

JOB SHEET

PEMROGRAMAN BERORIENTASI OBJEK WITH "JAVA"

(Teori & Implementasi Java)

Java adalah salah satu Pemrograman Berorientasi Obyek yang layak untuk dipelajari, karena Java mempunyai kemampuan dapat berjalan diberbagai platform dan memiliki fitur yang sesuai dengan kebutuhan para pengembang perangkat lunak serta bersifat *open source*. Oleh karena itu java banyak dipakai untuk kepentingan pembelajaran khususnya dibidang informatika.

Dalam buku JOBSHEET java dasar ini, penulis memberikan banyak latihan untuk bisa langsung dipraktekkan dengan percobaan mandiri secara sederhana dilengkapi dengan sedikit teori penunjang untuk mengerjakan latihan-latihan. Banyak latihan sederhana dan mudah untuk dipraktekkan bagi pengguna yang awam dengan bahasa pemrograman yang berorientasi obyek. Namun dengan panduan yang singkat, jelas dan sederhana ini mudah-mudahan bisa membantu dalam menyelesaikan program dengan Java.



JOB SHEET

PEMROGRAMAN BERORIENTASI OBJEK WITH "JAVA"

(Teori & Implementasi Java)



SEKRETINGSIH NITA

JOB SHEET
PEMROGRAMAN BERORIENTASI OBJEK
WITH "JAVA"
(Teori & Implementasi Java)

Sekreningsih Nita



**JOB SHEET PEMROGRAMAN BERORIENTASI OBJEK WITH “JAVA”
(Teori & Implementasi Java)**

Penulis:

Sekreningsih Nita

Editor:

Sekreningsih Nita

Perancang Sampul:

Sekreningsih Nita

Penata Letak:

Tim Kreatif UNIPMA Press

Cetakan Pertama, Desember 2021

Diterbitkan Oleh:

UNIPMA Press Universitas PGRI Madiun

Jl. Setiabudi No. 85 Madiun Jawa Timur 63118

E-Mail: upress@unipma.ac.id

Website: kwu.unipma.ac.id

Anggota IKAPI: No. 207/Anggota Luar Biasa/JTI/2018

ISBN: 978-623-6318-43-0

Hak Cipta dilindungi oleh Undang-Undang

All right reserved

PRAKATA

Java adalah salah satu Pemrograman Berorientasi Obyek yang layak untuk dipelajari, karena Java mempunyai kemampuan dapat berjalan diberbagai platform dan memiliki fitur yang sesuai dengan kebutuhan para pengembang perangkat lunak serta bersifat *open source*. Oleh karena itu java banyak dipakai untuk kepentingan pembelajaran khususnya dibidang informatika.

Dalam buku JOBSHEET java dasar ini, penulis memberikan banyak latihan untuk bisa langsung dipraktekkan dengan percobaan mandiri secara sederhana dilengkapi dengan sedikit teori penunjang untuk mengerjakan latihan-latihan. Banyak latihan sederhana dan mudah untuk dipraktekkan bagi pengguna yang awam dengan bahasa pemrograman yang berorientasi obyek. Namun dengan panduan yang singkat, jelas dan sederhana ini mudah-mudahan bisa membantu dalam menyelesaikan program dengan Java.

Penulis menyadari buku JOHSHEET JAVA perdana ini belum sempurna, oleh karena itu perlu adanya masukan dari pengguna. Kami sangat terbuka untuk menerima semua masukan. Semoga bisa lebih baik lagi untuk penerbitan buku JOBSHEET berikutnya.

Insya Allah, Amin YRA...

