

Modul Praktikum

Object Oriented Programming



Program Studi Teknik Informatika

STMIK STIKOM Indonesi

DAFTAR ISI

MODUL 1 Pengenalan Java disertai First Simple Program	3
MODUL 2 Pengenalan Variable, Tipe data, Operator dan metode input di JAVA.....	11
MODUL 3 Perulangan, Percabangan dan Array Pada Java.....	20
MODUL 4 Class, Object, Elemen dalam Class Pada Java	34
MODUL 5 Enkapsulasi, Constructor, overloading constructor	42
Modul 6 Inheritance, Overloading, Overriding.....	47
MODUL 7 JAVA GUI Swing.....	56
MODUL 8 JavaGUI , JDBC Connector & MySQL	70
MODUL 9 JavaGUI Database Dengan iReport.....	107

MODUL 1
Pengenalan Java disertai First Simple Program
(Pertemuan 1)

Tujuan :

1. Mengenal dan mempersiapkan lingkungan kerja Java
2. Membuat program sederhana dengan Java dengan Netbean
3. Mengkompilasi dan menjalankan program Java Netbean
4. Menganalisa beberapa problem yang terjadi saat pemrograman dan memberikan solusi

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang Netbeans
2. Sebutkan keunggulan bahasa pemrograman JAVA

DASAR TEORI

Bahasa pemrograman Java pada awalnya dikembangkan oleh Sun Microsystems, yang digagas oleh James Gosling dan dirilis pada tahun 1995 sebagai komponen inti dari Sun Microsystems.s platform Java (Java 1.0 [J2SE]).

Sun Microsystems telah berganti nama menjadi J2 versi baru sebagai Java SE, Java EE dan Java ME masing-masing. Dalam java semuanya adalah Obyek. Java dapat dengan mudah diperpanjang karena didasarkan pada model Obyek. Platform independen: Tidak seperti banyak bahasa pemrograman lain termasuk C dan C + + ketika Java dikompilasi, tidak dikompilasi ke dalam mesin platform tertentu, bukan menjadi platform independen kode byte. Ini kode byte didistribusikan melalui web dan ditafsirkan oleh Mesin virtual (JVM) di mana platform itu yang dijalankan.

Java dirancang agar mudah untuk belajar. Jika Anda memahami konsep dasar OOP java akan mudah untuk menguasai. Dengan fitur aman Java memungkinkan untuk mengembangkan bebas virus, sistem tamper-bebas. Teknik otentifikasi didasarkan pada enkripsi kunci public. Java compiler menghasilkan format file objek arsitektur netral yang membuat kode dikompilasi menjadi executable pada banyak prosesor, dengan hadirnya sistem runtime Java. Menjadi arsitektur netral dan tidak memiliki implementasi aspek tergantung dari spesifikasi Java membuat portabel. Compiler dan Java ditulis dalam ANSI C dengan batas portabilitas bersih yang merupakan subset POSIX. Java membuat upaya untuk menghilangkan situasi rawan kesalahan dengan menekankan terutama pada waktu kompilasi kesalahan pengecekan dan pemeriksaan runtime.

KEGIATAN PRAKTIKUM

1. Persiapan

1.1 Peralatan yang Diperlukan

Pada pelatihan Java Dasar ini, peralatan yang diperlukan adalah :

1. Java Development Kit versi 1.6 keatas.
2. Java Runtime Environtment versi 1.6 keatas.
3. NetBeans IDE versi 6.9 keatas.

1.2 Java Development Kit

Java Development Kit merupakan perangkat lunak yang digunakan untukmelakukan proses kompilasi dari kode Java menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh Java Runtime Environtment.

1.3 Java Runtime Environtment

Java Runtime Environtment merupakan perangkat lunak yang digunakan untukmenjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dari JDK yang digunakan untuk membangun aplikasi agar aplikasi dapat berjalan sesuai dengan yang diharapkan.

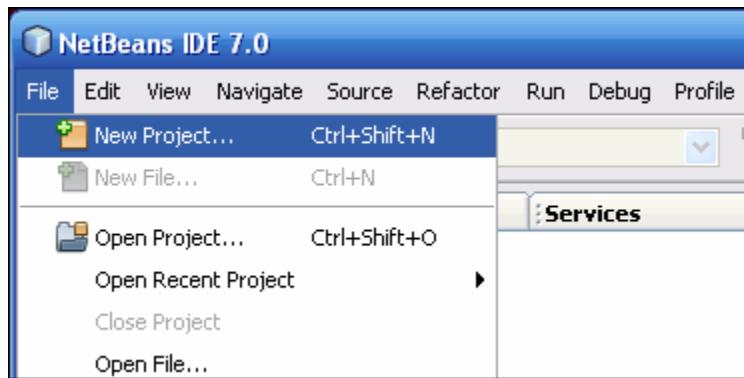
1.4 NetBeans IDE

NetBeans IDE merupakan perangkat lunak yang digunakan untuk membangun perangkat lunak yang lain. NetBeans IDE dapat digunakan untuk membangun perangkat lunak berbasis Java Standard Edition, Java Enterprise Edition, Java Micro Edition, JavaFX, PHP, C/C++, Ruby, Groovy dan Python.

2.Sintaks cetak string sederhana

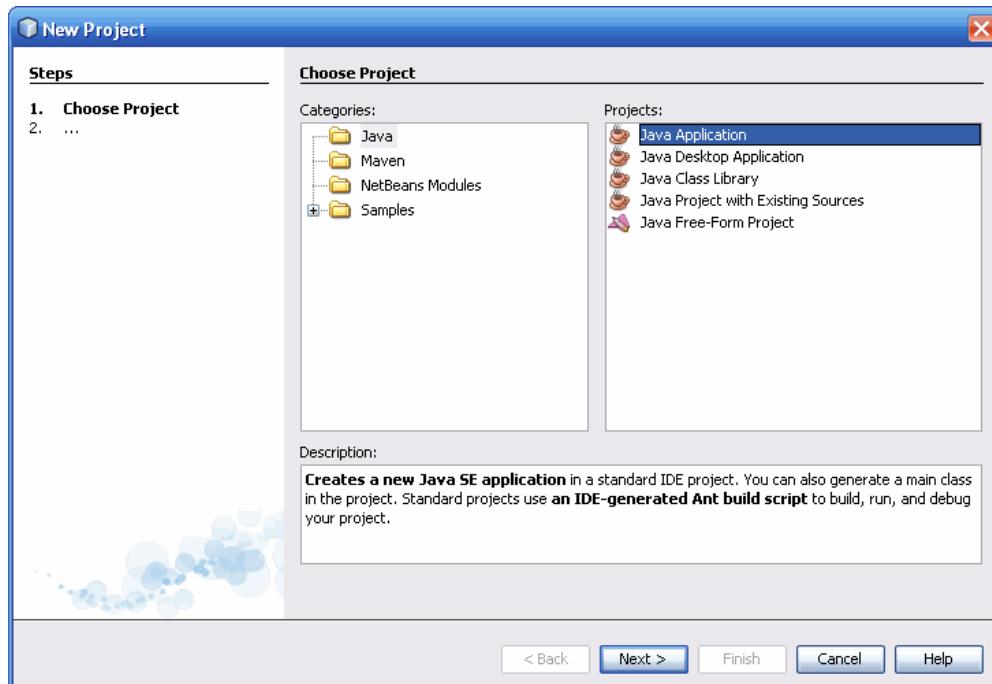
2.1 Membuat Project Sederhana

Buka Netbeans pada PC anda, kemudian Klik **File | New Project** Seperti contoh pada **Gambar 1.1** berikut :



Gambar 1.1 Create New Project

Dalam New Project wizard, memperluas kategori Java dan pilih Java Application seperti yang ditunjukkan pada **Gambar 1.2** berikut:

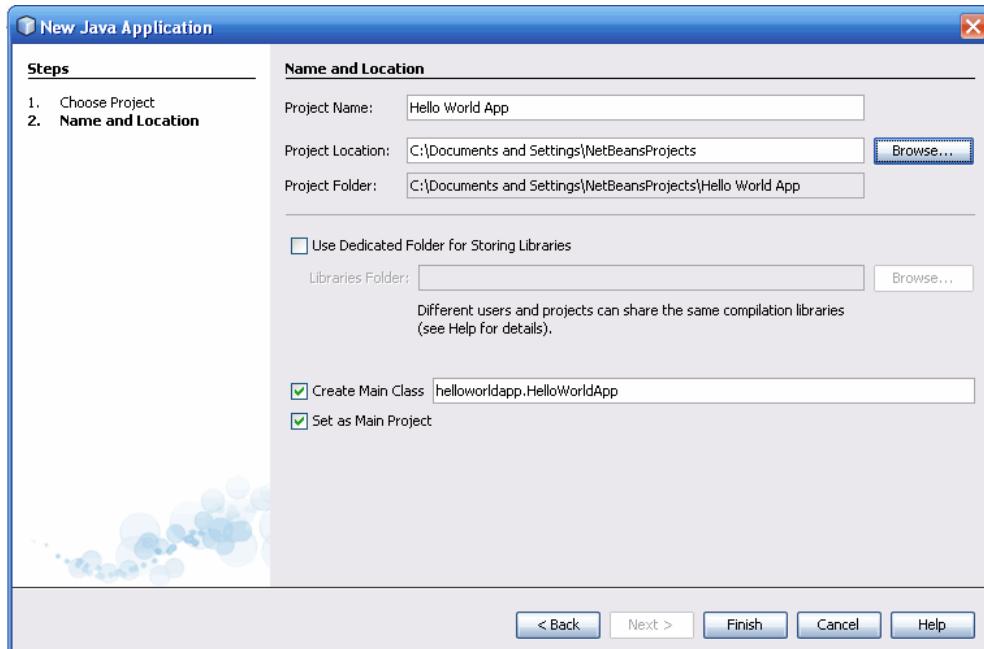


Gambar 1.2 Create New Project 2

Dalam Nama dan lokasi halaman wizard, lakukan hal berikut (seperti yang ditunjukkan pada gambar di bawah):

- Pada Project Name field, Ketik Hello World App.
- Pada Create Main Class field, Ketik helloworldapp.HelloWorldApp.
- Biarkan kotak centang Set sebagai Main Project dipilih.

Seperti Pada contoh **Gambar 1.3** berikut:

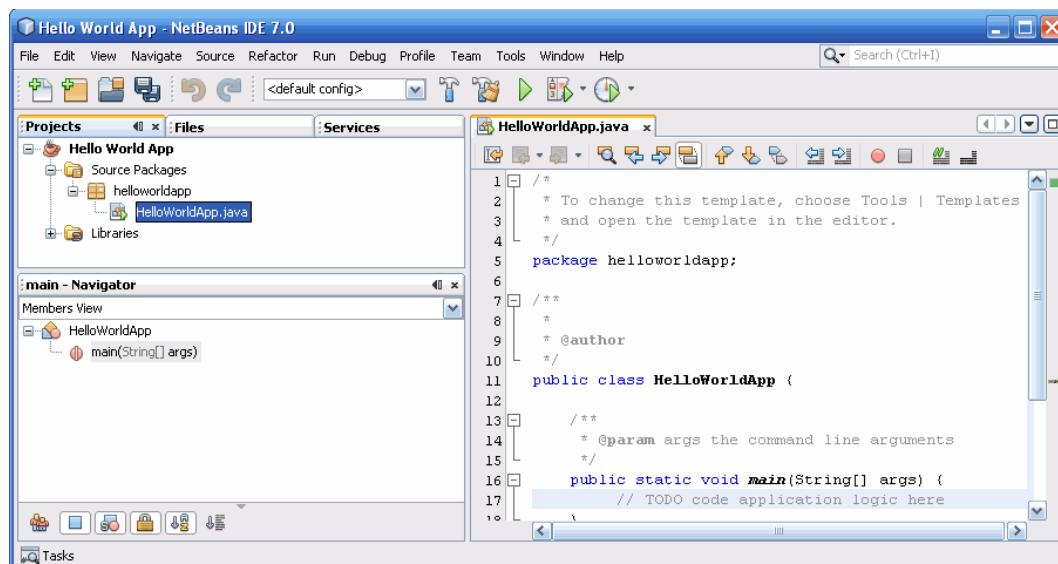


Gambar 1.3 Create New Project 3

Proyek dibuat dan dibuka dalam IDE. Anda akan melihat komponen-komponen berikut :

- Jendela Projects, yang berisi tampilan pohon dari berbagai komponen proyek, berisi file sumber, pustaka yang mendasari kode Anda tergantung pada, dan sebagainya.
- Jendela Source Editor dengan sebuah file yang bernama HelloWorldApp terbuka.
- Jendela Navigator, yang dapat Anda gunakan untuk navigasi cepat di antara elemen-elemen dalam kelas yang dipilih.

Seperi Pada Contoh Gambar 1.4 Berikut :



Gambar 1.4 Create New Project 4

Selanjutnya lakukan pengkodean pada listing Project, masukan syntax berikut tepat di main [string]

```
args {  
    System.out.println("Hello World!");  
}
```

Atau tepat di bawah komentar // TODO code application logic here

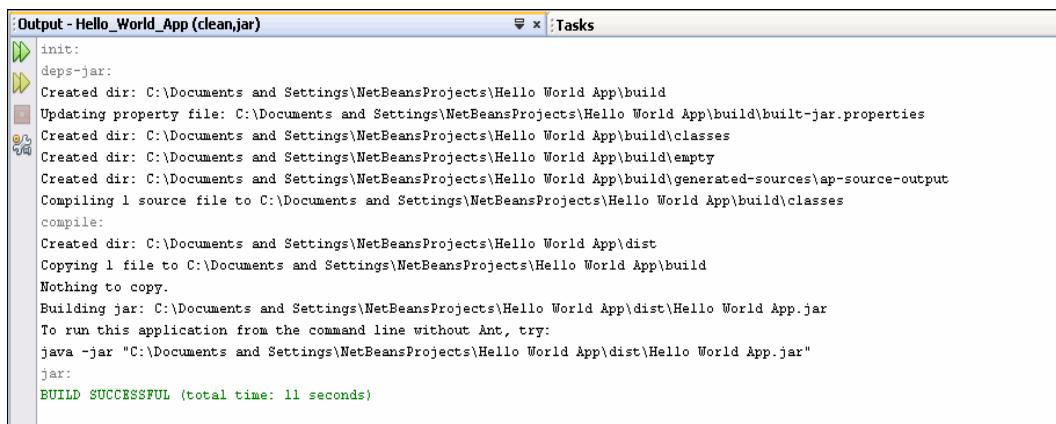
Seperti contoh :

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package helloworldapp;  
  
/**  
 * The HelloWorldApp class implements an application that  
 * simply prints "Hello World!" to standard output.  
 */  
public class HelloWorldApp {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
  
        // TODO code application logic here  
        System.out.println("Hello World!");  
    }  
}
```

Pemrograman Java merupakan salah satu bahasa **pemrograman yg Case Sensitive** maka dari itu perhatikan huruf besar dan kecil huruf seperti contoh a (a kecil) dan A (A Besar), untuk memberikan komentar pada listing code dengan menggunakan tanda // atau untuk mengawali komentar /* dan untuk mengakhiri komentar */ untuk menuliskan String, anda dapat memberikan tanda petik dua pada awal kaliamat dan akhir kalimat seperti contoh “Hello World!”.

Setelah itu lakukan compile program dengan cara :

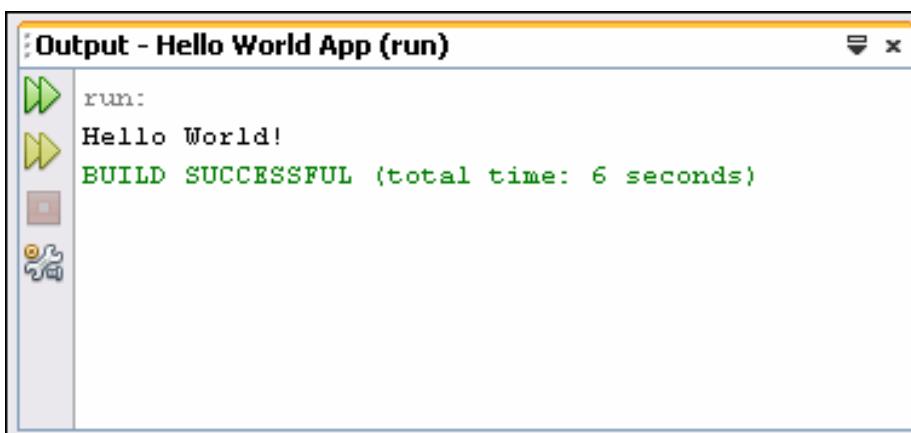
Untuk mengkompilasi file sumber Anda, pilih **Run | Build Main Project** dari menu utama IDE atau menggunakan shortcut Compile File (**F9**).Jendela Output terbuka dan menampilkan hasil yang serupa dengan apa yang Anda lihat pada **gambar 1.5** berikut:



```
:Output - Hello_World_App (clean,jar) Tasks
init:
deps-jar:
Created dir: C:\Documents and Settings\NetBeansProjects\Hello World App\build
Updating property file: C:\Documents and Settings\NetBeansProjects\Hello World App\build\built-jar.properties
Created dir: C:\Documents and Settings\NetBeansProjects\Hello World App\build\classes
Created dir: C:\Documents and Settings\NetBeansProjects\Hello World App\build\empty
Created dir: C:\Documents and Settings\NetBeansProjects\Hello World App\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Documents and Settings\NetBeansProjects\Hello World App\build\classes
compile:
Created dir: C:\Documents and Settings\NetBeansProjects\Hello World App\dist
Copying 1 file to C:\Documents and Settings\NetBeansProjects\Hello World App\build
Nothing to copy.
Building jar: C:\Documents and Settings\NetBeansProjects\Hello World App\dist\Hello World App.jar
To run this application from the command line without Ant, try:
java -jar "C:\Documents and Settings\NetBeansProjects\Hello World App\dist\Hello World App.jar"
jar:
BUILD SUCCESSFUL (total time: 11 seconds)
```

Gambar 1.5 Running Project

Jika hasil pembangunan diakhiri dengan pernyataan BUILD SUCCESSFUL, selamat! Anda telah berhasil mengkompilasi program! Jika hasil Build diakhiri dengan pernyataan BUILD FAILED, anda mungkin memiliki kesalahan sintaks dalam kode Anda. Kesalahan dilaporkan dalam jendela Output sebagai teks hyperlink. Anda klik dua kali seperti hiper-link untuk menavigasi ke sumber kesalahan. Anda kemudian dapat memperbaiki kesalahan dan sekali lagi pilih **Run | Build Main Project**. jika proses compile sukses maka langkah selanjutnya adalah RUN project anda dengan cara pilih **Run | Run Main Project** atau menggunakan shortcut Run File (**Shift-F6**). Setelah itu anda akan melihat tampilan Hello World seperti pada **gambar 1.6** berikut :



```
:Output - Hello World App (run) Tasks
run:
Hello World!
BUILD SUCCESSFUL (total time: 6 seconds)
```

Gambar 1.6 Build Project Sukses

Percobaan 1 : Menampilkan suatu tulisan ke layar

```
Hallo.java

public class Hallo {
    public static void main(String args[]) {
        System.out.println("Hallo...");
    }
}
```

Percobaan 2 : Melibatkan suatu class dalam program

```
TestGreeting.java
public class TestGreeting {
    public static void main (String[] args) {
        Greeting hello = new Greeting();
        hello.greet();
    }
}

Greeting.java
public class Greeting {
    public void greet() {
        System.out.println("hi");
    }
}
```

Latihan

Latihan 1 : Menganalisa dan membenahi kesalahan pada program

Tulislah program berikut ini dan simpanlah dengan nama tertentu.Test.java

```
Greeting.java
public class Testing {
    public static void main(String[] args) {
        System.out.println("What's wrong with this program?");
    }
}
```

Lakukan kompilasi pada file tersebut dan amati hasilnya. Kenapa terjadi kegagalan pada saat kompilasi?. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

Latihan 2 : Menganalisa dan membenahi kesalahan pada program

Tulislah program dibawah ini dan simpanlah dengan nama tertentu. Lakukan kompilasi pada file tersebut dan amati hasilnya. Kenapa terjadi kegagalan pada saat kompilasi?. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("What's wrong with this program?");  
    }  
}
```

Latihan 3 : Menganalisa dan membenahi kesalahan pada program

```
public class TestAnother {  
    public static void main(String[] args) {  
        System.out.println("What's wrong with this program?");  
    }  
}
```

Tulislah program berikut ini dan simpanlah.

```
public class Test {  
  
    public static void main(String args) {  
  
        System.out.println("What's wrong with this program?");  
  
    }  
}
```

Lakukan kompilasi pada program tersebut dan jalankan. Kenapa terjadi kesalahan pada saat menjalankan program tersebut. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.

Latihan 4 : Menganalisa dan membenahi kesalahan pada program

Tulislah program berikut ini dan simpan.

```
public class Test {  
    public void main(String args[]) {  
        System.out.println("What's wrong with this program?");  
    }  
}
```

Lakukan kompilasi pada program tersebut dan jalankan. Kenapa terjadi kesalahan pada saat menjalankan program tersebut? Benahilah kesalahan di atas sehingga program tersebut dapat berjalan dengan baik.

TUGAS

1. Ganti sintaks “println” dengan “print”, perhatikan output yang dihasilkan, analisa hasilnya.
2. Buatlah sebuah program untuk menampilkan string yang berisikan biodata diri masing
Nama, nim, alamat, no telp

MODUL 2
Pengenalan Variable, Tipe data, Operator
dan metode input di JAVA
(Pertemuan 2)

Tujuan :

1. Mengetahui tipe data di java dan cara menggunakannya
2. Mengetahui cara mendeklarasikan variable di java dan cara menggunakannya
3. Mengetahui dan memahami conditional statement di java juga mampu untuk menggunakannya
4. Mengetahui cara kerja reader input dijava

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang variable di java dan apa kegunaannya ?
2. Apa yang anda ketahui tentang metode input di Java dan kegunaanya ?

DASAR TEORI

1. Variabel Java

Variabel merupakan tempat atau wadah untuk menyimpan nilai / value pada bahasa pemrograman. Pada Pemrograman java, semua variabel harus dideklarasikan sebelum mereka dapat digunakan. Bentuk dasar dari sebuah deklarasi variabel yang ditampilkan di sini:

```
type identifier [ = value]
```

Jenis ini merupakan salah satu tipe data Java. Identifier adalah nama variabel. Menyatakan lebih dari satu variabel dari jenis tertentu, menggunakan daftar dipisahkan koma. Berikut adalah beberapa contoh deklarasi variabel dari berbagai jenis. Perhatikan bahwa beberapa mencakup inisialisasi.

```
int a, b, c; // deklarasi 3 variabel a,b,c bertipe integer.  
long d = 3, e, f = 5; // deklarasi 3 variabel d,e,f bertipe long  
byte z = 22; // deklarasi dan inisialisasi variable z.  
char x = 'x'; // variable x yang diberikan value / nilai x.  
String nama = "adi"; // variable nama yang diberikan value / nilai  
adhi.
```

2. Metode Input Java

Pada pemrograman java tentunya memiliki metode input yang digunakan sebagai tempat menangkap value atau nilai pada variabel sehingga mampu menjadikan project anda bersifat dinamis, dalam pemrograman java memiliki banyak metode input namun diantara lain sebagai berikut :

- BufferedReader

- JoptionPane
- Scanner

3. Operator Java

Java menyediakan banyak set operator untuk memanipulasi variabel. Kita dapat membagi semua operator Java ke dalam kelompok berikut:

- Arithmetic Operators

Operator	Hasil
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus
++	Increment
--	Decrement
+=	Persamaan penjumlahan
-=	Persamaan pengurangan

- Relational Operators

Operator	Penggunaan	Keterangan
>	op1 > op2	op1 lebih besar dari op2
>=	op1 >= op2	op1 lebih besar dari atau sama dengan op2
<	op1 < op2	op1 kurang dari op2
<=	op1 <= op2	op1 kurang dari atau sama dengan op2
==	op1 == op2	op1 sama dengan op2
!=	op1 != op2	op1 tidak sama dengan op2

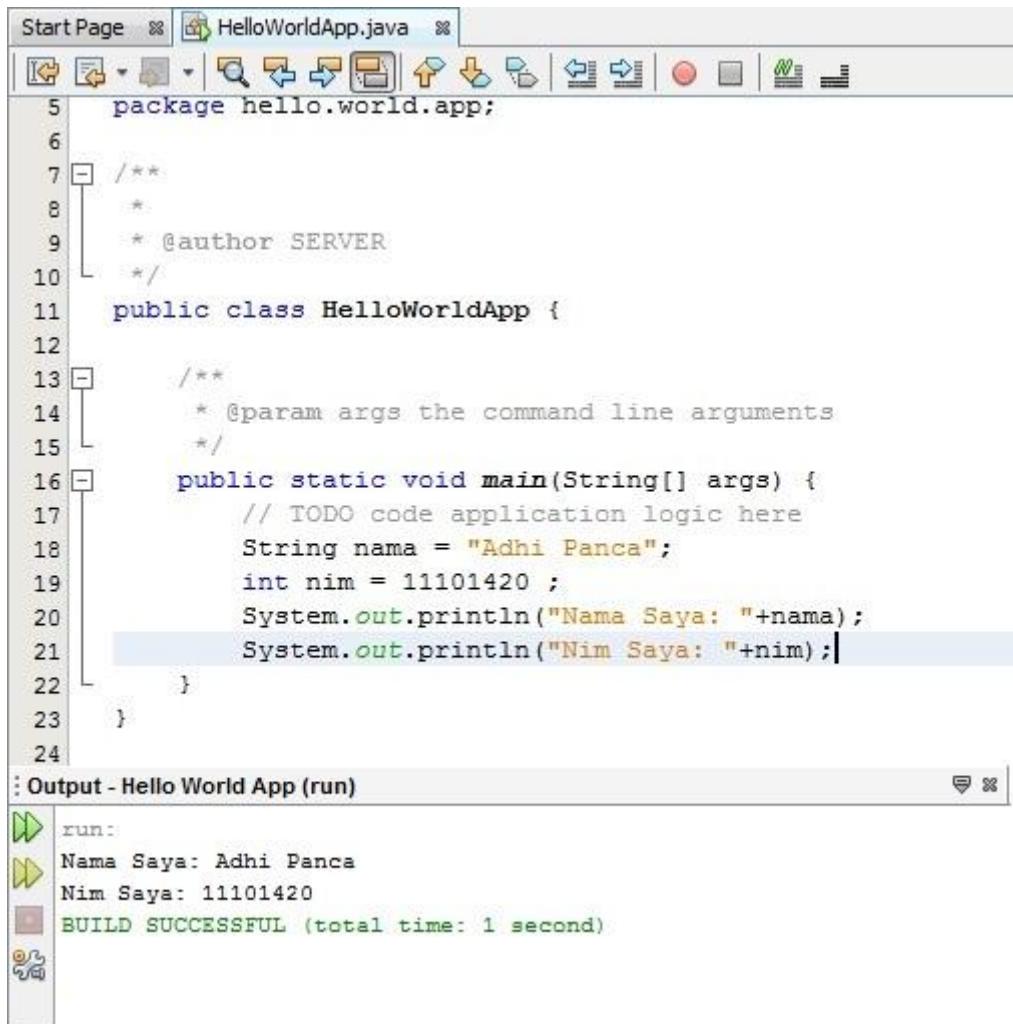
- Logical Operators

Operator	Keterangan
&&	Operasi AND
 	Operasi OR
^	Operasi XOR (exclusive OR)
!	Operasi NOT (negasi)

KEGIATAN PRAKTIKUM

1. variabel

Berikut merupakan contoh project dari variabel yang sudah dijelaskan.



The screenshot shows a Java IDE interface. At the top, there's a toolbar with various icons. Below it is a code editor window containing the following Java code:

```
5 package hello.world.app;
6
7 /**
8 * @author SERVER
9 */
10 public class HelloWorldApp {
11
12     /**
13      * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         // TODO code application logic here
17         String nama = "Adhi Panca";
18         int nim = 11101420 ;
19         System.out.println("Nama Saya: "+nama);
20         System.out.println("Nim Saya: "+nim);
21     }
22 }
23
24
```

Below the code editor is an output window titled ': Output - Hello World App (run)'. It displays the following text:

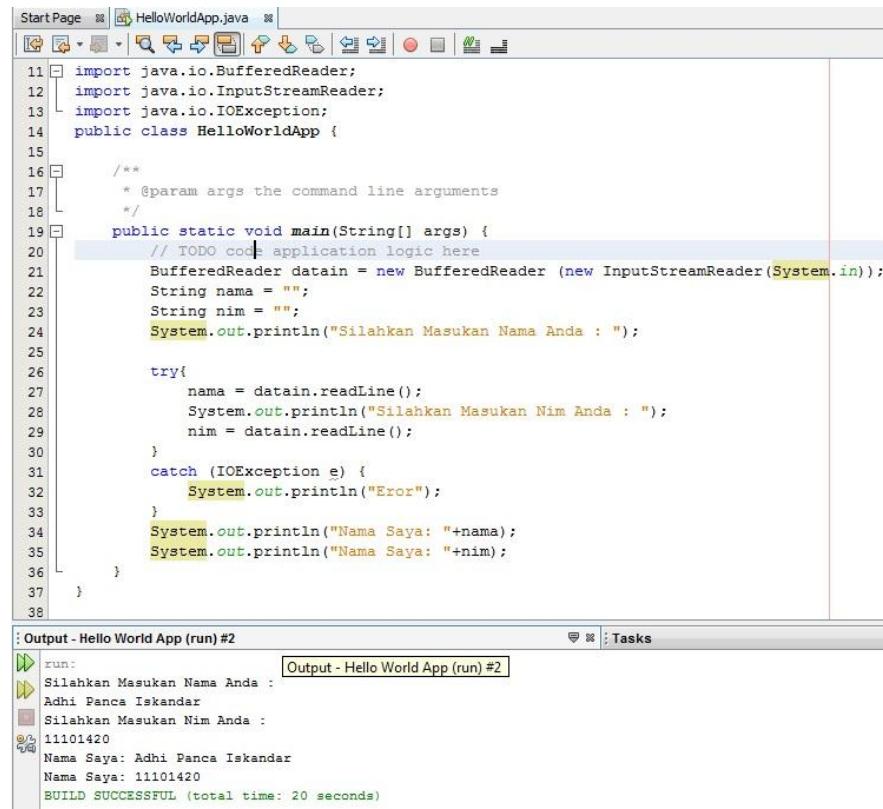
```
run:
Nama Saya: Adhi Panca
Nim Saya: 11101420
BUILD SUCCESSFUL (total time: 1 second)
```

2. Metode Input BufferedReader

Pada bufferedReader memerlukan file bawaan java dimana kita harus mengimport file–file:

- java.io.BufferedReader;
- java.io.InputStreamReader;
- java.io.IOException;

di luar main class atau pada baris 1,2,dan 3. Dan pada BufferedReader memiliki variabel dengan tipe Buffer yaitu Try dan catch yang berfungsi untuk menangkap value / nilai dari variabel utama. Seperti pada contoh berikut:



```

11 import java.io.BufferedReader;
12 import java.io.InputStreamReader;
13 import java.io.IOException;
14 public class HelloWorldApp {
15
16     /**
17      * @param args the command line arguments
18     */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         BufferedReader datain = new BufferedReader (new InputStreamReader(System.in));
22         String nama = "";
23         String nim = "";
24         System.out.println("Silahkan Masukan Nama Anda : ");
25
26         try{
27             nama = datain.readLine();
28             System.out.println("Silahkan Masukan Nim Anda : ");
29             nim = datain.readLine();
30         }
31         catch (IOException e) {
32             System.out.println("Error");
33         }
34         System.out.println("Nama Saya: "+nama);
35         System.out.println("Nama Saya: "+nim);
36     }
37 }
38

```

: Output - Hello World App (run) #2

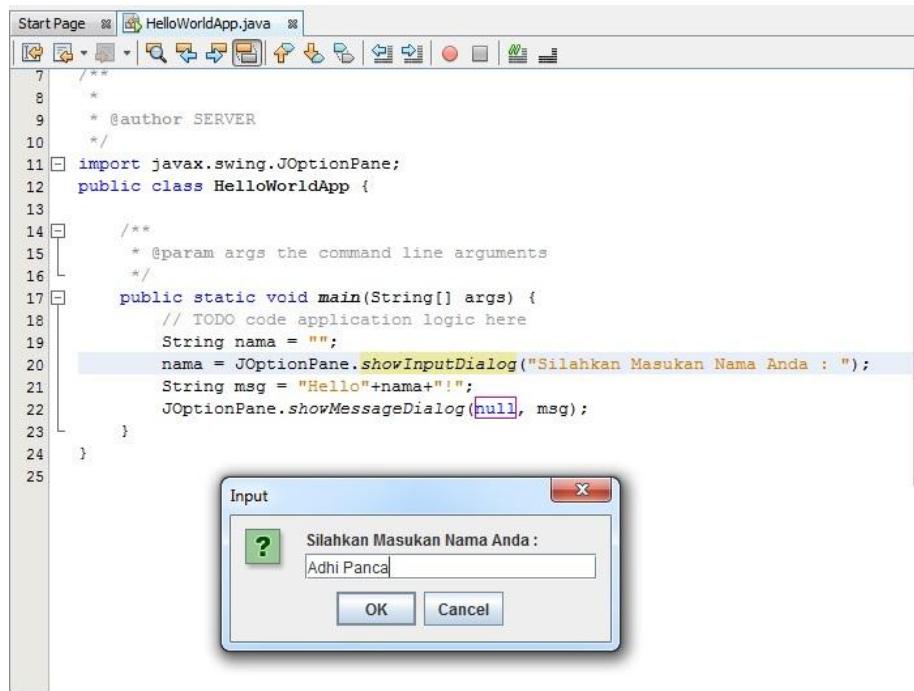
```

run: [Output - Hello World App (run) #2]
Silahkan Masukan Nama Anda :
Adhi Panca Iskandar
Silahkan Masukan Nim Anda :
11101420
Nama Saya: Adhi Panca Iskandar
Nama Saya: 11101420
BUILD SUCCESSFUL (total time: 20 seconds)

```

3. Metode Input JoptionPane

Pada JoptionPane memerlukan file bawaan java dimana kita harus mengimport file javax.swing.JOptionPane; tersebut diluar main class atau pada baris pertama. Seperti pada contoh berikut :



```

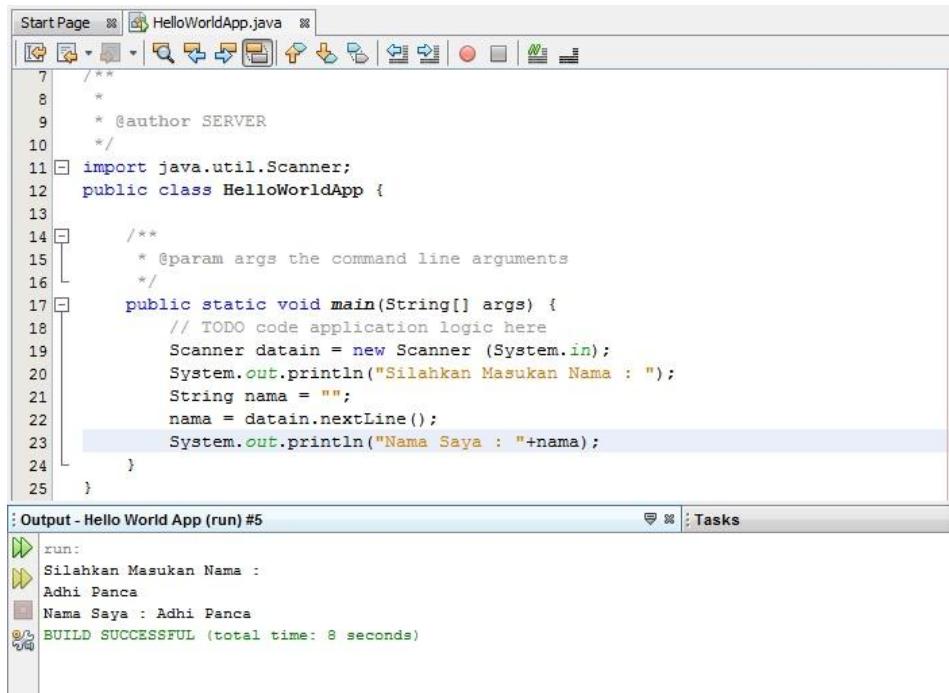
Start Page  HelloWorldApp.java
7  /**
8   *
9   * @author SERVER
10  */
11 import javax.swing.JOptionPane;
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         String nama = "";
20         nama = JOptionPane.showInputDialog("Silahkan Masukan Nama Anda : ");
21         String msg = "Hello"+nama+"!";
22         JOptionPane.showMessageDialog(null, msg);
23     }
24 }
25

```

The screenshot shows a Java code editor with the file name 'HelloWorldApp.java'. The code uses `JOptionPane.showInputDialog` to prompt the user for a name. A modal dialog box titled 'Input' appears, asking 'Silahkan Masukan Nama Anda :'. The user has typed 'Adhi Panca' into the input field. Below the dialog, the code continues with `JOptionPane.showMessageDialog` to display the greeting message.

