "Tennis", "Memancing", and "Lain-lain".
- Keluar:** A button at the bottom left of the form.

Gambar 4: Tampilan program komponen

## Bab X

### Koneksi Database dengan JDBC

#### Mengenal JDBC

Java Database Connectivity (JDBC) adalah API yang di gunakan Java untuk melakukan koneksi dengan aplikasi lain atau dengan berbagai macam database. JDBC memungkinkan kita untuk melakukan 3 hal, yaitu : Konek ke sumber data, Mengirimkan query ke database, menerima dan mengolah resultset yang di peroleh dari database.

JDBC memiliki 4 komponen, yaitu :

#### 1. JDBC API

JDBC API menyediakan metode akses yang sederhana ke Database. Dengan menggunakan JDBC API, kita bisa membuat program yang dapat mengeksekusi SQL, menerima hasil Resultset dan mengolah data dalam database. JDBC API juga mempunyai kemampuan untuk berinteraksi dengan lingkungan terdistribusi dari jenis sumber data yang berbeda-beda.

#### 2. JDBC Driver Manager

JDBC Driver Manager bertugas untuk mendefinisikan object-object yang dapat di gunakan untuk melakukan koneksi ke sebuah Database.

#### 3. JDBC Test Suite

JDBC Test Suite membantu kita untuk mencari driver mana yang cocok di gunakan untuk melakukan sebuah koneksi ke Database. Test yang di lakukan cukup test-test sederhana yang memastikan fitur-fitur penting JDBC berjalan lancar.

#### 4. JDBC-ODBC Bridge

Bridge ini menyediakan fasilitas JDBC untuk melakukan koneksi ke sumber data menggunakan ODBC (Open Database Connectivity) driver. Kita perlu meload driver ODBC di setiap komputer client untuk dapat menggunakan bridge ini. Sebagai konsekuensi nya, cara ini hanya cocok di lakukan di lingkungan intranet, di mana isu instalasi tidak menjadi masalah.

#### Database Driver

JDBC memerlukan Database Driver untuk melakukan koneksi ke sebuah Database. Database driver ini bersifat spesifik untuk setiap jenis database. Database driver biasa nya di buat oleh pihak pembuat database nya, namun tidak jarang juga komunitas atau pihak ketiga menyediakan database driver untuk database tertentu.

Pada kesempatan ini, kita menggunakan Mysql sebagai database yang di hubungkan dengan Java dan sebagai Driver atau connector nya kita menggunakan Mysql Connector/J sebagai JDBC Driver resmi untuk Mysql. Connector Mysql ini dapat di download di website Mysql di link <http://dev.mysql.com/downloads/connector/j/> dan dapat di pergunakan untuk semua sistem operasi populer, seperti Windows, Mac dan Linux. Untuk manual nya, dapat kita peroleh di website Mysql juga, di link : <http://dev.mysql.com/doc/refman/5.1/en/connector-j.html>, namun di modul ini juga di terangkan sedikit banyak mengenai penggunaan Mysql di dalam Java.

Pada saat modul ini di buat, kita akan menggunakan Connector 5.1 yang merupakan connector stabil terbaru. Connector 5.1 ini dapat di pergunakan untuk Mysql versi 4.1,5.0,5.1,5.4 dan 5.5.

#### Mysql

Untuk dapat menghubungkan program Java ke dalam Database Mysql, tentu nya harus terinstall Database Mysql di dalam komputer kita. Kita dapat mendownload file instalasi Mysql di situs

[www.mysql.com](http://www.mysql.com) atau dengan menginstall Xampp, yaitu aplikasi bundle yang terdapat php, mysql, phpmyadmin dan aplikasi lainnya. Xampp umum di gunakan oleh php programmer, namun di sini kita dapat menggunakan fasilitas Mysql dan PhpMyadmin nya untuk memanager Mysql dengan GUI.

### **Instalasi Driver**

Pada kesempatan kali ini, akan di jelaskan mengenai instalasi di dalam Linux.