Madiun, Oktober 2021

(Sekreningsih Nita)

DAFTAR ISI

PRAKATA	iii
DAFTAR ISI	v
BAB 1 PENGENALAN DASAR JAVA	1
BAB 2 STRUKTUR KONDISI (IF, NESTED IF, SWITCH)	10
BAB 3 STRUKTUR PERULANGAN	22
BAB 4 STRUKTUR BREAK DAN CONTINUE	34
BAB 5 STRUKTUR TIPE DATA ARRAY	42
BAB 6 PROCEDURE TANPA PARAMETER	55
BAB 7 PROCEDURE DENGAN PARAMETER	61
BAB 8 FUNCTION TANPA PARAMETER	68
BAB 9 FUNCTION DENGAN PARAMETER	77
BAB 10 REKURSIF	83
BAB 11 INHERITANCE (PEWARISAN)	92
BAB 12 ENCAPSULATION	102
BAB 13 POLYMORPHISM	110
DAFTAR PUSTAKA	121
GLOSARIUM	122
INDEKS	125
BIOGRAFI PENULIS	128

BAB 1

PENGENALAN DASAR JAVA

TUJUAN:

1. Dapat melakukan instalasi dan konfigurasi path untuk membuat program Java.
2. Struktur Penulisan Java.
3. Mengetahui Perintah Dasar Input dan Output pada Java.
4. Mengetahui fungsi operator aritmatika pada Java.
5. Mampu mengoperasikan operator aritmatika pada Java.

PERSIAPAN:

1. Perangkat lunak Java JDK 1.6.0_25.
2. Perangkat Lunak Command Prompt.
3. Perangkat lunak Notepad.

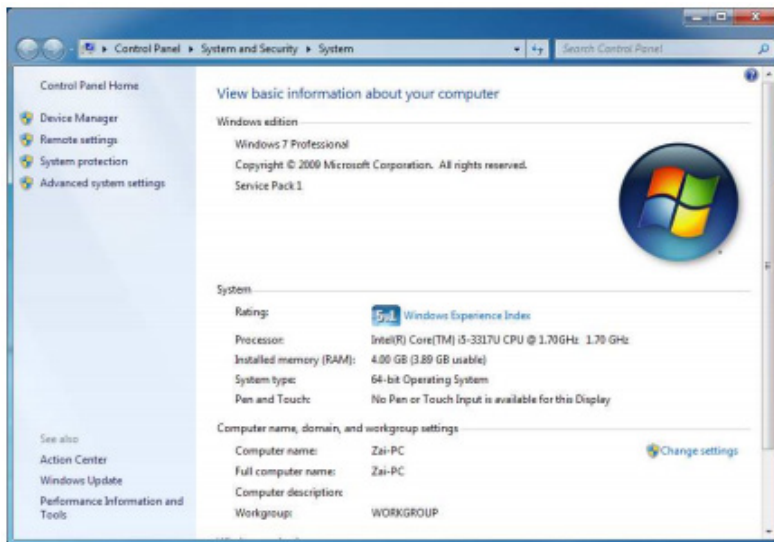
DASAR TEORI:

1. Instalasi dan konfigurasi path untuk membuat program Java.

Langkah pertama dalam pembuatan program Java adalah meng-install program JDK, diusahakan memakai JDK versi terbaru. Atau bisa juga melalui link seperti berikut :: <http://www.oracle.com/technetwork/java/javase/downloads/java-sejdk-7-download-432154.html>.

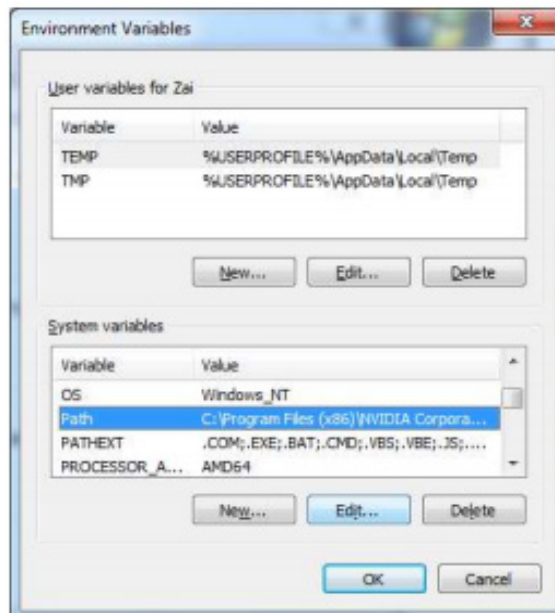
Langkah kedua, diperlukan pengaturan pada JDK tersebut tujuannya agar program java dapat di compile dan dijalankan melalui command prompt atau melalui cmd. Pengaturan jdk dalam java tersebut dinamakan CLASSPATH.