4. Metode Input Scanner

Pada Scanner memerlukan file bawaan java dimana kita harus mengimport file `java.util.Scanner`; tersebut diluar main class atau pada baris pertama. Seperti pada contoh berikut :



```

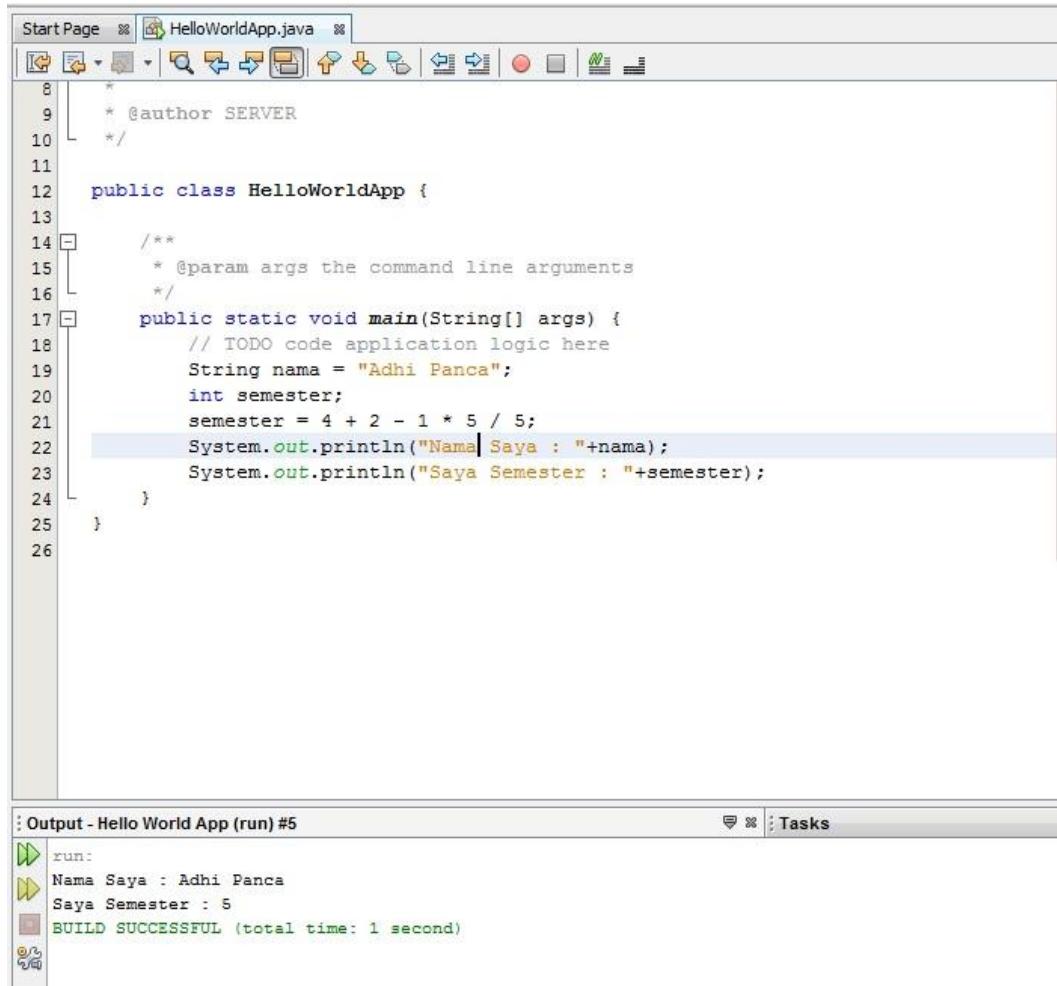
Start Page  HelloWorldApp.java
7  /**
8   *
9   * @author SERVER
10  */
11 import java.util.Scanner;
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         Scanner datain = new Scanner (System.in);
20         System.out.println("Silahkan Masukan Nama : ");
21         String nama = "";
22         nama = datain.nextLine();
23         System.out.println("Nama Saya : "+nama);
24     }
25 }

```

The screenshot shows a Java code editor with the file name 'HelloWorldApp.java'. The code uses `Scanner` to read input from the console. In the 'Output - Hello World App (run) #5' panel, the program prompts the user for a name ('Silahkan Masukan Nama :'), receives the input 'Adhi Panca', and then prints the response ('Nama Saya : Adhi Panca'). At the bottom, it shows a successful build message ('BUILD SUCCESSFUL (total time: 8 seconds)').

5. Operator Pada Java

Pada contoh Operator Java kita gunakan operator aritmatika, karena operator relational dan operator logika biasa digunakan pada sebuah kondisi atau perulangan, maka untuk contoh operator tersebut akan di cantumkan pada penjelasan percabangan dan perulangan.



```
Start Page  ❁ HelloWorldApp.java  ❁
[File] [Edit] [View] [Search] [Run] [Build] [Project] [Help]
8   *
9   * @author SERVER
10  */
11
12 public class HelloWorldApp {
13
14 	/*
15 	* @param args the command line arguments
16 	*/
17 	public static void main(String[] args) {
18 		// TODO code application logic here
19 		String nama = "Adhi Panca";
20 		int semester;
21 		semester = 4 + 2 - 1 * 5 / 5;
22 		System.out.println("Nama| Saya : "+nama);
23 		System.out.println("Saya Semester : "+semester);
24 	}
25 }
26

: Output - Hello World App (run) #5
run:
Nama Saya : Adhi Panca
Saya Semester : 5
BUILD SUCCESSFUL (total time: 1 second)

```

Percobaan

Percobaan 1 : Melakukan increment dan decrement nilai

```
class IncDec {
    public static void main (String args[]) {
        int x = 8, y = 13;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x = " + ++x);
        System.out.println("y = " + y++);
        System.out.println("x = " + x--);
```

```
        System.out.println("y = " + --y);  
    }  
}
```

Percobaan 2 : Melakukan operasi bit

```
class Bitwise {  
    public static void main (String args[]) {  
        int x = 5, y = 6;  
        System.out.println("x = " + x);  
        System.out.println("y = " + y);  
        System.out.println("x & y = " + (x & y));  
        System.out.println("x | y = " + (x | y));  
        System.out.println("x ^ y = " + (x ^ y));  
    }  
}
```

Percobaan 3 : Melakukan operasi komplemen

```
class BitwiseComplement {  
    public static void main (String args[]) {  
        int x = 8;  
        System.out.println("x = " + x);  
        int y = ~x;  
        System.out.println("y = " + y);  
    }  
}
```

Percobaan 4 : Melakukan operasi shift

```
class Shift {  
    public static void main (String args[]) {  
        int x = 7;  
        System.out.println("x = " + x);  
        System.out.println("x >> 2 = " + (x >> 2));  
        System.out.println("x << 1 = " + (x << 1));  
        System.out.println("x >>> 1 = " + (x >>> 1));  
    }  
}
```

Percobaan 5 : Menggunakan logical operator

```
class LogicalOperator {  
    public static void main (String args[]) {  
        int x = 7, y = 11, z = 11;  
        System.out.println("x = " + x);  
        System.out.println("y = " + y);  
        System.out.println("z = " + z);  
        System.out.println("x < y = " + (x < y));  
        System.out.println("x > z = " + (x > z));  
        System.out.println("y <= z = " + (y <= z));  
        System.out.println("x >= y = " + (x >= y));  
    }  
}
```

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

Percobaan 6 : Menggunakan operator boolean and

```
public class BooleanAnd {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a<2) & (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 7 : Menggunakan operator boolean and short-circuit

```
public class ShortCircuitBooleanAnd {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a<2) && (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 8 : Menggunakan boolean or

```
public class BooleanOr {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a>2) | (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 9 : Menggunakan boolean or short-circuit

```
public class ShortCircuitBooleanOr {
    public static void main(String args[]) {
        int a=5, b=7;
        if ((a>2) || (b++<10)) b+=2;
        System.out.println(b);
    }
}
```

Percobaan 10 : Menggunakan operator kondisi

```
class Conditional {
    public static void main (String args[]) {
        int x = 0;
        boolean isEven = false;
        System.out.println("x = " + x);
```

```
x = isEven ? 4 : 7;  
System.out.println("x = " + x);  
}  
}
```

TUGAS

1. Buatlah sebuah program untuk menghitung luas segitiga dengan metode BufferedReader
2. Buatlah sebuah program untuk menampilkan bilangan genap dan ganjil dengan menggunakan metode Scanner.

MODUL 3
Perulangan, Percabangan dan Array Pada Java
(Pertemuan 3)

Tujuan :

1. Mahasiswa mengetahui dan memahami perulangan dan percabangan di java
2. Mahasiswa mengetahui dan memahami array satu dimensi dan multi dimensi
3. Mahasiswa memahami cara mengakses array

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang perulangan dan percabangan di Java
2. Sebutkan kegunaan dari perulangan dan percabangan di Java

DASAR TEORI

1.1. Percabangan

1.1.1 Percabangan if

Pernyataan if merupakan salah satu bentuk pernyataan yang berguna untuk mengambil keputusan terhadap sebuah kemungkinan. Bentuk pernyataan if berupa

```
if(kondisi){  
    // yang akan dijalankan  
}
```

1.1.2. Percabangan if-else

Percabangan if-else merupakan percabangan yang sama dengan percabangan if namun memiliki kondisi false, artinya jika kondisi pada if tidak terpenuhi maka perintah pada else akan dijalankan. Bentuk pernyataan if-else berupa:

```
if(kondisi){  
    // jalankan jika kondisi true  
}else{  
    // jalankan jika kondisi false  
}
```

1.1.3. Percabangan if bersarang

Percabangan if bersarang merupakan gabungan beberapa if dan dapat pula digabung

dengan if–else. Bentuk pernyataan if bersarang adalah sebagai berikut:

```
if(kondisi1){  
    // perintah kondisi1  
}else if(kondisi2){  
    // perintah kondisi2  
}else if(kondisi3){  
    // perintah kondisi3  
}else{  
    // perintah jika semua kondisi tidak ada yang  
    benar}
```

1.1.4. Percabangan switch-case

Percabangan switch–case merupakan percabangan yang kondisinya hanya dapat menggunakan perbandingan `= =` (sama dengan). Bentuk pernyataan percabangan switch–case adalah sebagai berikut :

```
switch(variabel){ case nilai1:  
    // jalankan instruksi break; // hentikan  
case nilai2:  
    // jalankan instruksi break; // hentikan  
case nilai2:  
    // jalankan instruksi break; // hentikan  
case nilai4:  
    // jalankan instruksi break; // hentikan  
default:  
    // jalankan instruksi break; // hentikan}
```

Pada percabangan switch pertama, diperlukan sebuah variabel, setelah itu pada bagian case dibandingkan, jika sama, maka instruksi akan dijalankan sampai menemui tanda break.

1.2. Perulangan

1.2.1. Perulangan while

Pernyataan while berguna untuk melakukan proses perulangan untuk sebuah kondisi, selama kondisi tersebut bernilai benar (true), maka perulangan akan terus berjalan, dan terhenti ketika kondisi bernilai salah (false). Bentuk pernyataan while

seperti berikut ini :

```
while(kondisi){  
    // isi instruksi  
}
```

1.2.2. Perulangan do-while

Perulangan do-while merupakan perulangan yang hampir mirip dengan perulangan while namun perbedaannya, pada perulangan do-while, maka minimal instruksi akan dijalankan sekali. Bentuk pernyataan do-while sebagai berikut :

```
do{  
    // instruksi  
}while(kondisi);
```

1.2.3. Perulangan for

Perulangan for merupakan perulangan yang memiliki variabel untuk melakukan pengkondisian, berbeda dengan while dan do-while yang kita harus membuat sebuah variabel diluar untuk melakukan penkondisian, pada perulangan for, ditempatkan sebuah blok untuk membuat variabel dan melakukan proses pengkondisian. Bentuk pernyataan for seperti berikut :

```
for(inisialisasi;kondisi;penaikan/penurunan){  
  
    instruksi  
  
}
```

1.2.4. Perintah break

Perintah break merupakan perintah yang dapat digunakan untuk menghentikan proses perulangan.

1.2.5. Perintah continue

Perintah continue dapat digunakan untuk meloncati sebuah perulangan, maksudnya adalah instruksi yang seharusnya dapat dilewat, hal ini berarti instruksi tidak akan dijalankan.

1.3.1. Array

Suatu array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama,

dimana masing-masing elemen variabel mempunyai nilai indeks. Setiap elemen array mampu untuk menyimpan satu jenis data (variabel).

Array merupakan sekumpulan obyek yang memiliki tipe data yang sama dan dapat di akses secara random dengan menggunakan index. Array mempunyai panjang yang tetap, artinya ketika kita mendeklarasikan suatu array dengan panjang 10, maka array tersebut panjangnya akan tetap 10 walaupun kita hanya memakai 5 elemen.

1.3.2. Deklarasi Array satu dimensi

Untuk mendeklarasikan array dapat dilakukan dengan cara berikut ini :

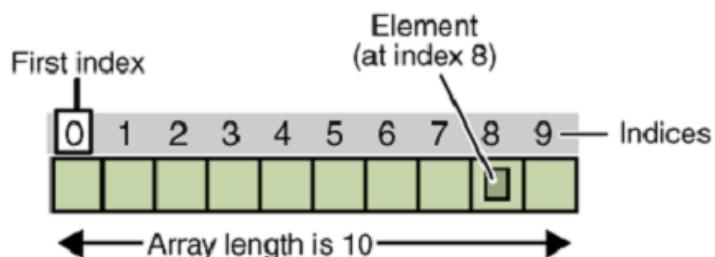
```
int value [];  
value = new int [100]; // menginisialisasi  
// [100] merupakan batas ruang array
```

atau bisa deklarasi sekaligus menginisialisasikan

```
Int value []=new int [100];
```

Cara pendeklarasian tanpa batas

```
int value []={1,2,3}; // {1,2,3} adalah isi dari array
```



Gambar visual suatu array

Untuk mengetahui panjang array yaitu gunakan syntax **nama_array.length** berdasarkan contoh diatas nama array = value maka untuk mengetahui panjang value kita gunakan syntax **value.length**. cara pemanggilan yaitu value[0] akan menampilkan angka 1.

1.3.3. Array Multidimensi

Array tidak hanya terdiri dari 1 dimensi, array juga bisa terdiri dari 2 dimensi, 3 dimensi dan n-dimensi. Array di atas 3 dimensi sangat jarang digunakan karena sangat sulit untuk di gambarkan. Array multidimensi diimplementasikan sebagai array dalam array. Untuk lebih jelas mengenai cara pendeklarasian array multidimensi dan penggunaanya, silakan ketikkan program di bawah ini.

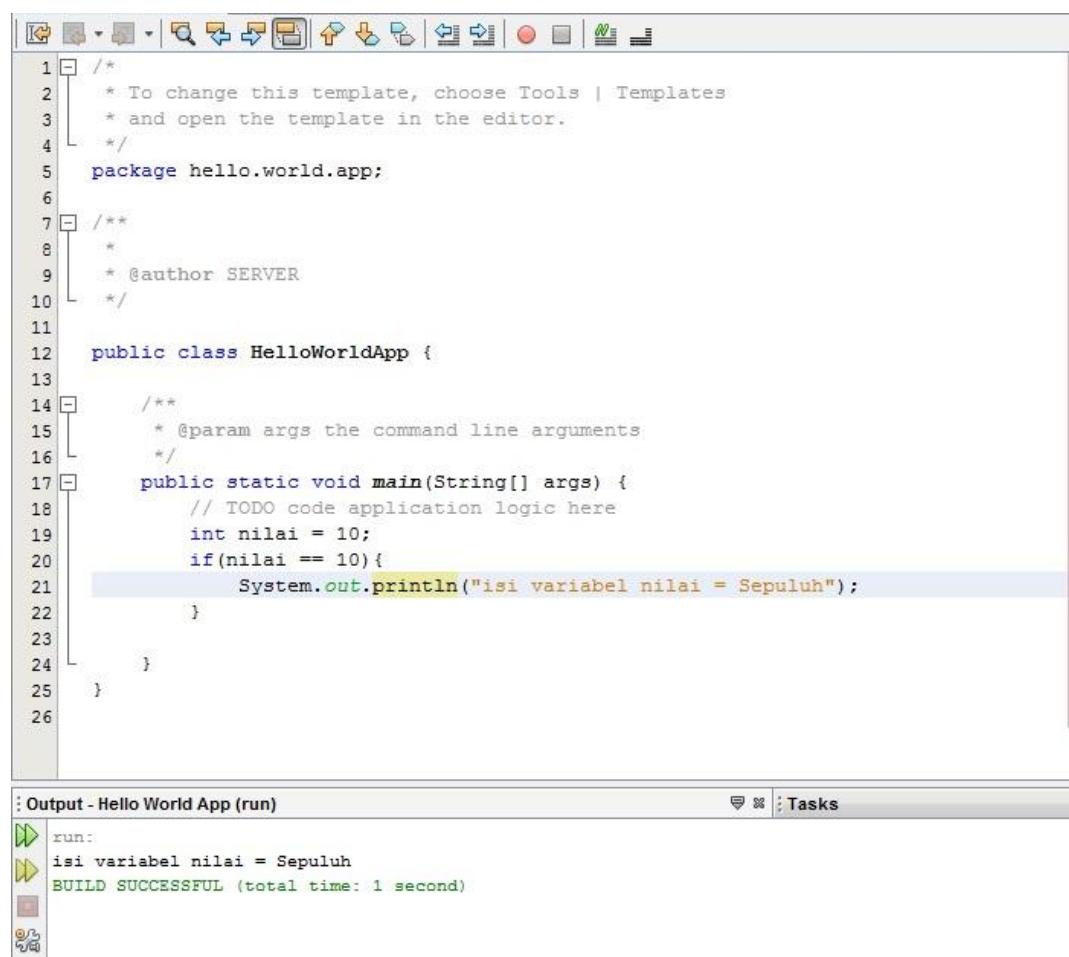
```
Array 2 dimensi:  
char multiChar[ ][ ]=new char[10][5];  
char multiChar2[ ][ ]={{'a','b','c'},{'d','e','f'}}; inisialisasi array 2 dimensi  
Array 3 dimensi:  
int tigaDimensi[ ][ ][ ]=new int[10][10][10];  
int tigaDimensi2[ ][ ][ ]={{{{1,2,3},{4,5,6}},{{1,2,3},{4,5,6}}}}; inisialisasi array 3 dimensi
```

Tambahan:

Array lebih dari 2-dimensi sangat jarang dipakai.

KEGIATAN PRAKTIKUM

2.1. Simple Project If



The screenshot shows an IDE interface with a code editor and a terminal window. The code editor displays a Java file named `HelloWorldApp.java` containing the following code:

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

/**
 *
 * @author SERVER
 */

public class HelloWorldApp {

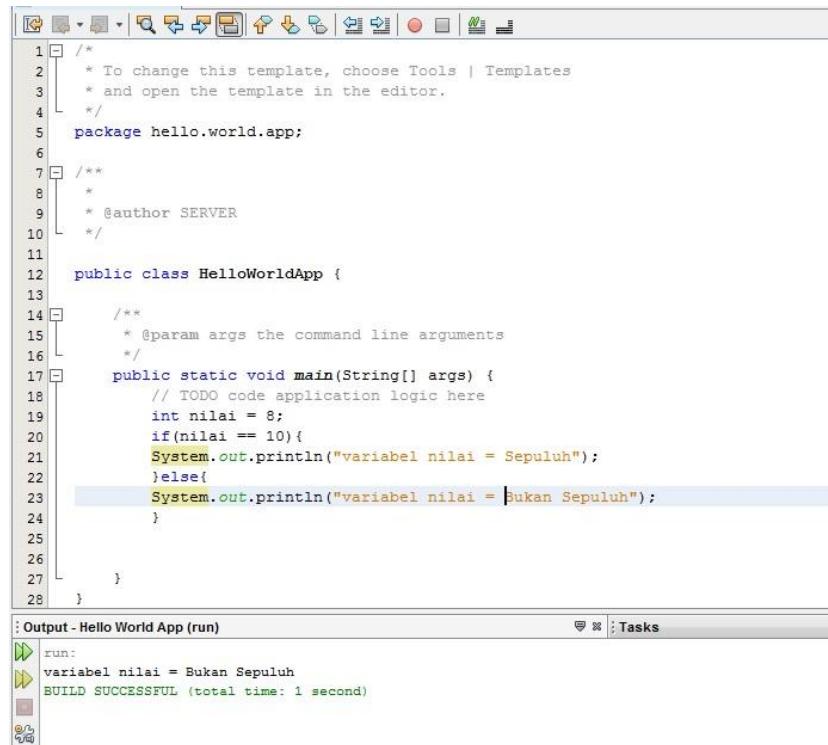
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int nilai = 10;
        if(nilai == 10){
            System.out.println("isi variabel nilai = Sepuluh");
        }
    }
}
```

The line `System.out.println("isi variabel nilai = Sepuluh");` is highlighted in yellow. The terminal window below shows the output of the run command:

```
: Output - Hello World App (run)
run:
isi variabel nilai = Sepuluh
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan “variabel nilai = Sepuluh” karena kondisi pada if bernilai true, jika kondisi bernilai salah, misal nilai == 100, maka program tidak akan menghasilkan tulisan apa-apa.

2.2. Simple Project If else



The screenshot shows a Java code editor with the following code:

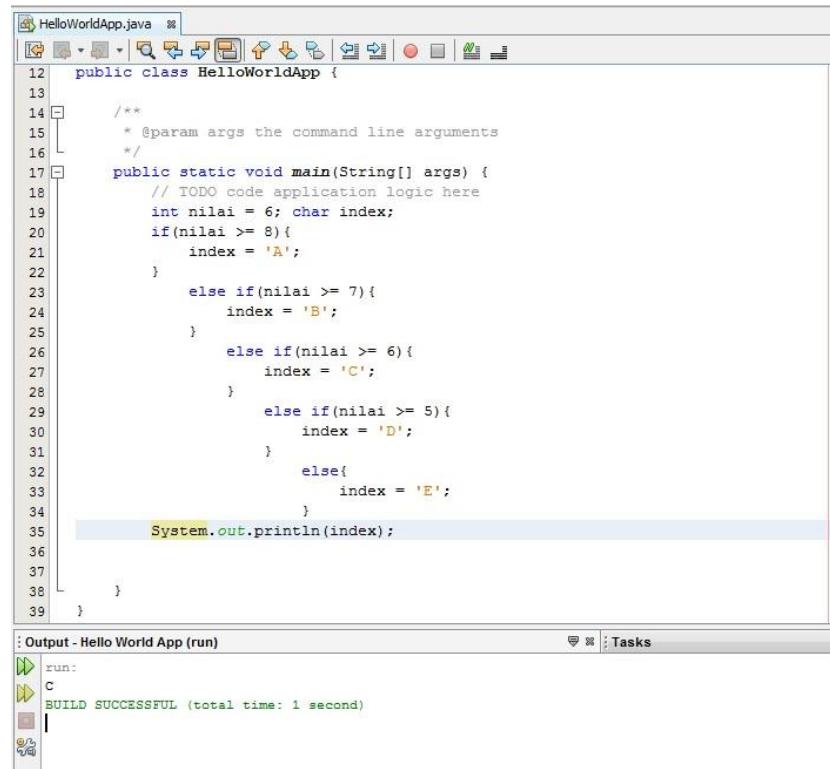
```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package hello.world.app;
6
7  /**
8   *
9   * @author SERVER
10  */
11
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int nilai = 8;
20         if(nilai == 10){
21             System.out.println("variabel nilai = Sepuluh");
22         }else{
23             System.out.println("variabel nilai = Bukan Sepuluh");
24         }
25
26     }
27 }
28 }
```

The code contains an if-else statement where the condition is set to false (nilai == 10). The output window below shows the result of running the application:

```
: Output - Hello World App (run)
run:
variabel nilai = Bukan Sepuluh
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan “variabel nilai = Bukan Sepuluh”, hal ini dikarenakan nilai bernilai 8, bukan 10.

2.3. Simple Project if else if



```

12  public class HelloWorldApp {
13
14      /**
15      * @param args the command line arguments
16      */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int nilai = 6; char index;
20          if(nilai >= 8){
21              index = 'A';
22          }
23          else if(nilai >= 7){
24              index = 'B';
25          }
26          else if(nilai >= 6){
27              index = 'C';
28          }
29          else if(nilai >= 5){
30              index = 'D';
31          }
32          else{
33              index = 'E';
34          }
35          System.out.println(index);
36
37      }
38  }
39

```

: Output - Hello World App (run)

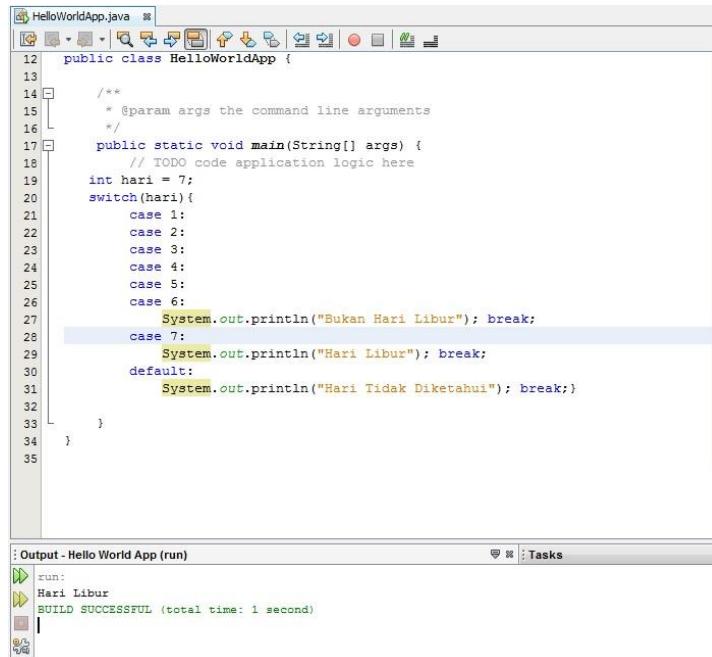
```

run:
C
BUILD SUCCESSFUL (total time: 1 second)

```

Jika program diatas dijalankan, maka hasilnya adalah ‘C’.

2.4. Simple Project Switch – Case



```

12  public class HelloWorldApp {
13
14      /**
15      * @param args the command line arguments
16      */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int hari = 7;
20          switch(hari){
21              case 1:
22              case 2:
23              case 3:
24              case 4:
25              case 5:
26              case 6:
27                  System.out.println("Bukan Hari Libur"); break;
28              case 7:
29                  System.out.println("Hari Libur"); break;
30              default:
31                  System.out.println("Hari Tidak Diketahui"); break;
32          }
33      }
34  }
35

```

: Output - Hello World App (run)

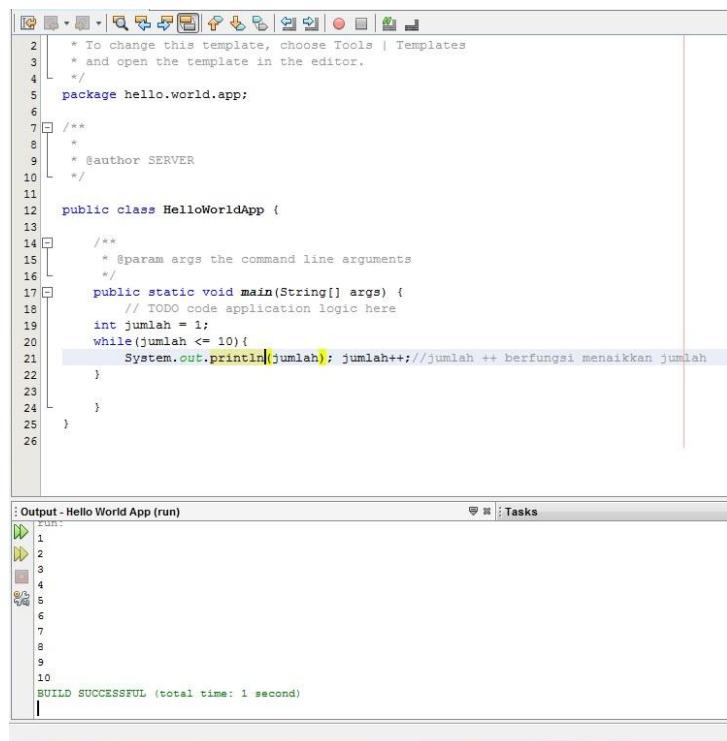
```

run:
Hari Libur
BUILD SUCCESSFUL (total time: 1 second)

```

Jika program diatas dijalankan, maka program akan menghasilkan tulisan “Hari Libur”.

2.5. Simple Project While



```

2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
8  *
9  * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int jumlah = 1;
20         while(jumlah <= 10){
21             System.out.println(jumlah); jumlah++; //jumlah ++ berfungsi menaikkan jumlah
22         }
23     }
24 }
25
26

```

: Output - Hello World App (run)

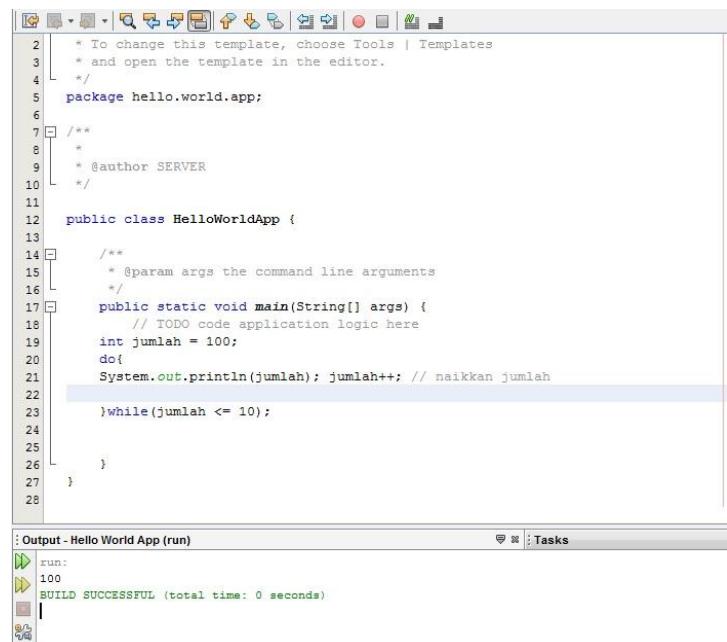
```

run:
1
2
3
4
5
6
7
8
9
10
BUILD SUCCESSFUL (total time: 1 second)

```

Jika program tersebut dijalankan, maka hasilnya adalah tulisan dari 1 sampai 10. Dan saat jumlah bernilai 11, maka perulangan akan terhenti dikarenakan kondisi bernilai false ($11 \leq 10$).

2.6. Simple Project Do – while



```

2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
8  *
9  * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int jumlah = 100;
20         do{
21             System.out.println(jumlah); jumlah++; // naikkan jumlah
22         }while(jumlah <= 10);
23     }
24 }
25
26

```

: Output - Hello World App (run)

```

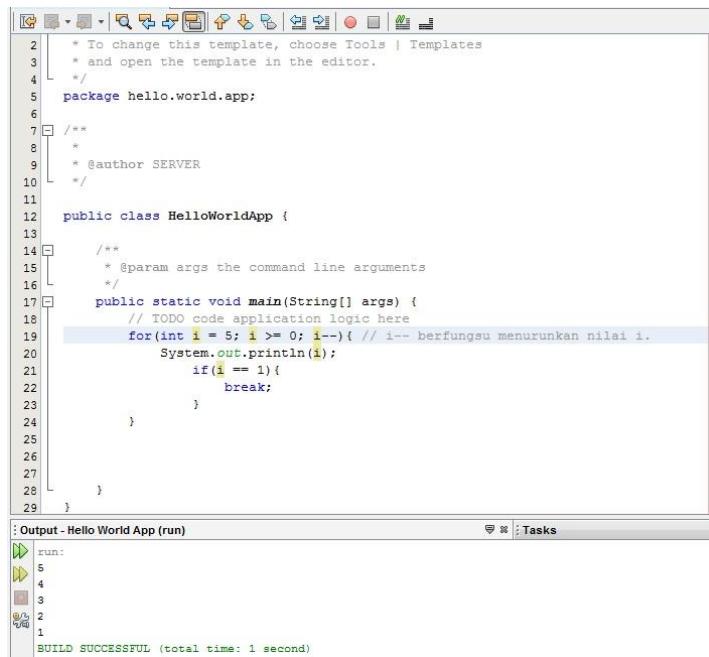
run:
100
BUILD SUCCESSFUL (total time: 0 seconds)

```

Jika program tersebut dijalankan, maka akan menghasilkan keluaran 100, artinya walaupun kondisi salah, namun minimal isi instruksi akan dijalankan sekali, hal ini dikarenakan proses do-while berbeda dengan while, dimana do-while pertama melakukan instruksi baru mengecek

kondisi, sedangkan while pertama mengecek kondisi baru melakukan instruksi.

2.7. Simple Project Break



```

2 * To change this template, choose Tools | Templates
3 * and open the template in the editor.
4 */
5 package hello.world.app;
6
7 /**
8 *
9 * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14 /**
15 * @param args the command line arguments
16 */
17 public static void main(String[] args) {
18     // TODO code application logic here
19     for(int i = 5; i >= 0; i--){ // -- berfungsi menurunkan nilai i.
20         System.out.println(i);
21         if(i == 1){
22             break;
23         }
24     }
25
26
27 }
28
29 }

```

: Output - Hello World App (run)

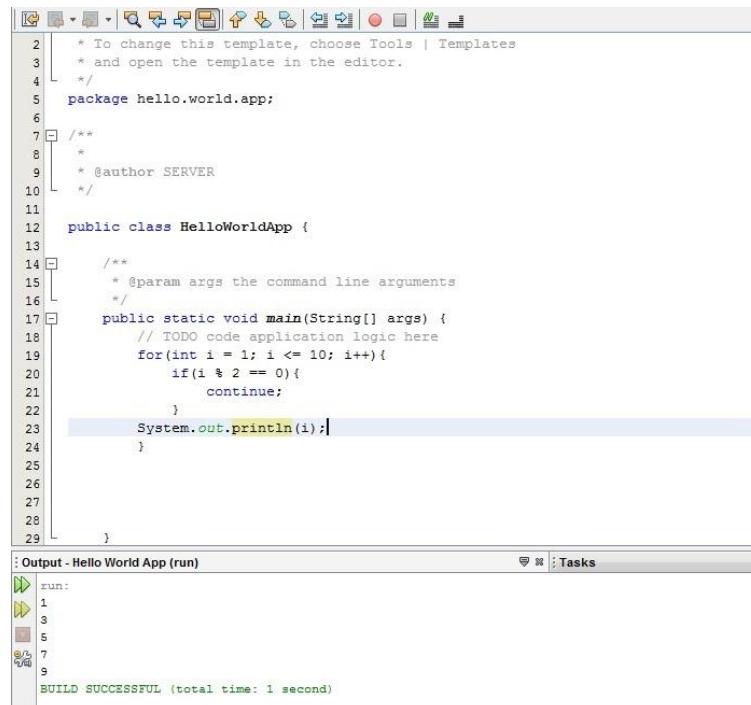
```

run:
5
4
3
2
1
BUILD SUCCESSFUL (total time: 1 second)

```

Maka program hanya akan menampilkan angka dari 5 sampai 1, karena pada saat i mencapai 1, program dihentikan oleh perintah break.

