

LAPORAN PRAKTIKUM
PBO (PEMODELAN BERORIENTASI OBJEK)
(PENGENALAN BAHASA JAVA DENGAN NOTEPAD++ DAN IDE NETBEANS)



RIKI HISBULLAH
0701192023

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UIN SUMATERA UTARA MEDAN
2020/2021

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penyusun, sehingga dapat menyelesaikan LAPORAN PRAKTIKUM PEMODELAN BERORIENTASI OBJEK ini. Sholawat dan salam juga kita curahkan kepada baginda yang mulia Rosulullah SAW beserta keluarga, sahabat dan pengikutnya hingga akhir zaman.

Laporan ini merupakan salah satu bahan ajar pendukung untuk mata kuliah PBO dengan Pengenalan Bahasa Java. Dari laporan ini diharapkan dapat dengan mudah mempelajari, memahami, dan mempraktikkan materi – materi yang telah diajarkan pada kelas teori mata kuliah PBO. Kemudian Laporan ini diharapkan dapat menjadi referensi untuk pemecahan permasalahan umum di luar materi perkuliahan. Sebagian besar isi dari Laporan ini merupakan rangkuman dari sumber-sumber yang telah dibuat penulis lain. Penyusun berharap agar laporan ini dapat bermanfaat bagi semua kalangan pembaca. Terima kasih untuk semuanya yang telah memberikan banyak kritik dan saran serta dukungan dalam penyusunan laporan ini.

Rumah Masing-masing, 13 Des. 2021

RIKI HISBULLAH

DAFTAR ISI

MODUL 1	PENGANTAR BAHASA JAVA	Halaman 5
MODUL 2	VARIABEL DAN TIPE DATA	Halaman 7
MODUL 3	OPEATOR DALAM JAVA	Halaman 8
MODUL 4	STRUKTUR KONTROL	Halaman 12
MODUL 5	ARRAY	Halaman 16
MODUL 6	CLASS METHOD DAN OBJEK DAN KONSTRUKTOR	Halaman 19
MODUL 7	PEWARISAN DAN POLIMORPISME	Halaman 23
MODUL 8	GUI DENGAN NETBEANS	Halaman 26
MODUL 9	PENUTUP	Halaman 32
MODUL 10	PENGESAHAN	Halaman 33

DAFTAR GAMBAR

1	Gambar 1	Halaman 6
2	Gambar 2	Halaman 11
3	Gambar 3	Halaman 11
4	Gambar 4	Halaman 14
5	Gambar 5	Halaman 15
6	Gambar 6	Halaman 15
7	Gambar 7	Halaman 15
8	Gambar 8	Halaman 17
9	Gambar 9	Halaman 18
10	Gambar 10	Halaman 19
11	Gambar 11	Halaman 20
12	Gambar 12	Halaman 20
13	Gambar 13	Halaman 24
14	Gambar 14	Halaman 24
15	Gambar 15	Halaman 36
16	Gambar 16	Halaman 27
17	Gambar 17	Halaman 27
18	Gambar 18	Halaman 27
19	Gambar 19	Halaman 28
20	Gambar 20	Halaman 28
21	Gambar 21	Halaman 30
22	Gambar 22	Halaman 30

DAFTAR TABEL

1	Tabel 1	Halaman 7
2	Tabel 2	Halaman 8
3	Tabel 3	Halaman 8
4	Tabel 4	Halaman 10
5	Tabel 5	Halaman 10
6	Tabel 6	Halaman 10
7	Table 7	Halaman 11

MODUL I

PENGANTAR BAHASA JAVA

1. Pengertian

Java merupakan salah satu bahasa pemrograman tingkat tinggi berorientasi objek, dikembangkan pertamakali oleh Sun Microsystems yang dimulai oleh James Gosling dan dirilis pada tahun 1995. Java bersifat Write Once, Run Anywhere (Program Aplikasi yang ditulis satu kali dan dapat berjalan pada banyak platform). “Jubilee Enterprise, Belajar java Database, dan Netbeans dari Nol.”

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

Untuk membuat program java, yang dibutuhkan yaitu :

J2 SDK : Java 2 Standar Development Kit (J2SDK) adalah seperangkat tool (kit) yang berisi compiler dan Java yang dirilis oleh Sun dan bersifat free. Istilah J2SDK merupakan akronim dari Java 2 Software Development Kit. Compiler ini tersedia untuk berbagai platform sistem operasi; DOS, Linux, dan Solaris.

Netbeans : Netbeans IDE merupakan software IDE (Integrated Development Environment) yang digunakan untuk membuat aplikasi / software yang lain (edit + compiler + debugger + built automation tool + versioning system + dst). Netbeans IDE dapat digunakan untuk membuat aplikasi Java Standard Edition, Java Micro Edition, Java Enterprise Edition, JavaFX, PHP, C/C++, Ruby, Groovy dan Python.

Bahasa pemrograman java yang telah dirilis adalah :

- 1) Produk Sun Microsystem : Java 2 Platform Standard Edition (J2SE), Java 2
- 2) Micro Edition (J2ME), Java 2 Enterprise Edition (J2EE).
- 3) Produk Borland : Borland JBuilder
- 4) Produk Microsoft : Microsoft Visual J/J++

Spesifikasi J2EE antara lain Java Database Connectivity, JDOM (untuk XML), Remote Method Invocation, Enterprise Java Beans, Java Messaging, Java Server Pages, Java Servlet, Socket Programming, dan SQLJ. Java/Standard Development Kit (JDK/SDK) merupakan alat-alat utama bagi programmer untuk membuat dan menjalankan java. Development Kit dapat didownload di :

<http://java.sun.com/product/JDK/index.html> atau <http://java.sun.com/j2se/1.5/index.html> atau, <http://java.sun.com/cgi-bin/java-ports.cgi> Komponen JDK antara lain compiler(javac), interpreter(java) disebut juga

java virtual machine atau java runtime environment, applet viewer(appletviewer), debugger(jdb), java class library(jcl), header dan stub generator(javah), dan yang penting yaitu java documentation(javadoc).

Penjelasan penggunaan komponen JDK :

- 1) Kompilator (javac)

Berfungsi untuk kompilasi file source code : *.java menjadi *.class

Syntax umum : javac nama_file.java

- 2) Interpreter (java)

Bertugas untuk menjalankan bytecode (*.class)

Syntax umum : java nama_file_class

- 3) Applet Viewer

Digunakan untuk menjalankan applet viewer, namun sekarang sudah digantikan browser.

Syntax umum : appletviewer nama_file.html

4) 4. Java Debugger

Bertugas untuk melakukan debugging aplikasi java. Syntax umum : jdb option

5) 5. Java Class File Diassembler (javap)

Bertugas membuat daftar method dan attribute public dari suatu kelas.

Syntax : javap namaKelas

6) 6. Java Header and Stub Generator

Bertugas menerjemahkan bahasa yang ditulis dalam bahasa Java menjadi bahasa pemrograman C.

Syntax umum : javah namaKelas

7) 7. Java Documentation Generator

Menampilkan pustaka kelas, interface, constructor, dan method standard yang telah dibuat vendor.

Dari hasil instalasi JDK, dokumentasi ini dapat dilihat pada

C:\java\docs\api\index.html dan Dari hasil instalasi Netbeans, dapat dilihat pada C:\Program Files\NetBeans3.6\doc\junit\index.html

8) 8. Source Code Java API

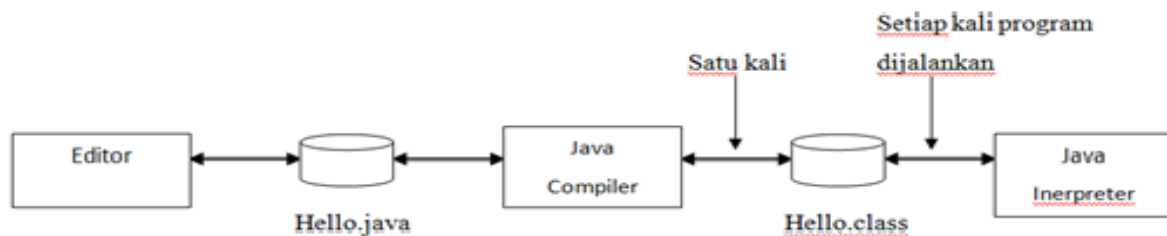
Source code ini dapat diperoleh dari file src.zip.

Untuk pemrogram pemula, lingkungan pemrograman java dapat diringkas menjadi :

- a) Editing source code menggunakan editor teks, seperti Notepad atau TextPad
- b) Compiling menggunakan keyword javac melalui command prompt (dapat juga dari editor teks seperti TextPad).
- c) Executing menggunakan :
 - i. Command prompt untuk java application(atau dari editor teks TextPad)
 - ii. Browser atau appletviewer untuk java applet

2. Tahap Pemrograman Java

Gambar dibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java:



Penjelasan gambar :

1. Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada text editor. Missal Notepad ++ Notepad, emacs, dll.
2. Kode program yang dibuat tersimpan dalam sebuah berkas berekstensi .Java
3. Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari kompilasi berupa berkas bytecode dengan ekstensi.class
4. Bahasa yang mengandung bytecode tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan platform yang digunakan.

MODUL II

VARIABLE DAN TIPE DATA

1. Variabel dan Tipe Data

Variable merupakan container yang digunakan untuk menyimpan suatu nilai pada sebuah program dengan tipe tertentu. Pada dasarnya ada dua macam tipe variabel data dalam bahasa Java, yakni tipe primitif dan tipe reference.

Adapun tipe primitif meliputi:

- Tipe boolean

Dalam Java, setiap variabel yang bertipe boolean hanya akan memiliki nilai true atau false.