7. Download file driver dari link : <http://dev.mysql.com/downloads/connector/j/>
8. File hasil download mysql-connector-java-5.1.18.tar.gz (Pada saat modul ini di download, sesuaikan dengan nama file yang anda peroleh pada proses install selanjut nya)
9. Pindahkan ke folder /usr/local
10. Extract file tersebut. Gunakan user root :  
# gunzip mysql-connector-java-5.1.18.tar.gz  
Akan menghasilkan file mysql-connector-java-5.1.18.tar
11. Install file tar tersebut dengan cara : # tar -xvf mysql-connector-java-5.1.18.tar
12. Akan membentuk folder bar bernama : mysql-connector-java-5.1.18
13. Selesai

### **Testing Driver**

Untuk menguji driver tersebut di dalam program Java, dapat kita pergunakan code sbb :

Nama file : jdbc\_select

Nama Package : latihan

1. package latihan;
- 2.
3. import java.sql.\*;
- 4.
5. public class jdbc\_select{
6. public static void main(String[] args) {
7. System.out.println("MySQL Connect Example.");
8. Connection conn = null;
9. String url = "jdbc:mysql://localhost:3306/";
10. String dbName = "test";
11. String driver = "com.mysql.jdbc.Driver";
12. String userName = "root";
13. String password = "opansan";
14. try {
15. Class.forName(driver).newInstance();
16. conn = DriverManager.getConnection(url+dbName,userName,password);
17. System.out.println("Connected to the database");
18. try{
19. DatabaseMetaData dbm = conn.getMetaData();
20. String[] types = {"TABLE"};
21. ResultSet rs = dbm.getTables(null,null,"%",types);
22. System.out.println("Table name:");
23. while (rs.next()){
24. String table = rs.getString("TABLE\_NAME");
25. System.out.println(table);
26. conn.close();
27. }
28. }
29. catch (SQLException s){

```
30. System.out.println("No any table in the database");
31. }
32. }
33. catch (Exception e){
34. e.printStackTrace();
35. }
36. }
37. }
```

**Keterangan :**

3. *import java.sql.\*;*

Ket : Untuk menggunakan database dengan JDBC, kita perlu mengimport *java.sql.\**, yang mana di dalam *java.sql.\** tersebut kita akan menggunakan class-class di dalam nya, seperti *DriverManager*, *Class.forName* dan *DatabaseMetaData*.

8. *Connection conn=null*

Ket : Pemberian nilai awal null untuk variabel *conn* dengan tipe *connection*.

9. *String url = "jdbc:mysql://localhost:3306/";*

10. *String dbName = "test";*

11. *String driver = "com.mysql.jdbc.Driver";*

12. *String userName = "root";*

13. *String password = "opansan";*

Ket : Memberi nilai pada variabel *url* (*Url jdbc*), *dbName* (*Nama database yang akan di gunakan di Mysql*), *driver* (*Driver jdbc mysql*), *userName* (*user name Mysql*), *password* (*Password Mysql*)

14. *try {*

Ket : Kita menggunakan *syntax try and catch*

15. *Class.forName(driver).newInstance();*

Ket : Menggunakan *Class.forName* method, meload secara *dynamic class com.mysql.jdbc.Driver* yang tersimpan nama nya di dalam *String driver*.

16. *conn = DriverManager.getConnection(url+dbName,userName,password);*

Ket : *Class DriverManager*, service dasar untuk mengatur *driver JDBC*. Menghubugnkan *Mysql* dengan menyertakan *url,dbName, userName dan password*

19. *DatabaseMetaData dbm = conn.getMetaData();*

Ket : Menggunakan *interface DatabaseMedataData* untuk object *dbm*

20. *String[] types = {"TABLE"};*

21. *ResultSet rs = dbm.getTables(null,null,"%",types);*

Ket : *Persiapan untuk mendapatkan nama tabel di sebuah database.*

23. *while (rs.next()){*

24. *String table = rs.getString("TABLE\_NAME");*

25. *System.out.println(table);*

26. *conn.close();*

26. }

Ket : Mencetak nama tabel

29. *catch (SQLException s){*

```
30. System.out.println("No any table in the database");
```

Ket : Menutup try dengan catch dan mencetak output jika tidak ada tabel di dalam database.

Jika program di atas sukses terhubung dengan Mysql dan ada tabel yang tercetak di dalam database yang di pilih, maka output yang keluar dari program di atas adalah sbb :

MySQL Connect Example.

Connected to the database

Table name:

data

data\_parent

math

mhs

pages

pemasaran

person

tbl\_user

**BUILD SUCCESSFUL** (total time: 2 seconds)

## Bab XI

### Mengambil dan Memanipulasi data dari Database

Proses pengambilan data dari database memerlukan suatu class untuk menampung data yang berhasil di ambil, class tersebut harus mengimplement interface Resultset.