Adapun cara bagaimana men-setting CLASSPATH di windows 7 adalah pertama pilih / klik Computer kemudian lakukan klik kanan dan klik Properties, berikutnya akan tampil seperti pada gambar berikut. Pada gambar di bawah ini menunjukkan langkah-langkah berupa windows properties, windows environment variables dan administrator classpath yang digunakan untuk setting konfigurasinya.



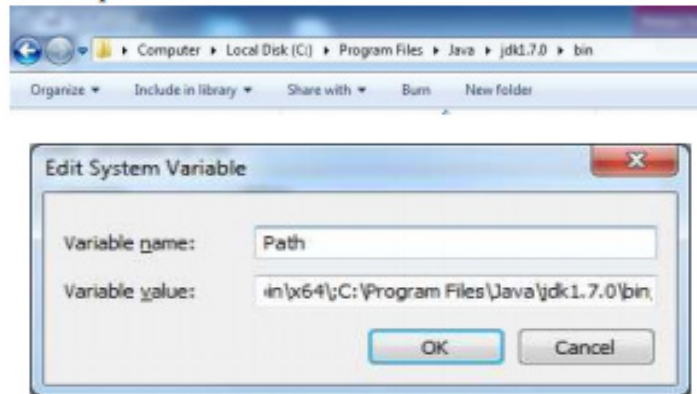
Gambar 1.1. *Window Properties*

Pilih *Advanced System Settings*. Pada bagian tab *Advanced* tekan tombol *Environment Variables*. Pilih *Path* pada bagian *System Variables* dan tekan *Edit*.



Gambar 1.2. *Environment Variables*

Tambahkan letak folder bin hasil instalasi JDK pada bagian *Variable value* dengan cara *copy address* pada *windows explorer*.



Gambar 1.3. *Edit System Variable*

Selanjutnya penempatan CLASSPATH JDK, menggunakan tanda titik koma ; tanpa spasi diikuti classpath yang lain sudah berada. Berikutnya lakukan cek keberhasilan setting apakah classpath sudah berhasil dengan membuka lewat cmd (*command prompt*) kemudian coba dijalankan dengan perintah JAVAC. Jika berhasil dalam setting classpathnya, maka akan tampil seperti pada gambar window di bawah ini :



Gambar 1.4. *Administrator Classpath*

2. Struktur Penulisan Java

Struktur penulisan Java adalah sebagai berikut:

```
public class namakelas
{
    public static void main (String args[])
    {
        kode program
    }
}
```

Struktur penyimpanan file Java : **namakelas.java**

Kompilasi (Compile) program Java adalah : **javac namakelas.java**

Menjalankan (Run) program java: **java namakelas**

3. Mengenal Perintah Dasar IO (Input – Output) di Java

Perintah Input Scanner : digunakan untuk mengambil input dari keyboard.

Berikut struktur scanner di coding java :

```
import java.util.Scanner ;
public class BelajarScanner (
    public static void main (String[] args) {
        Scanner input = new Scanner(System.in);
        String nama = input.nextLine()
        System.out.println("Hai"+nama);
    }
}
```

Perintah pada program di atas jika dijalankan belum ada hasil yang ditampilkan di layar, karena belum dilakukan inputan dari keyboard. Oleh karena itu masukkan inputan dengan *keyboard* maka program akan menerima perintah inputan dari *user*. dan tekan enter, sehingga program akan menampilkan tulisan : Hai <nama_yg_diinputkan> di layar monitor.

Catatan penting ! :

Perintah *Scanner* dalam *coding* di atas (*import java.util.Scanner*) sangat diperlukan karena fungsi *Scanner* berguna untuk kepastian dalam menggunakan *class Scanner* yang berada dalam *package* yang berbeda-beda antara lain *package* : *java.util*. Jika perintah *import java.util.Scanner* tidak ditemukan keberadaannya program tidak bisa dijalankan.