- Tipe numeric (Byte, Short, Int, Long, Char, Float, Double)

Adapun tipe numerik dalam Java ada tujuh macam seperti yang telah dijelaskan sebelumnya yakni:

Nama Tipe	Ukuran	Range Nilai
Byte	8 bit	-128, 127
Short	16 bit	-32768, 32767
Int	32 bit	-2147482648, 2147483647
Long	64 bit	-9223372036854775808, 9223372036854775808
Char	16 bit	0..65535 ('\u0000' ... '\uffff')
Float	32 bit IEEE	
Double	64 bit IEEE	

Tipe data variabel reference terdiri :

- Tipe class
- Tipe array
- Tipe interface

2. Konstanta dalam Java

Variabel dalam Java bisa dijadikan konstanta, sehingga nilainya tidak akan dapat diubah-ubah dengan mendeklarasikannya sebagai variabel final seperti ini:

```
final int dataInt = 10;
final char charData = '\u103';
final float x = 12,67;
final byte y = 2;
```

3. Penamaan Variabel dalam Java

Harus terdiri atas sederetan karakter unicode yang diawali oleh karakter huruf atau garis bawah. Unicode merupakan sistem pengkodean karakter yang dapat dibaca oleh berbagai bahasa manusia (Jepang, Yunani, Cyrillic, dan Hebrew). ASCII hanya terbatas pada karakter latin.

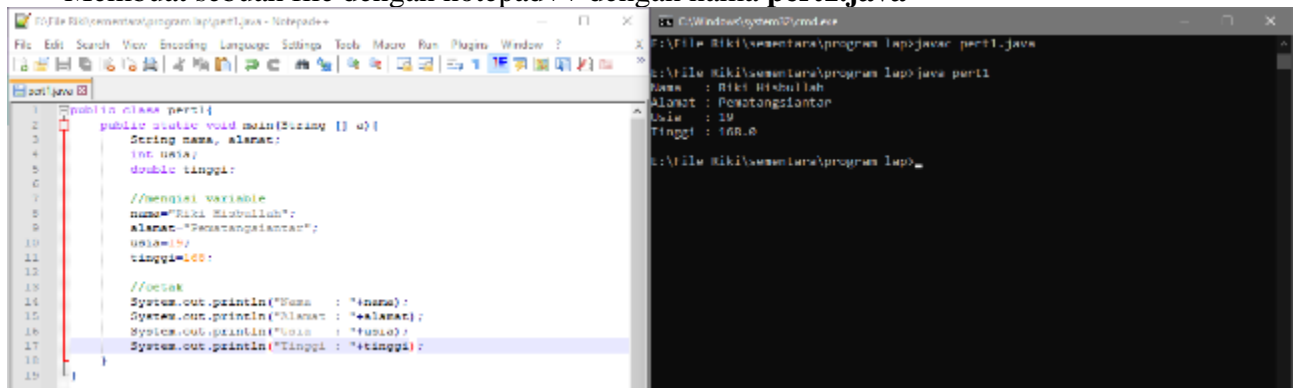
- Tidak boleh berupa keyword (kata yang dicadangkan), null, atau literatur true/false.
- Harus unik dalam suatu scope

Contoh pendeklarasian variabel yang salah :

```
Int 8k;           /* salah, karena nama variabel diawali dengan angka */
Char null;        /* salah, karena nama variabel = null */
String public;    /* salah, karena public merupakan keyword dalam Java */
```

Praktikum program

Membuat sebuah file dengan notepad++ dengan nama **pert1.java**



The image shows two windows side-by-side. The left window is Notepad++ editing a file named 'pert1.java'. The code is as follows:

```
1 public class pert1 {
2     public static void main(String[] args) {
3         String nama, alamat;
4         int usia;
5         double tinggi;
6
7         //mengisi variable
8         nama="Riki Rihullah";
9         alamat="Pematangsiantar";
10        usia=19;
11        tinggi=168.8;
12
13        //output
14        System.out.println("Nama : "+nama);
15        System.out.println("Alamat : "+alamat);
16        System.out.println("Usia : "+usia);
17        System.out.println("Tinggi : "+tinggi);
18    }
19 }
```

The right window is a terminal window showing the output of the Java program:

```
C:\Windows\system32\cmd.exe
F:\File Riki\semantara\program lap3\java pert1
Nama : Riki Rihullah
Alamat : Pematangsiantar
Usia : 19
Tinggi : 168.8
F:\File Riki\semantara\program lap3_
```

Pada program diatas menyimpan sebuah nilai pada variable dengan tipe data yang tertera pada program.

MODUL III

OPERATOR DALAM JAVA

1. Operator Aritmatika

Merupakan operator yang paling umum digunakan bahkan didunia nyata, kita sering menggunakan operator tersebut.

Operator aritmatika digunakan pada ekstensi matematika seperti pada operasi aljabar. Operator yang termasuk kedalam jenis aritmatika antara lain. Operator +, -, *, /, %, ++, --.

Operator dalam Aritmatika terbagi menjadi dalam dual kelompok yaitu:

a. Operator Binary

Adalah operator yang dalam penggunaannya melibatkan 2 buah operand, misalnya 7+2.

Berikut adalah jenis operator binary:

No	Simbol	Nama Operator	Deskripsi
1	+	Penjumlahan	Menjumlahkan dua buah Operand
2	-	Pengurangan	Mengurangkan dua buah Operand
3	*	Pengalian	Melakukan Perkalian dari dua buah operand
4	/	Pembagian	Membagi suatu operand dengan operand lainnya
5	%	Modulus	Menghasilkan nila sisa bagi dari proses pembagian

b. Operator Unary

Operator Unary adalah operator yang melibatkan satu buah operand, misalkan variabel A bernilai negatif 5, maka Anda bisa menggunakan Operator negatif sebagai penanda bahwa variabel A bernilai negatif(-A).

No	Simbol	Nama Operator	Deskripsi
1	+	Positif	Menandakan suatu bilangan menjadi bernilai positif
2	-	Negatif	Menandakan suatu bilangan menjadi bernilai negatif
3	++	Increment	Menambah 1 nilai keatas pada suatu operand(variabel)
4	--	Decrement	Penurunan: mengurangi 1 nilai kebawah pada operand

Coba perhatikan dan pahami, Operator '+' memiliki 2 makna, yaitu Operator + sebagai penjumlahan dan Operator + sebagai penanda bahwa suatu nilai bernilai positif. Lalu bagaimana jika Operator + digunakan untuk menjumlahkan String?.

2. Operator String Concetenation

Jika Operator dengan simbol '+' pada Aritmatika bisa dimaknai sebagai Operator Penjumlahan dan Operator Positif, maka berikut akan dijelaskan fungsi lain dari Operator dengan simbol plus(+).

a. Menggabungkan 2 buah String

Fungsi pertama dari Operator plus(+) yaitu bisa digunakan untuk menggabungkan 2 nilai yang berisi tipe data String. Berikut contoh program java untuk menggabungkan 2 buah String.

b. Menggunakan angka dengan String

Fungsi kedua dari Operator plus(+) yaitu, Anda bisa menggunakan-nya untuk menggabungkan 2 nilai, nilai pertama berisi Angka(int) dan nilai kedua berisi String.

c. Fungsi lainnya

Fungsi lainnya yaitu, Anda bisa menggunakan Operator plus(+) untuk menggabungkan 2 nilai yang salah satu nilainya merupakan variabel referensi dengan nilai null.

Variabel Referensi adalah variabel yang berisi tipe data yang didefinisikan sendiri oleh programmer, atau bisa disebut dengan 'Tipe Data Referensi'. Berikut contoh operator + untuk menjumlahkan nilai NULL.

3. Operator Penugasan/Assingment pada Java

Pada dasarnya, Operator Assignment adalah operator yang digunakan untuk memasukkan nilai ke dalam variabel. Operator tersebut disimbolkan dengan tanda '=', misal `<code>int jumlah = 25;`

Lalu bagaimana jika penulisannya seperti ini 'A += 3;', Hal tersebut hasilnya akan sama dengan 'A = A + 3'. Nah, Operator += disebut operator gabungan dari tanda plus dan assignment.

a. Operator Sama Dengan

Sama seperti pada pembahasan diatas, Operator sama dengan(=) disebut sebagai Operator Assignment. Operator tersebut digunakan untuk memasukkan nilai ke dalam variabel.

Contoh : `int jumlah = 25;`

b. Operator Gabungan

Operator-operator yang termasuk dalam jenis Operator Gabungan disini, pada dasarnya adalah gabungan dari Operator Assignment dan Operator Aritmatika. berikut daftar operator gabungan.

No	Operator	Nama Operasi	contoh
1	+=	Penjumlahan	A += B sama dengan A=A+B
2	-=	Pengurangan	A -= B sama dengan A=A-B
3	*=	Perkalian	A *= B sama dengan A=A*B
4	/=	Pembagian	A /= B sama dengan A=A/B
4	%=	Modulus	A %= B sama dengan A=A%B

4. Operator Relasi Pada Java

Semua Operator yang termasuk dalam jenis Operator Relasional adalah operator yang digunakan untuk membandingkan 2 buah operand, hasil yang akan didapat hanya 2 nilai yaitu TRUE dan FALSE.

Semua Operator Relasional lebih sering dimanfaatkan dalam pembuatan struktur if else, dan berikut tabel berisi daftar operator yang termasuk dalam kategori Operator Relasional.

No	Operator	Contoh
1	==	(10 == 20) akan menghasilkan nilai FALSE
2	!=	(10 != 20) akan menghasilkan nilai TRUE
3	>	(10 > 20) akan menghasilkan nilai FALSE
4	<	(10 < 20) akan menghasilkan nilai TRUE
5	>=	(10 >= 20) akan menghasilkan nilai FALSE

6	<=	(10 <= 20) akan menghasilkan nilai TRUE
---	----	---

5. Operator Logika Pada Java

Berbeda dengan operator Relasional, operator logika digunakan untuk perbandingan 2 buah operand, dimana kedua buah operand tersebut bernilai logika yang true dan false.

Nilai yang dihasilkan dari penggunaan operand logika adalah boolean (True dan False), dan berikut table berisi daftar Operator logika dalam lingkup Java.