2.8. Simple Project Continue



```

2 * To change this template, choose Tools | Templates
3 * and open the template in the editor.
4 */
5 package hello.world.app;
6
7 /**
8 *
9 * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14 /**
15 * @param args the command line arguments
16 */
17 public static void main(String[] args) {
18     // TODO code application logic here
19     for(int i = 1; i <= 10; i++){
20         if(i % 2 == 0){
21             continue;
22         }
23         System.out.println(i);
24     }
25
26
27 }
28
29 }

```

: Output - Hello World App (run)

```

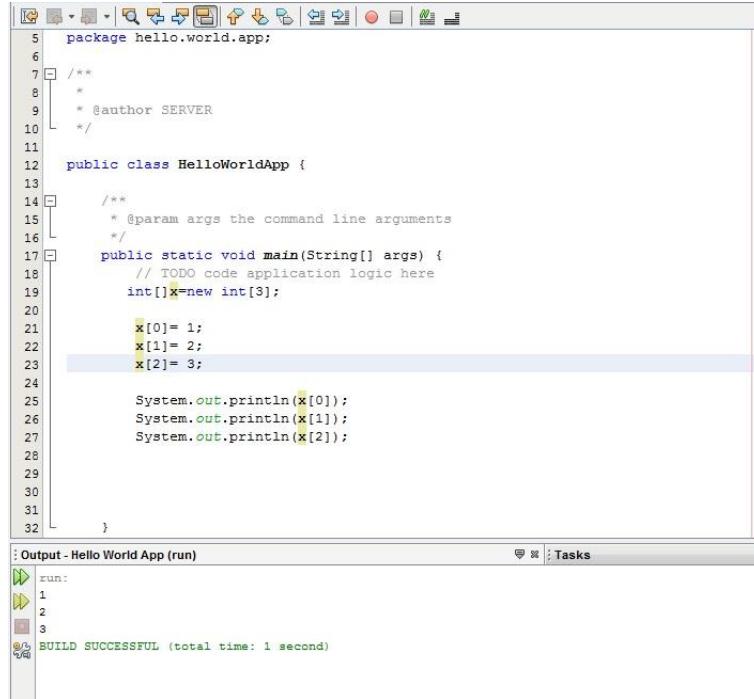
run:
1
3
5
7
9
BUILD SUCCESSFUL (total time: 1 second)

```

Jika program diatas dijalankan, maka hasilnya akan menampilkan angka–angka ganjil saja, hal ini dikarenakan saat nilai i merupakan angka genap, maka perintah continue membuat program

tidak menampilkan angka genap.

2.9. Simple Project Array 1 Dimensi



The screenshot shows a Java code editor with the following code:

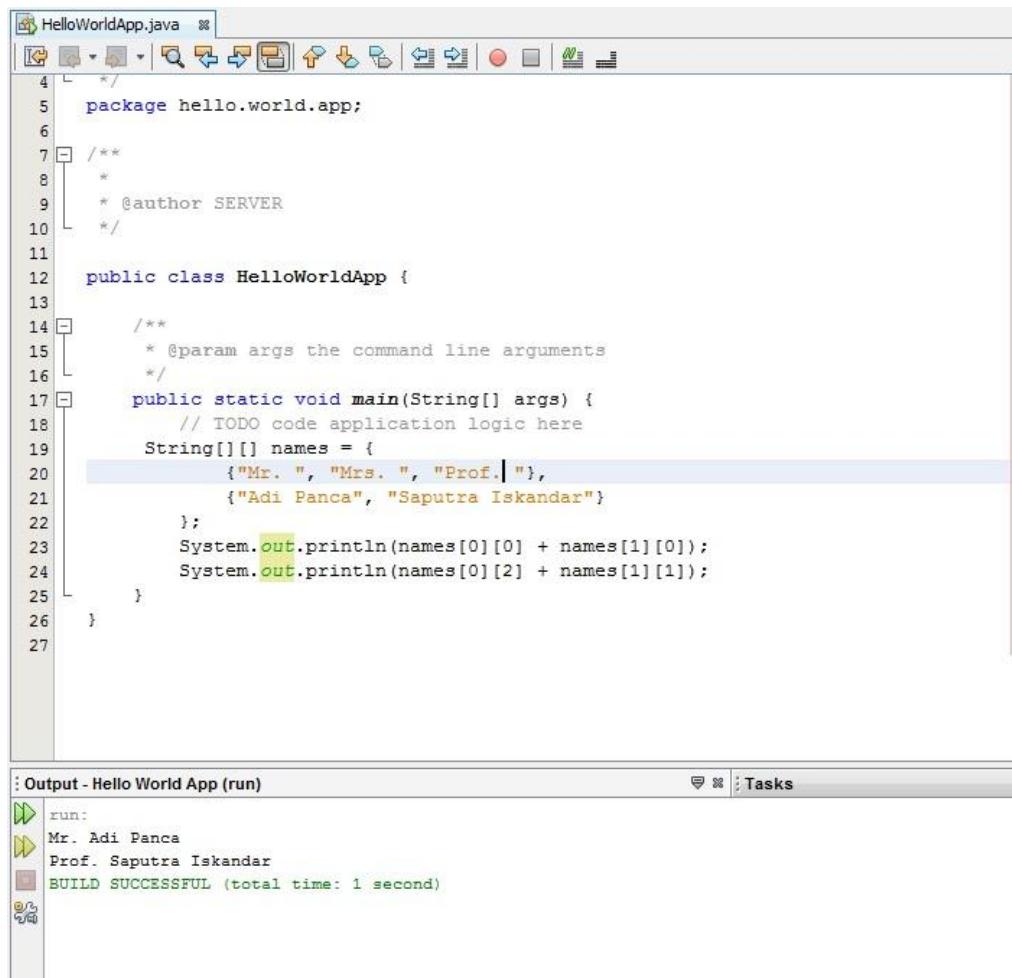
```
5 package hello.world.app;
6
7 /**
8 * @author SERVER
9 */
10
11 public class HelloWorldApp {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18         int[] x=new int[3];
19
20         x[0]= 1;
21         x[1]= 2;
22         x[2]= 3;
23
24         System.out.println(x[0]);
25         System.out.println(x[1]);
26         System.out.println(x[2]);
27
28
29
30
31     }
32 }
```

Below the code editor is an output window titled "Output - Hello World App (run)" showing the following results:

```
: Output - Hello World App (run)
run:
1
2
3
BUILD SUCCESSFUL (total time: 1 second)
```

Program yang ditampilkan adalah 1,2, dan 3 sebab array x telah dideklarasikan 1,2,3.

2.10. Simple Project Array 2 Dimensi / Multidimensi



```

1  /*
2  package hello.world.app;
3
4  /**
5   * @author SERVER
6  */
7
8  public class HelloWorldApp {
9
10    /**
11     * @param args the command line arguments
12     */
13    public static void main(String[] args) {
14        // TODO code application logic here
15        String[][] names = {
16            {"Mr. ", "Mrs. ", "Prof. "},
17            {"Adi Panca", "Saputra Iskandar"}
18        };
19        System.out.println(names[0][0] + names[1][0]);
20        System.out.println(names[0][2] + names[1][1]);
21    }
22 }
23
24
25
26
27

```

: Output - Hello World App (run)

- run:
- Mr. Adi Panca
- Prof. Saputra Iskandar
- BUILD SUCCESSFUL (total time: 1 second)

Program yang ditampilkan adalah Mr. Adi Panca | Prof. Saputra Iskandar sebab pemanggilan pada array names [0] [0] = **Mr** dan names [1][0] = **Adi Panca**. Demikian pun dengan pemanggilan kedua pada Prof. Saputra Iskandar.

Percobaan

Percobaan 1 : Percabangan menggunakan if, if-else dan else-if

```

class IfElseName {
    public static void main (String args[]) {
        char firstInitial = 'a';

        if (firstInitial == 'a')
            System.out.println("Nama anda pasti Asep!");
        else if (firstInitial == 'b')
            System.out.println("Nama anda pasti Brodin!");
        else if (firstInitial == 'c')
            System.out.println("Nama anda pasti Cecep!");
        else
            System.out.println("Nama anda tidak terkenal!");
    }
}

```

Percobaan 2 : Percabangan menggunakan switch

```
class SwitchName {
    public static void main (String args[]) {
        char firstInitial = 'a';

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

Percobaan 3 : Percabangan menggunakan switch dengan break

```
class SwitchNameBreak {
    public static void main (String args[]) {
        char firstInitial = 'a';

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
                break;
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
                break;
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
                break;
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

Percobaan 4 : Perulangan menggunakan for

```
class ForCount {
    public static void main (String args[]) {
        int count=1;
        for (int i=0; i<9; i++) {
            for (int j=0; j<i+1; j++) {
                System.out.print(count);
            }
            count++;
            System.out.println();
        }
    }
}
```

Percobaan 5 : Perulangan menggunakan while

```
class WhileCount {  
    public static void main (String args[]) {  
        int count=1;  
        int i=0;  
        while (i<9) {  
            int j=0;  
            while (j<i+1) {  
                System.out.print(count);  
                j++;  
            }  
            count++;  
            System.out.println();  
            i++;  
        }  
    }  
}
```

Percobaan 6 : Perulangan dengan break

```
class BreakLoop {  
    public static void main (String args[]) {  
        int i = 0;  
        do {  
            System.out.println("Iterasi ke " + i);  
            i++;  
            if (i > 10) break;  
        }  
        while (true);  
    }  
}
```

Percobaan 7 : Perulangan dengan continue

```
public class ContinueLoop {  
    public static void main(String args[]) {  
        int a, b;  
        for(a=0;a<2;a++)  
            for(b=0;b<3;b++) {  
                if (b==1) continue;  
                System.out.println("a=" + a + " ; b=" + b);  
            }  
    }  
}
```

Percobaan 8 : Pemakaian label pada kondisi break

```
public class BreakLabel {  
    public static void main(String args[]) {  
        int a, b;  
        Mulai:  
        for(a=0;a<2;a++)  
            for(b=0;b<3;b++) {  
                if (b==1) break Mulai;  
                System.out.println("a=" + a + " ; b=" + b);  
            }  
    }  
}
```

```
        if (b==1) break Mulai;
        System.out.println("a=" + a + " ; b=" + b);
    }
}
}
```

Percobaan 9 : Pemakaian label pada kondisi continue

```
public class ContinueLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
                if (b==1) continue Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

TUGAS

1. Buatlah program mencari nilai rata rata,nilai terbesar dimana nilai yg diinput oleh user, minimal inputan 10x.
2. Buatlah program mencari bilangan berpangkat, dengan ketentuan user menginput bilangan bulat, kemudian user menginput bilangan pangkat, maka akan muncul hasil dari kedua bilangan tersebut.
3. Buatlah program untuk menentukan suatu tahun kabisat atau bukan dimana tahun dibatasi mulai dari tahun 1900 sampai dengan tahun 2020

Contoh tampilan:

Masukkan tahun (1900-2005) : 1923
1923 bukan tahun kabisat

Masukkan tahun (1900-2005) : 1898
Maaf, tahun input dibawah 1900

Masukkan tahun (1900-2005) : 1996
1996 adalah tahun kabisat

Masukkan tahun (1900-2005) : 2008
Maaf, tahun input diatas 2005

MODUL 4
Class, Object, Elemen dalam Class Pada Java
(Pertemuan 4)

Tujuan :

1. Mahasiswa mampu memahami dan mempraktekkan konsep Class dan Objek di java
2. Mahasiswa mengetahui elemen Class di Java

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang Class dan objek di Java?
2. Apa keunggulan penggunaan konsep Class dan objek dibandingkan konsep pemrograman terstruktural

DASAR TEORI

- Deklarasi class dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> class <nama_class> {  
    [deklarasi_atribut]  
    [deklarasi_konstruktor]  
    [deklarasi_metode]  
}
```

Contoh:

```
public class Siswa {  
    ...  
}
```

- Deklarasi atribut dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <tipe> <nama_atribut> ;
```

Contoh:

```
public class Siswa {  
    public int nrp;  
    public String nama;  
}
```

- Deklarasi metode dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <return_type> <nama_metode> ([daftar_argumen])
```

```
{  
    [<statement>]  
}
```

Contoh:

```
public class Siswa {  
    public int nrp;  
    public String nama;  
    public void info() {  
        System.out.println("Ini siswa PENS");  
    }  
}
```

```
    }
}

• Untuk dapat mengakses anggota-anggota dari suatu obyek, maka harus dibuat instance dari class tersebut terlebih dahulu. Berikut ini adalah contoh pengaksesan anggota-anggota dari class Siswa:  
public class Siswa {  
    public static void main(String args[]) {  
        Siswa it=new Siswa();  
        it.nrp=5;  
        it.nama="Andi";  
        it.info();  
    }  
}
```

1.2. Metode (Method)

Dalam java terdapat dua buah metode

1. Fungsi (Non Void), merupakan metode yang memiliki nilai balik jika metode tersebut dipanggil, cara pembuatan sebuah fungsi adalah dengan cara menentukan nilai baliknya, lalu membuat nama metodenya.

```
class Manusia { String nama;  
  
    // fungsi  
  
    Public String ambilNama() {  
  
        // untuk mengembalikan nilai gunakan kata kunci return  
  
        return nama; }
```

2. Prosedur (Void), merupakan metode yang tidak memiliki nilai balik, cara pembuatan prosedur sama dengan fungsi namun bedanya, nilai baliknya menggunakan kata kunci void.

Contoh :

```
// prosedur  
  
Public void hapusNama() {  
  
    nama = "";  
  
}
```

Pada kode diatas, kelas manusia memiliki 2 buah metode yaitu ambilNama() dan hapusNama(). Dimana ambilNama() merupakan sebuah fungsi karena mengembalikan nilai String, sedangkan hapusNama() merupakan prosedur karena tidak mengembalikan nilai.

Saat membuat sebuah fungsi maka untuk mengembalikan nilainya, harus

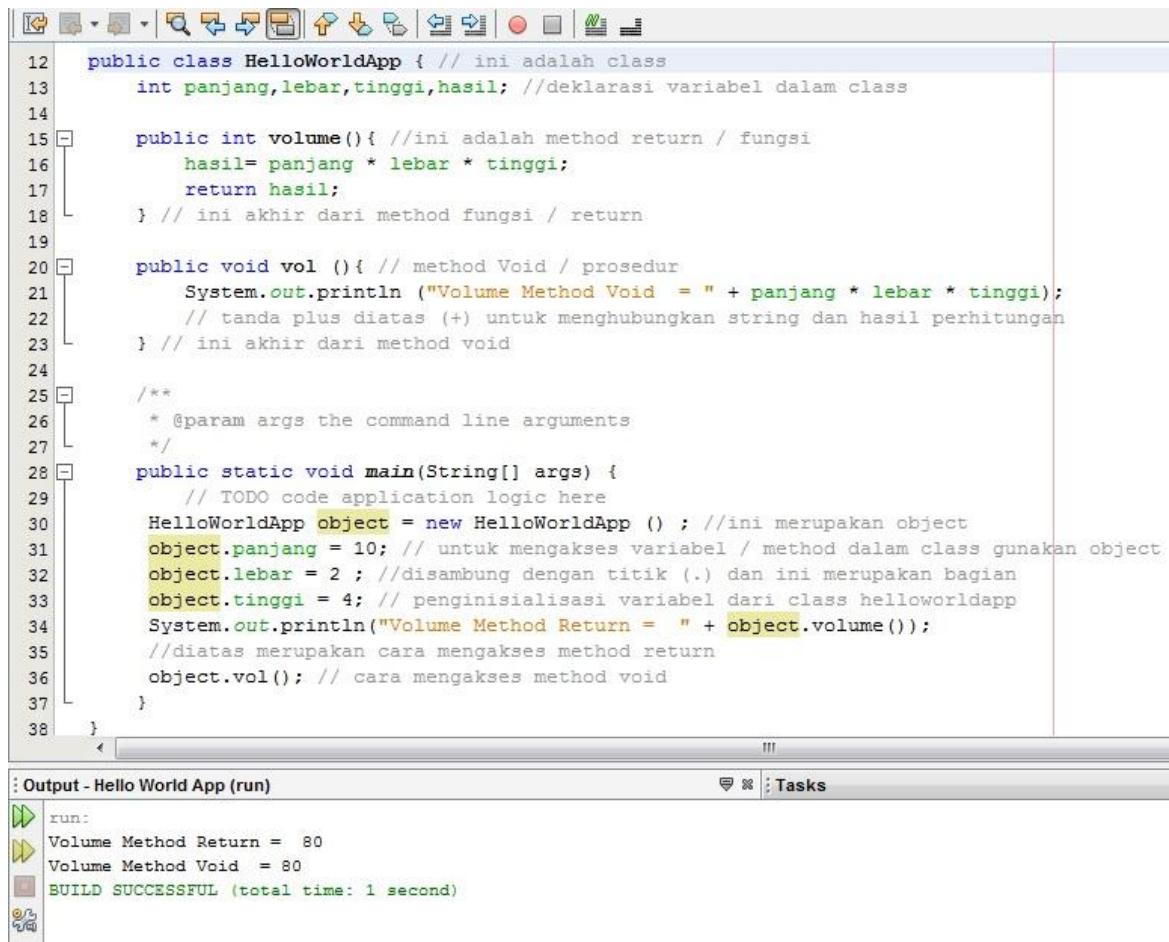
menggunakan kata kunci return, diikuti nilai yang akan dikembalikannya. Untuk mengambil nilai balik dari fungsi dapat dilihat pada contoh sebagai berikut.

```
Manusia manusia = new Manusia(); manusia.nama = "Eko
Kurniawan Khannedy";

// mengambil nilai dari fungsi String nama =
manusia.ambilNama();
```

KEGIATAN PRAKTIKUM

3.1. Simple Project Class, Object dan Method



```
12 public class HelloWorldApp { // ini adalah class
13     int panjang,lebar,tinggi,hasil; //deklarasi variabel dalam class
14
15     public int volume(){ //ini adalah method return / fungsi
16         hasil= panjang * lebar * tinggi;
17         return hasil;
18     } // ini akhir dari method fungsi / return
19
20     public void vol (){ // method Void / prosedur
21         System.out.println ("Volume Method Void = " + panjang * lebar * tinggi);
22         // tanda plus diatas (+) untuk menghubungkan string dan hasil perhitungan
23     } // ini akhir dari method void
24
25     /**
26      * @param args the command line arguments
27      */
28     public static void main(String[] args) {
29         // TODO code application logic here
30         HelloWorldApp object = new HelloWorldApp () ; //ini merupakan object
31         object.panjang = 10; // untuk mengakses variabel / method dalam class gunakan object
32         object.lebar = 2 ; //disambung dengan titik (.) dan ini merupakan bagian
33         object.tinggi = 4; // penginisialisasi variabel dari class helloworldapp
34         System.out.println("Volume Method Return = " + object.volume());
35         //diatas merupakan cara mengakses method return
36         object.vol(); // cara mengakses method void
37     }
38 }
```

: Output - Hello World App (run)

- run:
- Volume Method Return = 80
- Volume Method Void = 80
- BUILD SUCCESSFUL (total time: 1 second)

Percobaan 1 : Mengakses anggota suatu class

Amati program dibawah ini:

```
public class Siswa {
    int nrp;
    public void setNrp(int i) {
        nrp=i;
    }
}
public class Test {
    public static void main(String args[]) {
```

```
Siswa anak=new Siswa();
anak.setNrp(5);
System.out.println(anak.nrp);
}
}
```

Percobaan 2 : Mengakses anggota suatu class

Amati program dibawah ini:

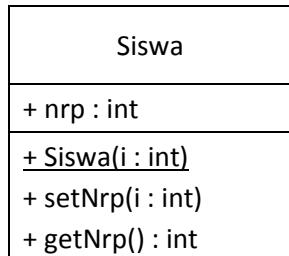
```
public class Siswa {
    int nrp;
    String nama;

    public void setNrp(int i) {
        nrp=i;
    }

    public void setNama(String i) {
        nama=i;
    }
}
```

Percobaan 3 : Mengimplementasikan UML class diagram dalam program

Berikut adalah sebuah UML class diagram dari suatu kasus:



Dari class diagram tersebut, dapat diimplementasikan ke dalam program sebagai berikut:

```
public class Siswa {
    public int nrp;

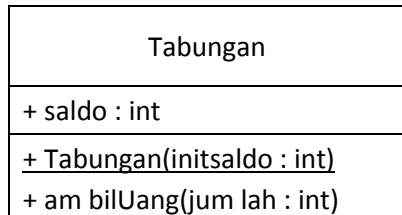
    public Siswa(int i) {
        nrp=i;
    }

    public void setNrp(int i) {
        nrp=i;
    }

    public int getNrp() {
        return nrp;
    }
}
```

Latihan

Latihan 1 : Mengimplementasikan UML class diagram dalam program untuk class Tabungan



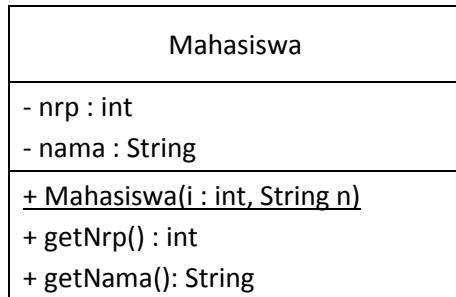
Transformasikan class diagram diatas ke dalam bentuk program!. Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesLatihan1 {
    public static void main(String args[]) {
        Tabungan tabungan=new Tabungan(5000);
        System.out.println("Saldo awal : "+tabungan.saldo);
        tabungan.ambilUang(2300);
        System.out.println("Jumlah uang yang diambil : 2300");
        System.out.println("Saldo sekarang : " + tabungan.saldo);
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Saldo awal : 5000
Jumlah uang yang diambil : 2300
Saldo sekarang : 2700
```

Latihan 2 : Mengimplementasikan UML class diagram dalam program untuk class Mahasiswa



Transformasikan class diagram diatas ke dalam bentuk program!. Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesLatihan2 {
    public static void main(String args[]) {
        Mahasiswa mhs=new Mahasiswa(12345, "Jono");
        System.out.println("NRP : " + mhs.getNrp());
```

```
        System.out.println("Nama : " + mhs.getNama());  
    }  
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti di atas.

```
NRP : 12345  
Nama : Jono
```

Latihan 3 : Mengimplementasikan UML class diagram dalam program untuk class Truk

Truk
- muatan : double
- muatanmaks : double
+ Truk(beratmaks : double)
+ getMuatan() : double
+ getMuatanMaks() : double
+ tambahMuatan(berat : double)

Transformasikan class diagram diatas ke dalam bentuk program?. Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesLatihan3 {  
    public static void main(String args[]) {  
        Truk truk=new Truk(1000);  
        System.out.println("Muatan maksimal : " +  
        truk.getMuatanMaks());  
        truk.tambahMuatan(500.0);  
        System.out.println("Tambah muatan : 500");  
        truk.tambahMuatan(350.0);  
        System.out.println("Tambah muatan : 350");  
        truk.tambahMuatan(100.0);  
        System.out.println("Tambah muatan : 100");  
        truk.tambahMuatan(150.0);  
        System.out.println("Muatan sekarang = " + truk.getMuatan());  
    }  
}
```

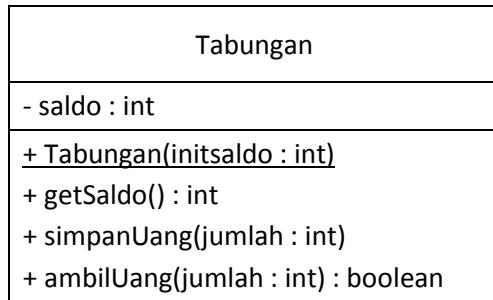
Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti di atas.

```
Muatan maksimal : 1000.0  
Tambah muatan : 500  
Tambah muatan : 350  
Tambah muatan : 100  
Tambah muatan : 150
```

Muatan sekarang = 950.0

Tugas

Tugas 1 : Mengimplementasikan UML class diagram dalam program untuk class Tabungan



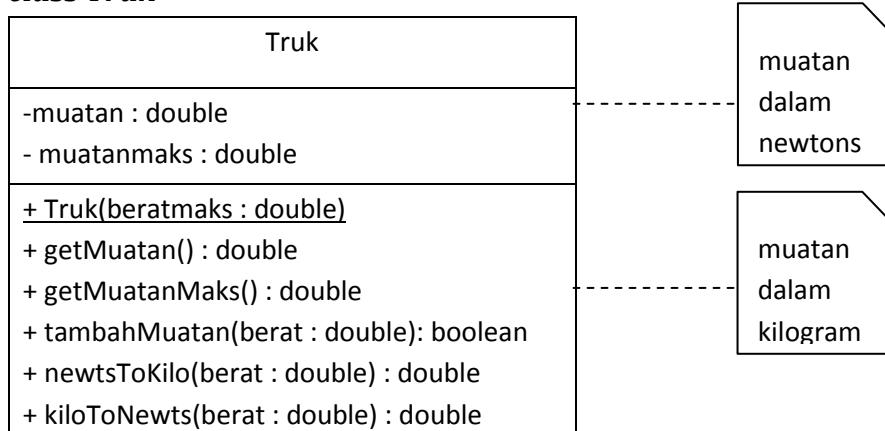
Transformasikan class diagram diatas ke dalam bentuk program?. Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesTugas1 {  
    public static void main(String args[]) {  
        boolean status;  
        Tabungan tabungan=new Tabungan(5000);  
        System.out.println("Saldo awal : "+tabungan.getSaldo());  
        tabungan.simpanUang(3000);  
        System.out.println("Jumlah uang yang disimpan : 3000");  
        status=tabungan.ambilUang(6000);  
        System.out.print("Jumlah uang yang diambil : 6000");  
        if (status)  
            System.out.println("    ok");  
        else  
            System.out.println("    gagal");  
        tabungan.simpanUang(3500);  
        System.out.println("Jumlah uang yang disimpan : 3500");  
        status=tabungan.ambilUang(4000);  
        System.out.print("Jumlah uang yang diambil : 4000");  
        if (status)  
            System.out.println("    ok");  
        else  
            System.out.println("    gagal");  
        status=tabungan.ambilUang(1600);  
        System.out.print("Jumlah uang yang diambil : 1600");  
        if (status)  
            System.out.println("    ok");  
        else  
            System.out.println("    gagal");  
        tabungan.simpanUang(2000);  
        System.out.println("Jumlah uang yang disimpan : 2000");  
        System.out.println("Saldo sekarang = " + tabungan.getSaldo());  
    }  
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000    ok
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000    ok
Jumlah uang yang diambil : 1600    gagal
Jumlah uang yang disimpan : 2000
Saldo sekarang = 3500
```

Tugas 2 : Mengimplementasikan UML class diagram dalam program untuk class Truk



Keterangan : 1 kilogram = 9,8 newtons

Transformasikan class diagram diatas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan

```
public class TestTugas2 {
    public static void main(String args[]) {
        Truk truk=new Truk(900);
        boolean status;
        System.out.println("Muatan maksimal : " +
        truk.getMuatanMaks());
        status=truk.tambahMuatan(500.0);
        System.out.print("Tambah muatan : 500");
        if (status)
            System.out.println(" ok");
        else
            System.out.println(" gagal");
        status=truk.tambahMuatan(300.0);
        System.out.print("Tambah muatan : 300");
        if (status)
            System.out.println(" ok");
        else
            System.out.println(" gagal");
    }
}
```

```
status=truk.tambahMuatan(150.0);
System.out.print("Tambah muatan : 150");
if (status)
    System.out.println("  ok");
else
    System.out.println("  gagal");
status=truk.tambahMuatan(50.0);
System.out.print("Tambah muatan : 50");
if (status)
    System.out.println("  ok");
else
    System.out.println("  gagal");
System.out.println("Muatan sekarang = " + truk.getMuatan());
}
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti di atas.

```
Muatan maksimal : 900.0
Tambah muatan : 500  ok
Tambah muatan : 300  ok
Tambah muatan : 150  gagal
Tambah muatan : 50  ok
Muatan sekarang = 849.999999999999
```

MODUL 5
Enkapsulasi, Constructor, overloading constructor
(Pertemuan 5)

Tujuan :

Dengan praktikum ini mahasiswa diharapkan dapat:

- Menerapkan konsep enkapsulasi pada class
- Mendeklarasikan suatu constructor

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang konsep enkapsulasi di java?
2. Apa yang anda ketahui tentang konstruktor di java

DASAR TEORI

- Kita dapat menyembunyikan information dari suatu class sehingga anggota-anggota class tersebut tidak dapat diakses dari luar. Adapun caranya adalah cukup dengan memberikan akses kontrol private ketika mendeklarasikan suatu atribut atau method. Contoh:
private int nrp;
- Encapsulation (Enkapsulasi) adalah suatu cara untuk menyembunyikan implementasi detail dari suatu class. Enkapsulasi mempunyai dua hal mendasar, yaitu:
 - information hiding
 - menyediakan suatu perantara (method) untuk pengaksesan data

Contoh:

```
public class Siswa {  
    private int nrp;  
  
    public void setNrp(int n) {  
        nrp=n;  
    }  
}
```

- Constructor (konstruktor) adalah suatu method yang pertama kali dijalankan pada saat pembuatan suatu obyek. Konstruktor mempunyai ciri yaitu:
 - mempunyai nama yang sama dengan nama class
 - tidak mempunyai return type (seperti void, int, double, dan lain-lain)

Contoh:

```
public class Siswa {  
    private int nrp;  
    private String nama;  
  
    public Siswa(int n, String m) {  
        nrp=n;  
        nama=m;  
    }  
}
```

- Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Contoh:

```
public class Siswa {
    private int nrp;
    private String nama;
    public Siswa(String m) {
        nrp=0;
        nama="";
    }
    public Siswa(int n, String m) {
        nrp=n;
        nama=m;
    }
}
```

Percobaan

Percobaan 1 : Melakukan enkapsulasi pada suatu class

Mahasiswa
+ nrp : int
+ nama : String
<u>+ Mahasiswa(nrp : int, nama : String)</u>

Jika enkapsulasi dilakukan pada class diagram di atas, maka akan berubah menjadi:

Mahasiswa
- nrp : int
- nama : String
<u>+ Mahasiswa(nrp : int, nama : String)</u>
+ getNrp() : int
+ getNama(): String
+ setNrp(nrp : int)
+ setNama(nama : String)

Percobaan 2 : Melakukan overloading constructor

Mahasiswa
- nrp : int
- nama : String
<u>+ Mahasiswa()</u>
<u>+ Mahasiswa(nama : String)</u>
<u>+ Mahasiswa(nrp : int, nama : String)</u>

Dari class diagram tersebut, dapat diimplementasikan ke dalam program sebagai berikut:

```
public class Mahasiswa {
    private int nrp;
    private String nama;
```

```

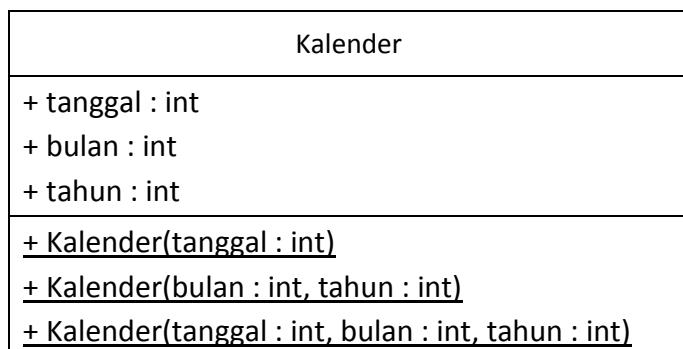
public Mahasiswa() {
    nrp=0;
    nama="";
}

public Mahasiswa(String nama) {
    nrp=0;
    this.nama=nama;
}

public Mahasiswa(int nrp, String nama) {
    this.nrp=nrp;
    this.nama=nama;
}
}
    
```

Latihan

Mengimplementasikan UML class diagram dalam program untuk class Kalender



Dari class diagram di atas, desainlah suatu class yang memenuhi konsep enkapsulasi. Untuk nilai inisialisasi, dipakai 1-1-2000. Pakailah kata kunci *this* untuk mempersingkat pengkodean. Tulislah listing program berikut ini sebagai pengetesan

```

public class TesLatihan {
    public static String getTime(Kalender kal) {
        String tmp;
        tmp=kal.getTanggal() + "-" +
            kal.getBulan() + "-" +
            kal.getTahun();
        return tmp;
    }

    public static void main(String args[]) {
        Kalender kal=new Kalender(8);
        System.out.println("Waktu awal : " + getTime(kal));
        kal.setTanggal(9);
        System.out.println("1 hari setelah waktu awal : " +
getTime(kal));
        kal=new Kalender(6,2003);
        System.out.println("Waktu berubah : " + getTime(kal));
        kal.setBulan(7);
    }
}
    
```

```
        System.out.println("1 bulan setelah itu : " + getTime(kal));
        kal=new Kalender(20,10,2004);
        System.out.println("Waktu berubah : " + getTime(kal));
        kal.setTahun(2005);
        System.out.println("1 tahun setelah itu : " + getTime(kal));
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda.

```
Waktu awal : 8-1-2000
1 hari setelah waktu awal : 9-1-2000
Waktu berubah : 1-6-2003
1 bulan setelah itu : 1-7-2003
Waktu berubah : 20-10-2004
1 tahun setelah itu : 20-10-2005
```

TUGAS

1. Buatlah sebuah program bebas yang menunjukkan penggunaan konsep enkapsulasi !
2. Buatlah sebuah program bebas yang menggunakan konsep overloading konstruktor!

Modul 6
Inheritance, Overloading, Overriding
(Pertemuan 6)

Tujuan

1. Untuk memahami konsep Inheritance dan cara penggunaanya
2. Untuk memahami konsep Overloading, overriding dan cara penggunaanya

Tugas Pendahuluan

1. Apa yang anda ketahui tentang konsep inheritance di java ?
2. Apa yang anda ketahui tentang konsep overloading, overriding di java?

Dasar Teori

- Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah class dapat mempunyai class turunan. Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class-nya, sehingga member dari suatu subclass adalah terdiri dari apa-apa yang ia punya dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.
- Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inheritance:

Contoh:

```
public class B extends A {  
...  
}
```

Contoh di atas memberitahukan kompiler Java bahwa kita ingin meng-extend class A ke class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

- Java hanya memperkenankan adanya single inheritance. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class. Dengan konsep single inheritance ini, pewarisan akan dapat diamati dengan mudah.
- Dalam konsep dasar inheritance dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya. Contoh :

```
public class Pegawai {  
public String nama;  
public double gaji;  
}  
public class Manajer extends Pegawai {  
public String departemen;  
}
```

Pada saat class Manajer menurunkan atau memperluas (extend) class Pegawai, maka ia mewarisi data member yang dipunyai oleh class Pegawai. Dengan demikian, class Manajer

mempunyai data member yang diwarisi oleh Pegawai (nama, gaji), ditambah dengan data member yang ia punyai (departemen).

- Pengaksesan member yang ada di parent class dari subclass-nya tidak jauh berbeda dengan pengaksesan member subclass itu sendiri.
- Suatu parent class dapat tidak mewariskan sebagian member-nya kepada subclass-nya. Sejauh mana suatu member dapat diwariskan ke class lain, ataupun suatu member dapat diakses dari class lain, sangat berhubungan dengan access control (kontrol pengaksesan). Di dalam java, kontrol pengaksesan dapat digambarkan dalam tabel berikut ini:

Modifier	Class yang sama	Package yang sama	Subclass	Class manapun
private	✓			
default	✓	✓		
protected	✓	✓	✓	
public	✓	✓	✓	✓

- Kata kunci *super* dipakai untuk merujuk pada member dari parent class, sebagaimana kata kunci *this* yang dipakai untuk merujuk pada member dari class itu sendiri. Adapun format penulisannya adalah sebagai berikut:

super.data_member % merujuk pada data member pada parent class
 super.function_member() % merujuk pada function member pada parent class
 super() % merujuk pada konstruktor pada parent class

Contoh:

```
public class Siswa {
    private int nrp;

    public setNrp(int nrp) {
        this.nrp=nrp;
    }
}
```

- Overloading adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda. Contoh penggunaan overloading dilihat dibawah ini:

Gambar(int t1) % 1 parameter titik, untuk menggambar titik

Gambar(int t1,int t2) % 2 parameter titik, untuk menggambar garis

Gambar(int t1,int t2,int t3)%3 parameter titik, untuk menggambar segitiga

Gambar(int t1,int t2,int t3,int t4) % 4 parameter titik, untuk menggambar persegi empat

Overloading ini dapat terjadi pada class yang sama atau pada suatu parent class dan subclass-nya. Overloading mempunyai ciri-ciri sebagai berikut:

1. Nama method harus sama
 2. Daftar parameter harus berbeda
 3. Return type boleh sama, juga boleh berbeda
- Overriding adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. Overriding mempunyai ciri-ciri sebagai berikut :
 1. Nama method harus sama

2. Daftar parameter harus sama
3. Return type harus sama

Berikut ini contoh terjadinya overriding dimana method Info() pada class Child mengOverride method Info() pada class parent:

```
class Parent {
    public void Info() {
        System.out.println("Ini class Parent");
    }
}

class Child extends Parent {
    public void Info() {
        System.out.println("Ini class Child");
    }
}
```

- Method yang terkena override (overridden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (overriding method).

Percobaan

Melakukan overloading pada method

Tulislah listing program berikut ini dan amati yang terjadi pada saat terjadinya overloading pada method.