JDBC menyediakan 3 tipe result set untuk tujuan berbeda :

1. TYPE\_FORWARD\_ONLY : Result set tersebut tidak bisa berjalan mundur, result set hanya bisa berjalan mundur, result set hanya bisa berjalan maju dari baris pertama hingga baris terakhir.
2. TYPE\_SCROLL\_INSENSITIVE : Result set dapat berjalan maju mundur. Result set dapat berjalan maju dari row pertama hingga terakhir atau bergerak bebas berdasarkan posisi relatif atau absolute.
3. TYPE\_SCROLL\_SENSITIVE : Result set dapat berjalan maju mundur. Result set dapat berjalan maju dari row pertama hingga terakhir atau bergerak bebas berdasarkan posisi relative atau absolute.

Instance dari object bertipe ResultSet di perlukan untuk menampung hasil kembalian data dari database. Sebelum kita bisa memperoleh instance dari ResultSet, kita harus membuat instance dari class Statement. Class Statement mempunyai method executeQuery yang di gunakan untuk menjalankan perintah Query dalam database, kemudian mengembalikan data hasil eksekusi query ke dalam object ResultSet.

Berikut ini adalah contoh kode untuk membuat instance class Statement, kemudian menjalankan query untuk mengambil data dari database, yang hasil nya di pegang oleh ResultSet :

```
Statement statement = conn.createStatement ( ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_READ_ONLY);  
ResultSet rs=statement.executeQuery("Select * from mhs");
```

ResultSet akan meletakkan kursornya (posisi pembacaan baris) di sebuah posisi sebelum baris pertama. Untuk menggerakkan kursor maju, mundur, ke suatu posisi relatif atau ke suatu posisi absolute tertentu, gunakan metode-metode dari ResultSet :

27. next() – Mengarahkan kursor maju satu baris.
28. previous () – Mengarahkan kursor mundur satu baris.
29. first () – Mengarahkan kursor ke baris pertama.
30. last () – Mengarahkan kursor ke baris terakhir.
31. BeforeFirst() – Mengarahkan kursor ke sebelum baris pertama.
32. AfterLast() – mengarahkan kursor ke setelah baris terakhir.
33. Relative (int rowNum) – mengarahkan kursor relatif dari posisinya yang sekarang. Set nilai rows dengan nilai positif untuk maju dan nilai negatif untuk mundur.
34. absolute(int rowNum) – Mengarahkan kursor ke posisi tertentu sesuai dengan nilai rowNum dan tentu saja nilai nya harus positif.

Interface Resultset menyediakan method getter untuk mengakses nilai dari setiap kolom dalam baris yang sedang aktif. Parameter fungsi getter bisa menerima nilai index dari kolom ataupun nama kolom nya.

Contoh kode untuk melakukan koneksi hingga mengambil data dari database :

```
package latihan;
```

```
import java.sql.*;
```

```
public class jdbc_bab12 {  
    public static void main (String args[])  
    {  
        Connection conn = null;  
        String url = "jdbc:mysql://localhost:3306/";  
        String dbName = "test";  
        String driver = "com.mysql.jdbc.Driver";  
        String userName = "root";  
        String password = "opansan";  
        try {  
            Class.forName(driver).newInstance();  
            conn = DriverManager.getConnection(url+dbName,userName,password);  
            Statement  
statement=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,  
                                ResultSet.CONCUR_READ_ONLY);  
            ResultSet rs=statement.executeQuery("select * from mhs");  
            while (rs.next())  
            {  
                System.out.print(rs.getString("nim")+ " ");  
                System.out.print(rs.getString("nama")+ " ");  
                System.out.println(rs.getString("no_telp"));  
            }  
        }  
        catch (Exception e)  
        {  
            e.printStackTrace();  
        }  
    }  
}
```



## Bab XII

### Mengambil dan Memanipulasi data dari Database (2)

Pada Pertemuan sebelumnya, kita telah membahas mengambil dan memanipulasi data dari database dengan menjalankan query select untuk mengambil data dari sebuah tabel. Pada kesempatan ini, kita akan membahas penggunaan query Update dan Delete untuk memanipulasi data di sebuah tabel.