No	Operator	Contoh
1	&&	(true && false) akan menghasilkan nilai FALSE
2		(true false) akan menghasilkan nilai TRUE
3	!	!(true && false) akan menghasilkan nilai TRUE

6. Operator Bitwise

Operator bitwise secara khusus digunakan untuk melakukan operasi dalam bentuk bit (biner). Beberapa operator bitwise antara lain :

No	Operator	Deskripsi
1	&	Operasi dalam bentuk bit dengan logika AND
2		Operasi dalam bentuk bit dengan logika OR
3	^	Operasi dalam bentuk bit dengan logika XOR
4	~	Operasi dalam bentuk bit dengan logika NOT/Negasi
5	>>	Pergeseran bit kekanan
6	<<	Pergeseran bit ke kiri

Uji coba program dengan Operator Aritmatika

```

1 public class OperatorAritmatika {
2     public static void main(String[] args) {
3         int A = 10;
4         int B = 5;
5         int kur, kur, bag, kul, mod;
6         kur = A/B;
7         bag = A%B;
8         kul = A*B;
9         mod = A%B;
10
11         System.out.println("penjumlahan : "+ kur);
12         System.out.println("pengurangan : "+ kur);
13         System.out.println("perkalian : "+ kul);
14         System.out.println("pembagian : "+ bag);
15         System.out.println("Mod : "+ mod);
16     }
17 }
  
```

```

C:\Windows\system32\cmd.exe
E:\File Riki\Semantara\program lap>javac OperatorAritmatika.java
E:\File Riki\Semantara\program lap>java OperatorAritmatika
penjumlahan : 15
pengurangan : 5
perkalian : 50
pembagian : 2
Mod : 0
E:\File Riki\Semantara\program lap>
  
```

Uji coba program dengan Operator Penugasan.

```

1 public class OperatorPenugasan {
2     public static void main(String[] args) {
3
4         int a = 10;
5         int b = 10;
6         //tanpa operator Penugasan
7         a = a + 20;
8         System.out.println(a);
9
10        //dengan operator penugasan
11        b += 20;
12        System.out.println(a);
13    }
14 }
  
```

```

C:\Windows\system32\cmd.exe
E:\File Riki\Semantara\program lap>javac OperatorPenugasan.java
E:\File Riki\Semantara\program lap>java OperatorPenugasan
30
30
E:\File Riki\Semantara\program lap>
  
```

Dengan program penugasan ini dapat dieksplorasi dengan symbol yang ada diatas.

MODUL IV

STRUKTUR KONTROL DALAM JAVA

1. Percabangan

Struktur Kontrol pada pemrograman java bertujuan untuk menentukan apa yang akan di lakukan oleh program selanjut nya berdasarkan kondisi yang ada saat itu.

Sebagai contoh sederhana nya pada suatu Aplikasi yaitu menu login

Dalam menu login biasa nya terdapat 2 kondisi, yaitu :

- Jika user dan password yang di input benar, maka aplikasi akan menampilkan menu selanjut nya
- Tetapi jika user atau password yang di input salah , maka aplikasi akan memunculkan warning atau pesan kesalahan.

Nah kira-kira seperti itu lah contoh kasus penggunaan struktur kontrol di dalam sebuah aplikasi. Sekarang mari kita mulai.

a. Statement IF

Menspesifikasikan sebuah statement (atau block dari code) yang akan dieksekusi jika dan hanya jika statement boolean bernilai true.

Form statement-if:

```
if( boolean_ekspresi )  
    statement;
```

atau

```
if( boolean_ekspresi ){  
    statement 1;  
    statement 2;  
}
```

dimana, boolean_ekspresi sama dengan boolean ekspresi atau boolean variabel.

b. Stattement IF-ELSE

Digunakan ketika kita akan mengeksekusi sebuah statement jika kondisinya true, dan statement yang lain jika berkondisi false.

Form statement if-else:

```
if( boolean_ekspresi ){  
    statement1;  
    statement2;  
    ...  
}  
else{  
    statement3;  
    statement4;  
    ...  
}
```

c. Statement IF-ELSE_IF

statement pada klausa else dari sebuah blok if-else dapat menjadi struktur if-else yang lain.

Struktur ini memperbolehkan kita untuk membuat pilihan yang lebih kompleks.

Form statement if-else-else if:

```
if( boolean_ekspresi1 ){
    statement1;}
else if( boolean_ekspresi2 ){
    statement2;}
else{
    statement3;}
```

d. Switch

Switch Memperbolehkan percabangan pada multiple outcomes.

Form statement-switch:

```
switch( switch_ekspresi ){
    case casepilihan1:
        statement1;//
        statement2;//blok 1
        break;
    case casepilihan2:
        statement1;//
        statement2;//blok 2
        break;
        :
    default:
        statement1;//
        statement2;//blok n
}
```

2. Perulangan

a. For

Digunakan untuk mengeksekusi code yang bernilai sama, berulang-ulang.

Form pengulangan-for:

```
for(InisialisasiEkspresi;KondisiPengulangan;StepEkspresi)
{
    statement1;
    statement2;
    ...
}
```

- InisialisasiEkspresi : meninisialisasi variabel pengulangan.
- KondisiPengulangan : membandingkan variabel pengulangan dengan nilai limit.
- StepEkspresi : memperbarui variabel pengulangan

b. While

Pengulangan while

Merupakan statement atau blok dari statement yang diulang selama kondisinya sesuai.

Form pengulangan while:

```
while( boolean_ekspresi ){  
    statement1;  
    statement2;  
    ...  
}
```

statement didalam pengulangan while akan dieksekusi selama boolean_ekspresi bernilai true.

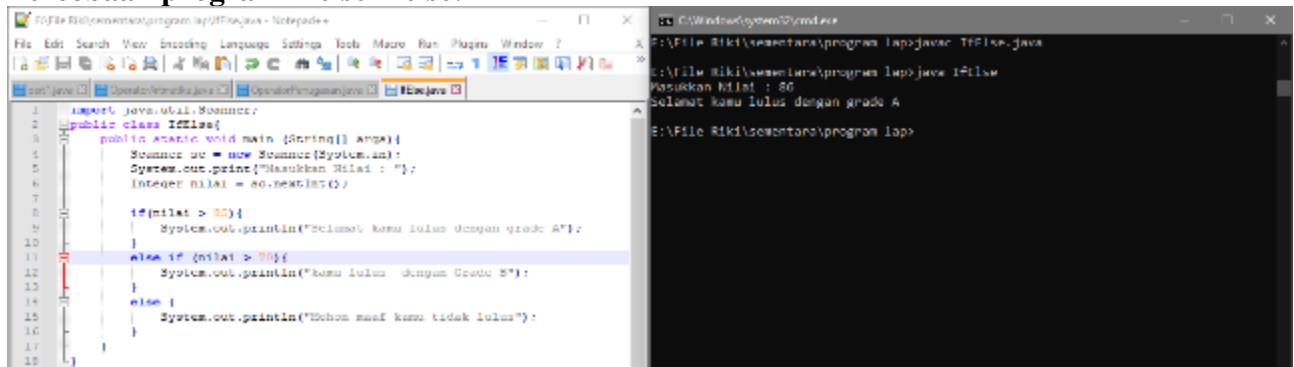
c. Do-While

- Sama dengan pengulangan-while
- Statement didalam pengulangan do-while akan dieksekusi beberapa kali selama kondisinya sesuai dengan ekspresi yang diberikan.
- Hal utama yang membedakan antara pengulangan while dan do-while:
- Statement didalam pengulangan do-while loop setidaknya dieksekusi satu kali.

Form pengulangan-do-while:

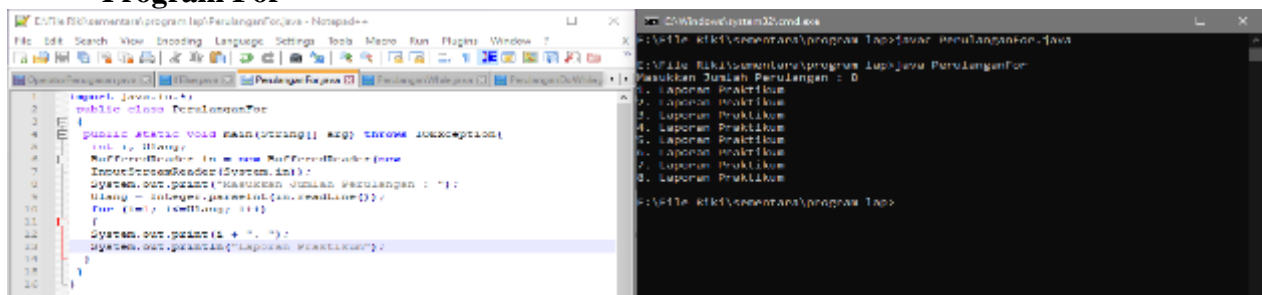
```
do{  
    statement1;  
    statement2;  
    ...  
}while( boolean_ekspresi );
```

Percobaan program if else if else:

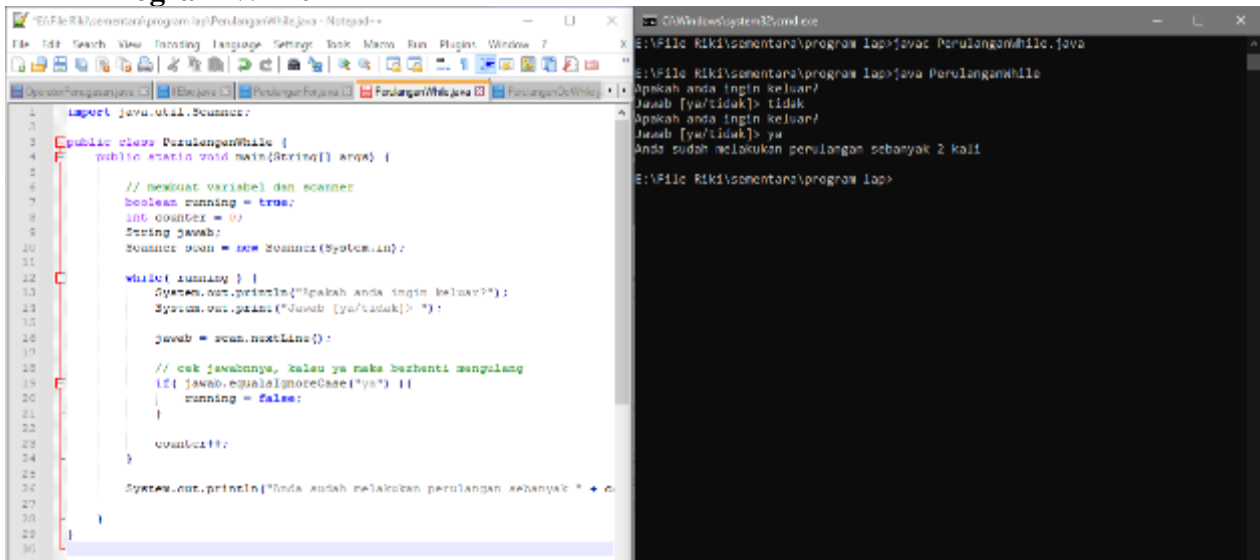


Percobaan Program dengan Perulangan

• Program For



• Program While



The screenshot shows a Java IDE with a file named 'PerulanganWhile.java'. The code is as follows:

```

1  import java.util.Scanner;
2
3  public class PerulanganWhile {
4      public static void main(String[] args) {
5
6          // membuat variabel dan scanner
7          boolean running = true;
8          int counter = 0;
9          String jawab;
10         Scanner scan = new Scanner(System.in);
11
12         while(running) {
13             System.out.println("Apakah anda ingin keluar?");
14             System.out.print("Jawab [ya/tidak]> ");
15
16             jawab = scan.nextLine();
17
18             // cek jawabannya, kalau ya maka berhenti mengulang
19             if (jawab.equalsIgnoreCase("ya")) {
20                 running = false;
21             }
22
23             counter++;
24         }
25
26         System.out.println("Anda sudah melakukan perulangan sebanyak " + counter);
27     }
28 }