```
import java.awt.Point;
public class Segiempat {
    int x1 = 0;
    int y1 = 0;
    int x2 = 0;
    int y2 = 0;

    public void buatSegiempat(int x1, int y1, int x2, int y2) {
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
    }

    public void buatSegiempat(Point topLeft, Point bottomRight) {
        x1 = topLeft.x;
        y1 = topLeft.y;
        x2 = bottomRight.x;
        y2 = bottomRight.y;
    }

    public void buatSegiempat(Point topLeft, int w, int h) {
        x1 = topLeft.x;
        y1 = topLeft.y;
        x2 = (x1 + w);
        y2 = (y1 + h);
    }
}
```

```
void cetakSegiempat(){
    System.out.print("Segiempat: <" + x1 + ", " + y1);
    System.out.println(", " + x2 + ", " + y2 + ">");
}

public static void main(String[] arguments) {
    Segiempat rect = new Segiempat();
    System.out.println("Buat segiempat dengan koordinat (25,25)
dan (50,50)");
    rect.buatSegiempat(25, 25, 50, 50);
    rect.cetakSegiempat();
    System.out.println();
    System.out.println("Buat segiempat dengan point (10,10) dan
point (20,20):");
    rect.buatSegiempat(new Point(10,10), new Point(20,20));
    rect.cetakSegiempat();
    System.out.println();
    System.out.print("Buat segiempat dengan 1 point (10,10),
koodinat (50,50)");
    rect.buatSegiempat(new Point(10,10), 50, 50);
    rect.cetakSegiempat();
}
}
```

Ketika program tersebut dijalankan, akan tampak hasil seperti dibawah ini :

```
Buat segiempat dengan koordinat (25,25) dan (50,50)
Segiempat: <25, 25, 50, 50>

Buat segiempat dengan point (10,10) dan point (20,20):
Segiempat: <10, 10, 20, 20>
Buat segiempat dengan 1 point (10,10), koodinat (50,50)Segiempat:
<10, 10, 60, 60>
```

Percobaan

Percobaan 1 : Menggunakan kata kunci super

Berikut ini listing penggunaan kata kunci super.

```
class Parent {
    public int x = 5;
}

class Child extends Parent {
    public int x = 10;

    public void Info(int x) {
        System.out.println("Nilai x sebagai parameter = " + x);
        System.out.println("Data member x di class Child = " +
this.x);
```

```
        System.out.println("Data member x di class Parent = "
"      +
super.x);
    }
}

public class NilaiX {
    public static void main(String args[]) {
        Child tes = new Child();
        tes.Info(20);
    }
}
```

Ketika program tersebut dijalankan, akan tampak hasil seperti dibawah ini :

```
Nilai x sebagai parameter = 20
Data member x di class Child = 10
Data member x di class Parent = 5
```

Percobaan 2 : Kontrol pengaksesan

Buatlah class Pegawai seperti dibawah ini:

```
public class Pegawai {
    private String nama;
    public double gaji;
}
```

Kemudian buatlah class Manajer seperti ini dibawah ini.

```
public class Manajer extends Pegawai {
    public String departemen;

    public void IsiData(String n, String d) {
        nama=n;
        departemen=d;
    }
}
```

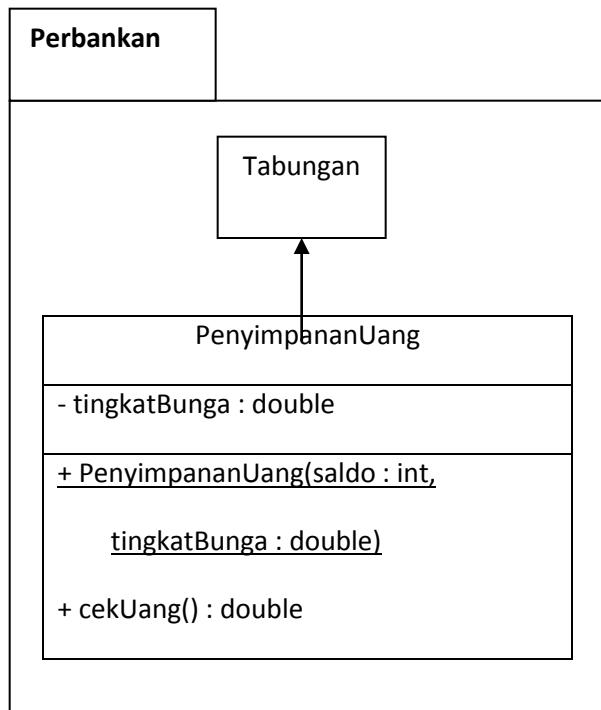
Sekarang cobalah untuk mengkompilasi class Manajer diatas. Apa yang terjadi?. Pesan kesalahan akan muncul seperti ini:

```
Manajer.java:5: nama has private access in Pegawai
        nama=n;
```

Ini membuktikan bahwa class Manajer tidak mewarisi data member nama dari parent class-nya (Pegawai).

Percobaan 3 : Konstruktor tidak diwariskan

Buatlah class kosong bernama Parent seperti di bawah:



Ubahlah mode akses atribut saldo pada Tabungan menjadi protected. Lalu transformasikan class diagram di atas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan.

```

import perbankan.*;

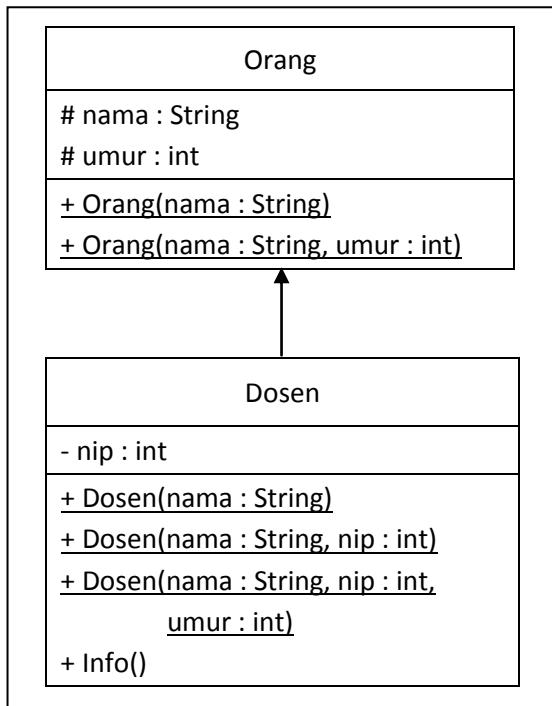
public class TesLatihan {
    public static void main(String args[]) {
        PenyimpananUang tabungan=new PenyimpananUang(5000,8.5/100);
        System.out.println("Uang yang ditabung : 5000");
        System.out.println("Tingkat bunga sekarang : 8.5%");
        System.out.println("Total uang anda sekarang : " +
        tabungan.cekUang());
    }
}
  
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti di atas.

```

Uang yang ditabung : 5000
Tingkat bunga sekarang : 8.5%
Total uang anda sekarang : 5425.0
  
```

Mengimplementasikan UML class diagram dalam program



Transformasikan class diagram diatas ke dalam bentuk program!. Tulislah listing program berikut ini sebagai pengetesan.

```

public class TesLatihan {
    public static void main(String args[]) {
        System.out.println("Masukkan identitas dosen 1 : Agus");
        Dosen dosen1=new Dosen("Agus");

        System.out.println("Masukkan identitas dosen 2 : Budi,
                           NIP. 1458");
        Dosen dosen2=new Dosen("Budi", 1458);
        System.out.println("Masukkan identitas dosen 3 : Iwan,
                           NIP. 1215, umur 47");
        Dosen dosen3=new Dosen("Iwan", 1215, 47);
        System.out.println();
        dosen1.Info();
        System.out.println();
        dosen2.Info();
        System.out.println();
        dosen3.Info();
    }
}
  
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti di atas.

```
Masukkan identitas dosen 1 : Agus
Masukkan identitas dosen 2 : Budi, NIP. 1458
Masukkan identitas dosen 3 : Iwan, NIP. 1215, umur 47

Nama : Agus
NIP : -
Umur : -

Nama : Budi
NIP : 1458
Umur : -

Nama : Iwan
NIP : 1215
Umur : 47
```

Tugas

Mengimplementasikan UML class diagram dalam program untuk package perbankan

Transformasikan class diagram diatas ke dalam bentuk program!. Tulislah listing program berikut ini sebagai pengetesan.

```
import perbankan.*;
public class TesTugas {
    public static void main(String args[]) {
        PengambilanUang tabungan=new PengambilanUang(5000,1000);
        System.out.println("Uang yang ditabung : 5000");
        System.out.println("Uang yang diproteksi : 1000");
        System.out.println("-----");
        System.out.println("Uang yang akan diambil : 4500 " +
        tabungan.ambilUang(4500));
        System.out.println("Saldo sekarang : " + tabungan.getSaldo());
        System.out.println("-----");
        System.out.println("Uang yang akan diambil : 2500 " +
        tabungan.ambilUang(2500));
        System.out.println("Saldo sekarang : " + tabungan.getSaldo());
    }
}
```

Lakukan kompilasi pada program di atas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Uang yang ditabung : 5000
Uang yang diproteksi : 1000
-----
Uang yang akan diambil : 4500 false
Saldo sekarang : 5000
-----
Uang yang akan diambil : 2500 true
Saldo sekarang : 2500
```

MODUL 7
JAVA GUI Swing
(Pertemuan 7 dan 8)

Tujuan :

1. Mahasiswa mengetahui dan mampu mempraktekkannya komponen Swing di java
2. Mahasiswa mampu membuat form sederhana dengan memanfaatkan komponen Swing diJava

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang java GUI Swing dan AWT?
2. Kenapa dalam GUI di java Swing lebih sering digunakan ?
3. Sebutkan komponen Swing yang ada di java?

DASAR TEORI

Java swing merupakan toolkit GUI pada Java yang sering dipakai untuk membuat aplikasi dengan Interface berbasis grafis.

Beberapa komponen dari Java Swing yaitu :

- Button : Tombol
- Label : teks untuk memberikan suatu keterangan
- Text Field : media input text sepanjang 1 baris
- Text Area : media input text dengan ukuran bisa lebih dari 1 baris
- Menu Bar : Bar yang biasanya menu utama suatu aplikasi
- Menu : Menu-menu pada aplikasi
- Table : untuk menampilkan data dalam bentuk table
- Combo box : media input untuk memilih 1 opsi dari beberapa opsi
- Radio box : seperti combo box namun semua opsi langsung ditampilkan
- Check Box : media input untuk memilih beberapa opsi dari opsi yang tersedia
- Tool bar : bar-bar untuk memilih tool-tool yang disediakan aplikasi dan bisa ditampilkan dalam bentuk ikon

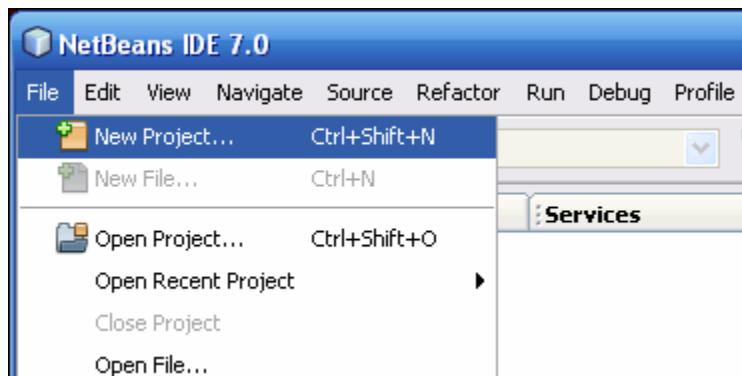
Setiap komponen memiliki metode setter maupun getter untuk mengakses atributnya. Semisal untuk JLabel terdapat metode setText() untuk mengubah tulisan pada label dan pada JTextField terdapat method getText() untuk mengambil data yang diinputkan ke dalam teks.

Setiap komponen juga memiliki yang disebut event listener (atau kadang disebut event handler) yaitu suatu aksi yang dilakukan ketika terjadi suatu event tertentu. Misal, ketika tombol ditekan, ketika teks ditulis dalam text field, dsb.

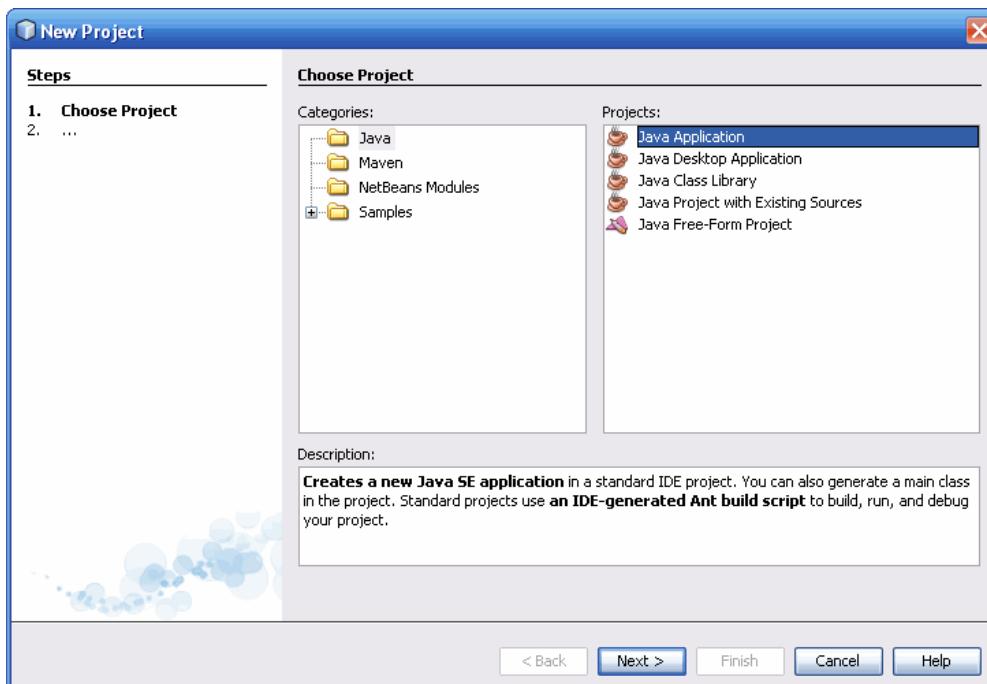
KEGIATAN PRAKTIKUM

1.1 Latihan Swing

Buka Netbeans pada PC anda, kemudian Klik **File | New Project** Seperti contoh pada **Gambar** berikut :



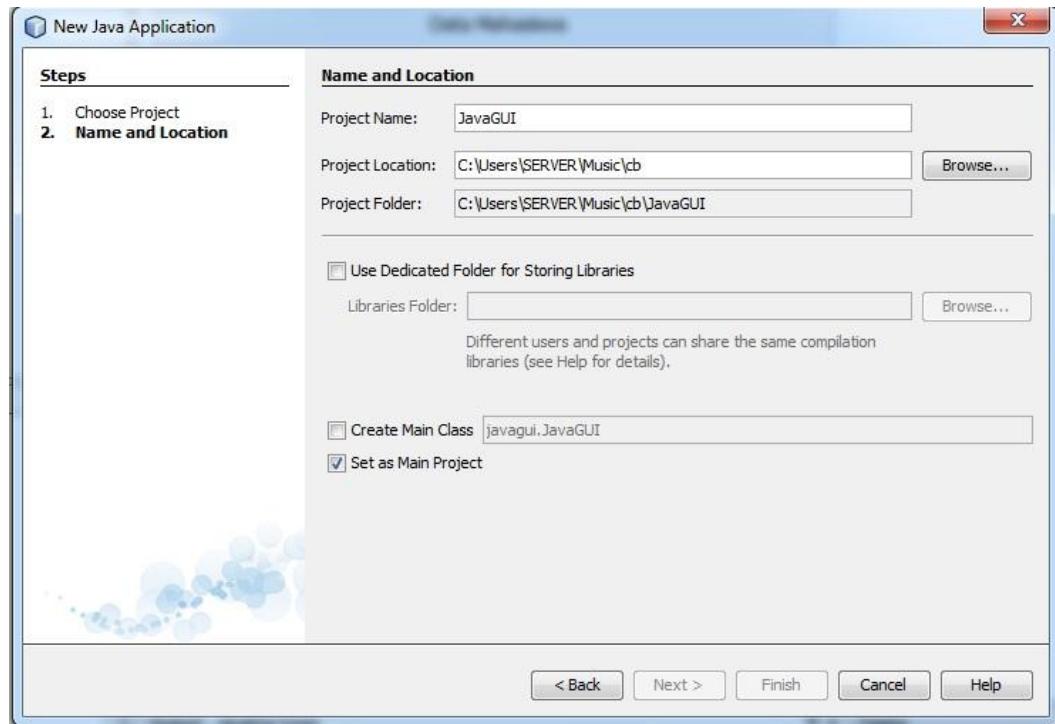
Dalam New Project wizard, memperluas kategori Java dan pilih Java Application seperti yang ditunjukkan pada **Gambar** berikut:



Dalam Nama dan lokasi halaman wizard, lakukan hal berikut (seperti yang ditunjukkan pada gambar di bawah):

- Pada Project Name field, Ketik JavaGUI.
- Pada Create Main Class field, Lepaskan Centang.
- Biarkan kotak centang Set sebagai Main Project dipilih.

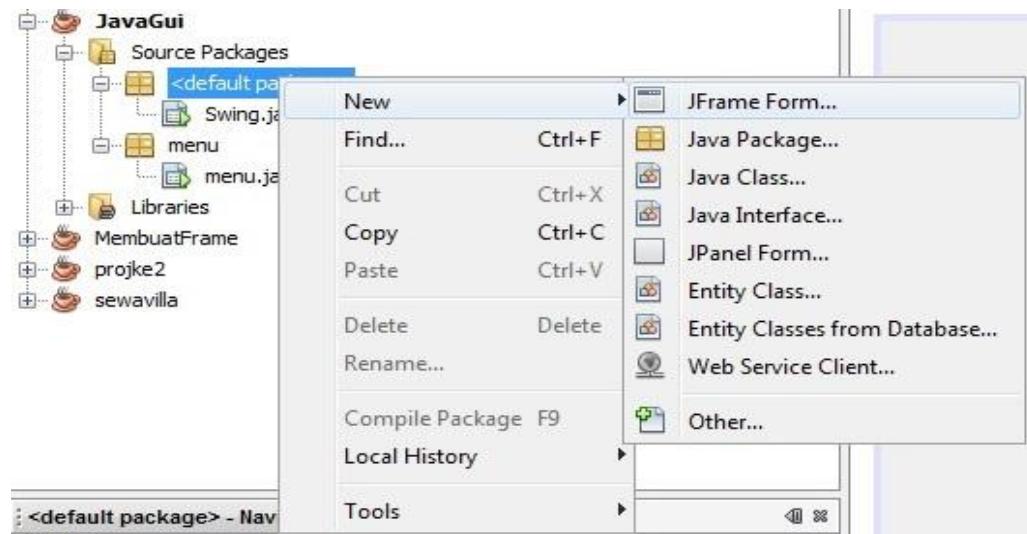
Seperti Pada contoh **Gambar** berikut :



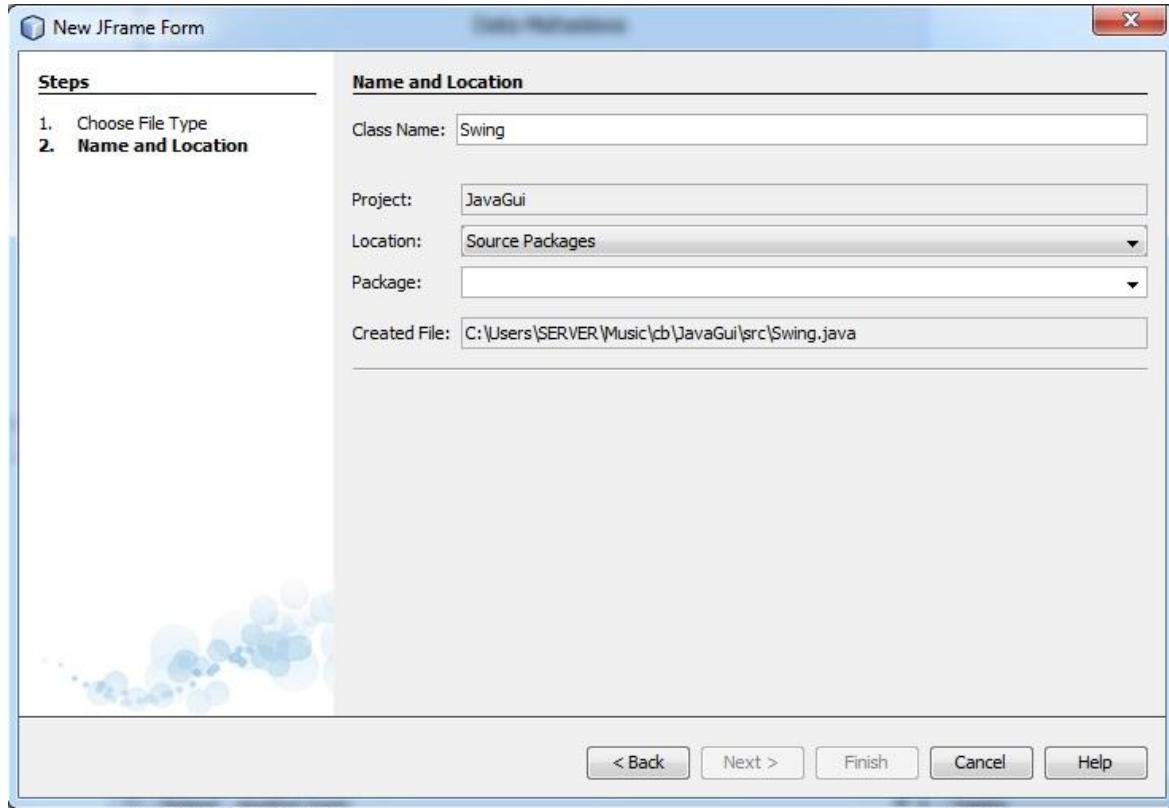
Proyek dibuat dan dibuka dalam IDE. Anda akan melihat komponen-komponen berikut :

- Jendela Projects, yang berisi tampilan pohon dari berbagai komponen proyek, berisi file sumber, pustaka yang mendasari kode Anda tergantung pada, dan sebagainya.
- Jendela Source Editor dengan sebuah file yang bernama JavaGUI terbuka.
- Jendela Navigator, yang dapat Anda gunakan untuk navigasi cepat di antara elemen-elemen dalam kelas yang dipilih.

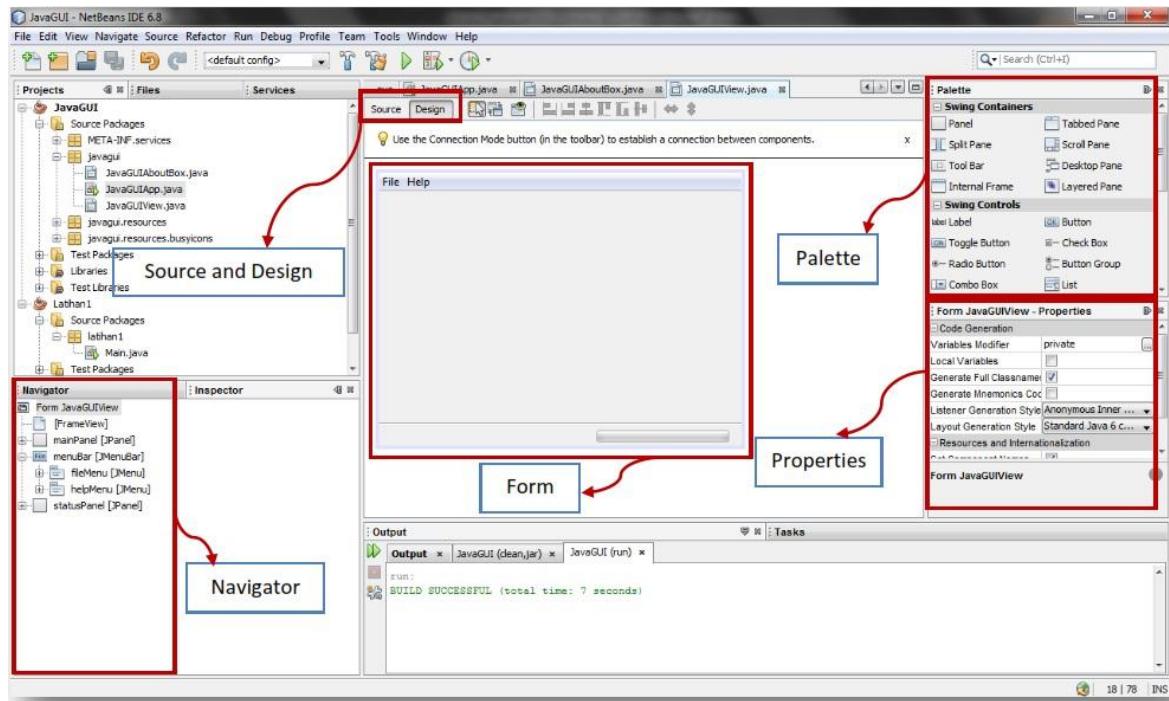
Klik kanan pada <default Package> untuk membuat frame. New | JFrame Form. Seperti pada gambar berikut.



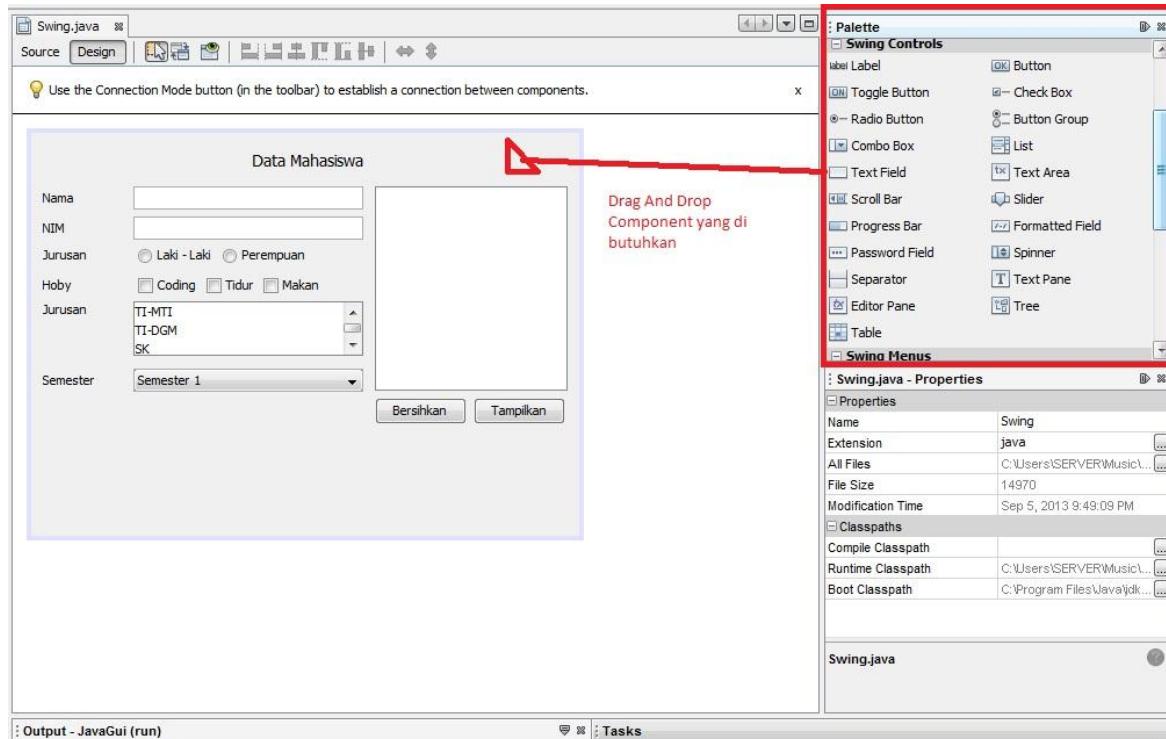
Maka akan muncul jendela new JFrame form. Pada field Class Name berikan nama Swing. Lalu klik Finish. Seperti pada gambar berikut.



Selanjutnya adalah perkenalan bagian netbeans.

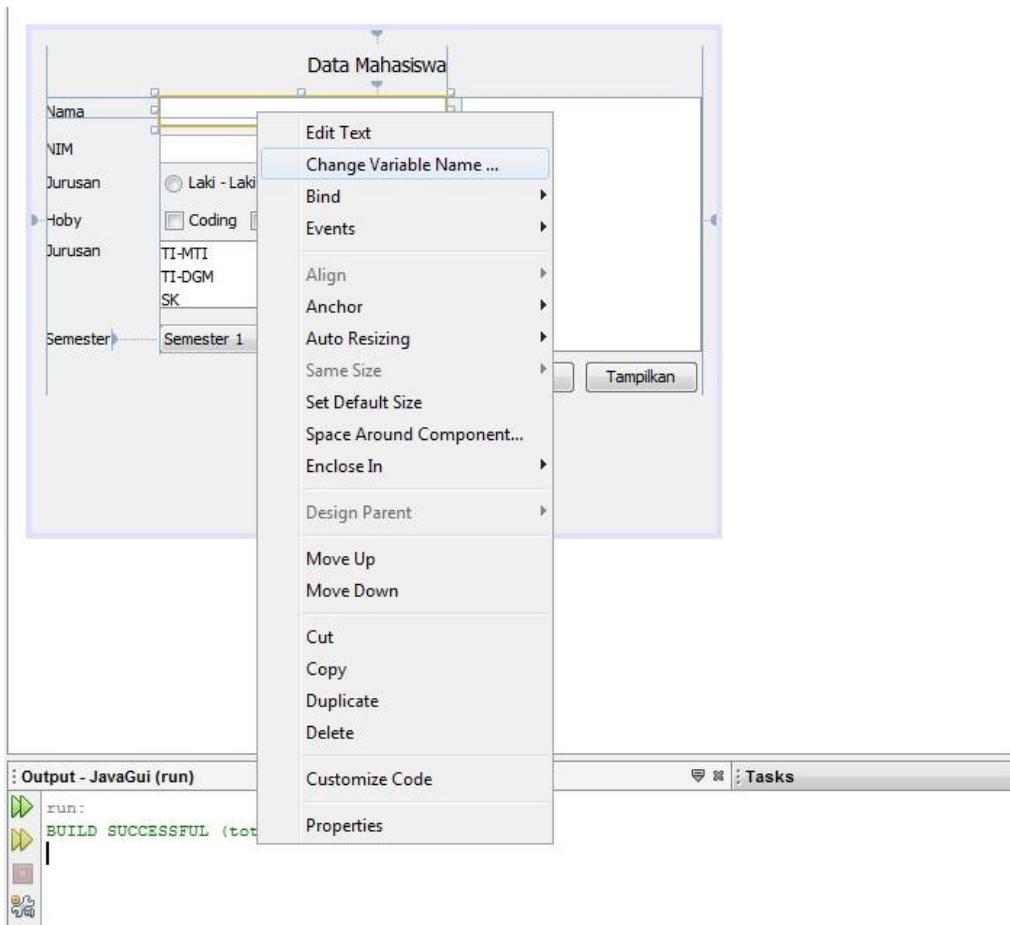


Langkah selanjutnya yaitu lakukan drag and drop swing yang dibutuhkan, lakukan seperti gambar berikut.

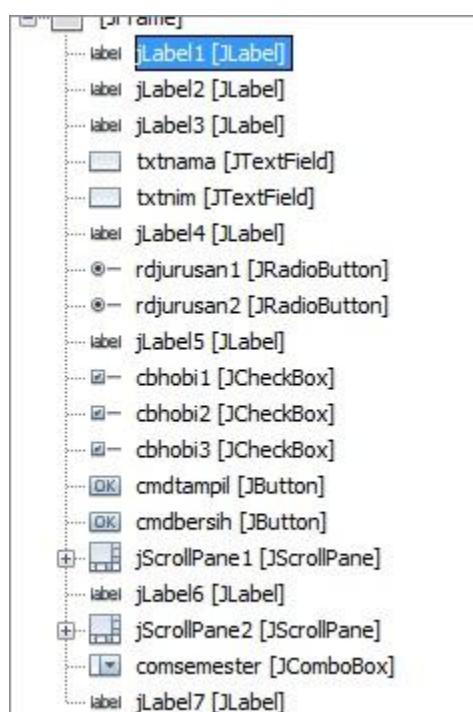


Buatlah design seperti data mahasiswa diatas dengan menggunakan :

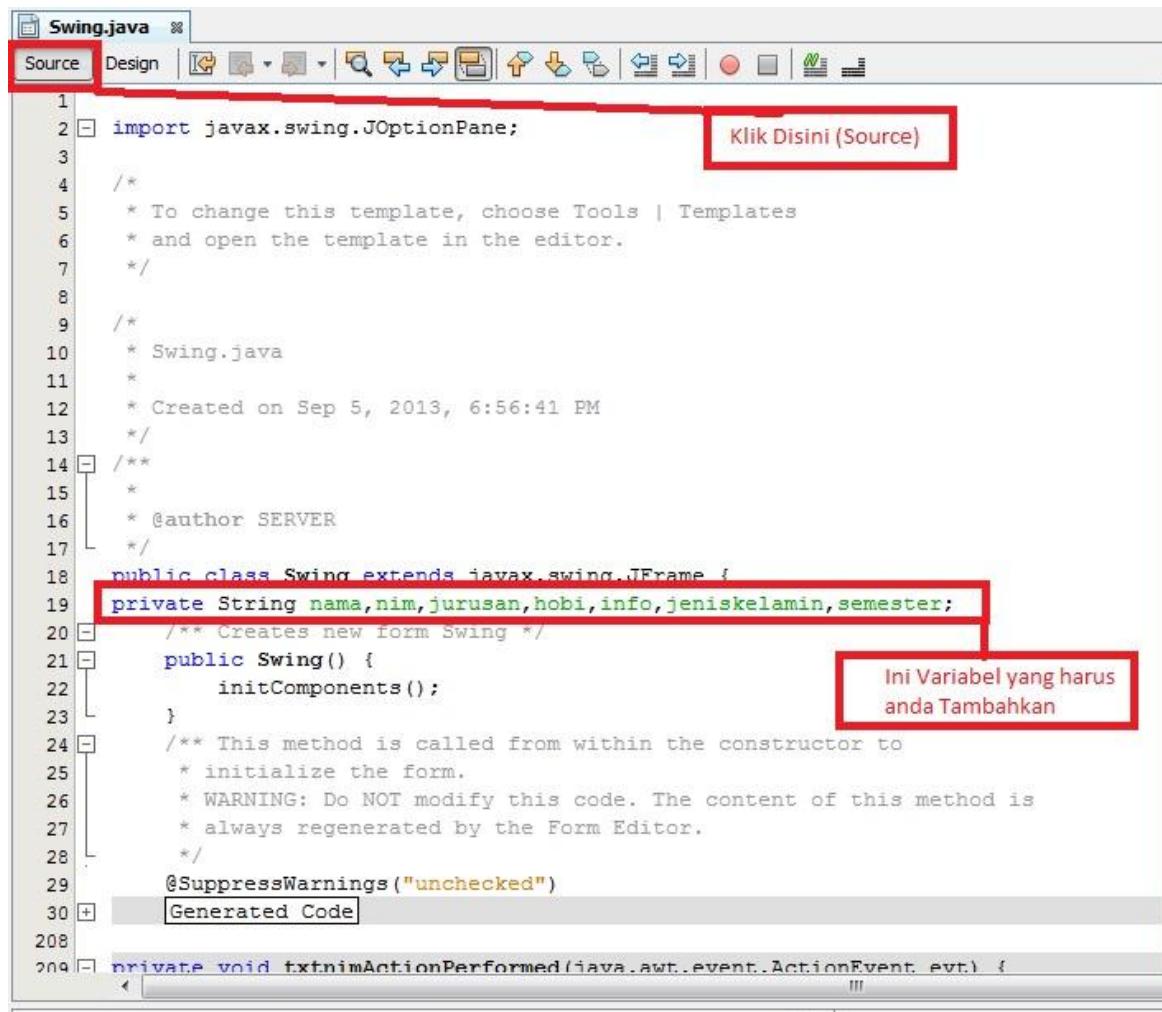
1. 2 buah JTextField
2. 2 buah JRadioButton
3. 3 buah JCheckBox
4. 1 buah JList
5. 1 buah JComboBox
6. 1 buah JTextArea
7. 2 buah JButton
8. 7 buah JLabel



Klik Kanan Pada setiap Swing pilih **Change Variable Name** setelah itu ubah nama **klik ok**, dan klik kanan lagi lalu pilih **Edit Text**, sesuaikan nama variabel dan text seperti pada gambar berikut:



Kemudian anda dapat memberikan variabel pada source code dengan cara klik source dan ketikan syntax berikut tepat di bawah class yang anda buat.