Kita akan menggunakan class statement.executeUpdate, sebagai contoh :

```
result=statement.executeUpdate ("update mhs set nama="Reni Harun" where  
nim='1011500101");  
result=statement.executeUpdate ("delete mhs where nim='1011500101");
```

#### **Contoh source code :**

```
import java.sql.*;  
  
public class bab13  
{  
    public static void main (String args[])  
    {  
        Connection conn = null;  
        String url = "jdbc:mysql://localhost:3306/";  
        String dbName = "test";  
        String driver = "com.mysql.jdbc.Driver";  
        String userName = "root";  
        String password = "opansan";  
        try {  
            Class.forName(driver).newInstance();  
            conn = DriverManager.getConnection(url+dbName,userName,password);  
  
            Statement  
statement=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,  
                                ResultSet.CONCUR_READ_ONLY);  
            statement.executeUpdate ("update mhs set nama='Budiman' where  
nim='1011500101' ");  
  
            ResultSet rs=statement.executeQuery("select * from mhs");  
  
            while (rs.next())  
            {  
                System.out.print(rs.getString("nim")+ " ");  
                System.out.print(rs.getString("nama")+ " ");  
                System.out.println(rs.getString("no_telp"));  
            }  
        }  
        catch (Exception e)  
        {  
            e.printStackTrace();  
        }  
    }  
}
```

}  
}

## **Bab XIII**

### **Membangun sebuah Aplikasi Database dengan Java Desktop Application**

Budiman adalah seorang mahasiswa Ilmu Komputer di sebuah Perguruan Tinggi di Jakarta. Saat ini ia terlihat sedang gelisah, karena kiriman uang dari Orang tua nya berkurang, karena orang tua nya sedang berkonsentrasi dalam persiapan pernikahan kakak nya, termasuk dalam hal biaya. Biaya kuliah Budiman agak terpengaruh. Tatkala ia sedang menyendiri di sebuah Masjid sambil memikirkan perekonomian nya yang terguncang, ia menerima sebuah sms dari Joko, teman nya yang berkuliah di Bandung. “Man, bisa tolong saya tidak, membuat sebuah aplikasi Entry Data sederhana, tapi jadi nya harus besok. Maaf man ya, mendadak, soal nya di butuhkan juga mendadak sich, untuk keperluan Organisasi. Kebutuhan dan proses informasi serta output nya serta lain nya sudah saya kirim via email. Oh iya, ada charge nya juga, budget kami hanya 600 ribu, cukup kan ?. Saya yakin kamu bisa, kamu kan emang hobi koding kan ?! :)”.

Betapa senang nya hati Budiman mendapat sms dari Joko tentang permintaan pembuatan aplikasi database dan segera ia pulang ke kost an nya untuk membuka e-mail. “Hmm ... harus jadi besok, singkat sekali”, gumam dia . Namun dia ingat akan cara Akhmad, rekan nya yang lain dalam membuat aplikasi database cepat menggunakan Java dan dia sempat mempelajari nya. “Aha ... !”, aku akan membuat seperti cara Akhmad membuat aplikasi database dengan cara cepat, Ya, dengan menggunakan Java Desktop Application di Netbeans.

Di bawah ini kita akan mempelajari pembuatan aplikasi Database dengan Java Desktop Application, seperti yang di kerjakan Budiman terhadap project komersial pertama nya.

-----

Cara membuat Aplikasi Database dengan Java Desktop Application :

Aplikasi yang di buat Budiman adalah aplikasi pendaftaran calon anggota BEM, di mana terdapat field entrian yang dapat di create, read, update dan delete. Field entrian nya adalah : nim, nama, alamat, e-mail dan no.hp.