```

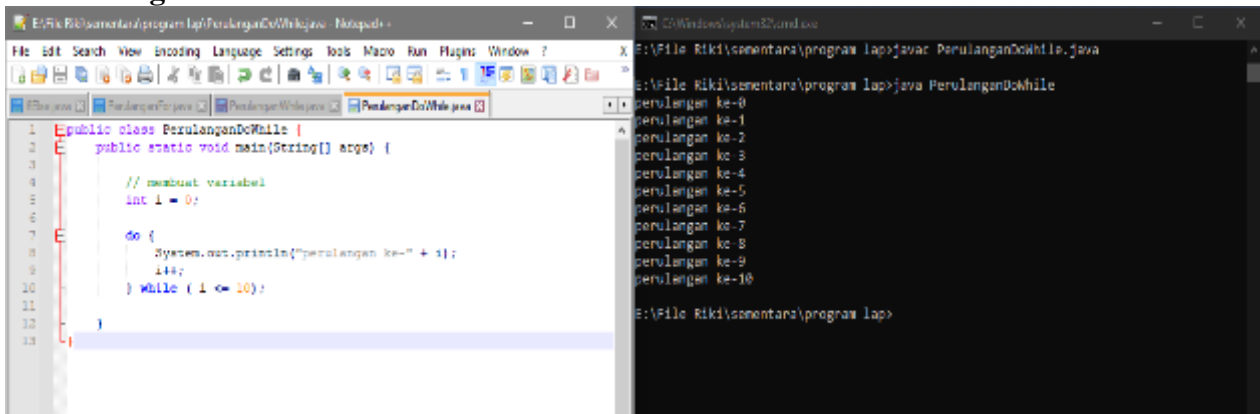
The terminal window shows the output of the program:

```

E:\File Riki\semantara\program lap>java PerulanganWhile.java
Apakah anda ingin keluar?
Jawab [ya/tidak]> tidak
Apakah anda ingin keluar?
Jawab [ya/tidak]> ya
Anda sudah melakukan perulangan sebanyak 2 kali
E:\File Riki\semantara\program lap>

```

• Program DoWhile



The screenshot shows a Java IDE with a file named 'PerulanganDoWhile.java'. The code is as follows:

```

1  public class PerulanganDoWhile {
2      public static void main(String[] args) {
3
4          // membuat variabel
5          int i = 0;
6
7          do {
8              System.out.println("perulangan ke-" + i);
9              i++;
10         } while (i <= 10);
11     }
12 }

```

The terminal window shows the output of the program:

```

E:\File Riki\semantara\program lap>java PerulanganDoWhile.java
perulangan ke-0
perulangan ke-1
perulangan ke-2
perulangan ke-3
perulangan ke-4
perulangan ke-5
perulangan ke-6
perulangan ke-7
perulangan ke-8
perulangan ke-9
perulangan ke-10
E:\File Riki\semantara\program lap>

```

MODUL V

ARRAY

1. Array

Array digunakan untuk menyimpan beberapa nilai dalam satu variabel, bukan mendeklarasikan variabel terpisah untuk setiap nilai. Untuk mendeklarasikan sebuah array, tentukan tipe variabel dengan tanda kurung siku. Biasanya, array adalah kumpulan elemen sejenis yang memiliki lokasi memori yang berdekatan.

a. Deklarasi Array Pada Java

Untuk menggunakan array dalam program, Anda harus mendeklarasikan variabel untuk mereferensikan array, dan Anda harus menentukan jenis array yang dapat direferensikan oleh variabel. Berikut adalah sintaks untuk mendeklarasikan variabel array

```
dataType[] arrayRefVar; // preferred way.  
or  
dataType arrayRefVar[]; // works but not preferred way.  
  
// contoh lainnya adalah  
double[] myList; // preferred way.  
Or  
double myList[]; // works but not preferred way.
```

Anda dapat membuat array dengan menggunakan operator baru dengan sintaks berikut

```
String[] cars;
```

Kami sekarang telah mendeklarasikan variabel yang menampung array string. Untuk memasukkan nilai ke dalamnya, kita dapat menggunakan literal array – tempatkan nilai dalam daftar yang dipisahkan koma, di dalam tanda kurung kurawal.

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Untuk membuat array bilangan bulat, Anda bisa menulis seperti ini

```
int[] myNum = { 10, 20, 30, 40};
```

Kemudian kita dapat membuat contoh sintaks lain dengan cara

```
arrayRefVar = new dataType[arraySize];
```

Pernyataan di atas melakukan dua hal yaitu

- Ini membuat sebuah array menggunakan dataType [arraySize] baru.
- Ini memberikan referensi dari array yang baru dibuat ke variabel arrayRefVar.

Mendeklarasikan variabel array, membuat array, dan menetapkan referensi array ke variabel dapat digabungkan dalam satu pernyataan, seperti yang ditunjukkan di bawah ini

```
dataType[] arrayRefVar = new dataType[arraySize];
```

Alternatifnya, Anda dapat membuat array sebagai berikut

```
dataType[] arrayRefVar = {value0, value1, ..., valuek};
```

Elemen array diakses melalui indeks. Indeks array berbasis 0; artinya, mereka mulai dari 0 hingga `arrayRefVar.length-1`. Pernyataan berikut mendeklarasikan variabel array, `myList`, membuat array 10 elemen tipe ganda dan memberikan referensinya ke `myList`

```
double[] myList = new double[10];
```

Gambar diatas mewakili array `myList`. Di sini, `myList` memiliki sepuluh nilai ganda dan indeksnya dari 0 hingga 9.

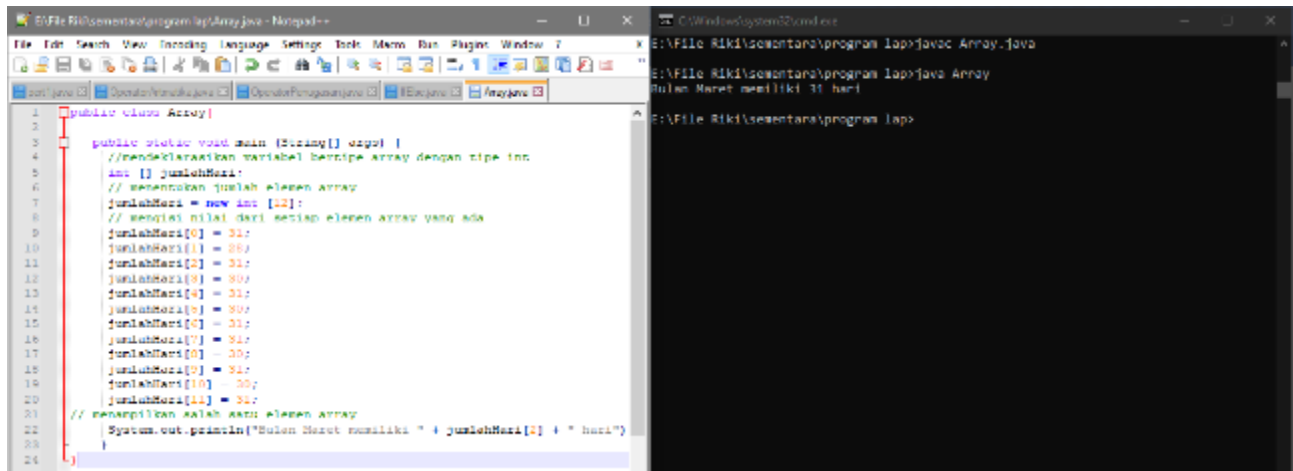
2. Percobaan Array Pada Notepad++

Dengan contoh koding sebagai berikut :

```
public class Array{

    public static void main (String[] args) {
        //mendeklarasikan variabel bertipe array
dengan tipe int
        int [] jumlahHari;
        // menentukan jumlah elemen array
        jumlahHari = new int [12];
        // mengisi nilai dari setiap elemen array
yang ada
        jumlahHari[0] = 31;
        jumlahHari[1] = 28;
        jumlahHari[2] = 31;
        jumlahHari[3] = 30;
        jumlahHari[4] = 31;
        jumlahHari[5] = 30;
        jumlahHari[6] = 31;
        jumlahHari[7] = 31;
        jumlahHari[8] = 30;
        jumlahHari[9] = 31;
        jumlahHari[10] = 30;
        jumlahHari[11] = 31;
        // menampilkan salah satu elemen array
        System.out.println("Bulan Maret
memiliki " + jumlahHari[2] + " hari");
    }
}
```