The screenshot shows a Java code editor with the file name "Swing.java". The "Source" tab is selected. A red box highlights the line "private String nama,nim,jurusan,hobi,info,jeniskelamin,semester;" with the annotation "Klik Disini (Source)". Another red box highlights the line "Generated Code" with the annotation "Ini Variabel yang harus anda Tambahkan".

```

1
2 import javax.swing.JOptionPane;
3
4 /**
5  * To change this template, choose Tools | Templates
6  * and open the template in the editor.
7 */
8
9 /**
10 * Swing.java
11 *
12 * Created on Sep 5, 2013, 6:56:41 PM
13 */
14 /**
15 *
16 * @author SERVER
17 */
18 public class Swing extends javax.swing.JFrame {
19     private String nama,nim,jurusan,hobi,info,jeniskelamin,semester;
20     /** Creates new form Swing */
21     public Swing() {
22         initComponents();
23     }
24     /** This method is called from within the constructor to
25      * initialize the form.
26      * WARNING: Do NOT modify this code. The content of this method is
27      * always regenerated by the Form Editor.
28      */
29     @SuppressWarnings("unchecked")
30     // Generated Code
31
32     private void txtnimActionPerformed(java.awt.event.ActionEvent evt) {
33         ...
34 }

```

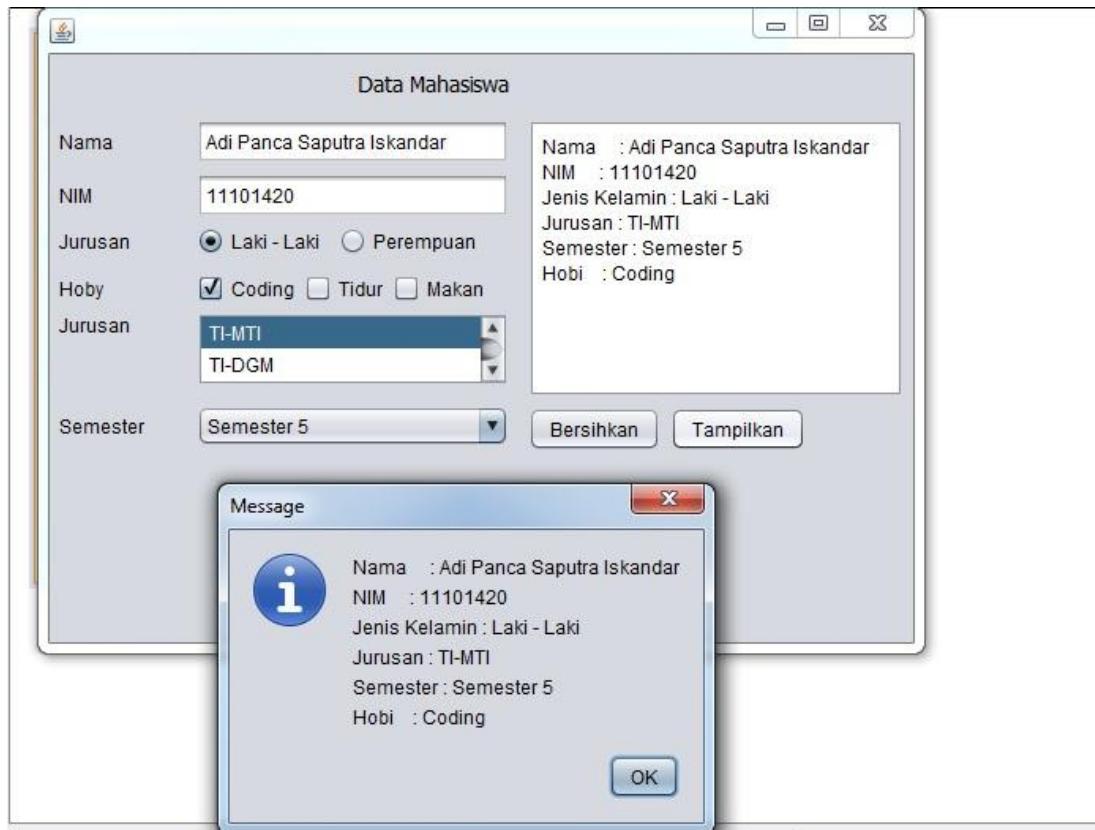
Kemudian kembali ke **Design** dan double klik pada cmdtampil atau klik kanan, kemudian pilih **Event | Action | actionPerformed**. Maka anda akan masuk pada bagian coding / source code. Dan ketikan syntax berikut.

```
[ private void cmdtampilActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    nama=txtnama.getText();  
    nim=txtnim.getText();  
    if (cbhobi1.isSelected())  
        hobi = "Coding";  
    if (cbhobi2.isSelected())  
        hobi += " ,Tidur";  
    if (cbhobi3.isSelected())  
        hobi += " ,Makan";  
  
    if (rdjurusan1.isSelected())  
        jeniskelamin = "Laki - Laki";  
    else  
        jeniskelamin = "Perempuan";  
  
    jurusan = listjurusan.getSelectedValue().toString();  
    semester = comsemester.getSelectedItem().toString();  
  
    info="Nama : "+nama+"\n";  
    info+="NIM : "+nim+"\n";  
    info+="Jenis Kelamin : "+jeniskelamin+"\n";  
    info+="Jurusan : "+jurusan+"\n";  
    info+="Semester : "+semester+"\n";  
    info+="Hobi : "+hobi+"\n";  
    hasil.setText(info);  
    JOptionPane.showMessageDialog(null, info);  
}  
]
```

Dan pada cmdbersih lakukan hal yang sama dan ketikan syntax berikut.

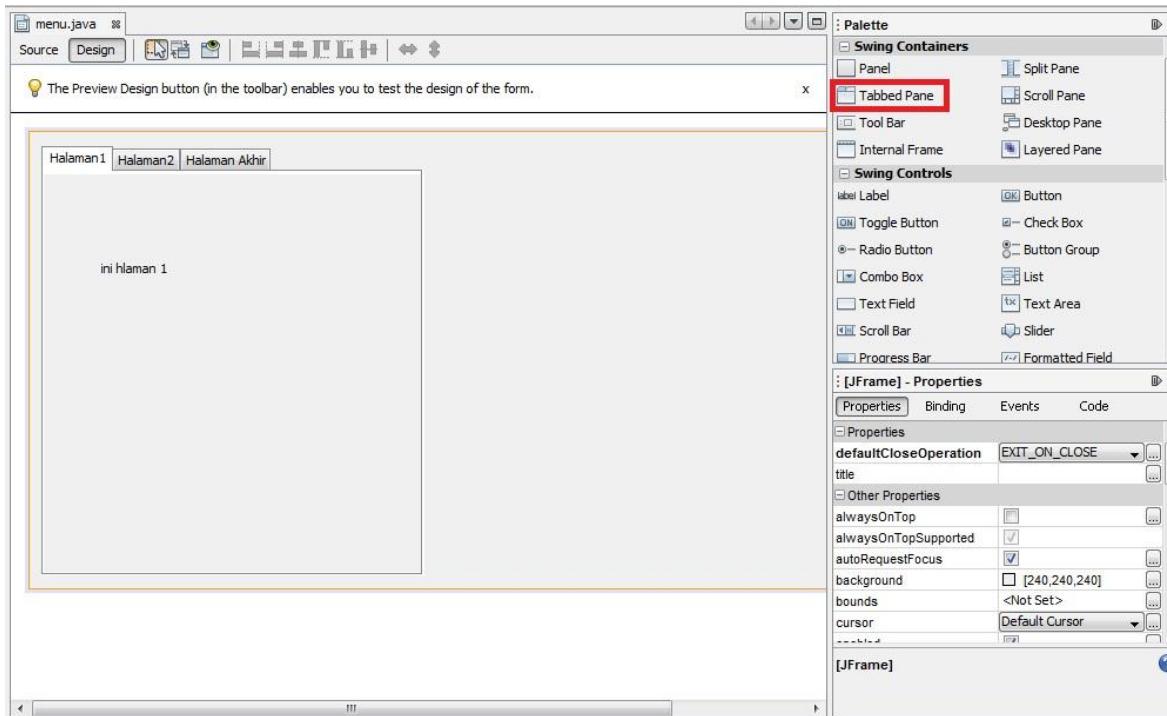
```
private void cmdbersihActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    txtnama.setText("");  
    txtnim.setText("");  
    cbhobi1.setSelected(false);  
    cbhobi2.setSelected(false);  
    cbhobi3.setSelected(false);  
    hasil.setText("");  
}  
]
```

Jika sudah maka project anda siap di **compile** dan di **run**.maka program anda akan tampil seperti gambar berikut dan silahkan diisi data anda dan klik **Tampilkan**.



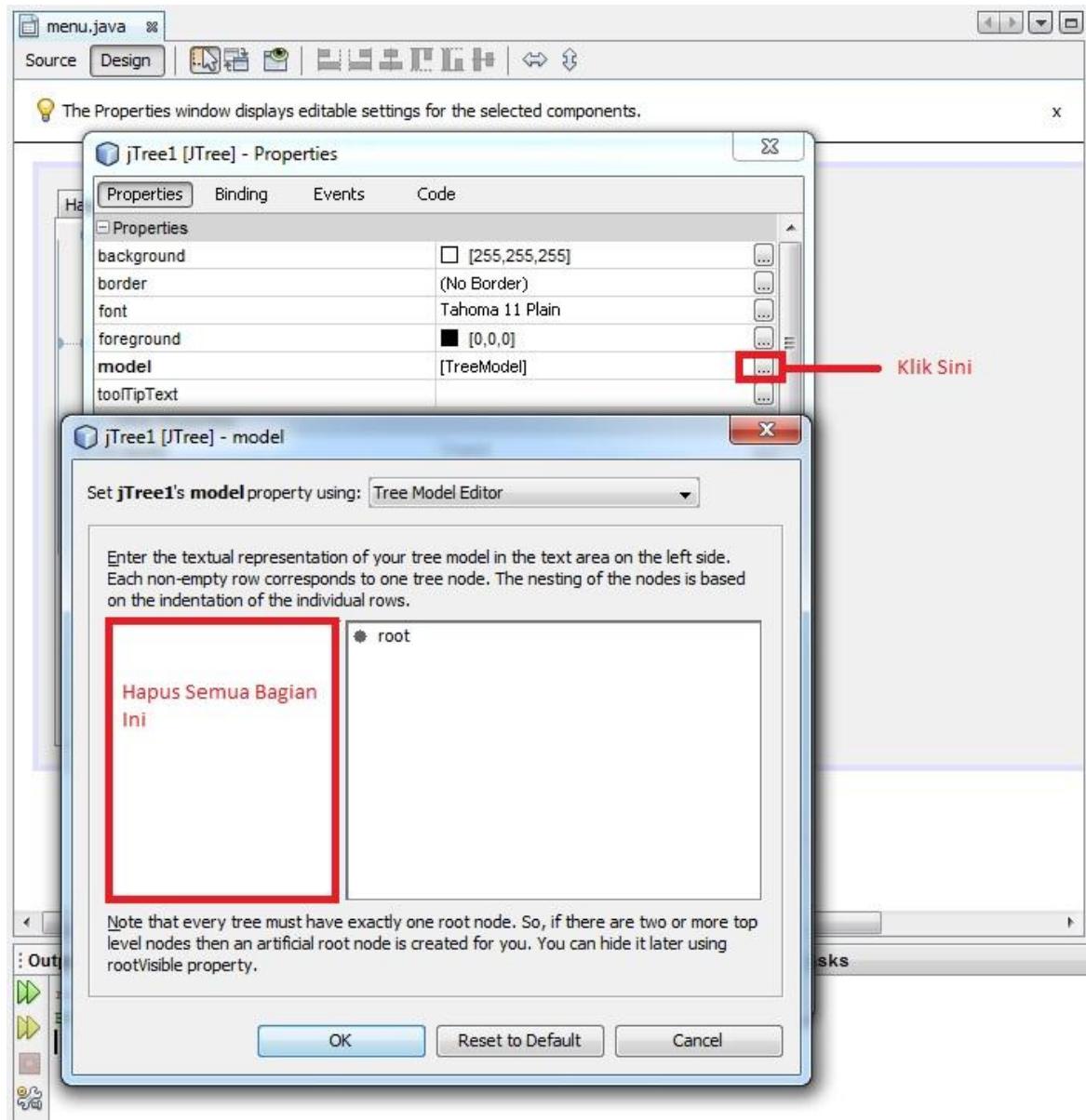
1.1. Latihan JTabPane, JTree, JTable

Dalam Pembuatan JTabPane lakukan hal yang sama **new project** hingga **new frame** kemudian pilih **Design** lalu **drag & drop JtabPane** Seperti pada gambar berikut.

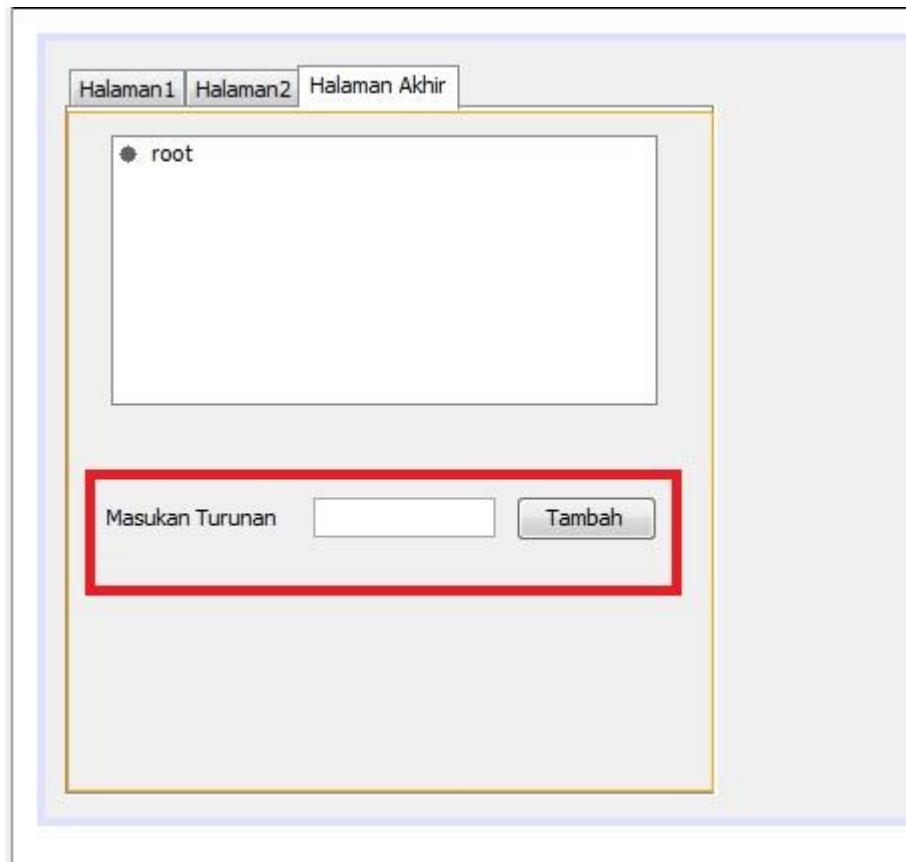


Setelah itu Drag & Drop **Panel** kedalam **JTabbedPane** maka Tab berhasil di buat.lakukan drag & drop **panel** hingga **JTabbedPane** memiliki 3 Tab. Setelah itu anda dapat mengedit Text pada Tab tersebut dengan cara yang sama yaitu klik kanan pada tab tersebut dan **Edit Text**. Ubahlah menjadi **Halaman1 , Halaman 2 dan Halaman Akhir**.

Pada pada halaman 1 anda dapat memberikan **JLabel** untuk menandai ini halaman1. Sedangkan pada halaman2 anda dapat memberikan **JTable** dengan cara darg & drop Jtabel ke Halaman2, dan pada halaman 3 anda dapat memberikan **JTree**. Kemudian anda dapat klik kanan pada **JTree** dan pilih **Property** selanjutnya klik pada Model kemudian klik titik2 pada bagian kanan model “....” Maka akan muncul jendela baru Dan hapus semua item di kolom sebelah kiri seperti gambar berikut.



Kemudian tambahkan 1 buah JLabel, 1 buah JTextField, dan 1 buah JButton. Seperti pada gambar berikut.

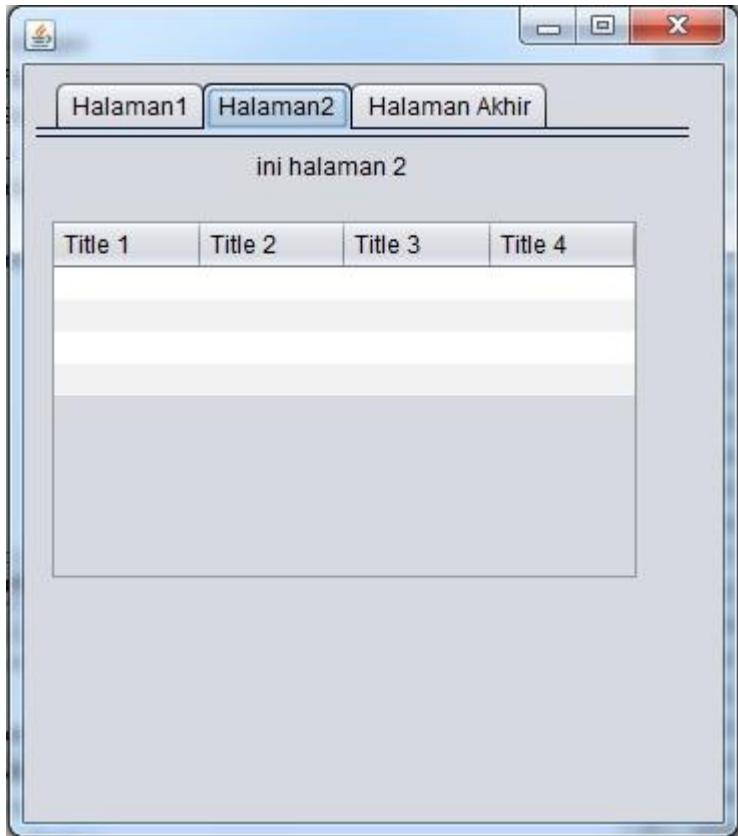


Kemudian double klik pada JButton atau klik kanan, kemudian **pilih Event | Action | actionPerformed**. Maka anda akan masuk pada bagian coding / source code.

Dan ketikan syntax berikut.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    DefaultMutableTreeNode admin = new DefaultMutableTreeNode(jTextField1.getText());  
    DefaultMutableTreeNode set = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();  
    DefaultTreeModel dt= (DefaultTreeModel) jTree1.getModel();  
    dt.insertNodeInto(admin, set, set.getChildCount());  
}
```

Setelah itu anda dapat Compile dan Run Project Anda. Maka Akan tampil sebagai berikut.

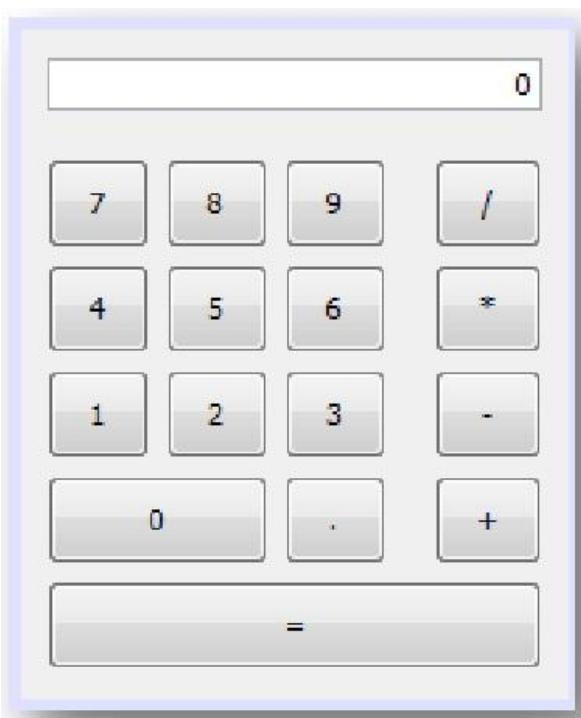


Pilih Halaman Akhir dan Klik pada **Jtree** ROOT. Dan ketikan anak pertama pada JTextField. Dan klik Tambah. Maka **Jtree** anda akan bertambah seperti pada gambar berikut.



TUGAS

Buatlah Sebuah Program yang mempunyai fungsi seperti kalkulator (mampu menjumlahkan,mengurangkan,mengalikan dan membagikan. Seperti berikut.



MODUL 8
JavaGUI , JDBC Connector & MySQL
(Pertemuan 9 dan 10)

Tujuan :

1. Mahasiswa dituntut untuk membuat form sederhana dengan memanfaatkan komponen Swing
2. Koneksi ke database Mysql dengan memanfaatkan JDBC Connector
3. Form yang telah terkoneksi dapat melakukan insert, update, delete

Tugas Pendahuluan :

1. Apa yang ketahui tentang JDBC Connector ?
2. Apa yang ada ketahui tentang Mysql?

DASAR TEORI

JDBC merupakan singkatan dari Java Database Connectivity yaitu API java yang membantu aplikasi java untuk mengeksekusi SQL statement. JDBC merupakan interface pemrograman aplikasi yang mendefinisikan bagaimana pemrogram java dapat mengakses database dalam format tabular dari kode2 java menggunakan sekumpulan interface standard an class-class yang tertulis dalam bahasa java.

Interface pemrograman aplikasi java menyediakan mekanisme untuk memuat package java beserta driver-driver dan memasang pada JDBC Driver Manager yang digunakan sebagai pembuat koneksi JDBC yang mendukung eksekusi syntax SQL seperti INSERT, UPDATE dan DELETE. Driver Manager adalah bagian itama dari JDBC.

Secara umum semua RDBMS (Relational Database Management System) dan Java merupakan platform yang independen, jadi JDBC memungkinkan untuk membuat sebuah aplikasi database yang dapat dijalankan pada platform-platform berbeda dan berinteraksi dengan DBMS.

Singkatnya, JDBC membantu programmer untuk membuat aplikasi java yang mengatur 3 aktivitas pemrograman ini :

1. Membantu menghubungkan dengan sumber data, seperti database
2. Membantu mengirim query dan perintah update ke database
3. Menerima dan memroses hasil yang diterima dari database sebagai respon dari query yang dikirim

KEGIATAN PRAKTIKUM

Kita akan mencoba membuat sebuah aplikasi sederhana menggunakan Java yang menggunakan sistem basis data **MySQL**. Programnya sederhana, hanya membuat sebuah Form dengan fasilitas CRUD (Create, Read, Update dan Delete).

Tabel yang akan kita buat sekarang adalah tabel MAHASISWA, dimana tabel tersebut

memiliki beberapa kolom, yaitu NIM, NAMA, TANGGAL_LAHIR, JURUSAN dan ALAMAT.

Membuat Database

Sebelum membuat program sederhana ini, hal yang pertama perlu kita buat adalah database yang akan kita gunakan. Misal kita akan membuat database UNIVERSITAS, maka kita bisa menggunakan perintah :

```
CREATE DATABASE UNIVERSITAS;
```

Membuat Tabel

Setelah membuat database, kita terlebih dahulu perlu membuat tabel MAHASISWA. Kita dapat menggunakan perintah sebagai berikut untuk membuat tabel MAHASISWA :

```
CREATE TABLE MAHASISWA(
    NIM VARCHAR(8) PRIMARY KEY,
    NAMA VARCHAR(50) NOT NULL,
    TANGGAL_LAHIR DATE NOT NULL,
    JURUSAN VARCHAR(50) NOT NULL,
    ALAMAT VARCHAR(500) NOT NULL
);
```

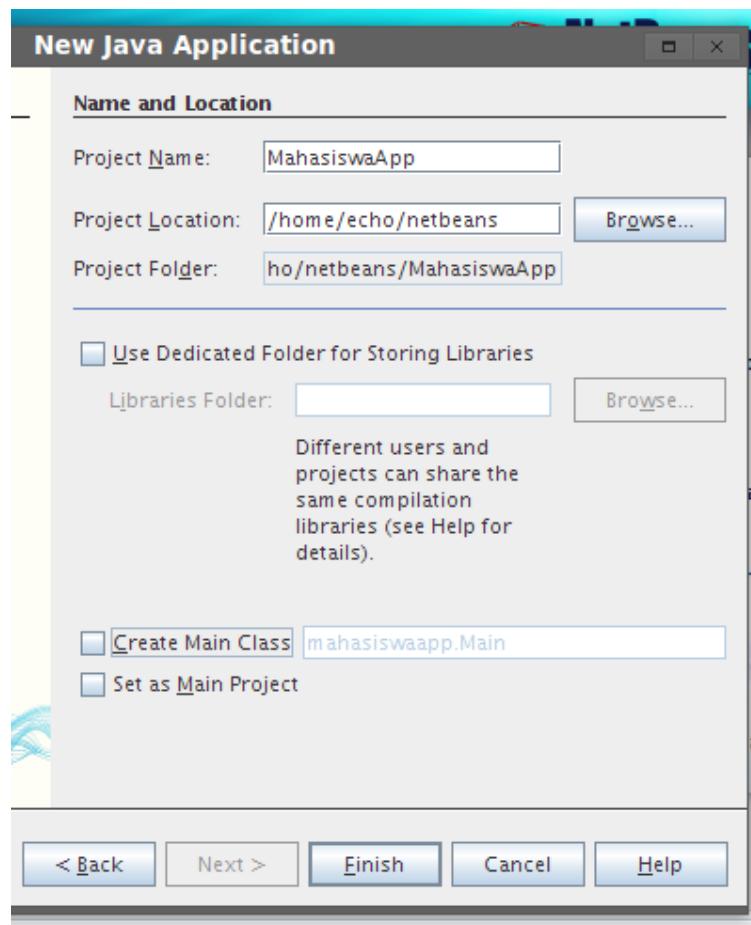
Sekarang kita sudah punya sebuah tabel dengan nama MAHASISWA. Saatnya kita lanjutkan membuat project Java-nya menggunakan NetBeans IDE.

Membuat Project

Sama seperti sebelumnya, untuk membuat sebuah project dalam NetBeans IDE kita dapat membuatnya menggunakan menu File -> New Project. Setelah itu pilih kategori Java dan pilih tipe project-nya Java Application.



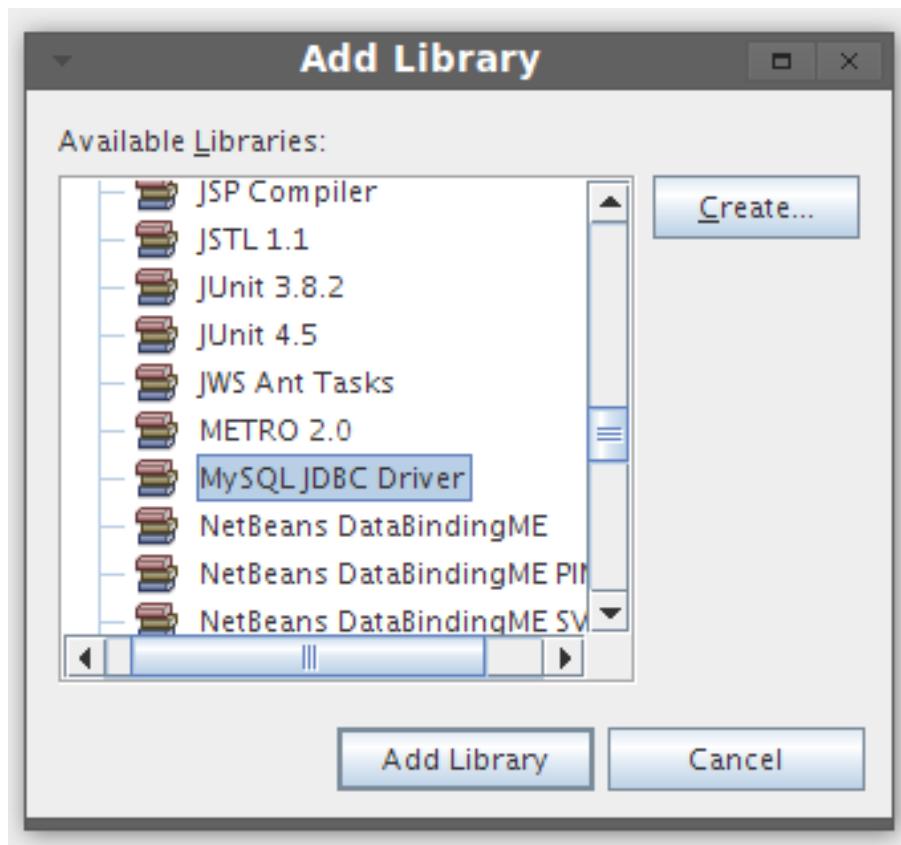
Klik Next > untuk melanjutkan pembuatan project.



Beri nama project dan jangan diceklis c eckbos Create Main Class. Hal ini dikarenakan kita tidak memerlukan dahulu membuat sebuah Main Class. Setelah itu klik tombol Finish, sekarang kita telah membuat project Java menggunakan NetBeans IDE.

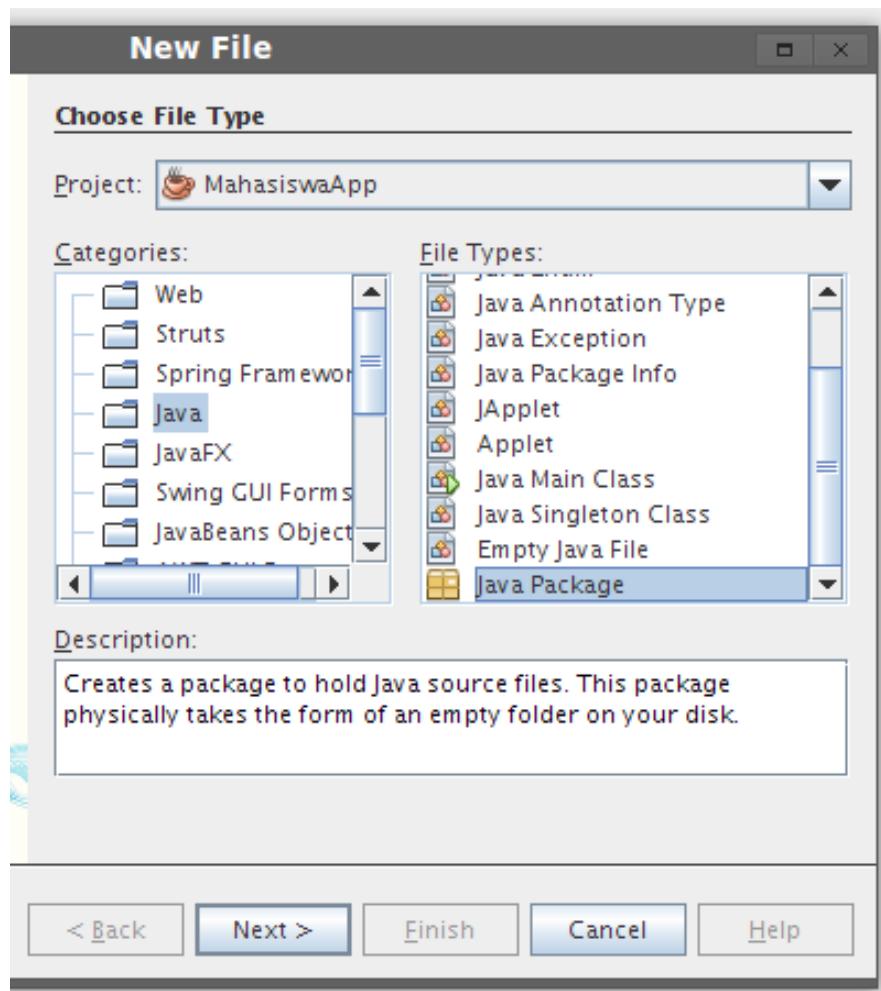
Membuat Koneksi MySQL

Setelah membuat project, saatnya membuat koneksi ke database UNIVERSITAS yang telah kita buat sebelumnya. Jadi hal yang pertama kita lakukan adalah menambah driver MySQL ke dalam project yang telah kita buat. Caranya klik kanan bagian Libraries project yang telah kita buat lalu pilih Add Library.

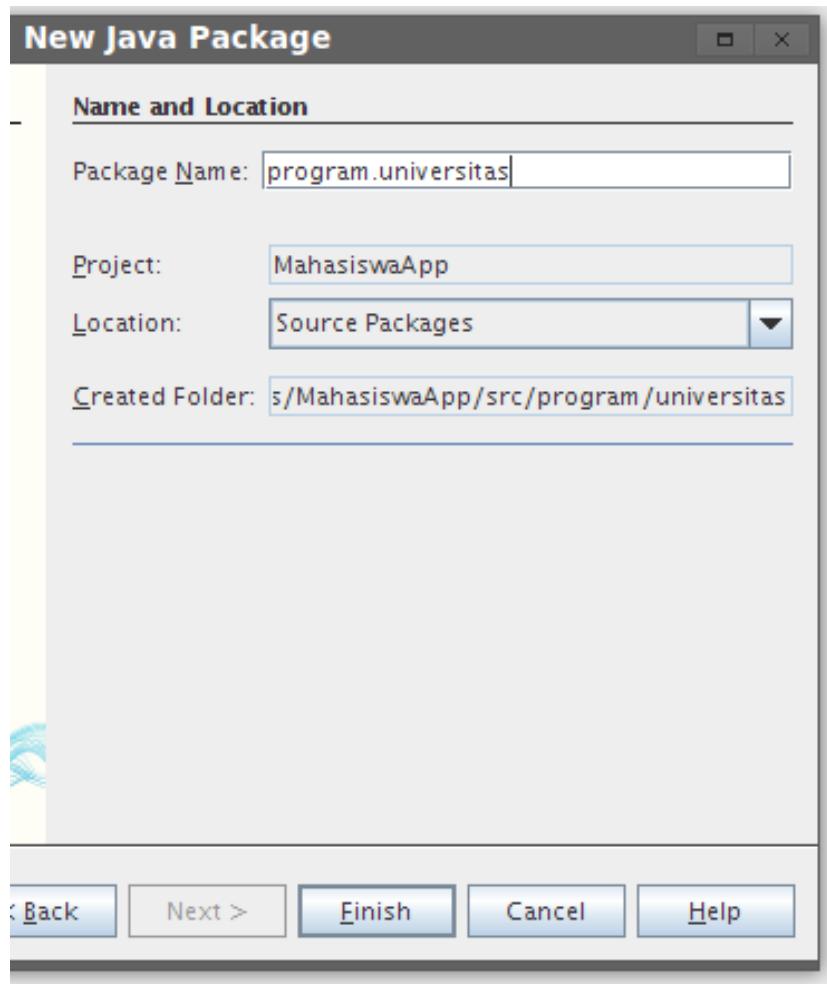


Setelah keluar dialog Add Library, pilih MySQL JDBC Driver lalu klik tombol Add Library untuk menambahkan MySQL Driver kedalam project.

Setelah menambahkan driver MySQL, sekarang saatnya membuat sebuah kelas yang akan kita gunakan untuk melakukan koneksi ke database MySQL. Tapi sebelum membuat sebuah kelas, pastikan kita membuat package dulu, caranya klik kanan bagian Source project yang telah kita buat lalu pilih Next → Other.

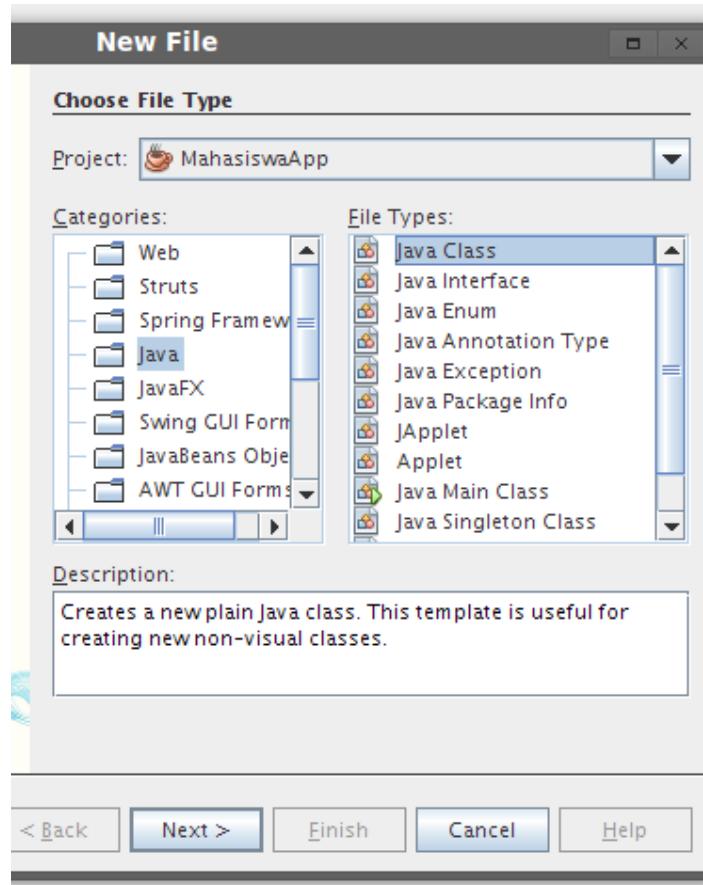


Setelah keluar dialog Next File, pilih kategori Java dan jenis file Java Package. Klik Next untuk melanjutkan membuat package.

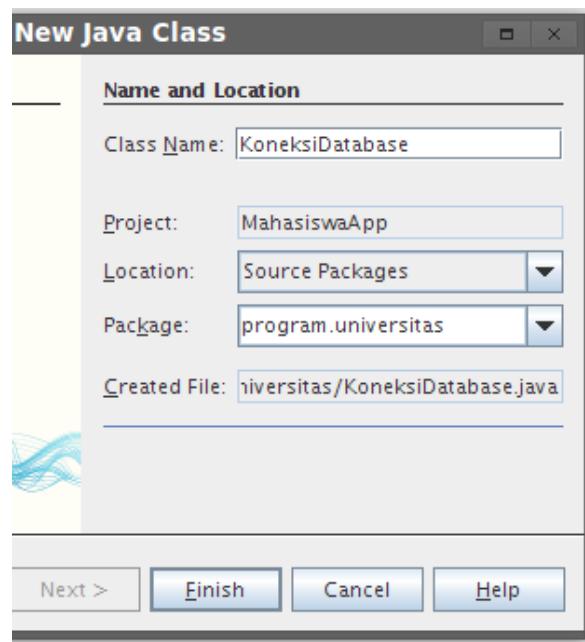


Setelah itu beri nama packagenya, misal **program.universitas**, setelah itu klik Finish untuk membuat package-nya.

Setelah membuat package **program.universitas**, sekarang kita buat sebuah kelas untuk melakukan koneksi ke MySQL. Caranya klik kanan package**program.universitas** lalu pilih Next -> Other.



Pilih kategori Java dan tipe filenya Java Class, setelah itu klik tombol Next untuk melanjutkan membuat sebuah kelas.



Beri nama kelas tersebut, misal KoneksiDatabase, setelah itu klik Finish agar kelas KoneksiDatabase terbuat.



Sekarang, saatnya melakukan proses pengkodean. Pertama buat sebuah variabel static yang bertipe java.sql.Connection, kita menggunakan static agar nanti aplikasi dapat mengakses koneksi secara langsung tanpa harus membuat object KoneksiDatabase.