Dengan menggunakan perintah *import java.util** maka program dapat menggunakan semua *class* yang ada didalam *package java.util*. Khusus pada perintah *import java.util.Scanner* mempunyai arti bahwa penggunaan hanya pada *class Scanner* yang berada dalam *package java.util* saja.

Pada prinsipnya, perintah *Scanner* untuk dapat melakukan pembacaan pada semua jenis karakter yang dimasukkan lewat keyboard. Tetapi jika akan mengambil data yang berupa angka, dibutuhkan metode khusus dari class *Scanner* tersebut, yaitu dengan *nextInt()* salah satu contohnya.

Berikut contoh lain dari input (next) :

Macam-macam input (next) :

- nextInt : menerima input bilangan bulat Integer
- nextBoolean : menerima input Boolean (true, false)
- nextFloat : menerima input pecahan Float
- nextDouble : menerima input pecahan Double
- nextByte : menerima bilangan bulat Byte
- nextLong : menerima bilangan bulat Long
- nextShort : menerima bilangan bulat Short

Output : System.out.println();

Untuk bisa melihat letak posisi output dalam *coding*, perhatikan *coding* di bawah ini

// Contoh *coding* untuk : Output → System.out.println();

//Package yang dibutuhkan untuk memuat Class *Scanner*

Berikut contoh perintah *Scanner* yang digunakan dalam program java :

```
import java.util.Scanner;
Public class input_example {
    Public static void main (String[] args){
        Scanner input = new Scanner(System.in); //membuat objek dari Class Scanner
        String nama; //variable string, untuk menyimpan input dari user
        System.out.print("Siapa Nama Kamu : ");
        nama = input.nextLine(); //mendapatkan input dari user
        System.out.println("Nama Saya "+nama); // mencetak output
    }
}
```

4. Mengenal Fungsi Operator Aritmatika pada Java.

Operator aritmatika merupakan operator yang berfungsi mengerjakan komputasi numerik. Berikut adalah tabel operator aritmatika yang susunan operator nya sudah berdasarkan Hirarchi Operator.

Tabel 1. Operator Aritmatika

Operator	Kegunaan
*	Perkalian
/	Pembagian
+	Penjumlahan
-	Pengurangan
%	Nilai hasil bagi

5. Mampu mengoperasikan operator aritmatika pada Java

Berikut contoh coding untuk penerapan operasi aritmatika :

```
Public class operator_aritmatika_bulat {  
    Public static void main(String[] args) {  
        System.out.println("Operasi aritmatika "+" pada tipe integer");  
        Int a = 2 + 5 ;  
        Int b = 25 - 3 ;  
        Int c = 7 * 4 ;  
        Int d = 9 / 3 ;  
        Int e = 20 % 5 ;  
  
        System.out.println("Nilai a : " + a);  
        System.out.println("Nilai b : " + b);  
        System.out.println("Nilai c : " + c);  
        System.out.println("Nilai d : " + d);  
        System.out.println("Nilai e : " + e);  
    }  
}
```

Output :

```
Operasi aritmatika pada tipe integer  
Nilai a : 7  
Nilai b : 22  
Nilai c : 28  
Nilai d : 3  
Nilai e : 0
```

LATIHAN 1.1 :

1. Buka notepad
2. Ketik source code berikut:

```
public class mobil
{
    public static void main(String args[])
    {
        System.out.println("Type Mobil    : Mini Bus");
        System.out.println("Merk Mobil   : Suzuki Ignis");
        System.out.println("Harga: Rp 150.000.000");
    }
}
```

3. Simpan dengan nama **mobil.java**, kemudian Compile dan jalankan dengan **Command Prompt** (misal: → D:\>javac mobil.java), **RUN Program** → menghasilkan output :

```
public class nilaiku
{
    public static void main(String args[])
    {
        double tugas, uts, uas, na;
        tugas=78;
        uts=83;
        uas=90;
        na=(0.3*tugas)+(0.3*uts)+(0.4*uas);
        System.out.println(" Nilai Akhir Anda adalah =" +na);
    }
}
```

LAT

- 1.
- 2.