Hasil Program jika dijalankan



The image shows a screenshot of a Java IDE (NetBeans) with two windows. The left window displays the source code of a Java program, and the right window shows the output of the program.

```
1 public class Array {
2
3     public static void main (String[] args) {
4         //mendeklarasikan variabel bertipe array dengan tipe int
5         int [] jumlahHari;
6         // menentukan jumlah elemen array
7         jumlahHari = new int [12];
8         // mengisi nilai dari setiap elemen array yang ada
9         jumlahHari[0] = 31;
10        jumlahHari[1] = 28;
11        jumlahHari[2] = 31;
12        jumlahHari[3] = 30;
13        jumlahHari[4] = 31;
14        jumlahHari[5] = 30;
15        jumlahHari[6] = 31;
16        jumlahHari[7] = 31;
17        jumlahHari[8] = 30;
18        jumlahHari[9] = 31;
19        jumlahHari[10] = 30;
20        jumlahHari[11] = 31;
21        // menampilkan hasil dari elemen array
22        System.out.println("Bulan Maret memiliki " + jumlahHari[2] + " hari")
23    }
24 }
```

The right window shows the output of the program:

```
E:\File Riki\semantara\program lap>javac Array.java
E:\File Riki\semantara\program lap>java Array
Bulan Maret memiliki 31 hari
E:\File Riki\semantara\program lap>
```

MODUL VI

CLASS, METHOD, OBJEK DAN KONSTRUKTOR PADA JAVA

1. Class

Class merupakan blueprint/rancangan dari suatu objek. Class adalah gambaran umum dari suatu objek. Dikatakan bahwa Anjing adalah class, maka class Anjing memiliki karakteristik/atribute dan perilaku/behavioral yang dimiliki oleh anjing pada umumnya. Untuk lebih jelasnya perhatikan ilustrasi berikut:



Ilustrasi diatas memperlihatkan bahwa class Dog memiliki attribute (age) dan bahavioral (bark, eat, dan chase cat). Attribute dan bahavioral tersebut umum dimiliki oleh anjing sehingga setiap objek yang memiliki attribute dan bahavioral tersebut dikategorikan sebagai anjing. Di Java, untuk mendeklarasikan class menggunakan kata kunci “class” yang diikuti dengan nama class-nya.

Pada baris 1 dideklarasikan “class Dog”. Class Dog ini memiliki atribut “age” dan bahavioral “bark”, “eat”, dan “chaseCat” yang dideklarasikan di dalam kurung kurawal. Atribut dan bahavioral tersebut merupakan anggota/member dari class. Penjelasan detail tentang atribut dan bahavioral dibahas pada bahasan berikutnya.

Penamaan class mengikuti aturan penamaan seperti variable namun ada perbedaan sedikit, yaitu setiap kata harus diawali dengan huruf kapital/huruf besar. Misal, class “mydog” terdiri dari kata “my” dan “dog” maka huruf “m” pada “my” huruf kapital dan huruf “d” pada “dog” harus kapital juga. Dari “mydog” menjadi “MyDog”. Hal ini dimaksud supaya nama suatu class itu mudah untuk dibaca.

2. Method

Method merupakan sebutan untuk bahavioral/function di Java. Method selalu memiliki kurung lengkung atau “()”, kurung lengkung tersebut bisa juga disemati suatu variable atau parameter. Parameter sendiri adalah sebutan dari variabel yang terletak dalam kurung lengkung suatu method. Aturan penamaan method sama dengan aturan penamaan variable. Pembahasan tersebut bisa dilihat di penjelasan tentang variabel.

Method sendiri ada dua jenis: (1) void dan (2) return.

a. Void Method

void method adalah method yang tidak mengembalikan suatu nilai. Dilihat sekilas, void method dapat diidentifikasi dengan adanya kata kunci “void” di depan nama method.

```
void eat(){  
    System.out.println("anjing makan");  
}
```

Ilustrasi di atas adalah method void karena secara kasat mata method tersebut di depan nama method-nya (eat) memiliki kata kunci “void”. Void bisa disebut dengan method tidak mengembalikan nilai karena method ini ketika dipanggil tidak memiliki nilai yang bisa disimpan di suatu variabel.

b. Return Method

Return method adalah method yang mengembalikan nilai. Method ini bisa diidentifikasi dengan adanya data type di depan nama method-nya dan kata kunci return di dalam method-nya.

```
int getAge(){  
    return 3;  
}
```

Pada baris 1 di depan nama method, “getAge” terdapat data type “int” sehingga di dalam method ini harus mengembalikan “int” juga. Kata kunci “return” digunakan untuk mengembalikan nilai. Dilihat pada baris 2, method “getAge” ini mengembalikan nilai 3, nilai 3 tersebut adalah nilai yang ber-data type “int”.

3. Objek

Object adalah representasi dari class. Katakan bahwa Anjing adalah class-nya maka doggy, pretty, dan sweetie adalah objectnya. Class masih berupa blueprint/rancangan sedangkan object adalah wujud nyatanya. Cara mendeklarasikan object sebagai berikut:



Untuk membuat object terlebih dahulu harus tahu class yang akan dibuatkan object-nya. Pada ilustrasi di atas “Dog” adalah class yang akan dibuatkan object-nya. Object dari class “Dog” tersebut bernama “pretty”. Berikutnya, untuk benar-benar “pretty” adalah object dari class “Dog” maka pada ilustrasi diatas ditambahkan assignment operator (=) yang digunakan untuk memberikan nilai object “pretty” ini dengan object baru dari class Dog, yaitu caranya dengan menambahkan kata kunci “new” yang diikuti dengan “constructor Dog“ / “Dog()”, constructor akan dibahas pada bahasan berikutnya. Sedangkan, penamaan object mengikuti penamaan pada variable. Sekali object tersebut dibuat, semua anggota (attribute dan method) bisa diakses oleh object tersebut.

Percobaan Program dengan nama Dog.java:

```
package Class;  
class Dog {  
    int age;  
  
    int getAge(){  
        return 3;  
    }  
    void bark(){  
        System.out.println("anjing menggongong");  
    }  
}
```

```

void eat(){
System.out.println("anjing makan");
}

void chaseCat(){
System.out.println("anjing mengejar
kucing");
}
}

```

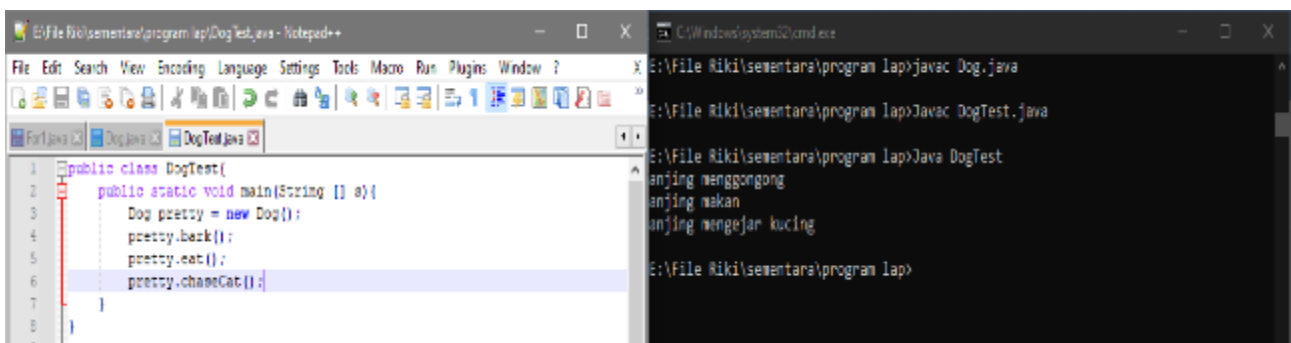
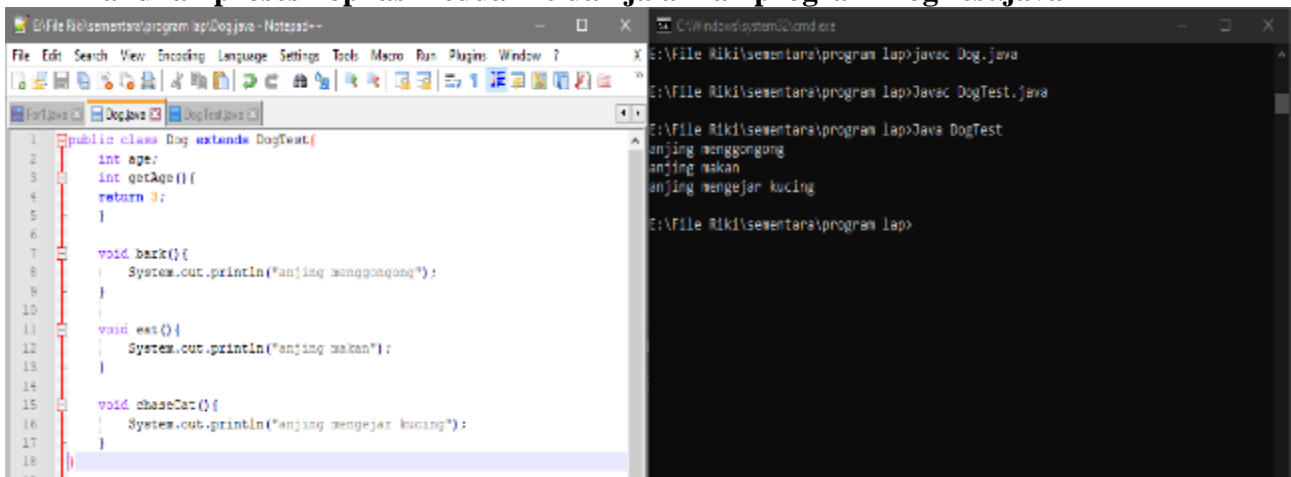
Lalu buat Program baru dengan nama DogTest.java

```

package Class;
public class DogTest {
    public static void main(String [] args){
        Dog pretty = new Dog();
        pretty.bark();
        pretty.eat();
        pretty.chaseCat();
    }
}

```

Lakukan proses kompilasi kedua file dan jalankan program DogTest.java



4. Konstruktor

Constructor adalah method yang memiliki nama yang sama dengan nama class-nya. Biasanya

method ini digunakan untuk memberikan nilai inisiasi dari suatu class. Setiap kali membuat class secara otomatis sistem membuat constructor secara implisit. Sebenarnya, method ini sering kali dipanggil sekali setiap kali membuat object baru.

```
class Dog {  
  
}  
class Dog {  
    Dog(){}  
}
```

Ilustrasi pertama dan kedua jika dihasilkan keluarannya sama. Yang pertama constructor tidak didefinisikan secara eksplisit sedangkan yang kedua constructor didefinisikan secara implisit.

```
Dog pretty = new Dog();  
class Dog {  
  
    String name;  
  
    Dog(){}  
  
    Dog(String name){  
        this.name = name;  
    }  
}
```

Ilustrasi di atas menunjukkan constructor yang overloading, sehingga ketika membuat object baru, programmer bisa memilih constructor yang baris ke 5 atau baris ke 7, seperti yang diilustrasikan sebagai berikut:

```
public class DogTest {  
    public static void main(String [] args){  
        Dog pretty = new Dog();  
        Dog doggy = new Dog("Doggy");  
    }  
}
```

Pada baris 3 memanggil constructor yang tidak memiliki parameter sedangkan pada baris 4 memanggil constructor yang memiliki parameter.