```
package
program.universitas;
import
java.sql.Connection;
public class
KoneksiDatabase {
private static
Connection koneksi;
}
```

Setelah itu buat sebuah metode static getKoneksi(), metode ini digunakan untuk mendapatkan koneksi itu sendiri.

```
package  
program.universitas;  
import  
java.sql.Connection;  
public class  
KoneksiDatabase {  
    private static Connection koneksi;  
  
    public static Connection  
        getKoneksi() { return  
            koneksi;  
        }  
}
```

Sekarang untuk mengambil koneksi kita dapat langsung menggunakan petintah KoneksiDatabase.getKoneksi(), namun pastinya jika kita melakukan hal itu akan terjadi error, kok bisa?

Tentu karena pada kelas KoneksiDatabase tersebut kita belum membuat konesinya, jadi sebelum return koneksi, pada metode getKoneksi() seharusnya kita cek dulu apakah koneksi-nya null, jika null, maka kita deklarasikan sebuah koneksi yang baru.

```
package program.universitas;

import java.sql.Connection;
import
java.sql.DriverManager;
import java.sql.SQLException;

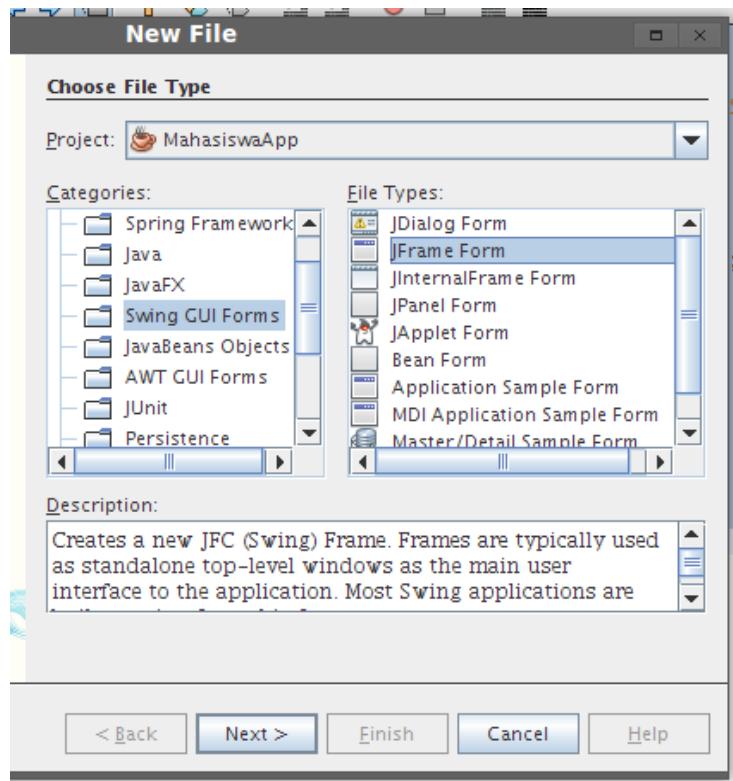
public class KoneksiDatabase {
    private static Connection koneksi;
    public static Connection getKoneksi()
    {
        // cek apakah koneksi
        null if (koneksi == null)
        {

            try {
                String url = "jdbc:mysql://khannedy.server:3306/UNIVERSITAS";
                String user = "echo";
                String password = "sssss";
                DriverManager.registerDriver(new
                com.mysql.jdbc.Driver()); koneksi =
                DriverManager.getConnection(url, user, password);
            } catch (SQLException t) {
                System.out.println("Error Membuat
                Koneksi");
            }
        }
        return koneksi;
    }
}
```

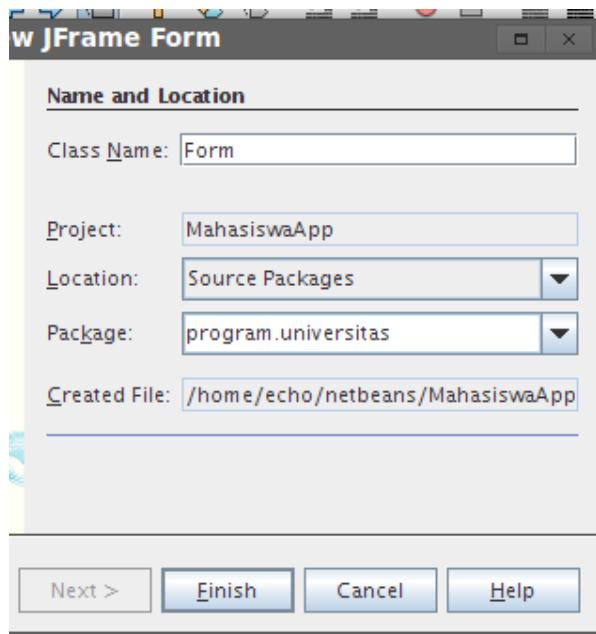
Sekarang, kita telah selesai membuat sebuah kelas untuk melakukan proses koneksi ke MySQL. Saatnya kita membuat Form aplikasinya.

Membuat Form Aplikasi

Pada program sederhana ini, kita tidak akan membuat program berbasis terminal (command line) lagi, tapi kita akan membuat aplikasi berbasis GUI. Dalam java teknologi untuk membuat program berbasis GUI disebut Java Swing. Sekarang untuk membuat sebuah Form, kita harus membuat JFrame, caranya dengan klik kanan package **program.universitas**, lalu pilih New -> Other.

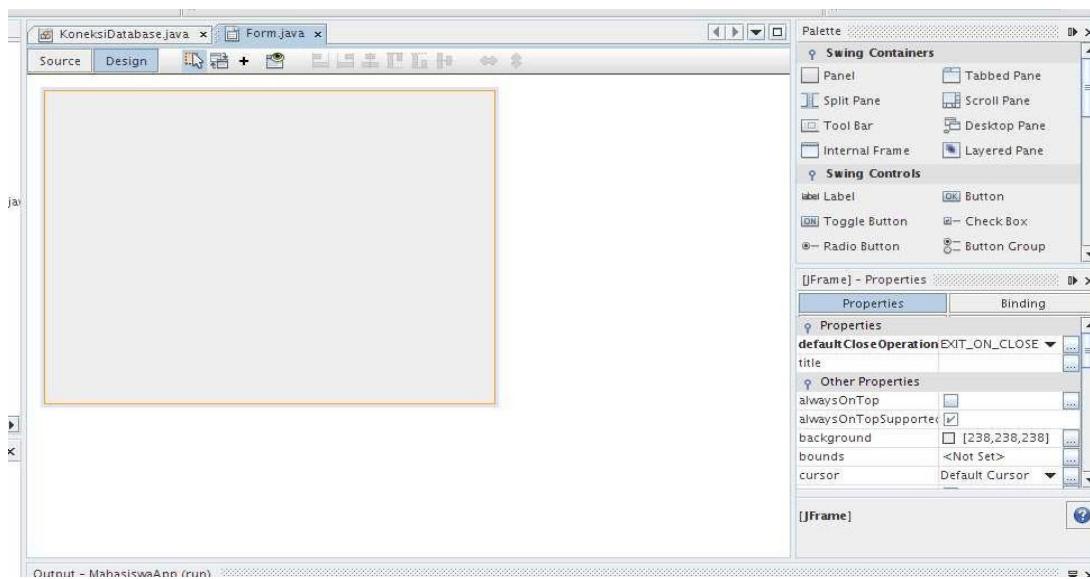


Pilih kategori String GUI Forms dan pilih tipe file JFrame Form. Lalu klik Nest untuk melanjutkan membuat Form.

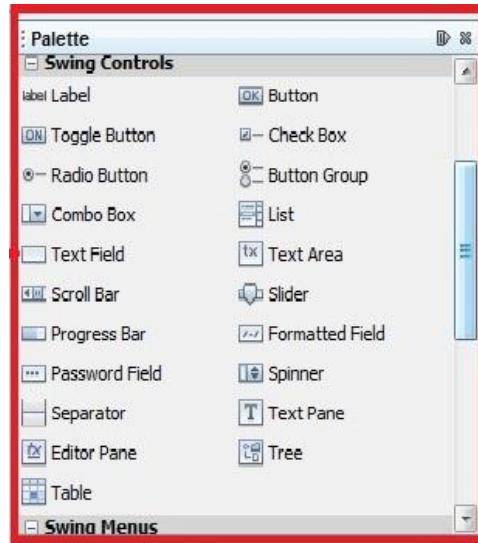


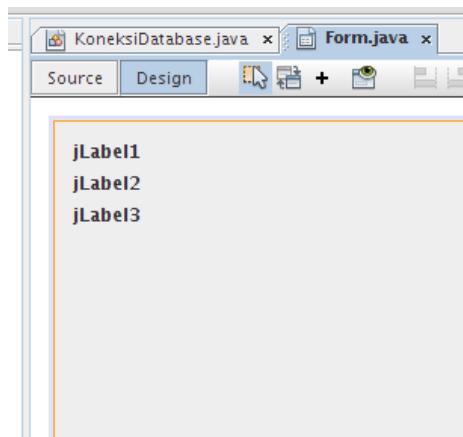
Beri nama Form tersebut, misal dengan nama Form, dengan begitu maka NetBeans akan membuatkan sebuah kelas dengan nama Form yang merupakan turunan dari kelas JFrame, dimana kelas JFrame ini merupakan kelas Java Swing.

Sekarang kita dapat melihat GUI builder pada editor NetBeans dan disebelah kanannya terdapat Pallete yang merupakan komponen-komponen GUI yang ada di Java dan Properties yang merupakan editor atribut-atribut komponen yang kita klik pada GUI Builder.

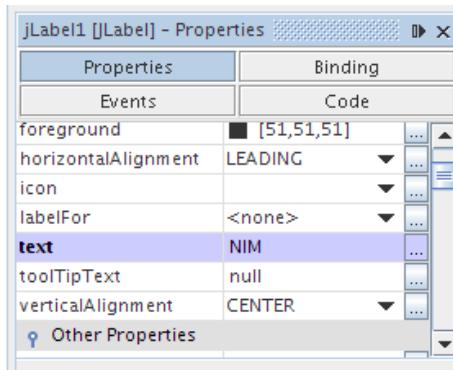


Untuk menambahkan komponen-komponen GUI lainnya, kita cukup mengklik dan mendrag salah satu komponen yang ada dalam Pallete ke dalam Form. Misal kita klik dan drag sebuah Label dari Pallete.

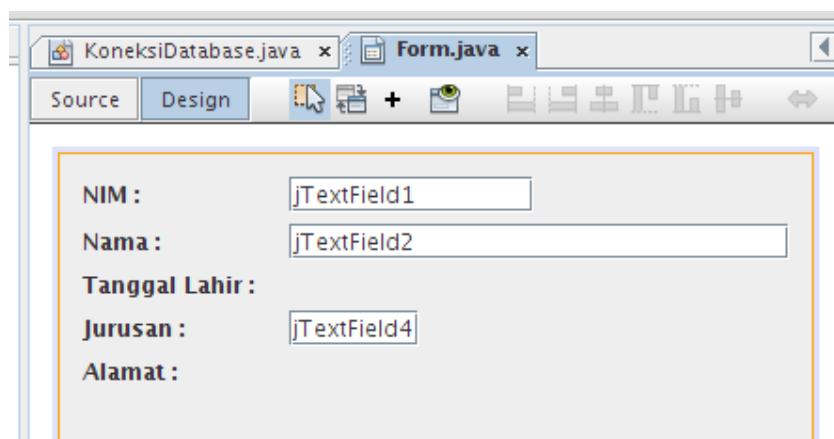




Untuk mengubah tulisan pada Label, kita dapat mengklik label tersebut, lalu lihat pada bagian Properties. Ubah atribut Text, misal menjadi NIM, Nama, Tanggal Lahir, Jurusan dan Alamat.

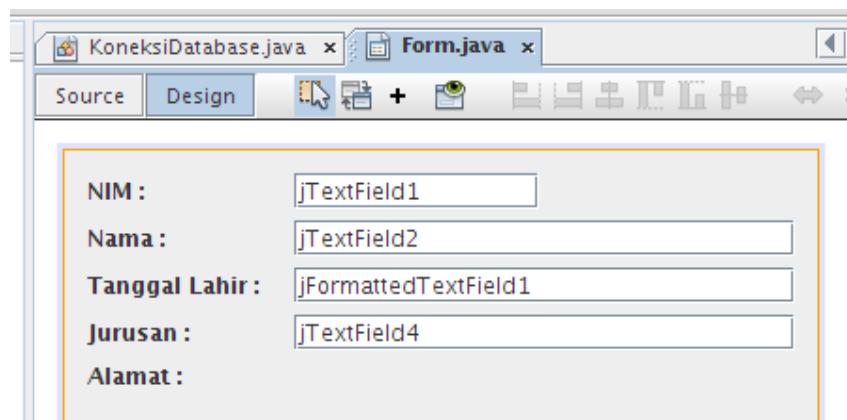


Setelah itu klik dan drag tiga buat Text Field yang ada dipallete ke Form, gunakan Text Field untuk Nim, Nama dan Jurusan.

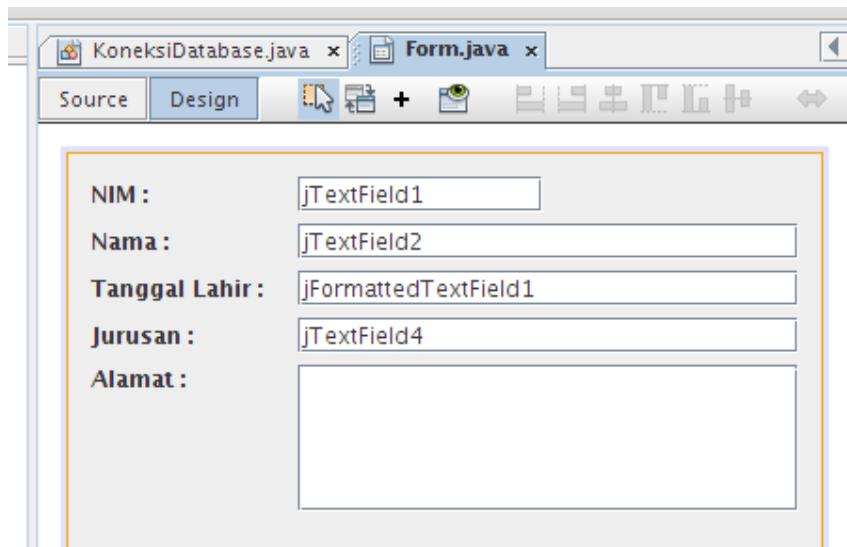


Untuk Tanggal Lahir dan Alamat kita tidak menggunakan Text Field, hal ini dikarenakan Tanggal Lahir memerlukan inputan berupa tanggal sedangkan Text Field hanya mendukung teks (string), sedangkan untuk Alamat, biasanya isi alamat itu panjang, sehingga lebih tidak cocok menggunakan Text Field, karena Text Field hanya mendukung satu baris.

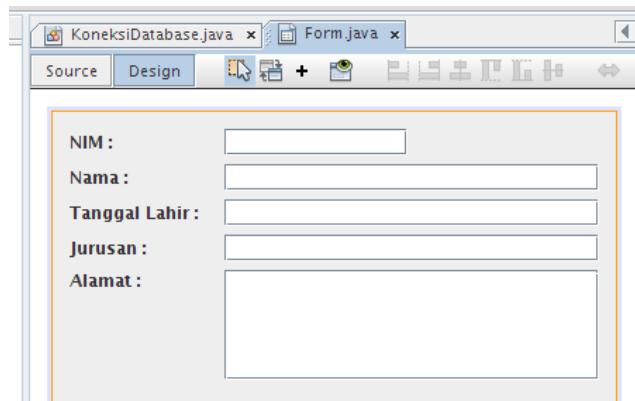
Dengan demikian, untuk Tanggal Lahir kita akan menggunakan Formatted Field, tinggal kita klik dan drag Formatted Field dari Palette ke dalam Form.



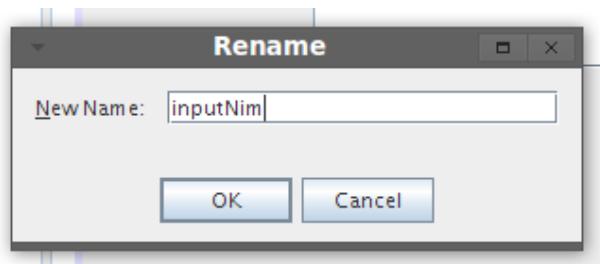
Dan untuk Alamat, gunakan komponen Text Area. Text Area hampir mirip dengan Text Field, namun mendukung lebih dari satu baris.



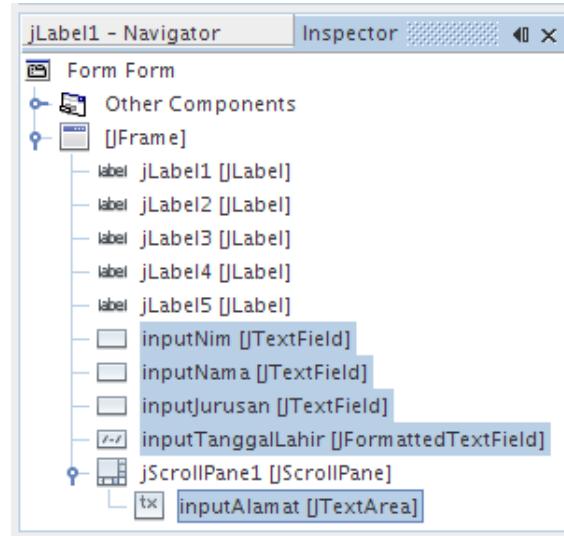
Untuk mengosongkan isi tulisan pada NIM, Nama, Tanggal Lahir dan Jurusan, kosongkan atribut Text pada setiap komponen pada Properties-nya.



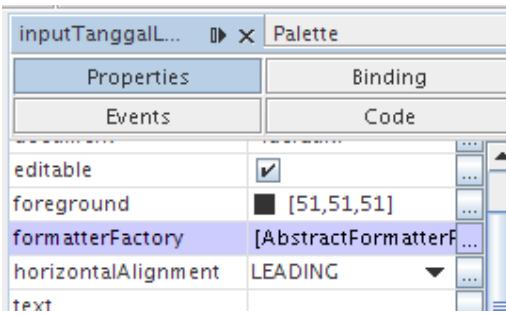
Setelah itu, sekarang saatnya kita mengubah setiap nama variabel komponennya, misal untuk Text Field NIM kita beri nama variabelnya dengan nama inputNim, untuk Text Field Nama dengan nama inputNama dan seterusnya, caranya dengan mengklik kanan komponennya lalu pilih menu **Change Variable Name**.



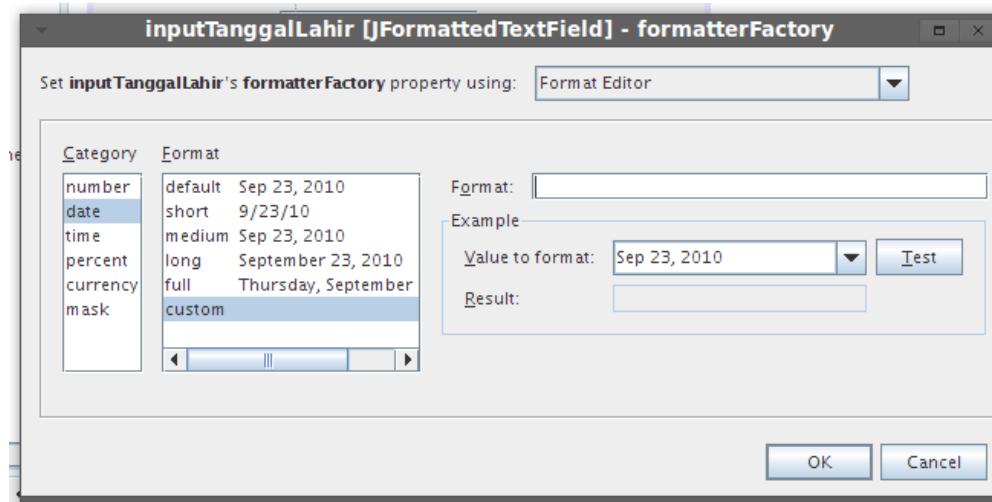
Untuk melihat seluruh nama variabelnya, kita dapat melihatnya pada bagian Inspector di sebelah kiri bawah Form NetBeans.



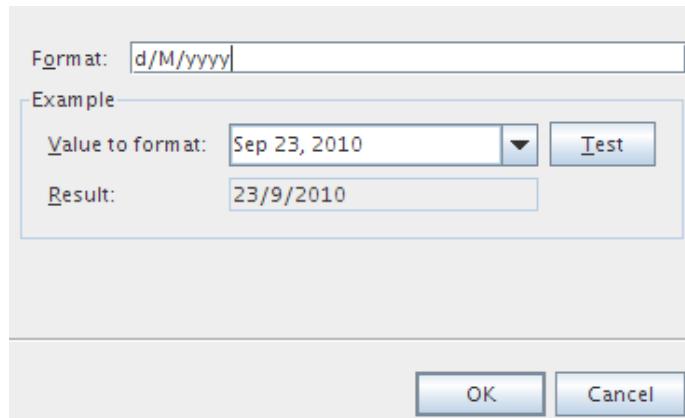
Secara default Formatted Field seperti Text Field, dia hanya menerima teks (String), agar Formatted Field hanya menerima input berupa tanggal, maka kita perlu memberitahukannya ke Formatted Field nya, caranya klik inputTanggalLahir, lalu pada bagian Properties, cari atribut formatterFactory, ubah atribut tersebut.



Pada saat mengklik tombol [...] pada atribut formatterFactory, maka akan muncul dialog formatterFactory.



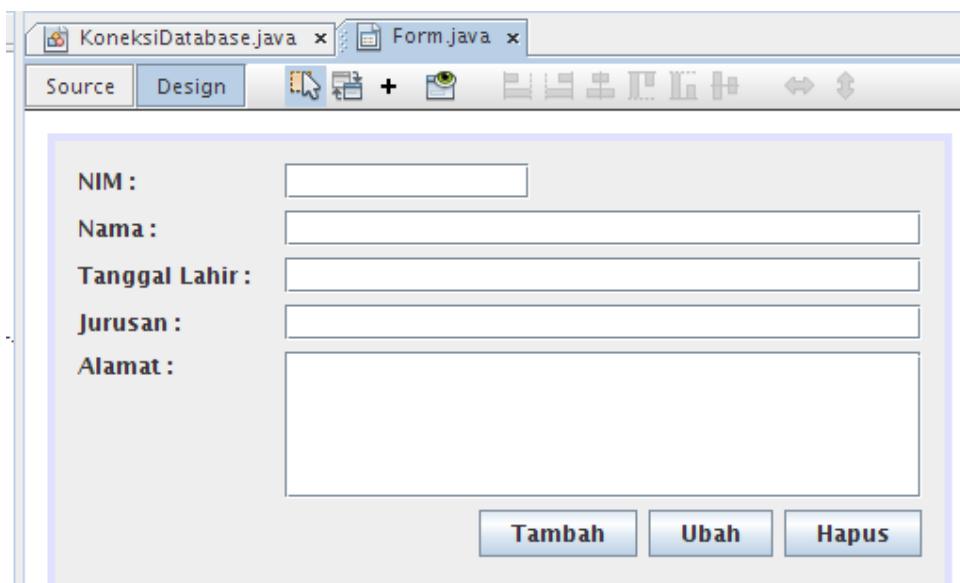
Agar Formatted Field hanya menerima input tanggal, maka ubah kategorinya menjadi date, formatnya menjadi custom, lalu pada input Format beri teks “d/M/yyyy”.



Maksud dari “d/M/yyyy” merupakan representasi tanggal/bulan/tahun dalam angka, misal jika tanggal 1 Januari 2010, maka input harus 1/1/2010 dan seterusnya. Klik tombol OK untuk mengkonfirmasi perubahan.

Menambah Tombol ke Form

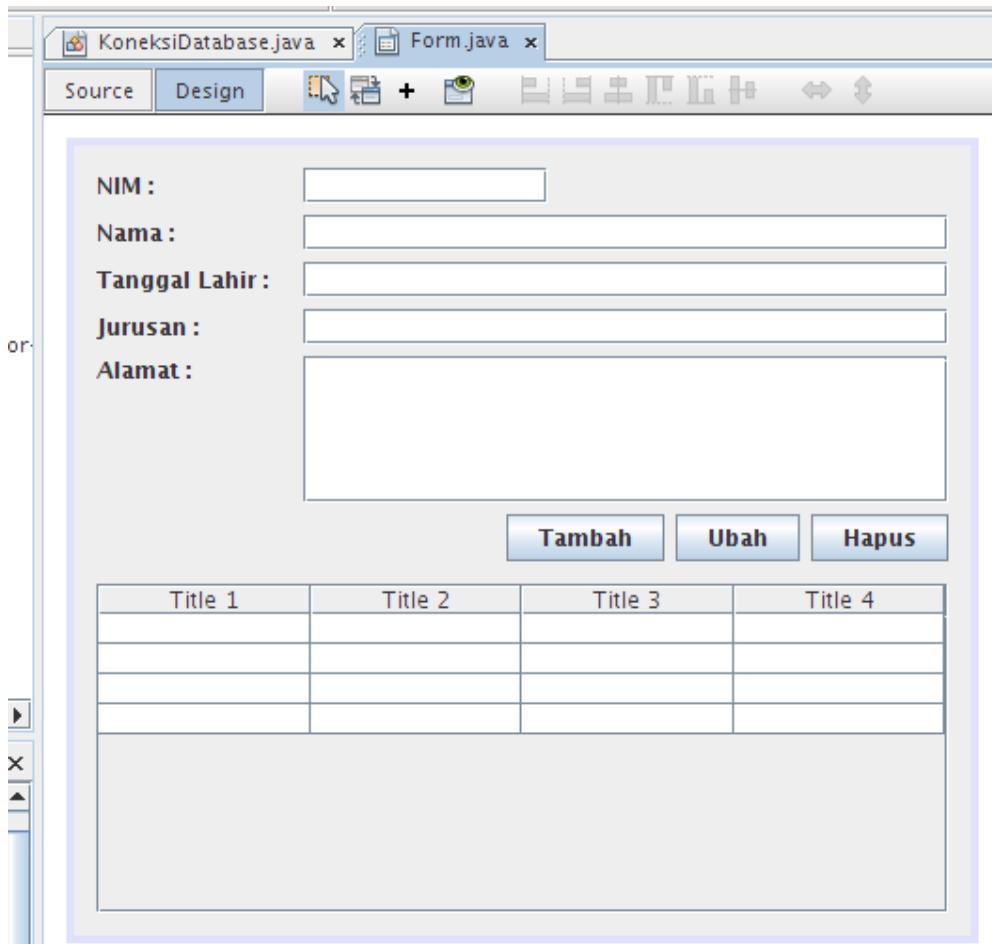
Setelah menambahkan input Form, sekarang saatnya kita menambahkan tombol ke dalam Form. Caranya dengan mengklik dan drag komponen Button pada Palette ke dalam Form.



Tambahkan 3 buah tombol, Tambah, Ubah dan Hapus. Untuk mengubah teks tombolnya caranya sama seperti Label, yaitu dengan mengubah atribut Text pada Properties.

Menambah Tabel ke Form

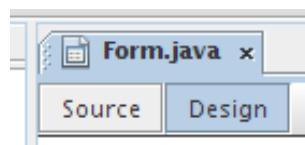
Setelah menambahkan input Form beserta tombolnya, sekarang saatnya menambahkan Tabel ke Form, caranya tinggal kita klik dan drag komponen Table dari Palette ke Form, hasilnya seperti terlihat pada gambar dibawah ini.



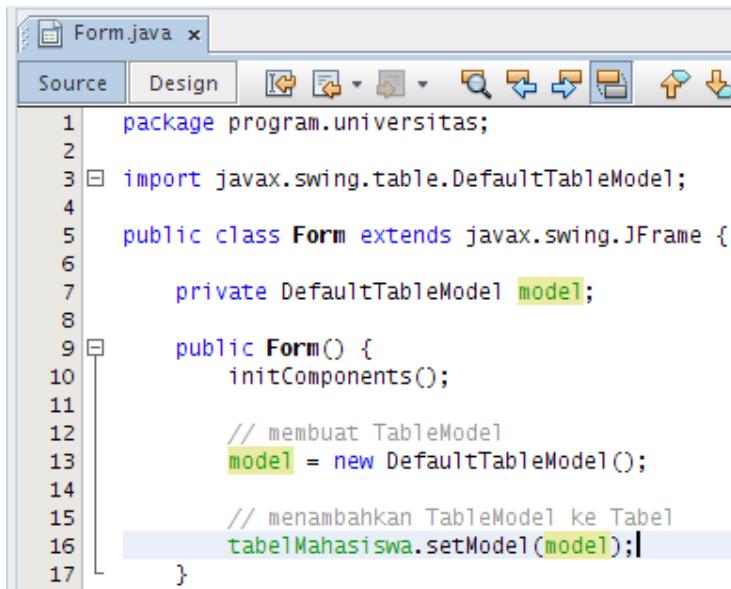
Jangan lupa untuk mengubah nama variabel Tabel yang tadi kita masukkan ke Form, caranya klik kanan Tabel nya lalu pilih Change Variabel Name, misal dengan nama tabelMahasiswa.



Sekarang saatnya mengubah kolom pada Tabel. Berbeda dengan komponen lain, untuk mengubah kolom pada komponen Tabel, kita memerlukan kelas lain, namanya kelas DefaultTableModel, sehingga kita perlu melakukan pengkodean, caranya masuk ke bagian Source.



Setelah itu tambahkan sebuah variabel DefaultTableModel pada kelas Form tersebut.



```
Form.java x
Source Design
1 package program.universitas;
2
3 import javax.swing.table.DefaultTableModel;
4
5 public class Form extends javax.swing.JFrame {
6
7     private DefaultTableModel model;
8
9     public Form() {
10         initComponents();
11
12         // membuat TableModel
13         model = new DefaultTableModel();
14
15         // menambahkan TableModel ke Tabel
16         tabelMahasiswa.setModel(model);
17     }
}
```

Untuk menambahkan kolom ke Tabel, maka kita dapat menggunakan metode addColumn(nama) milik kelas DefaultTableModel. Dan saat ini kita perlu menambahkan kolom Nim, Nama, Tanggal Lahir, Jurusan dan Alamat.



```
14
15
16
17
18     // menambahkan TableModel ke Tabel
19     tabelMahasiswa.setModel(model);
20
21     model.addColumn("Nim");
22     model.addColumn("Nama");
23     model.addColumn("Tanggal Lahir");
24     model.addColumn("Jurusan");
25     model.addColumn("Alamat");
26 }
```

Menambahkan Aksi

Sekarang kita telah selesai membuat Form, saatnya kita menambahkan aksi-aksi database, seperti load data dari database, menambah data ke database, mengubah data dari database dan menghapus data dari database.

Menambah Aksi Load Data

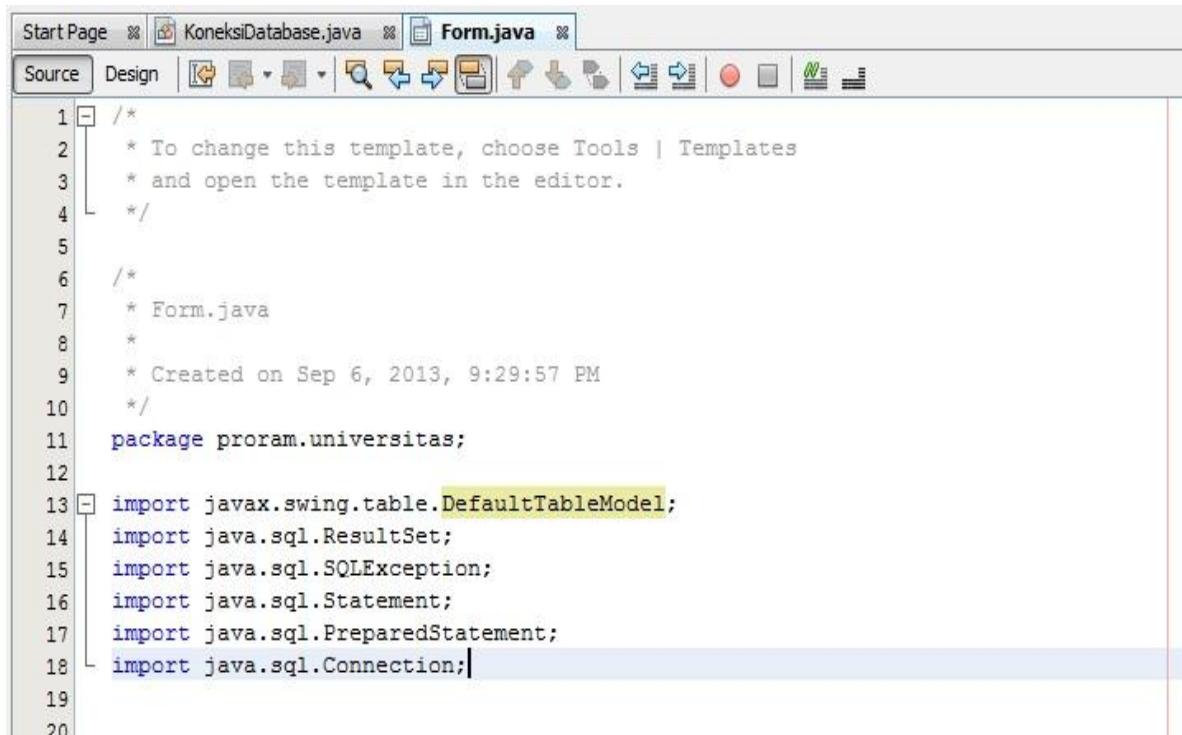
Saat pertama kali aplikasi muncul, maka otomatis kita harus mengambil seluruh data mahasiswa yang ada dalam tabel MAHASISWA dan ditampilkan ke dalam Table yang ada di Form. Dengan demikian, maka pertama kita perlu membuat sebuah aksi melakukan load data dari database.

Namun sebelum anda membuat method tersebut tambahkan beberapa import dibawah ini.

Tambahkan tepat di bawah coding package proram.universitas;

```
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.Connection;
```

import diatas merupakan bawaan dari java sehingga syntax yang akan anda buat tidak mengalami eror / user define.