3. Simpan dan beri nama file dengan **nama nilaiku.java**, selanjutnya kompilasi program java untuk dijalankan di **Command Prompt**, sehingga menghasilkan output seperti berikut:

```
D:\> java nilaiku
Nilai Akhir Anda adalah = 84.3
```

LATIHAN 1.3 :

1. Buka Notepad
2. Ketik source code berikut:

```
import java.util.Scanner;
public class persegi panjang
{
    public static void main(String args[])
    {
        Scanner input=new Scanner(System.in);

        System.out.print("Masukkan Panjang    :");
        double panjang=input.nextInt();

        System.out.print("Masukkan Lebar      :");
        double lebar=input.nextInt();

        System.out.println("LuasPersegiPanjang="+(panjang*lebar));
    }
}
```

3. Simpan file dan beri nama **persegi panjang.java**, selanjutnya kompilasi program serta jalankan dengan **Command Prompt**, sehingga menghasilkan output seperti berikut:

```
D:\> java persegi panjang
Masukkan Panjang    : 2
Masukkan Lebar      : 4
Luas Persegi Panjang = 8.0
```

PERCOBAAN MANDIRI-1 :

1. Buat program untuk menghitung total harga dengan diskon 5% dari total harga. Output program seperti di bawah ini :

Masukkan Harga Barang =
Masukan Jumlah Beli =
Total Bayar =


```
D:\>javac tot_harga.java
```

```
D:\>java tot_harga
```

```
Masukkan harga barang : 2400
```

```
Masukkan jumlah beli : 13
```

```
Total bayar = 15600.0
```

2. Buat program untuk menghitung gaji bersih karyawan. Dimana setiap karyawan dikenakan pajak 2% dari total gaji. Output program seperti berikut:

```
Masukkan Gaji Pokok      = .....
```

```
Masukan Tunjangan       = .....
```

```
Gaji Bersih (setelah dikurangi pajak 2% dari total gaji) = ..
```

```
D:\>javac tot_gaji.java
```

```
D:\>java tot_gaji
```

```
Masukkan Gaji Pokok Karyawan : 2000000
```

```
Masukkan Tunjangan Karyawan : 570000
```

```
Gaji Bersih = 2056000.0
```

BAB 2

STRUKTUR KONDISI (IF, NESTED IF, SWITCH)

TUJUAN:

1. Mengetahui perintah dasar struktur kontrol kondisi (If, Nested If) , (Switch..Case)
2. Mengetahui fungsi operator struktur kondisi If dan Switch
3. Mampu mengoperasikan operator struktur kondisi If dan Switch pada Java

PERSIAPAN:

1. Perangkat lunak Java JDK 1.6.0_25
2. Perangkat Lunak Command Prompt
3. Perangkat lunak Notepad

DASAR TEORI:

1. **Mengetahui perintah dasar struktur kontrol kondisi (IF..THEN..ELSE) dan (switch...case).**

Pada prinsipnya dengan memakai struktur kondisi yang berbentuk If maupun Nested If merupakan suatu perintah logika yang ada dalam program dengan cara menghubungkan beberapa baris perintah IF.. else yang lain sehingga menjadi satu kesatuan yang utuh. Algoritma pembacaannya bahwa if pertama jika kondisi tidak terpenuhi atau syarat kondisinya bersifat salah/ false maka perintah program tidak akan dijalankan dan akan lanjut ke if berikutnya (else) sampai menemukan yang sesuai dengan kondisi yang dimaksudkan. Jika sudah mendapatkan kondisi If yang sesuai (benar/true) maka proses program akan mengeluarkan hasil sesuai yang diinginkan (benar). Jadi pembacaan pada perintah If tersebut akan berjalan terus sampai menemukan kondisi (syarat)

terpenuhi dari program, sepanjang persyaratan dari kondisi yang ditentukan. Semakin panjang persyaratan biasanya kriteria yang diinginkan semakin detail / rinci, sehingga program lebih akurat / valid hasilnya.