MODUL VII

PEWARISAN DAN POLIMORPISME PADA JAVA

1. Pewarisan

inheritance (penurunan sifat / pewarisan), ini merupakan ciri khas dari OOP yang tidak terdapat pada pemrograman prosedural gaya lama. Dalam hal ini, inheritance bertujuan membentuk obyek baru yang memiliki sifat sama atau mirip dengan obyek yang sudah ada sebelumnya (pewarisan). Obyek turunan dapat digunakan membentuk obyek turunan lagi dan seterusnya. Setiap perubahan pada obyek induk, juga akan mengubah obyek turunannya. Susunan obyek induk dengan obyek turunannya disebut dengan hirarki obyek.

Atau Inheritance yaitu pewarisan sifat-sifat suatu object kepada object turunannya.

Percobaan Program dengan nama Dog.java:

```
package Class;
class Dog {
    int age;

    int getAge(){
        return 3;
    }
    void bark(){
        System.out.println("anjing menggongong");
    }

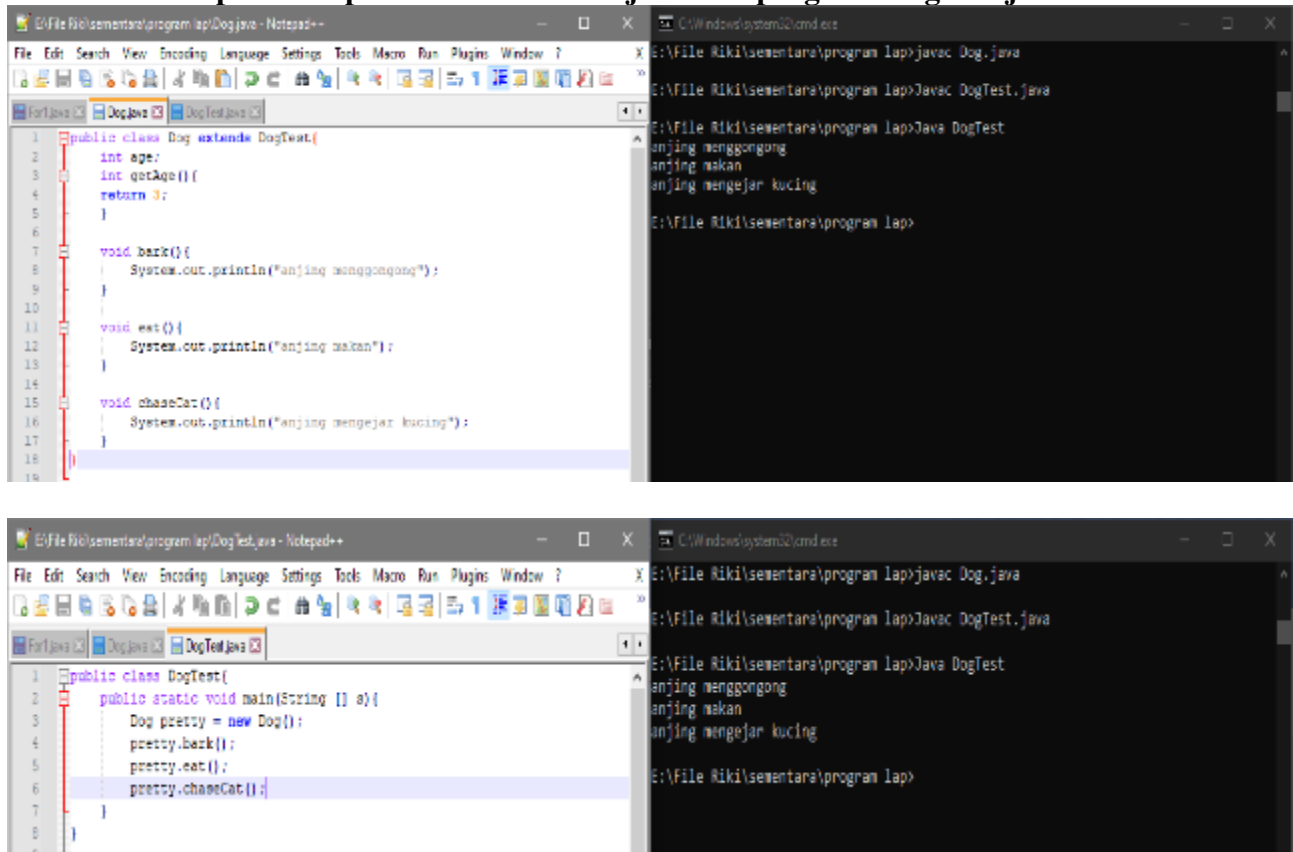
    void eat(){
        System.out.println("anjing makan");
    }

    void chaseCat(){
        System.out.println("anjing mengejar kucing");
    }
}
```

Lalu buat Program baru dengan nama DogTest.java

```
package Class;
public class DogTest {
    public static void main(String [] args){
        Dog pretty = new Dog();
        pretty.bark();
        pretty.eat();
        pretty.chaseCat();
    }
}
```

Lakukan proses kopilasi kedua file dan jalankan program DogTest.java



The first screenshot shows the IDE with two files: Dog.java and DogTest.java. Dog.java contains methods for getting age, barking, eating, and chasing cats. DogTest.java contains a main method that creates a Dog object and calls these methods. The command prompt shows the compilation of both files and the execution of DogTest.java, which produces the output: anjing menggonggong, anjing makan, and anjing mengejar kucing.

```
public class Dog extends DogTest{
    int age;
    int getAge(){
        return 3;
    }

    void bark(){
        System.out.println("anjing menggonggong");
    }

    void eat(){
        System.out.println("anjing makan");
    }

    void chaseCat(){
        System.out.println("anjing mengejar kucing");
    }
}

public class DogTest{
    public static void main(String [] a){
        Dog pretty = new Dog();
        pretty.bark();
        pretty.eat();
        pretty.chaseCat();
    }
}
```

```
E:\File Riki\sementara\program lap>javac Dog.java
E:\File Riki\sementara\program lap>javac DogTest.java
E:\File Riki\sementara\program lap>java DogTest
anjing menggonggong
anjing makan
anjing mengejar kucing
E:\File Riki\sementara\program lap>
```

2. Polimorpisme

Polymorphism, suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, di mana obyek yang berbeda memberikan tanggapan/respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek.

Atau Polymorphic dapat berarti banyak bentuk, maksudnya yaitu kita dapat menimpa (override), suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda.

Contoh program java nya Simpan dengan nama BentukWajah.java

```
import java.*;
import java.io.*;
/*class induk dengan nama Bentukwajah*/
class Bentukwajah{
    /*Membuat sebuah method respons pada class induk dimana mengembalikan
    *sebuah nilai String*/
    public String respons()
    {
        return("Perhatikan reaksi wajah saya\n");
    }
}
/*Membuat class-class lain dengan turunan dari class Bentukwajah*/
class Senyum extends Bentukwajah{
```



```

    public String respons()
    {
        return("Senyum karena senang\n");
    }
}
class Tertawa extends Bentukwajah{
    public String respons()
    {
        return("Tertawa karena gembira\n");
    }
}
class Marah extends Bentukwajah{
    public String respons()
    {
        return("kemarahan disebabkan bertangkar\n");
    }
}
class Sedih extends Bentukwajah{
    public String respons()
    {
        return("Sedih disebabkan cemburu\n");
    }
}
}

```

Simpan dengan nama Programpolimorphis.java

```

//program utama
public class Programpolimorphis
{
    /**Main method*/
    public static void main(String []args)
    {
        System.out.println("\n Program Polimorphis");
        System.out.println("=====");
        //membuat objek-objek raut(ekspresi) dari wajah
        Bentukwajah objbentuk=new Bentukwajah();
        Senyum objsenyum=new Senyum();
        Tertawa objtertawa=new Tertawa();
        Marah objmarah=new Marah();
        Sedih objsedih=new Sedih();
        //polimorphis dari bentuk wajah ada 5 dimulai dari 0-4
        Bentukwajah []Bentuk=new Bentukwajah[5];
        Bentuk[0]=objbentuk;
        Bentuk[1]=objsenyum;
        Bentuk[2]=objtertawa;
        Bentuk[3]=objmarah;
        Bentuk[4]=objsedih;
        System.out.println("Bentuk[0]:"+Bentuk[0].respons());
        System.out.println("Bentuk[1]:"+Bentuk[1].respons());
    }
}

```

```

        System.out.println("Bentuk[2]:"+Bentuk[2].respons());
        System.out.println("Bentuk[3]:"+Bentuk[3].respons());
        System.out.println("Bentuk[4]:"+Bentuk[4].respons());
    }
}

```

Kemudian compile dan run.

```

//File Riki\sementara\program lap\BentukWajah.java - Notepad++
1  import java.io.*;
2  import java.io.*;
3  /**class induk dengan nama BentukWajah*/
4  class BentukWajah{
5      /**Bentuk sebuah method respons pada class induk dimana menyumbangkan
6       *sebuah nilai String*/
7      public String respons()
8      {
9          return("Perhatikan reaksi wajah saya");
10     }
11 }
12
13 /**Mewakili class-class lain dengan turunan dari class BentukWajah*/
14 class Senyum extends BentukWajah{
15     public String respons()
16     {
17         return("Senyum karena senang");
18     }
19 }
20
21 class Tertawa extends BentukWajah{
22     public String respons()
23     {
24         return("Tertawa karena gembira");
25     }
26 }
27
28 class Marah extends BentukWajah{
29     public String respons()
30     {
31         return("Marahan disebabkan bertengkar");
32     }
33 }
34
35 class Sedih extends BentukWajah{
36     public String respons()
37     {
38         return("Sedih disebabkan cemburu");
39     }
40 }
41
length: 583 lines: 36 lns: 31 cols: 82 pos: 790 Windows (CRLF) UTF-8