#### **Langkah-langkah nya adalah sbb :**

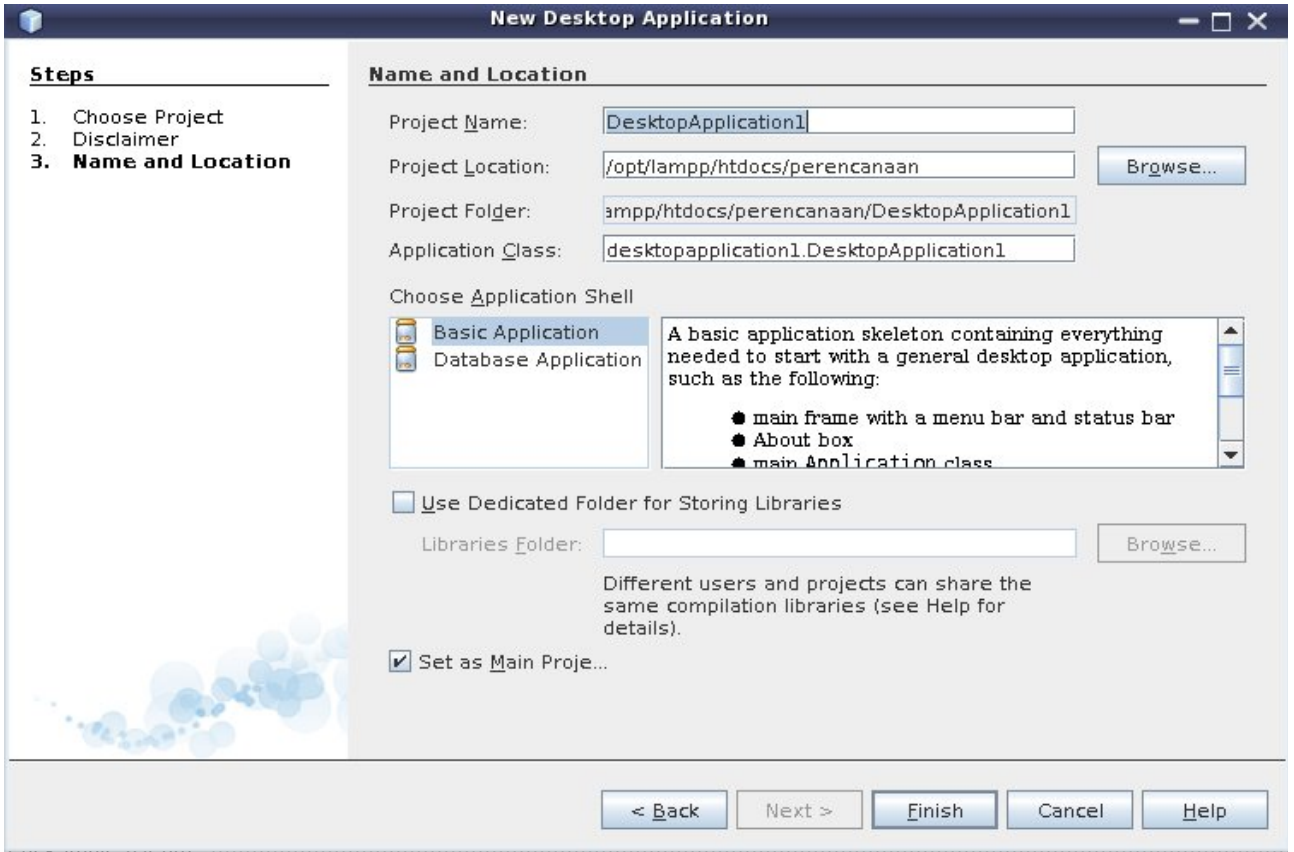
1. Membuat table anggota pada database Mysql.

<b>No</b>	<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	Int (4)	PK,AI
2	nim	Varchar (10)	
3	nama	Varchar (40)	
4	alamat	Varchar (60)	
5	email	Varchar (20)	
6	no_hp	Varchar (20)	
7	Keterangan	Varchar (60)	

Tabel 14.1 Struktur tabel anggota

2. Pilih File → New Project → Java Desktop Application

3.