```
Start Page ✘ KoneksiDatabase.java ✘ Form.java ✘
Source Design | 
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  /*
7   * Form.java
8   *
9   * Created on Sep 6, 2013, 9:29:57 PM
10  */
11 package proram.universitas;
12
13 import javax.swing.table.DefaultTableModel;
14 import java.sql.ResultSet;
15 import java.sql.SQLException;
16 import java.sql.Statement;
17 import java.sql.PreparedStatement;
18 import java.sql.Connection;
```

Sekarang kita buat sebuah metode dengan nama loadData() dimana metode tersebut dibuat dalam kelas Form dan dalam metode tersebut berisikan proses load data dari database.

```
public void loadData() {
```

```
}
```

Sebelum melakukan proses load data dari database, maka pertama kali, kita perlu menghapus seluruh isi baris yang ada pada Table yang ada di Form. Hal ini perlu dilakukan agar saat kita akan melakukan load ulang data, maka Tabel dikosongkan dulu. Untuk mengosongkan isi Table, kita harus menggunakan DefaultTableModel.

```
public void loadData() {  
  
    // menghapus seluruh data  
    model.getDataVector().removeAllElements();  
  
    // memberi tahu bahwa data telah kosong  
    model.fireTableDataChanged();  
  
}
```

Setelah itu baru kita melakukan load datanya, untuk mengambil data dari database, kita memerlukan Connection yang ada dalam kelas KoneksiDatabase. Setelah itu buat Statement dan ResultSet seperti biasanya.

```
public void loadData(){  
  
    // menghapus seluruh data  
  
    model.getDataVector().removeAllElements();  
  
    // memberi tahu bahwa data telah kosong  
  
    model.fireTableDataChanged();  
  
    try{  
  
        Connection c = KoneksiDatabase.getKoneksi();  
        Statement s = c.createStatement();  
  
        String sql = "SELECT * FROM MAHASISWA";  
  
        ResultSet r = s.executeQuery(sql);  
  
        while(r.next()){  
            // lakukan penelusuran baris  
        }  
  
        r.close();  
  
        s.close();  
  
    }catch(SQLException e){  
        System.out.println("Terjadi Error");  
    }  
}
```

Pada saat melakukan proses penelusuran data menggunakan ResultSet, maka kita dapat menambahkan data tersebut ke dalam Table yang ada dalam Form. Untuk menambah sebuah baris ke Table kita menambahkannya ke DefaultTableModel dengan menggunakan metode addRow(Object[]).

```
while(r.next()){

    // lakukan penelusuran baris
    Object[] o = new Object[5];
    o[0] = r.getString("NIM");

    o[0] = r.getString("NAMA");

    o[0] = r.getDate("TANGGAL LAHIR");

    o[0] = r.getString("JURUSAN");

    o[0] = r.getString("ALAMAT");

    model.addRow(o);

}
```

Lengkapnya metode loadData() akan berisi seperti pada kode dibawah ini.

```
public void loadData(){

    // menghapus seluruh data
    model.getDataVector().removeAllElements();

    // memberi tahu bahwa data telah kosong
    model.fireTableDataChanged();

    try{

        Connection c = KoneksiDatabase.getKoneksi();
        Statement s = c.createStatement();

        String sql = "SELECT * FROM MAHASISWA";

        ResultSet r = s.executeQuery(sql);

        while(r.next()){

            // lakukan penelusuran baris

            Object[] o = new Object[5];
            o[0] = r.getString("NIM");

            o[1] = r.getString("NAMA");

            o[2] = r.getDate("TANGGAL LAHIR");

            o[3] = r.getString("JURUSAN");
            o[4] = r.getString("ALAMAT");
        }
    }
}
```

```

        model.addRow(o);

    }

    r.close();

    s.close();

} catch (SQLException e){
    System.out.println("Terjadi Error");

}

}

```

Agar metode loadData() dipanggil ketika program berjalan, maka kita perlu memanggil metode loadData() dalam konstruktor Form.

```

24   model.addColumn("Nim");
25   model.addColumn("Jurusan");
26   model.addColumn("Alamat");
27
28   // panggil loadData()
29   loadData();
30 }
31
32 public void loadData() {
33     // menghapus seluruh dat
34     model.getDataVector().re
35     // memberi tahu bahwa da
36     model.fireTableDataChang
--
```

Menambah Aksi Tombol Tambah

Sekarang saatnya kita menambahkan aksi tombol, dimana aksi tombol itu akan berjalan ketika tombol Tambah diklik. Untuk menambah sebuah aksi ke tombol Tambah, pertama kita masuk lagi ke bagian Design, setelah itu tinggal klik kanan tombol Tambah-nya setelah itu pilih menu **Events -> Action -> actionPerformed** ed, maka otomatis NetBeans IDE akan membuatkan sebuah metode baru untuk aksi tombol Tambah.

```

197
198 private void jButton3ActionPerformed(java.awt.event
199     // TODO add your handling code here:
200 }
201

```

Dalam metode tersebutlah kita melakukan proses penambahan data ke dalam database. Untuk menambahkan data ke dalam tabel MAHASIWA, otomatis kita memerlukan data input dari pengguna. Untuk mendapatkan data tulisan dari Text Field dan Text Area, maka kita dapat menggunakan metode getText(), sedangkan untuk mendapatkan tanggal dari Formatted Field, kita dapat menggunakan metode getValue(), namun dikarenakan getValue() menghasilkan

Object, maka kita perlu mengkonversinya ke tanggal.

```
String nim = inputNim.getText(); String nama = inputNama.getText();
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();
String alamat = inputAlamat.getText();
```

Setelah mengambil seluruh data dari input, maka baru kita menyimpannya ke dalam database MySQL. Caranya adalah dengan membuat Connection dari kelas KoneksiDatabase setelah itu membuat PreparedStatement untuk menyimpan datanya.

```
String nim = inputNim.getText(); String nama = inputNama.getText();

java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();
String alamat = inputAlamat.getText();

try{
    Connection c = KoneksiDatabase.getKoneksi();
    String sql = "INSERT INTO MAHASISWA VALUES (?, ?, ?, ?, ?, ?)";
    PreparedStatement p = c.prepareStatement(sql);
    p.setString(1, nim);
    p.setString(2, nama);
    p.setDate(3, new java.sql.Date(tanggalLahir.getTime()));
    p.setString(4, jurusan);
    p.setString(5, alamat);
    p.executeUpdate();
    p.close();
} catch (SQLException e) {
    System.out.println("Terjadi Error");
} finally {
    loadData();
}
```

Pada blok finally, kita perlu memanggil metode loadData(), hal ini dilakukan agar setelah proses penyimpanan data ke database, maka data akan dimuat ulang ke Table yang ada di Form.

Menambah Aksi Tombol Ubah

Untuk aksi tombol Ubah, agak sedikit berbeda dengan aksi tombol Tambah, perbedaannya adalah pertama kita harus mendeteksi baris yang sedang diklik, setelah itu baru melakukan proses pengubahan data yang diklik dengan data baru yang ada dalam input Form.

Untuk menambah aksi tombol Ubah caranya sama dengan tombol Tambah, tinggal klik kanan tombol Ubah lalu pilih **Events -> Action -> actionPerformed**.



Seperti yang telah ditulis sebelumnya, pertama kita harus mendapatkan baris yang terseleksi pada Table, jika tidak ada baris yang terseleksi, maka proses Ubah dibatalkan. Untuk mendapatkan baris yang terseleksi kita dapat menggunakan metode getSelectedRow() milik Table, jika return-nya -1 artinya tidak ada baris yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();
if(i == -1) {
    // tidak ada baris terseleksi
    return;
}

// ambil nim yang terseleksi
String nim = (String) model.getValueAt(i, 0);
String nama = inputNama.getText();
java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();
String alamat = inputAlamat.getText();
```

Setelah mengambil data nim yang terseleksi dan data lainnya dari input, baru kita lakukan proses ubah data yang ada di database berdasarkan nim yang baris yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();
if(i == -1) {
    // tidak ada baris terseleksi
    return;
}
// ambil nim yang terseleksi
String nim = (String) model.getValueAt(i, 0);
String nama = inputNama.getText();

java.util.Date tanggalLahir = (java.util.Date) inputTanggalLahir.getValue();
String jurusan = inputJurusan.getText();

String alamat = inputAlamat.getText();

try{
    Connection c = KoneksiDatabase.getKoneksi();
```

```

String sql = "UPDATE MAHASISWA SET NAMA = ?, TANGGAL_LAHIR = ?, JURUSAN = ?, ALAMAT
= ? WHERE NIM = ?";

PreparedStatement p = c.prepareStatement(sql);

p.setString(1, nama);

p.setDate(2, new java.sql.Date(tanggalLahir.getTime()));

p.setString(3, jurusan);
p.setString(4, alamat);
p.setString(5, nim);

p.executeUpdate();
p.close();

}catch(SQLException e){

    System.out.println("Terjadi Error");

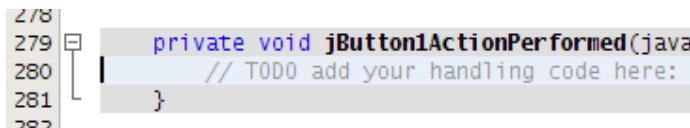
}finally{

    loadData();

}
    
```

Menambah Aksi Tombol Hapus

Untuk aksi hapus, kita tidak perlu menggunakan input Form, yang kita perlukan hanyalah baris yang terseleksi. Jika baris tidak ada yang terseleksi, maka proses penghapusan dibatalkan. Untuk menambah aksi pada tombol Hapus caranya sama seperti tombol Tambah dan Ubah, klik kanan tombol Hapus, lalu pilih menu **Events -> Action -> actionPerformed**.



Setelah itu sama seperti pada proses Ubah, kita cek dulu apakah ada baris yang terseleksi atau tidak, jika ada ambil nim yang terseleksi, jika tidak ada, maka batalkan proses Hapus.

```

int i = tabelMahasiswa.getSelectedRow();
if(i == -1){

    // tidak ada baris terseleksi
    return;

}

String nim =(String) model.getValueAt(i, 0);
    
```

Setelah itu, baru kita lakukan proses penghapusan data dari database berdasarkan data baris yang terseleksi.

```
int i = tabelMahasiswa.getSelectedRow();
if(i == -1){
    // tidak ada baris terseleksi
    return;
}
String nim = (String) model.getValueAt(i, 0);
try{
    Connection c = KoneksiDatabase.getKoneksi();

    String sql = "DELETE FROM MAHASISWA WHERE NIM = ?";
    PreparedStatement p = c.prepareStatement(sql);
    p.setString(1, nim);
    p.executeUpdate();
    p.close();
} catch(SQLException e){
    System.err.println("Terjadi Error");
} finally{
    loadData();
}
```

Menambahkan Aksi Baris Terseleksi

Aksi terakhir yang perlu kita tambahkan adalah aksi ketika baris Table terseleksi, misal jika baris pertama terseleksi, maka program akan menampilkan data yang terseleksi tersebut pada Form. Hal ini agar pengubahan lebih mudah, karena kita tidak perlu memasukkan seluruh datanya lagi.

Untuk menambahkan aksi ketika baris terseleksi, kita dapat menggunakan aksi Mouse Click, yaitu aksi yang dijalankan ketika mouse mengklik. Caranya, klik kanan komponen Table pada Form, setelah itu pilih **Events -> mouse -> MouseClicked**. Sekarang akan terbuat sebuah metode baru yang akan dipanggil ketika Table diklik.



312
313 private void tabelMahasiswaMouseClicked(java.awt.event.MouseEvent evt) {
314 // TODO add your handling code here:
315 }
316

Pertama yang harus dilakukan adalah melakukan pengecekan apakah ada baris yang terseleksi, jika ada maka ambil data yang terseleksi dari DefaultTableModel setelah itu tampilkan pada Form, namun jika tidak ada baris yang terseleksi, maka batalkan proses.

```

int i = tabelMahasiswa.getSelectedRow();
if(i == -1) {
    // tak ada baris terseleksi
    return;
}
String nim = (String) model.getValueAt(i, 0);
inputNim.setText(nim);
String nama = (String) model.getValueAt(i, 1);
inputNama.setText(nama);

java.util.Date tanggalLahir = (java.util.Date) model.getValueAt(i, 2);
inputTanggalLahir.setValue(tanggalLahir);