```

```

//File Riki\sementara\program lap\ProgramPolimorfis.java - Notepad++
1  import java.io.*;
2  import java.io.*;
3  /**class induk dengan nama BentukWajah*/
4  class BentukWajah{
5      /**Bentuk sebuah method respons pada class induk dimana menyumbangkan
6       *sebuah nilai String*/
7      public String respons()
8      {
9          return("Perhatikan reaksi wajah saya");
10     }
11 }
12
13 /**Mewakili class-class lain dengan turunan dari class BentukWajah*/
14 class Senyum extends BentukWajah{
15     public String respons()
16     {
17         return("Senyum karena senang");
18     }
19 }
20
21 class Tertawa extends BentukWajah{
22     public String respons()
23     {
24         return("Tertawa karena gembira");
25     }
26 }
27
28 class Marah extends BentukWajah{
29     public String respons()
30     {
31         return("Marahan disebabkan bertengkar");
32     }
33 }
34
35 class Sedih extends BentukWajah{
36     public String respons()
37     {
38         return("Sedih disebabkan cemburu");
39     }
40 }
41
length: 583 lines: 36 lns: 31 cols: 82 pos: 790 Windows (CRLF) UTF-8

```

```

E:\File Riki\sementara\program lap>javac BentukWajah.java
E:\File Riki\sementara\program lap>javac ProgramPolimorfis.java
E:\File Riki\sementara\program lap>java ProgramPolimorfis
Program Polimorfis
Bentuk[0]:Perhatikan reaksi wajah saya
Bentuk[1]:Senyum karena senang
Bentuk[2]:Tertawa karena gembira
Bentuk[3]:Marahan disebabkan bertengkar
Bentuk[4]:Sedih disebabkan cemburu
E:\File Riki\sementara\program lap>

```

```

//File Riki\sementara\program lap\ProgramPolimorfis.java - Notepad++
1  import java.io.*;
2  import java.io.*;
3  /**class induk dengan nama BentukWajah*/
4  class BentukWajah{
5      /**Bentuk sebuah method respons pada class induk dimana menyumbangkan
6       *sebuah nilai String*/
7      public String respons()
8      {
9          return("Perhatikan reaksi wajah saya");
10     }
11 }
12
13 /**Mewakili class-class lain dengan turunan dari class BentukWajah*/
14 class Senyum extends BentukWajah{
15     public String respons()
16     {
17         return("Senyum karena senang");
18     }
19 }
20
21 class Tertawa extends BentukWajah{
22     public String respons()
23     {
24         return("Tertawa karena gembira");
25     }
26 }
27
28 class Marah extends BentukWajah{
29     public String respons()
30     {
31         return("Marahan disebabkan bertengkar");
32     }
33 }
34
35 class Sedih extends BentukWajah{
36     public String respons()
37     {
38         return("Sedih disebabkan cemburu");
39     }
40 }
41
length: 583 lines: 36 lns: 31 cols: 82 pos: 790 Windows (CRLF) UTF-8

```

```

//File Riki\sementara\program lap\ProgramPolimorfis.java - Notepad++
1  import java.io.*;
2  import java.io.*;
3  /**class induk dengan nama BentukWajah*/
4  class BentukWajah{
5      /**Bentuk sebuah method respons pada class induk dimana menyumbangkan
6       *sebuah nilai String*/
7      public String respons()
8      {
9          return("Perhatikan reaksi wajah saya");
10     }
11 }
12
13 /**Mewakili class-class lain dengan turunan dari class BentukWajah*/
14 class Senyum extends BentukWajah{
15     public String respons()
16     {
17         return("Senyum karena senang");
18     }
19 }
20
21 class Tertawa extends BentukWajah{
22     public String respons()
23     {
24         return("Tertawa karena gembira");
25     }
26 }
27
28 class Marah extends BentukWajah{
29     public String respons()
30     {
31         return("Marahan disebabkan bertengkar");
32     }
33 }
34
35 class Sedih extends BentukWajah{
36     public String respons()
37     {
38         return("Sedih disebabkan cemburu");
39     }
40 }
41
length: 583 lines: 36 lns: 31 cols: 82 pos: 790 Windows (CRLF) UTF-8

```

```

E:\File Riki\sementara\program lap>javac BentukWajah.java
E:\File Riki\sementara\program lap>javac ProgramPolimorfis.java
E:\File Riki\sementara\program lap>java ProgramPolimorfis
Program Polimorfis
Bentuk[0]:Perhatikan reaksi wajah saya
Bentuk[1]:Senyum karena senang
Bentuk[2]:Tertawa karena gembira
Bentuk[3]:Marahan disebabkan bertengkar
Bentuk[4]:Sedih disebabkan cemburu
E:\File Riki\sementara\program lap>

```

MODUL VIII

GUI DENGAN NETBEANS

1. Install Netbeans dan JDK

Silahkan anda download dulu Netbeans dan JDK nya di sini

Download sesuai sistem operasi yang kalian gunakan di komputer atau laptop kalian

a. Java Development Kit(JDK) pada link berikut ini :

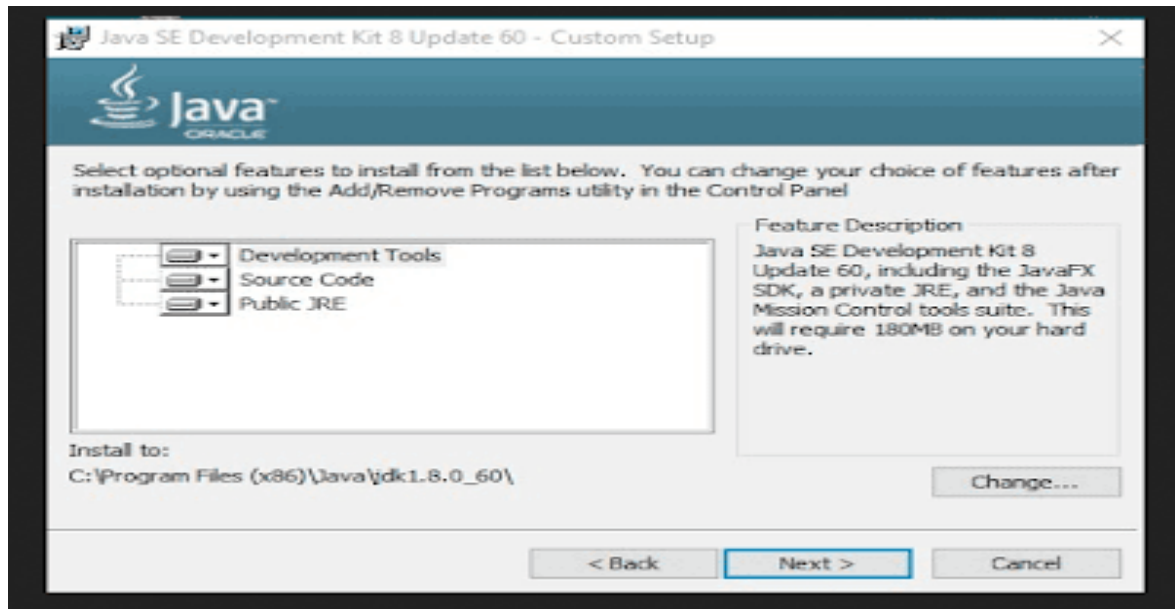
<https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

b. Netbeans 8.2 pada link berikut ini : <https://netbeans.org/downloads/?pagelang=pt> _

INSTALL Java Development Kit(JDK)

Klik dua kali file JDK yang sudah kalian download, tinggal ikuti proses instalasinya, silahkan klik Next setiap prosesnya

Kalau masih kurang jelas bisa lihat gambar di bawah ini

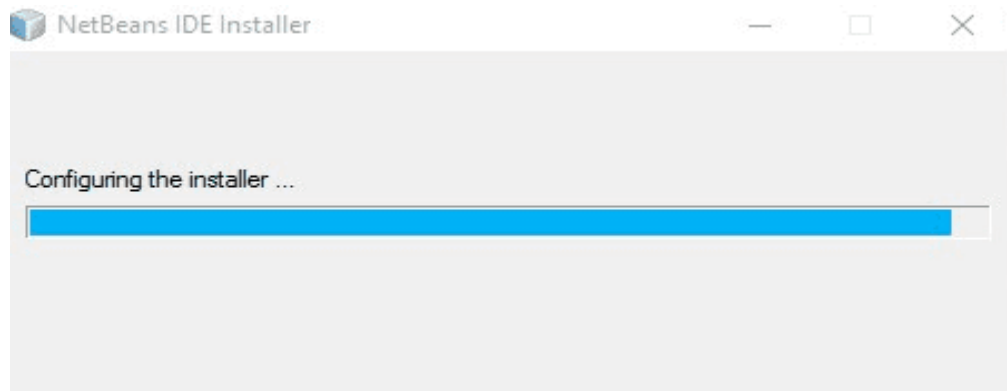


Lalu klik next dan install seperti installer biasa.