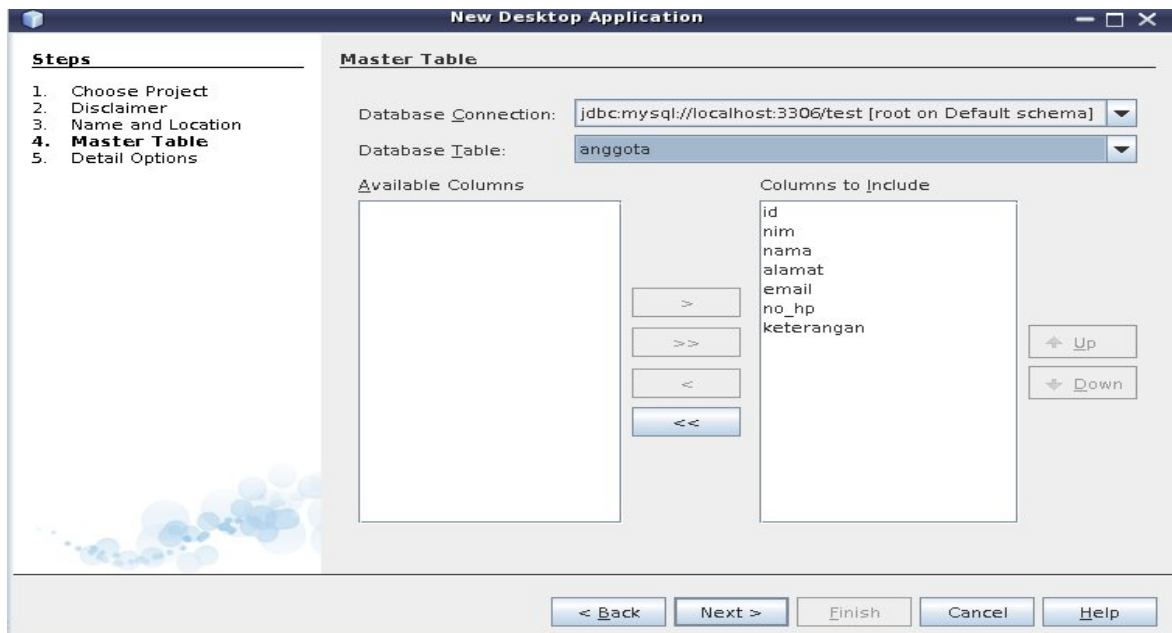
Gambar 14.1 : Pengisian data project

Tulis Project name : perdana

Project Location : bebas

Project Folder : bebas

Application class : Perdana.PerdanaApp



Application Shell : Database Application

4. Gambar 14.2 : Pemilihan Koneksi Database

Pilih Database Connection dan Database Table. Jika belum ada Database Connection, pilih pilihan “New Database Connection”, maka akan muncul window untuk membuat koneksi database yang baru. Gunakan Database Mysql dalam kesempatan ini.

5. Pada Detail Options, pada Create Detail Area as, pilih Textfields.

6. Klik next. Selesai. Maka akan muncul jendela tampilan seperti di bawah ini :

Gambar 14.3 : Hasil Form Aplikasi

Id	Nim	Nama	Alamat	Email	No Hp	Keterangan...
----	-----	------	--------	-------	-------	---------------

**Id:**

**Nim:**

**Nama:**

**Alamat:**

**Email:**

**No Hp:**

**Keterangan:**

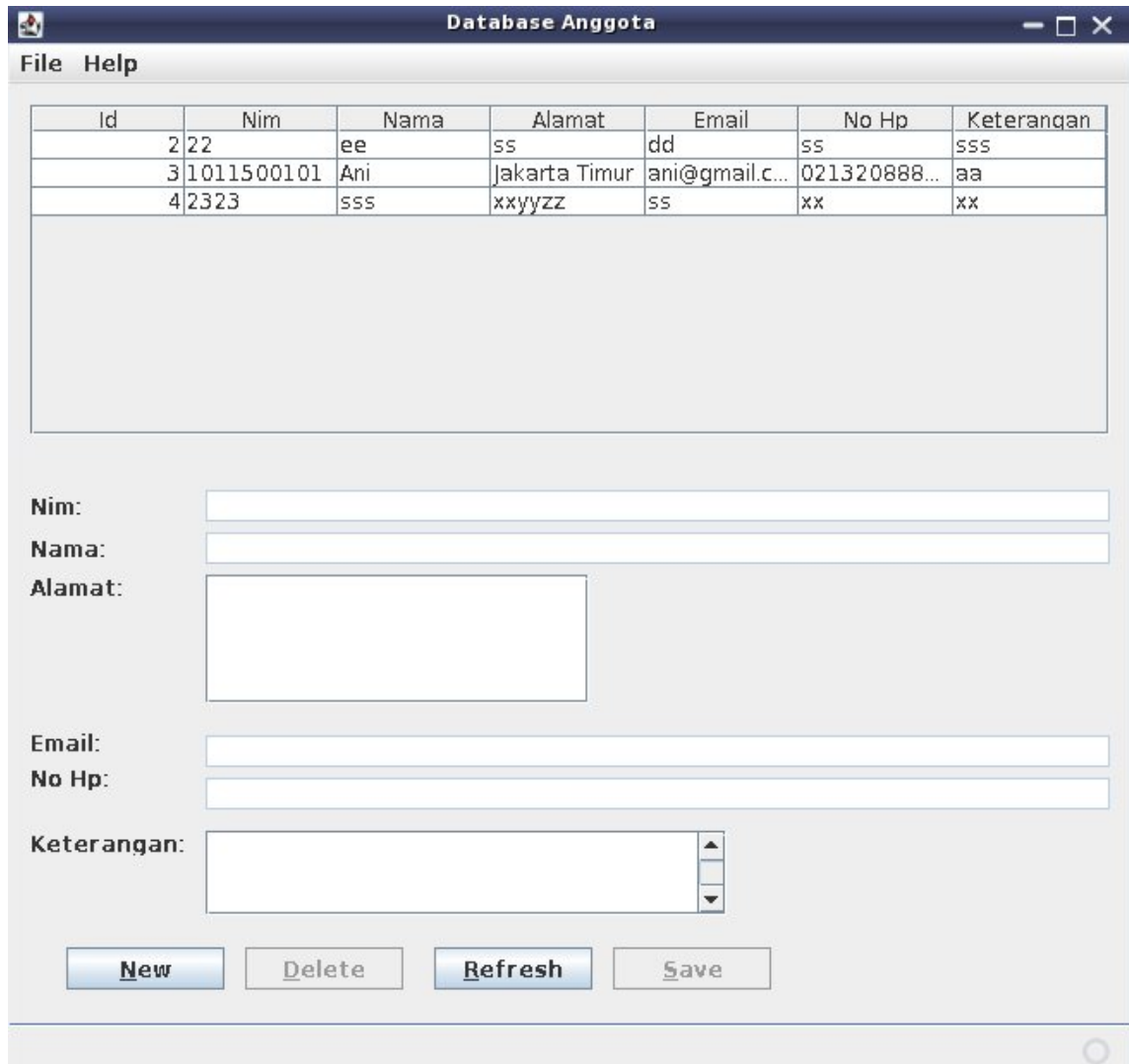
Pada tahap ini, program sudah selesai di buat dan sudah dapat di gunakan untuk menginput, menampilkan, update dan delete data. Namun kita perlu memodif form entri nya, yakni :