Format / syntax / dari If dan Nested If :

a. Deklarasi perintah kondisi **IF tunggal** :

```
If (kondisi)  
{  
    pernyataan1;  
    pernyataan2;  
    ...  
}
```

b. Deklarasi perintah kondisi **IF lebih dari Satu (Nested If)** :

```
If (kondisi)  
{  
    pernyataan1;  
}  
Else  
{  
    pernyataan2;  
}
```

Penerapan penulisan dalam program bisa bervariasi, tergantung dari programmer. Contoh program berikut ini adalah salah satu contoh penulisan yang standard. Programmer yang lain bisa menuliskan dengan cara yang berbeda-beda, tergantung dari algoritama masing-masing. Namun secara konsep dasar penulisan program java bisa dicoba dengan cara penulisan seperti pada contoh program sederhana di bawah ini.

Sebelum penulisan program dimulai , pastikan bahwa program java sudah terinstall secara lengkap di komputer dan dipastikan tidak ada kesalahan dalam melakukan installasi java nya, karena jika ada kesalahan dalam install terutama dalam pengaturan tata letak / classpath nya maka java tidak akan bisa merespon perintah-perintah yang sudah ditulis di tempat editor

program. Pilih salah satu tempat editor program ditulis. Saran pakailah yang mudah dan sederhana yaitu dengan memakai test editor yang sudah umum dipakai yaitu : Notepad

Program di bawah ini ditulis di dalam editor notepad.

```
import java.util.Scanner;
class BelajarJava {
    public static void main(String args[]){
        char nilai;
        Scanner input = new Scanner(System.in);
        System.out.print("Input Nilai Anda (A - E): ");
        nilai = input.next().charAt(0);

        if (nilai == 'A') {
            System.out.println("Pertahankan!");
        }
        else if (nilai == 'B') {
            System.out.println("Harus lebih baik lagi");
        }
        else if (nilai == 'C') {
            System.out.println("Belajar Lagi!");
        }
        else if (nilai == 'D') {
            System.out.println("Belajar Ditingkatkan!!");
        }
        else if (nilai == 'E') {
            System.out.println("Harus diUlang !...");
        }
        else {
            System.out.println("Maaf, format nilai tidak sesuai");
        }
    }
}
```

Hasil Output :

```
Input Nilai Anda (A..E) : A
Pertahankan !
```

```
Input Nilai Anda (A..E) : D
Belajar ditingkatkan !!
```

```
Input Nilai Anda (A..E) : E
Harus di Ulang !
```

c. Deklarasi perintah kondisi dengan : **Switch..Case**

Hampir sama dengan struktur kondisi If.. else, pemakaian perintah kondisi yang berupa Switch ... case juga sama fungsinya dengan if. Perbedaannya hanya dalam penulisan / susunan kalimat perintah. Dalam Switch case cara membandingkan data yang disimpan dalam variabel itu secara sekaligus yaitu beberapa nilai yang sudah dimasukkan atau yang sudah tersimpan dalam variabel tersebut. Jika dalam perbandingan menghasilkan kondisi / syarat kebenaran (true) maka blok program akan dijalankan/ diproses. Bentuk switch case ada 2 bagian, pertama switch untuk nama variabel yang akan cek kebenarannya (kondisi true/false) , yang kedua switch untuk perintah case yang dipakai untuk cek isi / data yang sesuai kriteria yang nantinya akan menghasilkan output / luaran. Jadi fungsi **switch..case** sama dengan **if..else if** , hanya saja **switch case** hanya terbatas pada satu variabel dan nilainya saja, **switch case** tidak bisa benar-benar menyatakan suatu kondisi seperti lebih dari, kurang dari seperti layaknya **if else if** .

```
Switch (nama_variabel)
{
    case(nilai1):
        pernyataan1;
        break;
    case(nilai1):
        break;
    default:
        pernyataan default;
}
```

atau :

```
Switch (nama_variabel) (
    Case 'nilai_1' :
        // Kode program yang dijalankan jika nama_variabel == nilai_1
        Break;
    Case 'nilai_2' :
        // Kode program yang dijalankan jika nama_variabel == nilai_2
        Break;
    Case 'nilai_3' :
        // Kode program yang dijalankan jika nama_variabel == nilai_3
        Break;
    .....
    .....
    Default :
        //Kode program yang dijalankan jika tidak ada kondisi yang terpenuhi
```