INSTALL Netbeans 8.2

Langkah penginstalannya adalah sebagaiberikut:

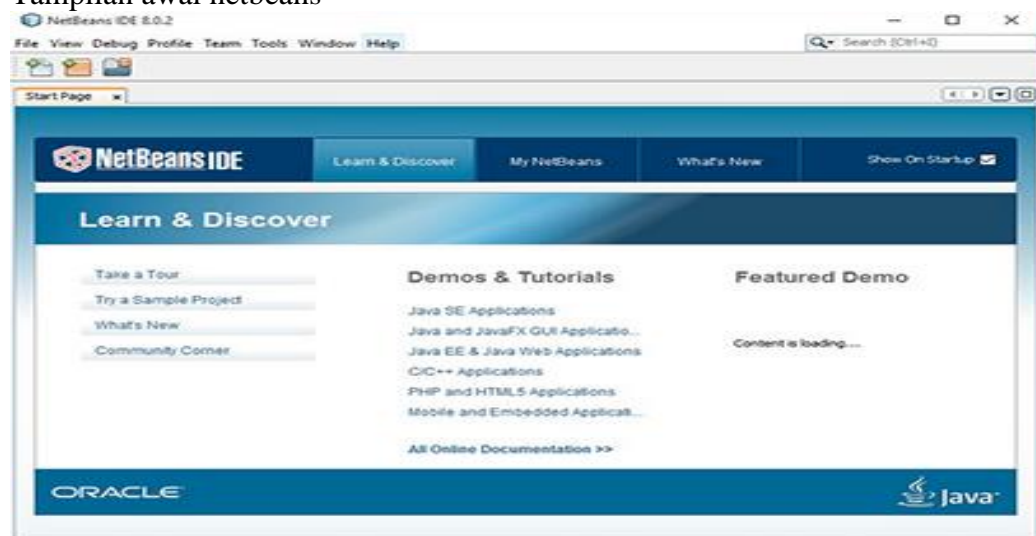
- Silahkan buka file netbeans klik dua kali akan muncul proses di bawah ini, silahkan tunggu sampai selesai.



ii. Klik Next untuk melanjutkan



- iii. Silahkan centang dan klik Next untuk melanjutkan
- iv. Silahka pilih I accept dan lalu pilih Next untuk melanjutkan
- v. Silahkan pilih letak instalasi netbeans dan arahka letak JDK ke folder instalasi JDK lalu klik Next untuk melanjutkan
- vi. Silahka tunggu proses hingga selesai lalu klik Next
- vii. Klik finish untuk membuka aplikasi netbeans
- viii. Tampilan awal netbeans

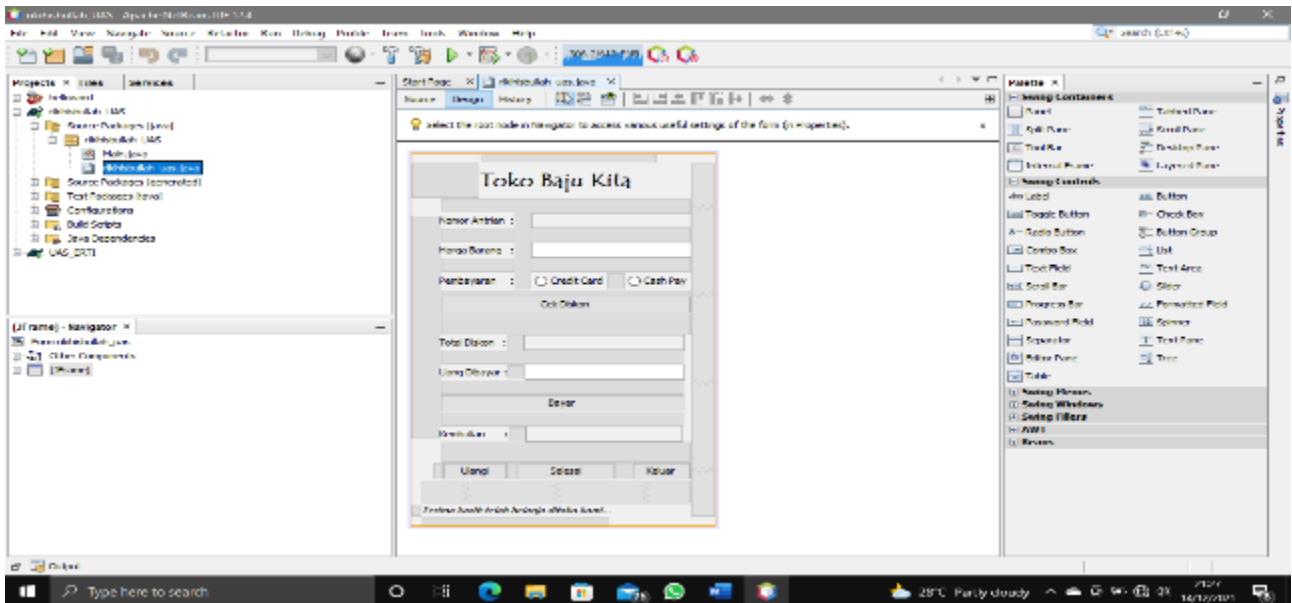


2. Testing Sebuah Program dengan Gui Netbeans

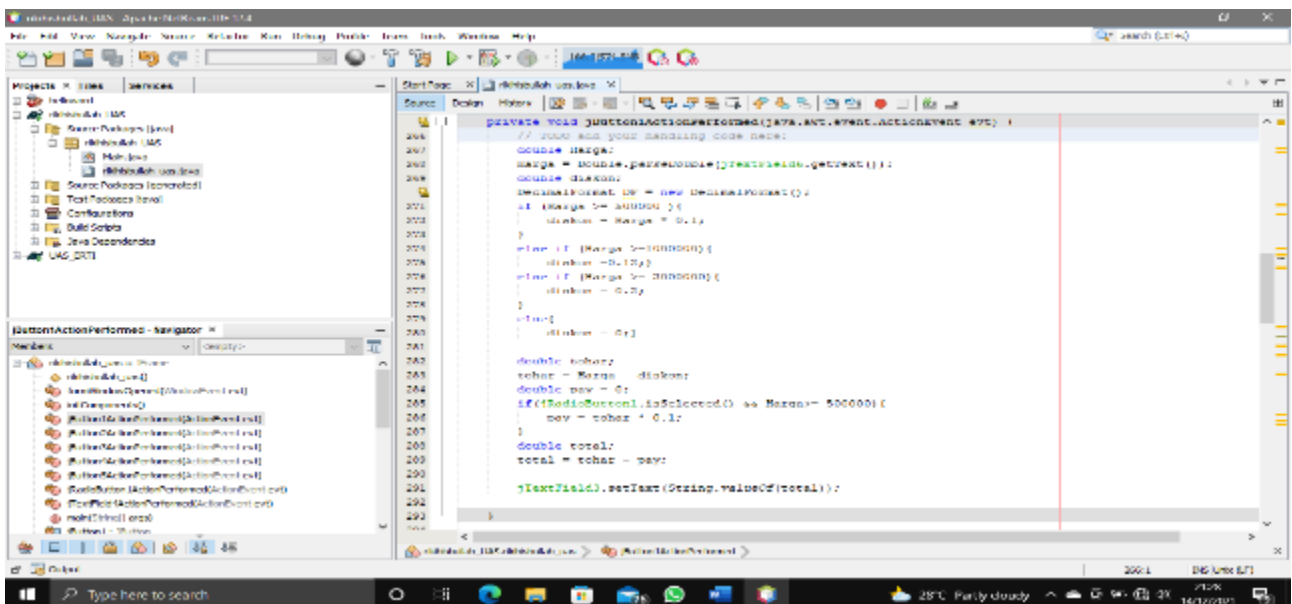
Dibuat sebuah Projek Java dengan nama bebas, dengan algoritma program Menghitung Diskon sebagai berikut:

- jika membeli >500000 mendapat potongan 10%
- jika membeli >1000000 mendapat potongan 12%
- jika membeli >2000000 mendapat potongan 20%
- jika menggunakan kartu kredit mendapat tambahan 10% diskon dari harga setelah dipotong

Sehingga didesain tampilannya sebagai berikut :



Dengan koding pada buton pertama adalah :



Dan pada buton/tombol kedua, adalah sebagai berikut :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    double Tot;  
    Tot = Double.parseDouble(jTextField3.getText());  
    double Uang;  
    Uang = Double.parseDouble(jTextField4.getText());  
    double kembalian;  
  
    kembalian = Uang - Tot;  
  
    jTextField5.setText(String.valueOf(kembalian));  
}
```

Lalu pada buton/tombol Ulang adalah :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    jTextField3.setText(String.valueOf(""));  
    jTextField4.setText(String.valueOf(""));  
    jTextField5.setText(String.valueOf(""));  
    jTextField6.setText(String.valueOf(""));  
}
```

Pada Buton/Tombol Selesai, kodingnya adalah :

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int i;  
    i= Integer.valueOf(jTextField1.getText());  
    i+=1;  
    jTextField1.setText(String.valueOf(i));  
    jTextField3.setText(String.valueOf(""));  
    jTextField4.setText(String.valueOf(""));  
    jTextField5.setText(String.valueOf(""));  
    jTextField6.setText(String.valueOf(""));  
}
```

Dan pada buton/tombol keluae adalah :

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

Maka ketika dijalankan namun dengan nilai 200000 akan tidak mendapatkan diskon sama sekali

Toko Baju Kita

Nomor Antrian : 1

Harga Barang : 200000

Pembayaran : ☒ Credit Card ☐ Cash Pay

Cek Diskon

Total Diskon : 200000.0

Uang Dibayar : 200000

Bayar

Kembalian : 0.0

Ulangi Selesai Keluar

Terima kasih telah belanja ditoko kami...

Toko Baju Kita

Nomor Antrian : 2

Harga Barang : 501000

Pembayaran : ☒ Credit Card ☐ Cash Pay

Cek Diskon

Total Diskon : 405810.0

Uang Dibayar : 406000

Bayar

Kembalian : 190.0

Ulangi Selesai Keluar

Terima kasih telah belanja ditoko kami...

*jika total belanja yang dimasukkan >500000.

:

PENUTUP

1. Kesimpulan

Pada praktikum kali ini dipaparkan materi dari dosen sebagai bahan ajar. Lalu setiap mahasiswa diminta untuk mempraktikkan dengan 2 aplikasi sebagai kompilernya. Yang pertama yaitu notepad++ sebagai media compiler dan Run file java nya. Yang kedua menggunakan aplikasi Gui berupa Netbeans IDE.

2. Saran

Diharap setiap materi yang disampaikan agar dapat disederhanakan terlebih dahulu sebelum disampaikan kepada mahasiswa, sehingga materi dapat dengan mudah diterima.

LEMBAR PENGESAHAN

Judul : LAPORAN PRAKTIKUM BASIS DATA
Nama : RIKI HISBULLAH
NIM : 0701192023
Program Studi : ILMU KOMPUTER 2

Medan, 15 Nov. 2021

Menyetujui,

KoordinatorAsisten

Asisten

() ()

Mengetahui,

Dosen Penanggung Jawab

()