1. Entrian Id, tidak perlu ada, karena id otomatis di tambahkan, Auto Increment.
2. Entrian Alamat adalah Text area, bukan Field text.
3. Entrian keterangan adalah Text area, bukan Field text.

Pada tahap ini, juga terbentuk secara otomatis file PerdanaApp.java, PerdanaAboutBox.java dan PerdanaView.java. PerdanaAboutBox.java dan PerdanaView.java, memiliki form design nya dan baik source code maupun tampilan design nya dapat di modif, kecuali beberapa bagian source code yang tidak dapat di modif, seperti deklarasi variabel.

Untuk memodif form entry, seperti di inginkan kondisi di atas, langkah-langkah nya sbb :

1. Hapus lah label id dan text field nya.
2. Hapus lah JTextField alamat, ganti dengan JTextArea.
3. Hapus lah JTextField keterangan, ganti dengan JTextArea.
4. Sesuaikan letak Swing Control pada form tsb terhadap perubahan di 3 langkah ini, termasuk ukuran JTextField dan JTextArea juga dapat di ubah.



Gambar 14.4 : Hasil form aplikasi setelah di modifikasi

### Menjalankan Aplikasi di luar Netbeans

Jika aplikasi sudah selesai di buat, kita dapat menjalankan aplikasi tersebut di luar dari Netbeans, yaitu dengan cara :

1. Pilih Tab AnggotaApp.java. Lalu pilih Run → Clean and Build Main Project (Shift F11)  
Akan terlihat output dari Script Ant. Proses tersebut adalah sbb :
  - a. Folder output yang terbentuk pada build sebelum nya di hapus (Umum nya folder build dan dist).
  - b. Folder build dan dist baru, di tambahkan ke dalam folder project.
  - c. Semua sumber di kompil ke dalam .class dan di tempatkan di PROJECT\_HOME/build
  - d. Sebuah file JAR yang berisi project kita di buat di dalam folder PROJECT\_HOME/dist

- e. Jika kita menspesifikasikan 1 atau beberapa library (Untuk di tambahkan di JDK), folder lib di buat di dalam folder dist. Library di copykan ke dalam dist/lib.
  - f. File manifest di dalam JAR di update.
2. Buka File manager, dan double click file jar yang ada di dalam folder PROJECT\_HOME/dist. Aplikasi akan running.