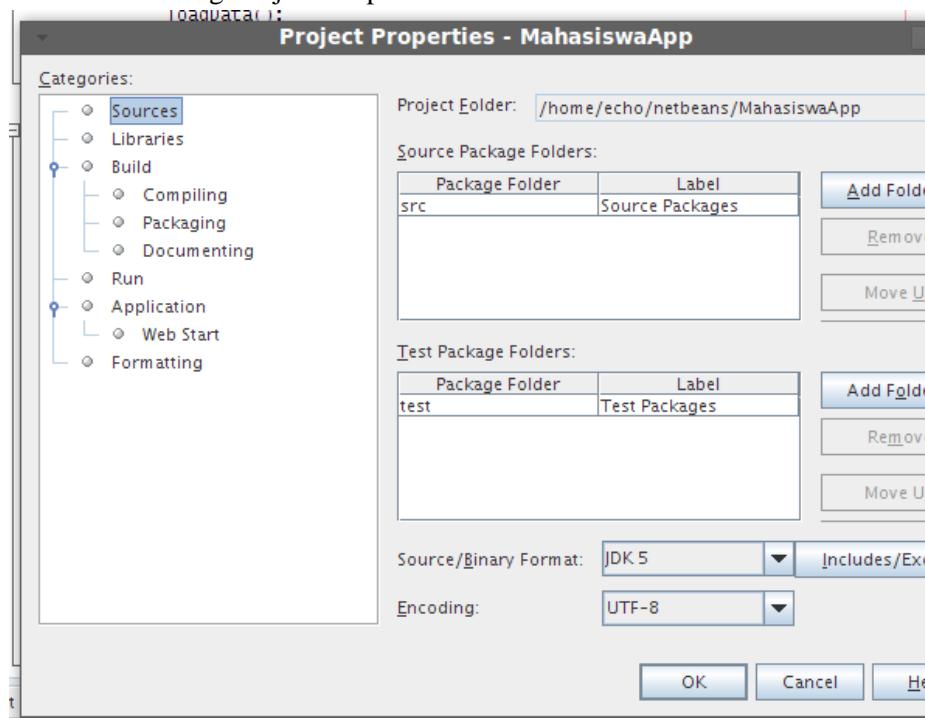
String jurusan = (String) model.getValueAt(i, 3);
inputJurusan.setText(jurusan);
String alamat = (String) model.getValueAt(i, 4);
inputAlamat.setText(alamat);

```

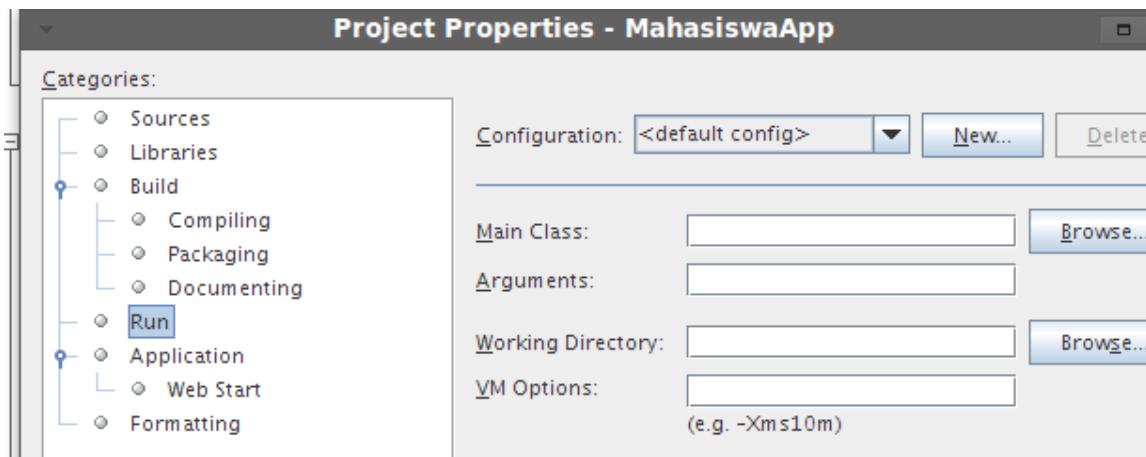
Run Program

Setelah pembuatan program telah selesai, saatnya menjalankan aplikasi. Untuk menjalankan aplikasi, pertama kita harus menentukan dahulu kelas yang akan digunakan sebagai program, dimana pada project yang telah kita buat, kelas program adalah kelas Form.

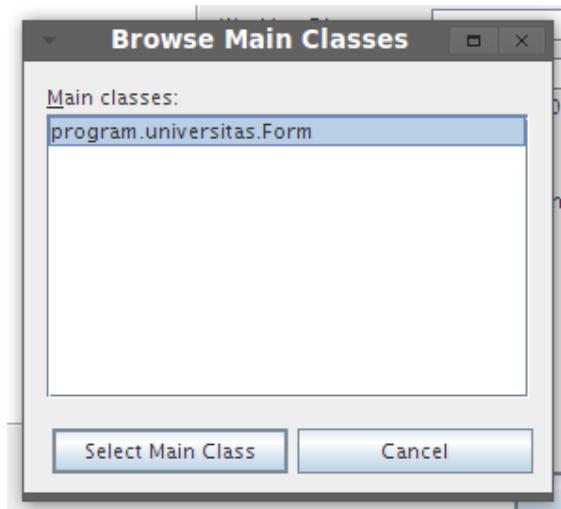
Untuk mengubah kelas program menjadi kelas Form, maka kita dapat mengubahnya dengan cara klik kanan Project yang telah kita buat, lalu pilih menu **Properties**, setelah itu akan keluar dialog Project Properties.



Pada bagian Categories, pilihlah menu Run, untuk mengubah kelas yang akan dijalankan sebagai kelas Program.



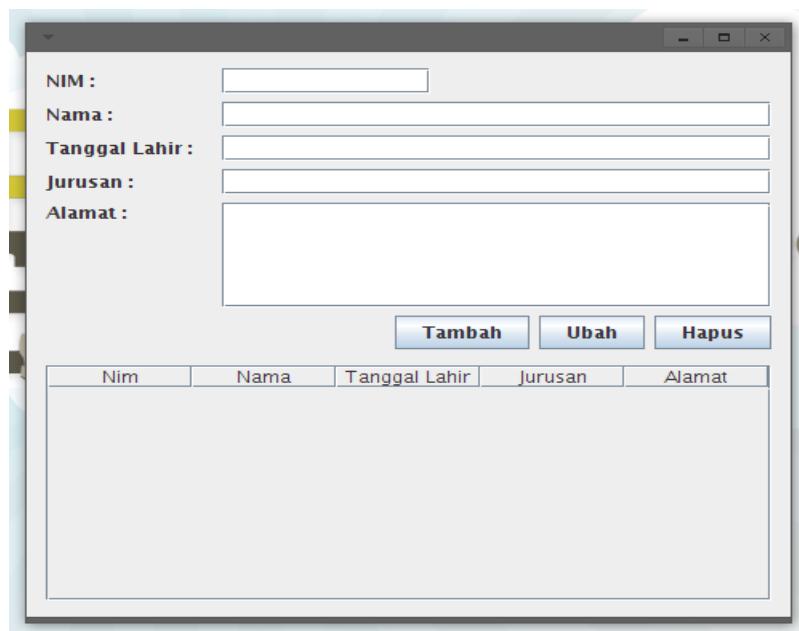
Pada input Main Class, klik tombol Browse, untuk menampilkan daftar kelas yang dapat dijadikan sebagai kelas program. Maka akan keluar dialog pemilihan kelas.



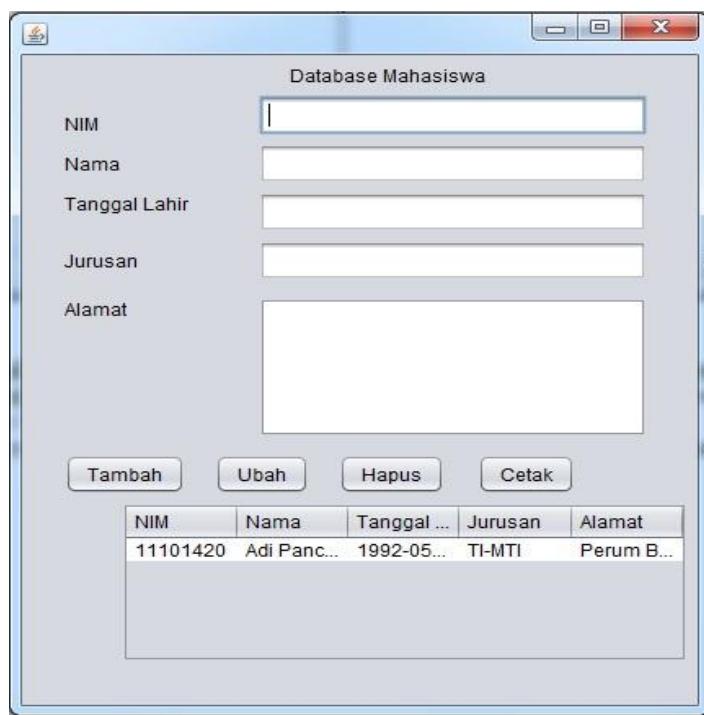
Pilih kelas Form yang telah dibuat tadi, setelah itu klik tombol Select Main Class. Untuk mengkonfirmasi perubahan, klik tombol OK. Sekarang kelas Form akan menjadi kelas yang dijalankan jika Project dijalankan.

Untuk menjalankan aplikasi, klik kanan project yang telah kita buat, setelah itu pilih menu

Run, maka otomatis program akan berjalan. Pada komputer saya, program terlihat seperti pada gambar dibawah ini.



Sekarang, kita dapat menambah data, mengubah data dan menghapus data yang telah kita masukkan.



TUGAS

Implementasikan materi yang telah didapat dalam project sistem informasi yang telah dirancang!

MODUL 9
Java GUI Database Dengan iReport
(Pertemuan 11, 12 dan 13)

Tujuan :

1. Mahasiswa bisa dituntun untuk membuat single form yang dilengkapi dengan report
2. Form yang telah terkoneksi dapat melakukan insert update delete

Tugas Pendahuluan :

Apa yang anda ketahui tentang iReport?

DASAR TEORI

Terdapat banyak tools untuk reporting dalam java. Diantaranya yang dapat digunakan adalah :

1. JasperReports
Merupakan software open source untuk reporting
2. iReport
Merupakan Visual Designer untuk membuat laporan yang kompleks menggunakan JasperReports library tanpa harus memiliki pengetahuan tentang XML .

Beberapa fitur iReport :

- 98% mendukung JasperReports tags
- Visual designer wysiwyg untuk menggambar rectangles, lines, ellipses, text fields fields, charts, sub reports...
- Built-in editor dengan syntax highlighting
- Mendukung Unicode dan bahasa non Latin (Russian, Chinese, Korean,...)
- Document structure browser
- Menggabungkan compiler dan exporter
- Mendukung semua JDBC compliant databases
- Memiliki Wizard untuk membuat report secara otomatis
- Mendukung sub reports
- Save backup
- Support for templates

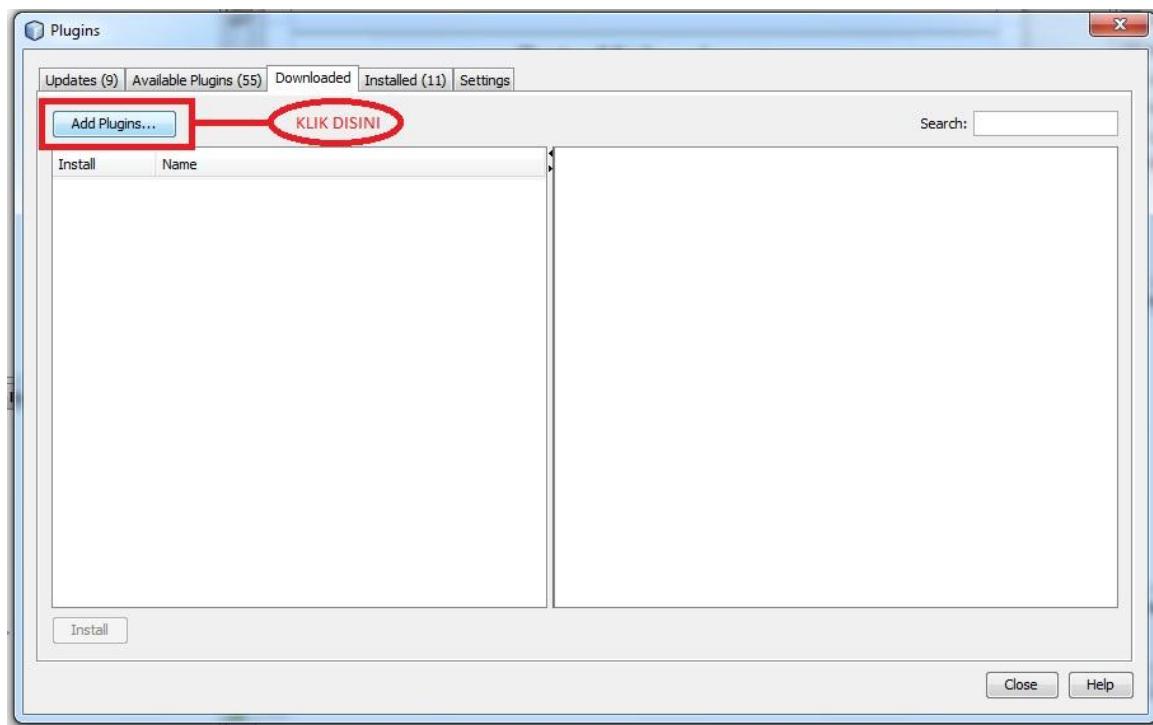
KEGIATAN PRAKTIKUM

Melanjutkan project pada module 9. Kita akan membuat report dengan iReport. Sebelum anda mempraktekan iReport. Anda perlu mendownload file yang sudah saya siapkan di link ini :

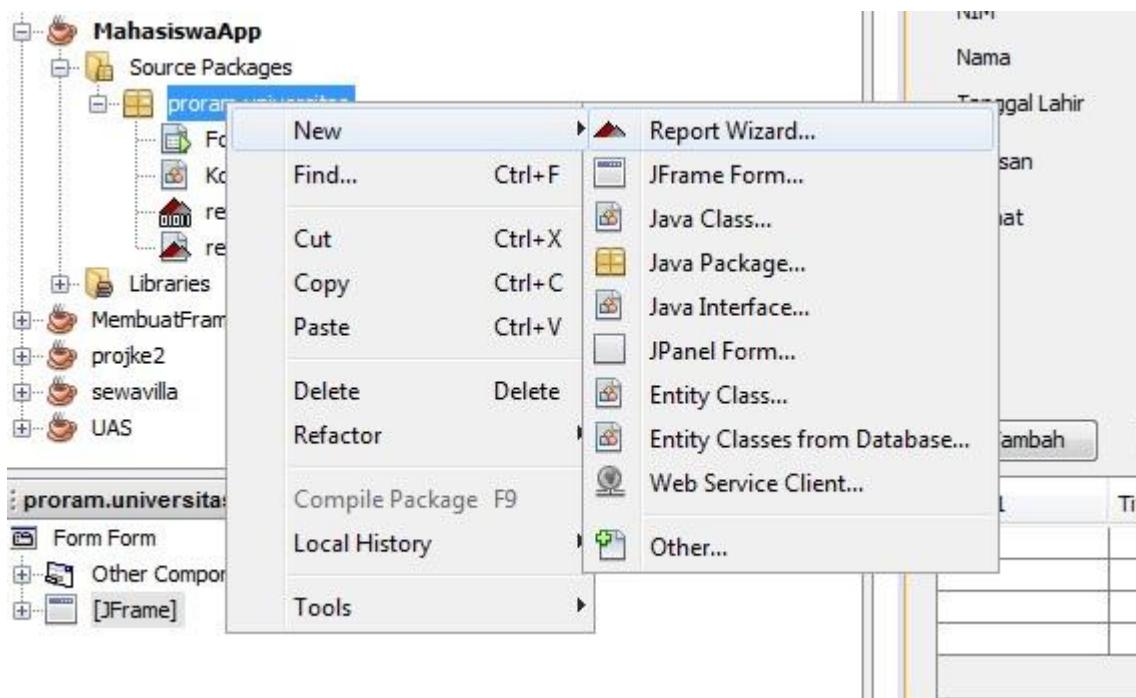
<http://www.4shared.com/rar/wbWFR7cb/Java-Report-file.html> atau

<https://docs.google.com/file/d/0BxYcW8LUKhiRMUpDVnVjSEIxM0k/edit?usp=sharing>

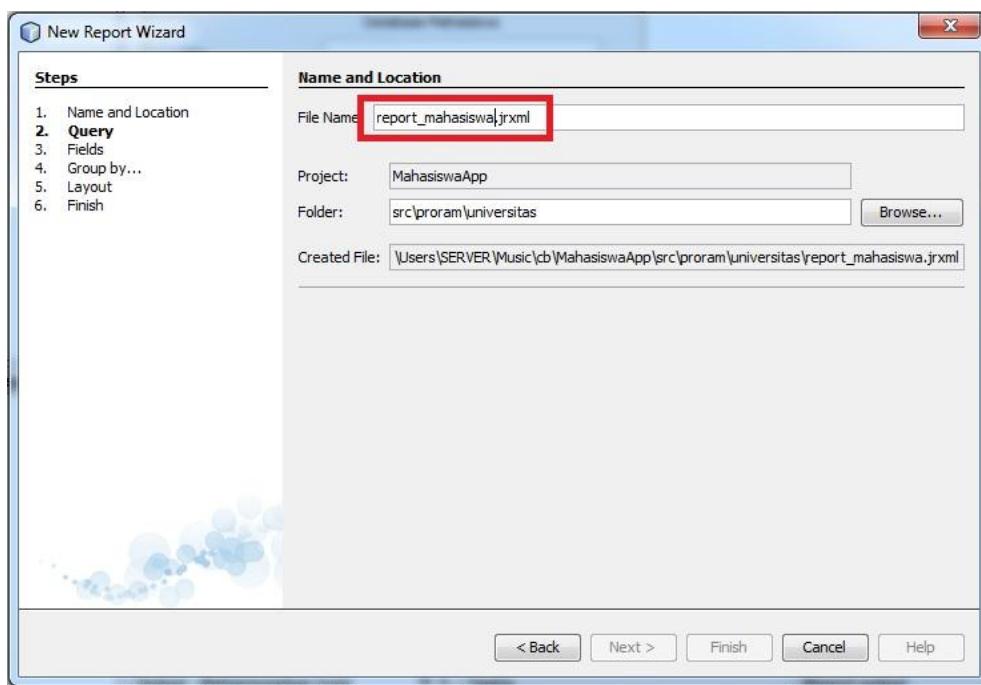
1. Buka Netbeans, selanjutnya pilih **Tools | Plugin**. Pada tab download klik **add plugin** dan masukkan plugin iReport yang sudah anda download tepatnya didalam folder **iReport-nb-3.5.2-plugin**.



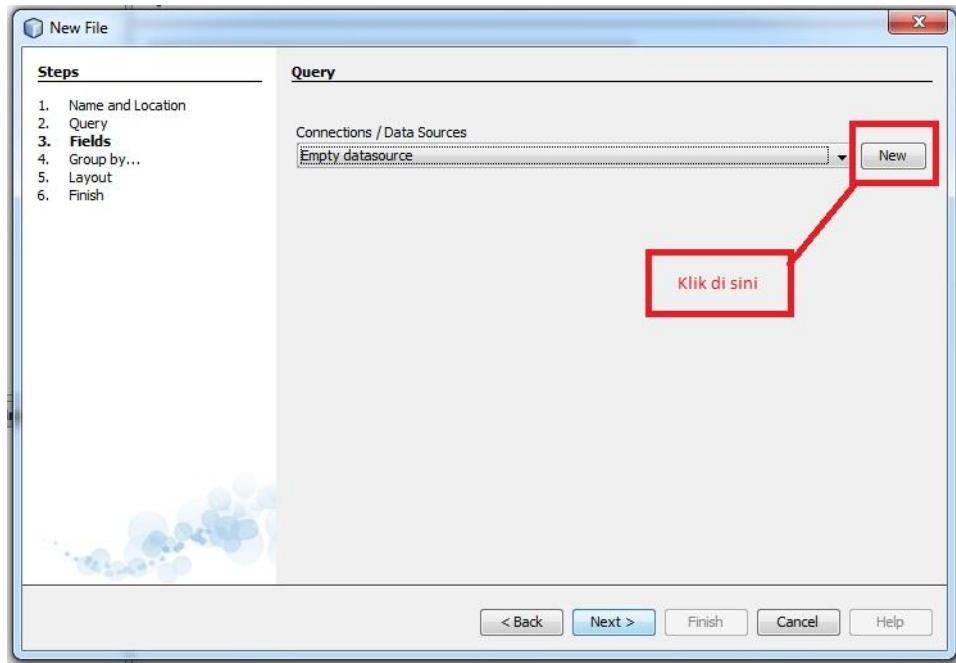
2. Klik Install dan ikuti langkah selanjutnya. Hingga anda diminta untuk merestart IDE Netbeans anda. Biarkan IDE di restart.
3. Setelah itu buka kembali IDE Netbeans anda dan buka pada project MahasiswaApp atau project module 9 yang sudah dibuat dan buat file Report dengan cara klik kanan pada **program.universitas** | pilih **New** | pilih **Report Wizard** atau pilih **other** | pilih **iReport** | pilih **Report Wizard** seperti gambar.



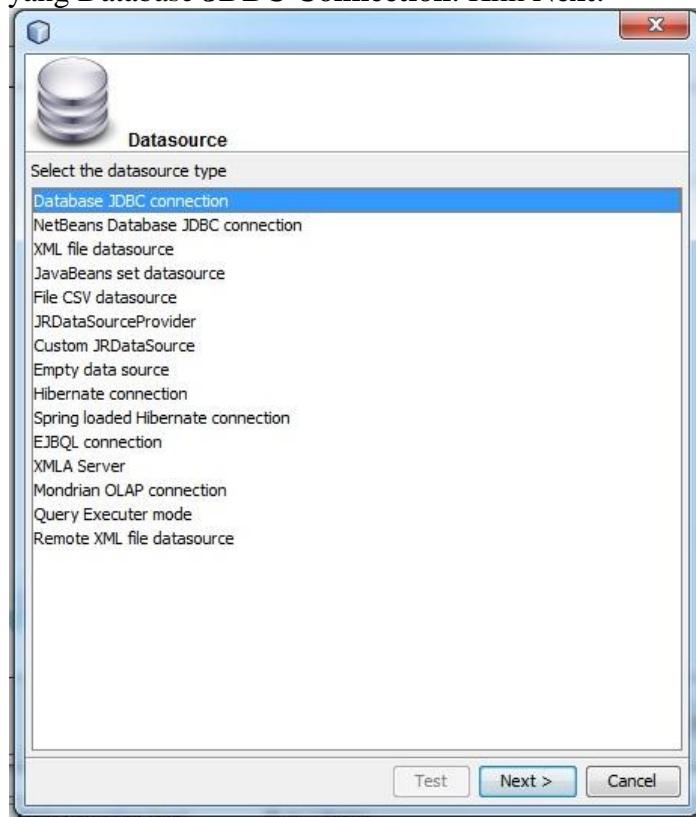
4. Beri nama File **report_mahasiswa.jrxml**



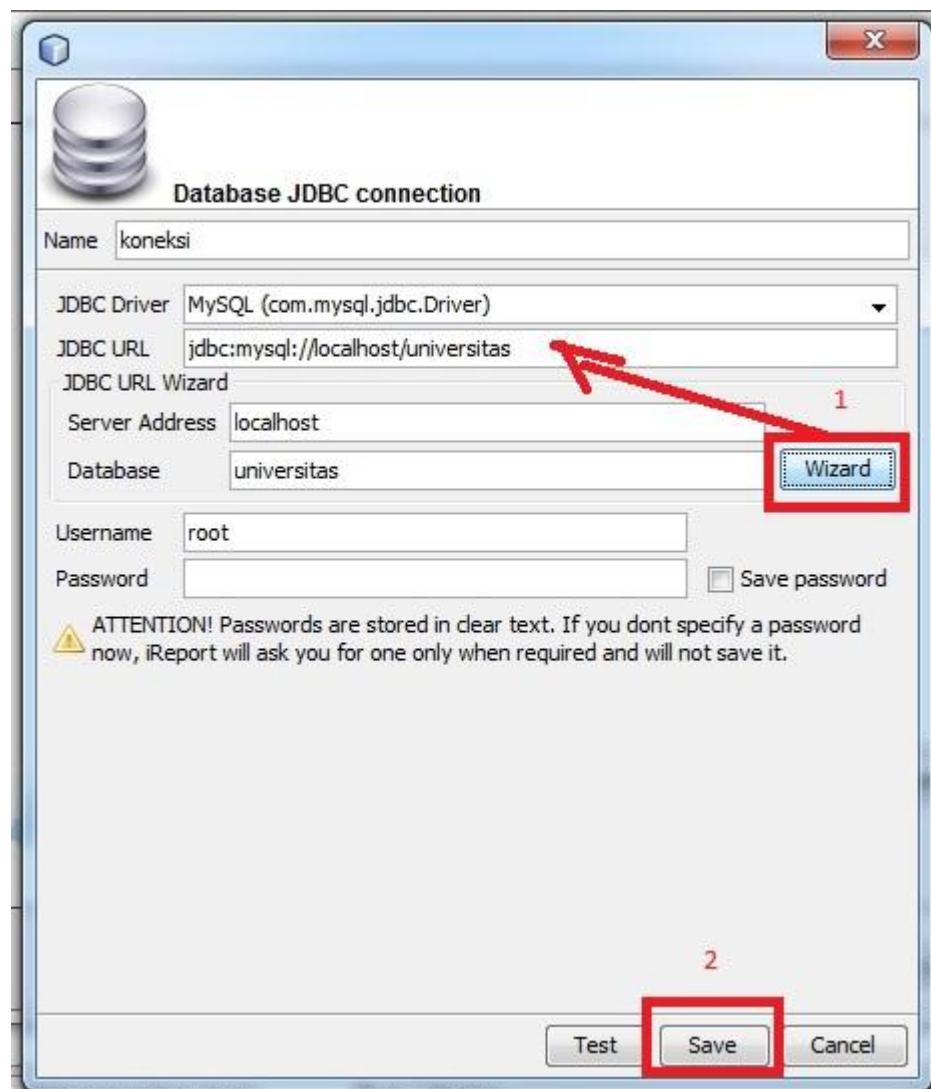
5. Selanjutnya koneksi ke database, pilih **New**.



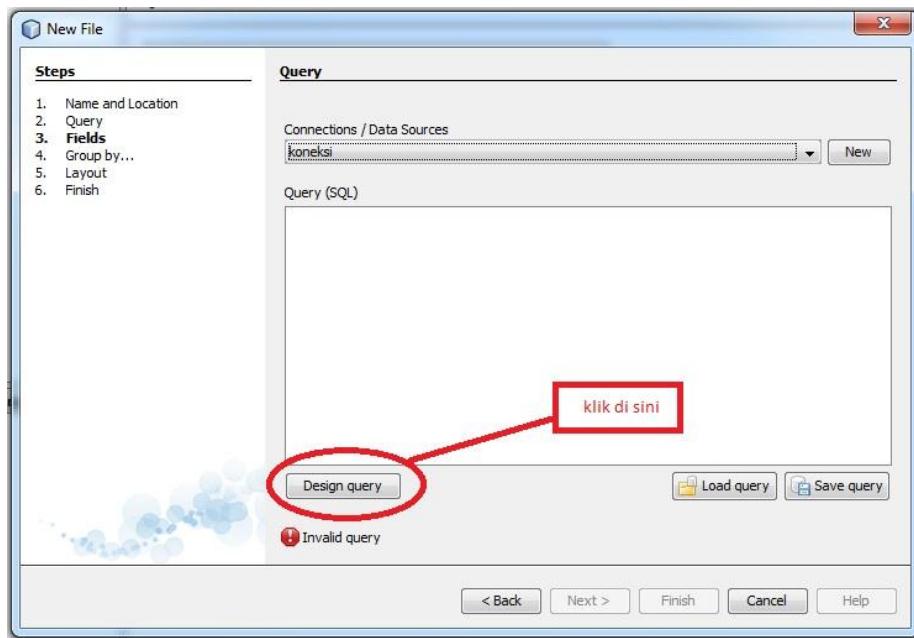
6. Setelah itu bakalan ada pilihan datasource, krn project MahasiswaApp kita pakai MySQL jadi pilih yang Database **JDBC Connection**. Klik Next.



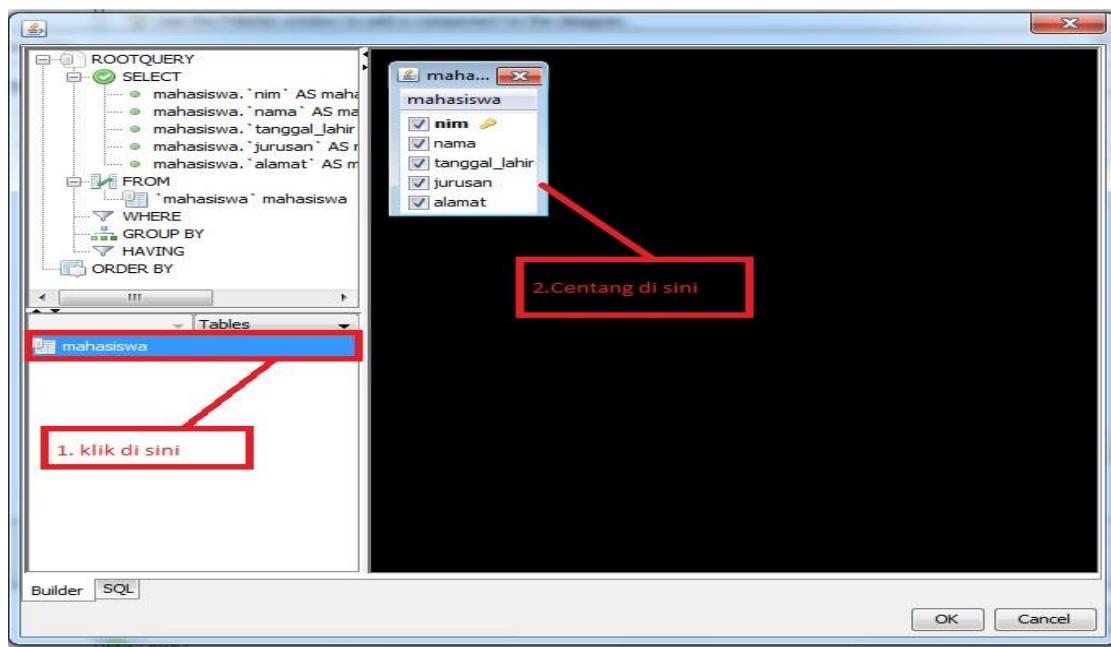
7. Step selanjutnya Setting koneksi, sesuaikan dengan database universitas. Berikan **nama: koneksi , server address : localhost , database universitas , username root, password : kosongkan**, Jika sudah klik **Wizard** agar url berubah dan klik **Save..**.



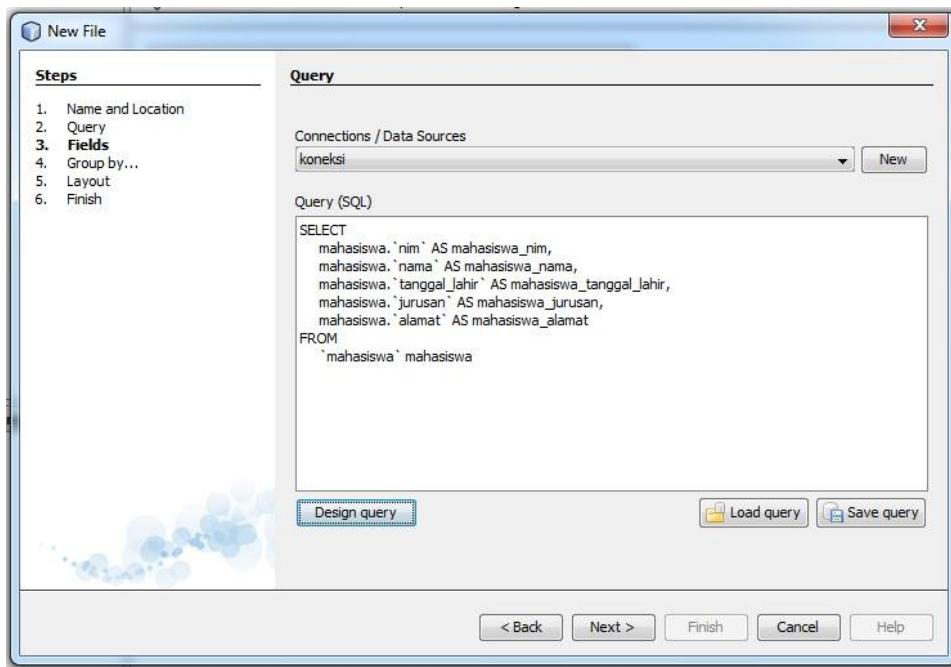
8. Setelah di Save, pilih Design Query..



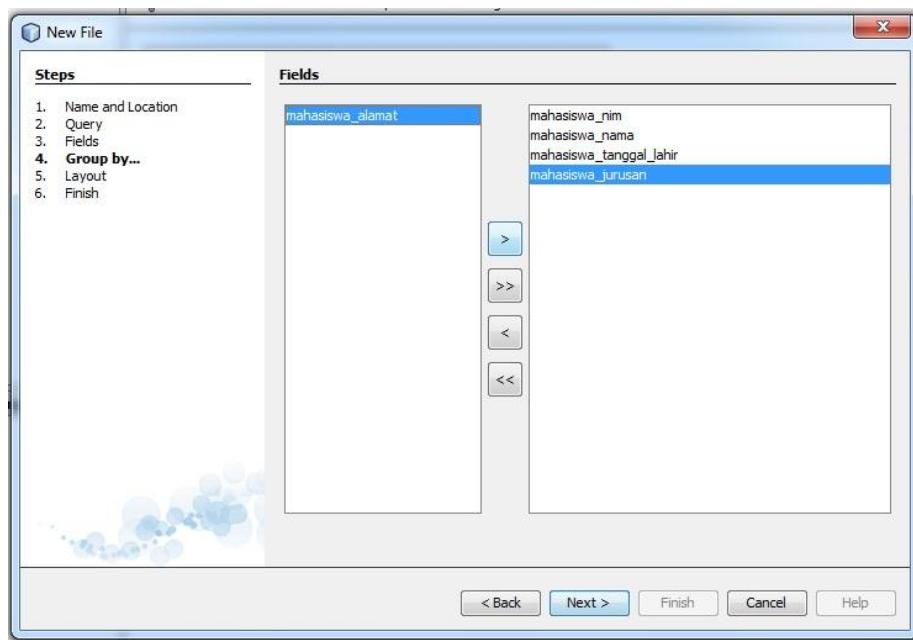
9. Selanjutnya klik oke jika muncul kotak dialog masukan password. Karena database kita tanpa password maka kosongkan dan klik OK. Selanjutnya klik table mahasiswa dan Centang field atau atribut dari tabel yang ingin di tampilkan.



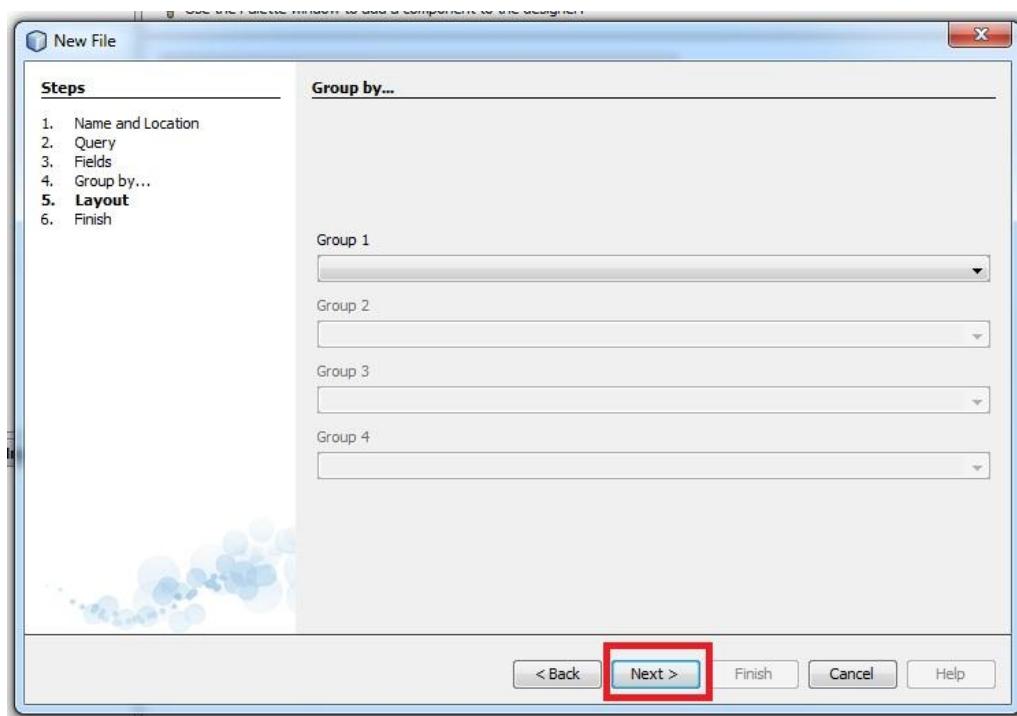
10. Setelah di klik OK. akan ada tampilan seperti dibawah. Lalu klik Next...



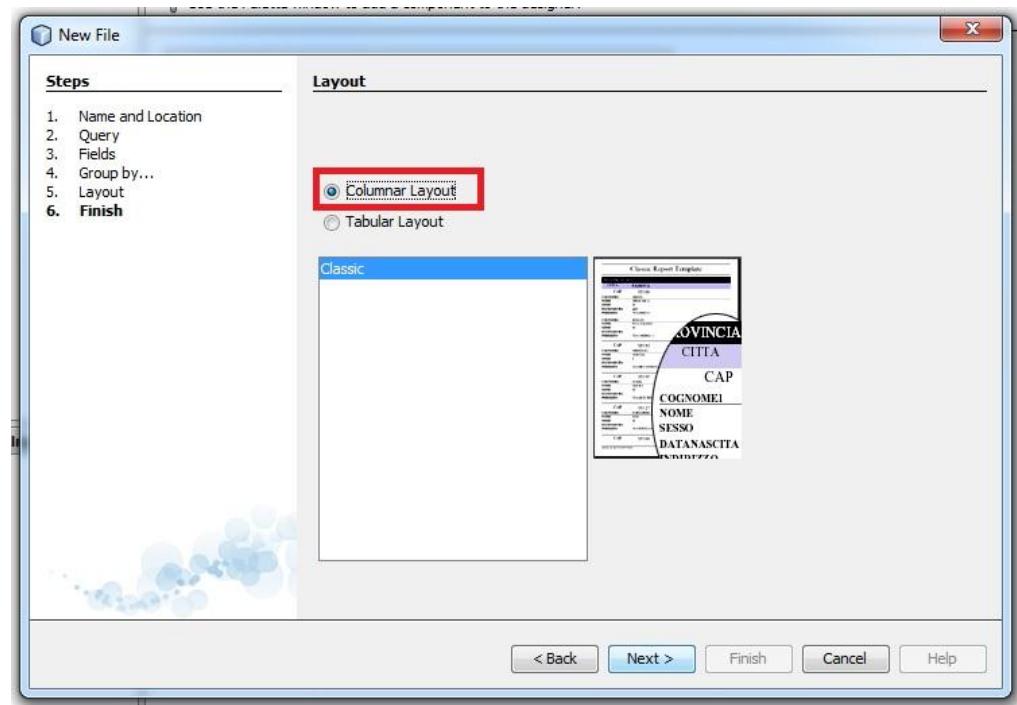
11. Pada tab ini pilih field yang ingin ditampilkan dilaporan dan pindahkan ke sisi kanan. Lalu klik Next..



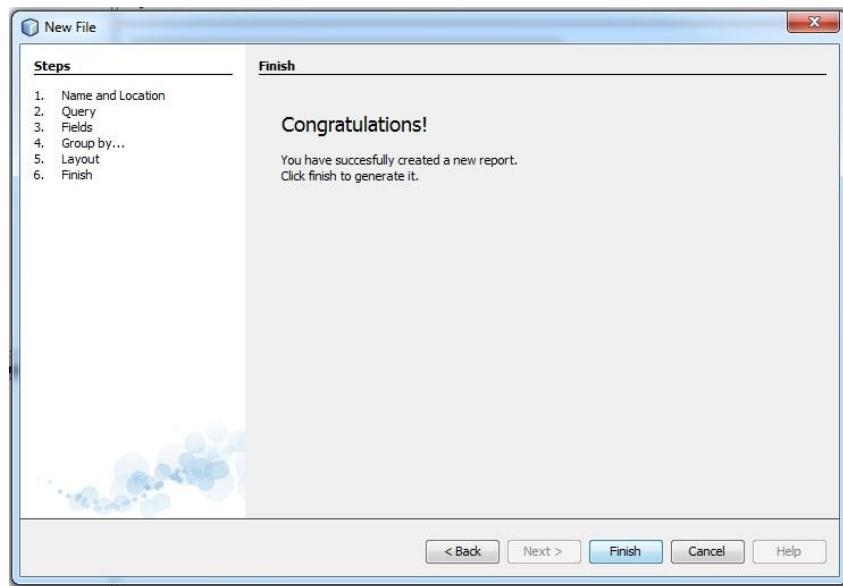
12. Step selanjutnya pilih berdasarkan group by.jika ingin membuat group, jika tidak klik Next, untuk saat ini kita langsung saja klik next.



13. Pilih sesuai yang anda inginkan. Untuk contoh kali ini kita gunakan columnar layout. Setelah itu klik next.



14. dan klik finish untuk menyelesaikan settingan wizard.



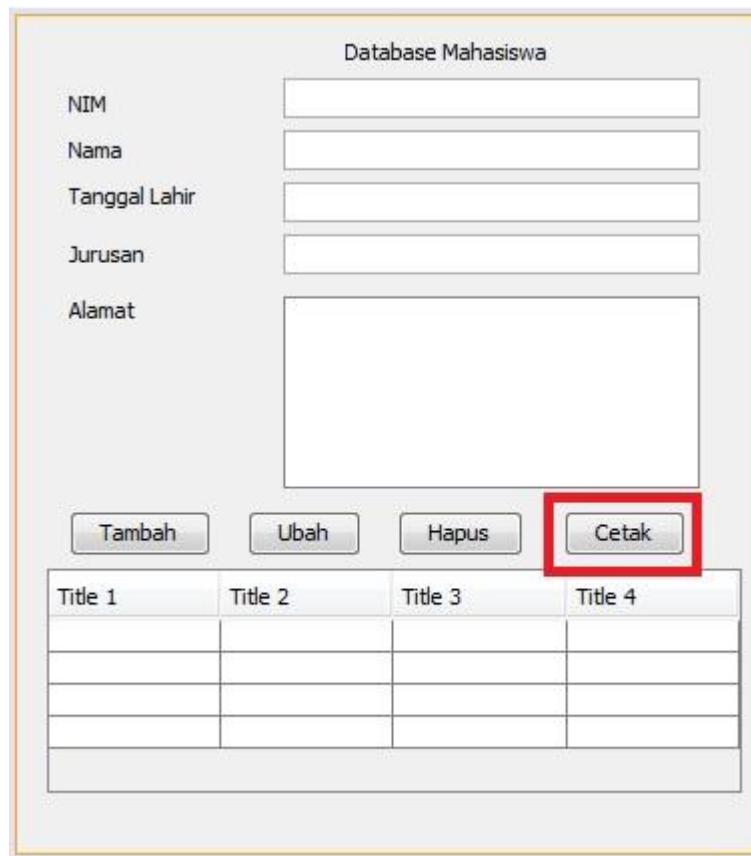
15. Selanjutnya atur format laporan sesuai gambar berikut. Klik 2x untuk mengubah label. Dan klik preview untuk compile report.

The screenshot shows the Jaspersoft Studio interface with the following details:

- File:** report_mahasiswa.jrxml
- Toolbars:** Designer, XML, Preview, SansSerif, 12pt font size.
- Designer View:**
 - Title:** Data Mahasiswa (highlighted by a red box).
 - Table:** Contains five rows with fields: NIM, Nama, Tanggal Lahir, Jurusan, and Alamat. The first row is also highlighted by a red box.
 - Page Footer:** Displays the page number.
- Palette:** Shows various report elements like Chart, Crosstab, Frame, Barcode, Line, Round Rectangle, Subreport, and Tools.
- Note:** A note in the palette states: "ini merupakan tools jika anda ingin berkreasi".
- Properties:** Shows Report name: null and Page size.

Text at the bottom left of the designer view: "Ubah Label sesuai kotak berwarna merah"

16. Selanjutnya kembali ke form GUI. Dan tambahkan JButton untuk mencetak data kita.

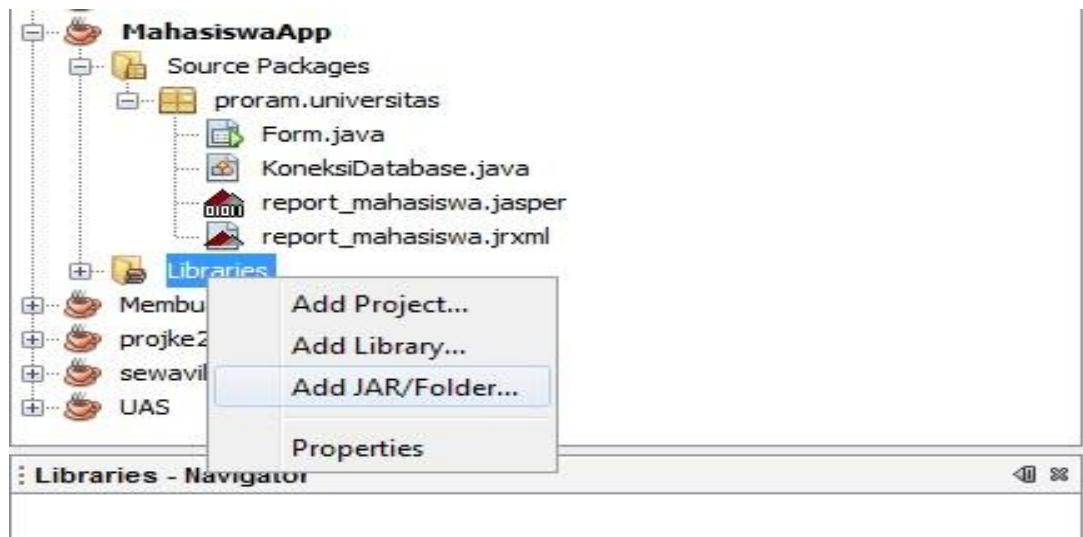


17. Klik 2x pada jButton **Cetak**, kemudian tambahkan coding berikut.

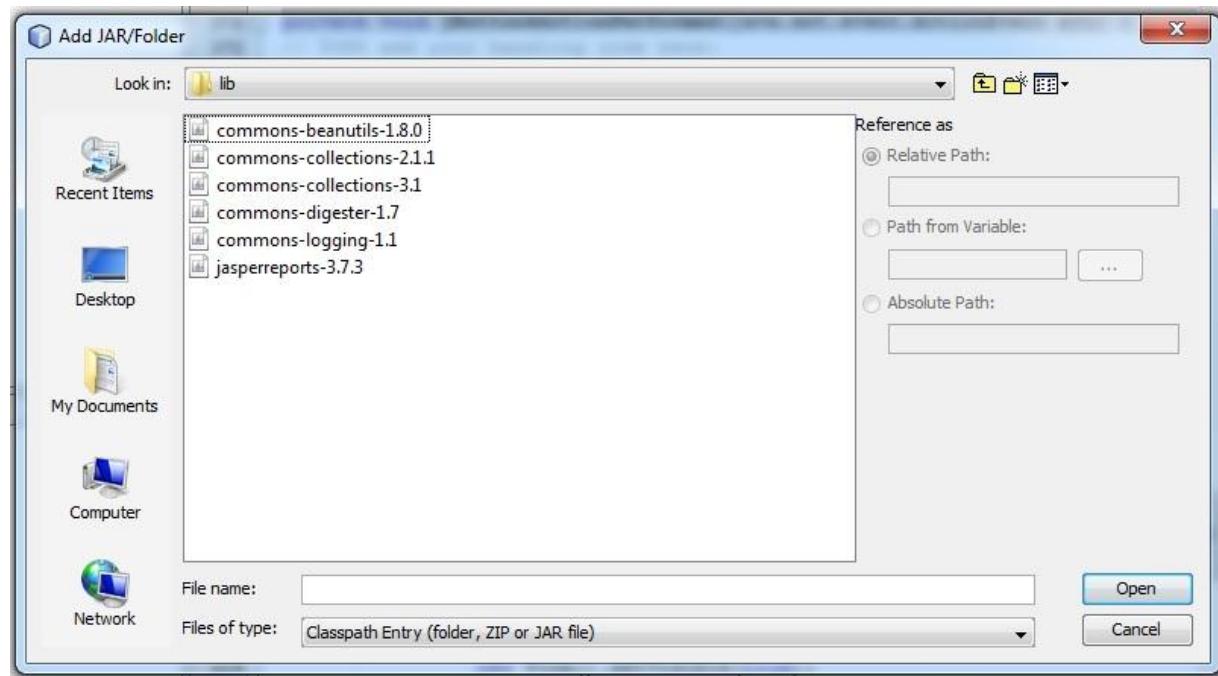
```
try {
    File file = new File("src/program/universitas/report_mahasiswa.jrxml ");
    jasperDesign = JRXmlLoader.load(file);
    param.clear();
    jasperReport = JasperCompileManager.compileReport(jasperDesign);
    jasperPrint = JasperFillManager.fillReport(jasperReport, param,
koneksi.getConnection());
    JasperViewer.viewReport(jasperPrint, false);
} catch (Exception e) {
    e.printStackTrace();
}
```

18. jika sudah pasti anda akan menemukan beberapa lampu eror dari netbeans. Tapi jangan khawatir anda hanya perlu menambahkan **libraries**, **variabel** serta **import** untuk

mengatasi masalah tersebut.pertama kita akan menambahkan **libraries**. Dengan cara klik kanan pada **libraries** | Pilih **Add JAR/Folder** seperti pada gambar berikut.



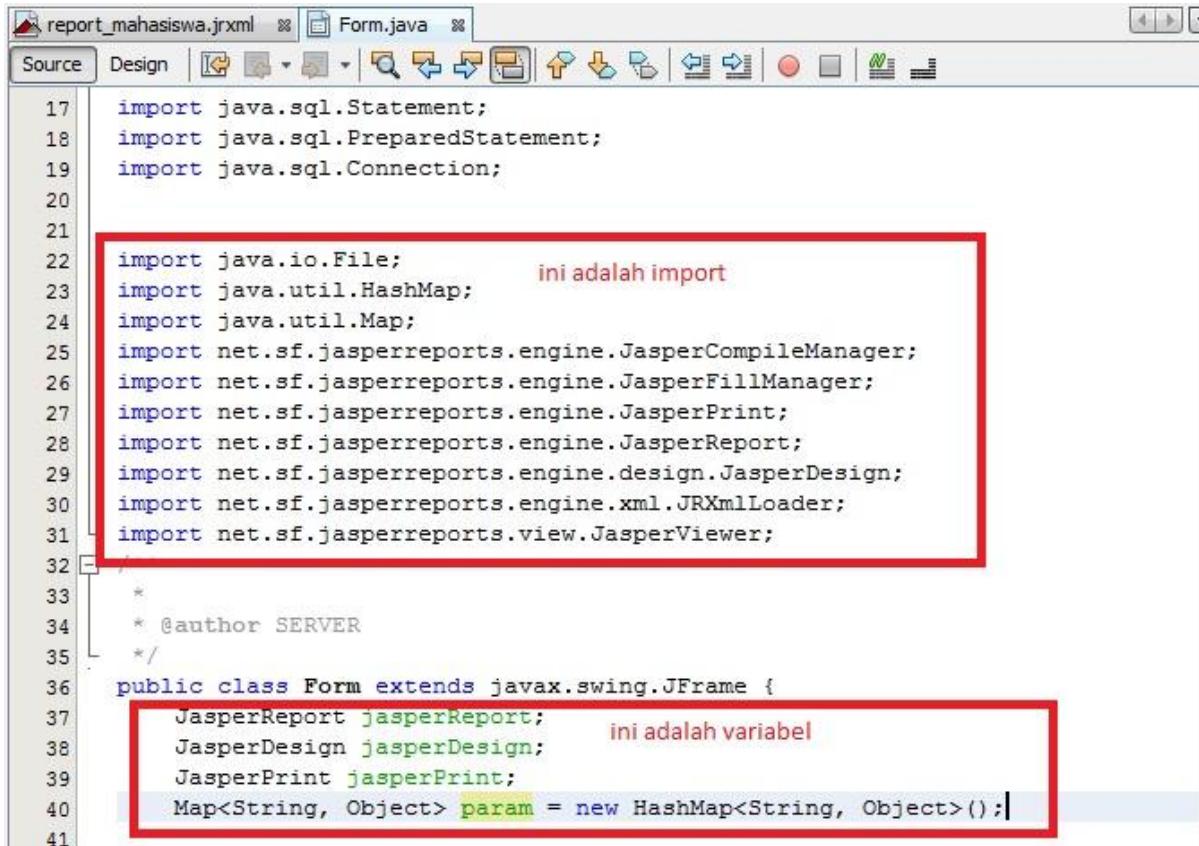
19. Kemudian pilih semua file yang telah anda download tepatnya di dalam folder **add-jar-library** tambahkan semua file seperti pada gambar berikut.



20. Selanjutnya tambahkan **variabel** berikut. Ketikan syntax berikut pada program anda.tepat dibawah class form.

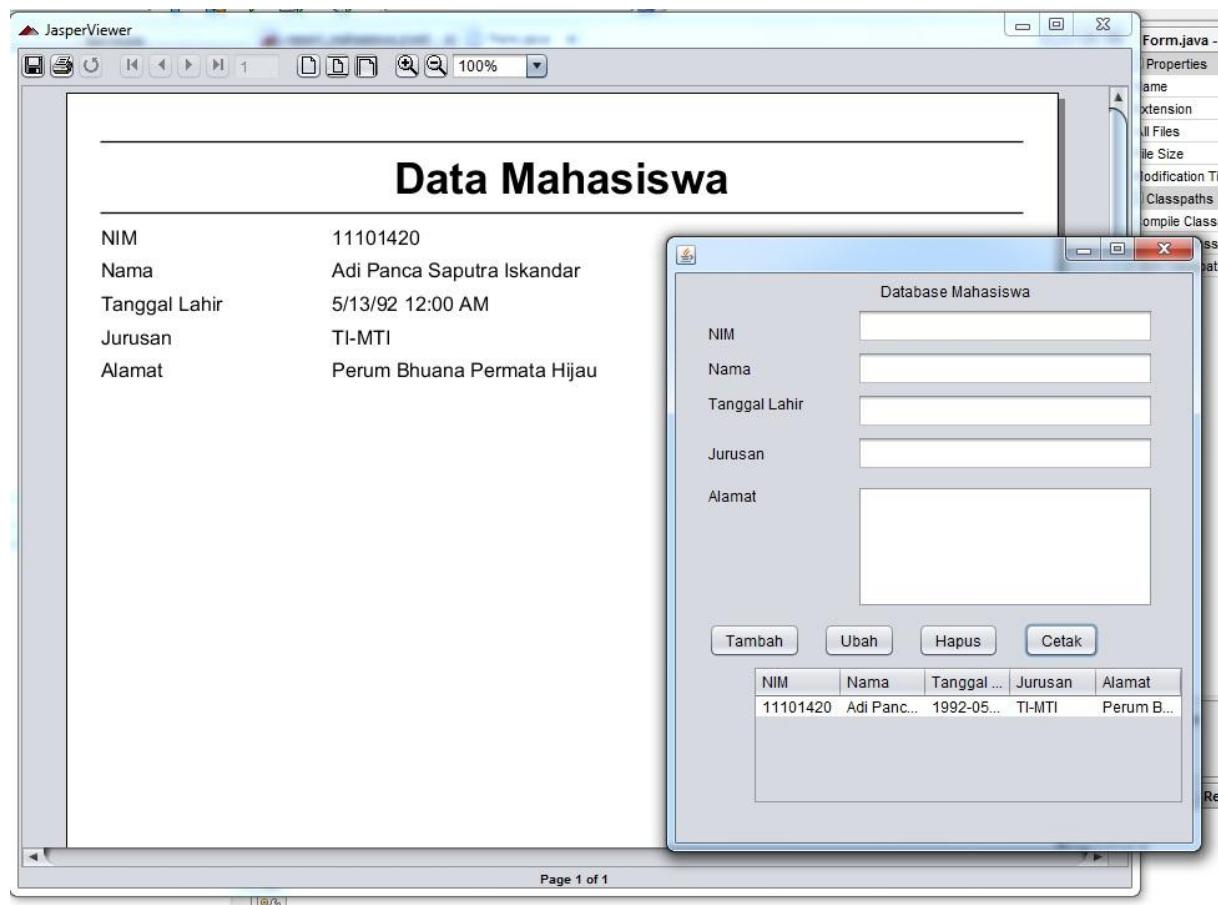
```
public class Form extends javax.swing.JFrame {  
    JasperReport jasperReport;  
    JasperDesign jasperDesign;  
    JasperPrint jasperPrint;  
    Map<String, Object> param = new HashMap<String, Object>();
```

21. Dan yang terakhir tambahkan beberapa import pada syntax anda. Ketikan syntax berikut. Tepat di bawah import sebelumnya atau di atas class form.



```
report_mahasiswa.jrxml  Form.java  
Source Design 17 import java.sql.Statement;  
18 import java.sql.PreparedStatement;  
19 import java.sql.Connection;  
20  
21  
22 import java.io.File;      ini adalah import  
23 import java.util.HashMap;  
24 import java.util.Map;  
25 import net.sf.jasperreports.engine.JasperCompileManager;  
26 import net.sf.jasperreports.engine.JasperFillManager;  
27 import net.sf.jasperreports.engine.JasperPrint;  
28 import net.sf.jasperreports.engine.JasperReport;  
29 import net.sf.jasperreports.engine.design.JasperDesign;  
30 import net.sf.jasperreports.engine.xml.JRXmlLoader;  
31 import net.sf.jasperreports.view.JasperViewer;  
32  
33 *  
34 * @author SERVER  
35 */  
36 public class Form extends javax.swing.JFrame {  
    JasperReport jasperReport;  ini adalah variabel  
    JasperDesign jasperDesign;  
    JasperPrint jasperPrint;  
    Map<String, Object> param = new HashMap<String, Object>();
```

Selanjutnya anda dapat compile dan Run project MahasiswaApp anda. Dan klik tombol cetak untuk melihat hasil dari report yang telah anda buat. Tampilan akan seperti gambar berikut.



TUGAS

Implementasikan materi yang telah di dapat dalam project sistem informasi yang telah dibuat