## BAB XIV

### Menampilkan data dari database di dalam Jtable

Setelah berhasil membuat form entrian data, kita ingin melihat hasil tampilan dan salah satu hasil tampilan yang umum dan baik, adalah menampilkan data dengan Table. Di Java Swing, kita dapat menggunakan Jtable dalam penampilan data. Jtable adalah salah satu komponen dari Java Swing Control.



The image shows a screenshot of a Java Swing window titled 'JTable'. The window contains a table with three columns: 'NIM', 'Nama', and 'No Telp'. The table has six rows of data. Below the table, there is a large empty rectangular area.

NIM	Nama	No Telp
1011500101	Budiman	021-3334455
1011500006	Muhammad Umar	021-3335599
1011500008	Aisyah Syarifah	021-3335588
1022500009	Akhmad Sofwan	021-3678977
1022500003	Julitawati	021-3337766
1011500145	Nawindah	735-090912

Gambar XV-1 : JTable

Berikut adalah coding sample menampilkan data ke dalam Jtable, berikut koneksi database dan menampilkan

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package latihan;

/**
 *
 * @author CHIPRODOY
 * Modified by Sofwan
 */

// Import Java package
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
```



```
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;

public class jtable2 {
    public jtable2 ()
    {
        // Variabel Koneksi DB
        String url="jdbc:mysql://localhost/";
        String driver="com.mysql.jdbc.Driver";
        String dbname="test";
        String username="root";
        String password="opansan";

        try {
            Connection conn=null;
            Class.forName(driver).newInstance();
            // Koneksi DB dengan DriverManager
            conn=DriverManager.getConnection(url+dbname,username,password);
            Statement stmt=conn.createStatement();
            // Query tampil data
            ResultSet rs=stmt.executeQuery("select * from mhs");

            ResultSetMetaData rsmd=rs.getMetaData();
            int intnum=rsmd.getColumnCount(); // Menghitung jumlah kolom
            String [] kolom={"NIM","Nama","No Telp"};
            DefaultTableModel model=new DefaultTableModel();
            model.setColumnIdentifiers(kolom); // Set Header Kolom //
            while (rs.next())
            {
                // Tampilkan setiap record dengan looping
                String [] data=new String [3];
                data[0]=rs.getString("nim");
                data[1]=rs.getString("nama");
                data[2]=rs.getString("no_telp");
                model.addRow(data);
            }
            JTable tbl2=new JTable(model);
            JFrame frm=new JFrame(); // Buat Object baru frm
            frm.add(new JScrollPane(tbl2));
            frm.setSize(500,300); // Set ukuran frame
            frm.setVisible(true);
        }
        catch(ClassNotFoundException e)
        {
            Logger.getLogger(jtable2.class.getName()).log(Level.SEVERE,null,e);
        }
        catch(Exception e)
        {
            System.err.println(e);
            Logger.getLogger(jtable2.class.getName()).log(Level.SEVERE,null,e);
        }
    }
}
```

```
    }  
    public static void main (String args[])  
    {  
        SwingUtilities.invokeLater(new Runnable()  
        {  
            public void run()  
            {  
                new jTable2();  
            }  
        });  
    }  
}
```

## Penutup

Demikianlah tutorial dasar-dasar pemrograman Java, semoga dasar-dasar ini dapat membantu anda dalam mengenal pemrograman Java dan menjadi landasan bagi pembelajaran pemrograman Java anda selanjutnya.

## Referensi

1. The Java Tutorials, <https://docs.oracle.com/javase/tutorial/>
2. Ifnu Bima, Materi pelatihan Java Swing versi :0.1-07.03
3. Java Tutorial, <https://www.tutorialspoint.com/java>

## Biografi Penulis

**Akhmad Sofwan** - Menyelesaikan Sarjana di Teknik Informatika Universitas Budi Luhur ,tahun 2001 dan Magister Ilmu Komputer, Fakultas Ilmu Komputer ,Universitas Indonesia, tahun 2017. Penulis saat ini adalah seorang Freelance Software Developer. Selain itu, penulis juga mengajar ilmu komputer di Perguruan Tinggi Raharja, Tangerang dan di Program Vokasi, Universitas Indonesia. Penulis juga aktif di Pusat Kajian Biostatistik dan Informatika Kesehatan (PKBIK) Fakultas Kesehatan Masyarakat, Universitas Indonesia sebagai Software Developer dan Peneliti.