Berikut implementasi Switch.. case dalam program Java secara lengkap :

```
import java.util.Scanner
class BelajarJava {
    public static void main (String args[]) {
        char nilai;
        Scanner input=new Scanner (System.in);
        System.out.[rint("Input Nilai Anda (A..D) : ");
        Nilai=input.next().charAt(0);
        Switch (nilai) {
            case 'A' :
                System.out.println("Pertahankan !");
                break;
            case 'B' :
                System.out.println("Harus lebih baik lagi !");
                break;
            case 'C' :
                System.out.println("Belajar Lagi !");
                break;
            case 'D' :
                System.out.println("Belajar ditingkatkan !");
                break;
            default;
                System.out.println("Maaf yang anda masukkan salah !!");
        }
    }
}
```

Hasil Output Program:

**Input Nilai Anda (A..D) : A
Pertahankan !**

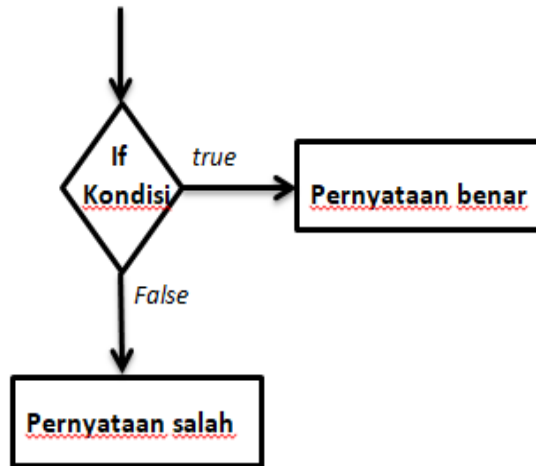
**Input Nilai Anda (A..D) : C
Belajar Lagi !**

**Input Nilai Anda (A..D) : E
Maaf yang anda masukkan salah !!**

2. Mengetahui fungsi operator struktur kondisi IF dan Switch.

Fungsi operator kondisi if tunggal menyatakan pilihan hanya satu, jadi if akan dikerjakan langsung jika kondisi / syarat dinyatakan benar / true. Namun apabila if pertama menyatakan kondisi salah / false maka perintah tidak dijalankan atau if tidak melakukan proses sehingga program akan berhenti atau perintah berlanjut ke proses berikutnya.

Struktur IF dalam bentuk flowchart bisa dilihat seperti di bawah ini :



Contoh :

Dalam studi kasus berikut bahwa toko buku “ABC” menjual barang dagangannya, Ketentuan dari toko akan memberikan hadiah kepada pembeli berupa perlengkapan sekolah, dengan ketentuan jika total belanja di atas Rp 100.000. Jika belanja di bawah Rp 100.000 maka tidak akan mendapatkan hadiah.

Di bawah ini penulisan program lengkap untuk studi kasus di atas :


```

import java.util.Scanner;
public class Hadiah {
    public static void main(String[] args) {
        // membuat variabel belanja dan scanner
        int belanja = 0;
        Scanner scan = new Scanner(System.in);
        // mengambil input
        System.out.print("Total Belanjaan: Rp ");
        belanja = scan.nextInt();
        // cek apakah dia belanja di atas 100000
        if (belanja > 100000 ) {
            System.out.println("Selamat, anda mendapatkan hadiah!");
        }
        System.out.println("Terima kasih...");
    }
}

```

Running :

```

Total Belanjaan : Rp 150000
Selamat, anda mendapatkan hadiah
Terima kasih...

```

3. *Mampu mengoperasikan operator struktur kondisi if dan switch pada Java*

Struktur kondisi digunakan untuk mengatur proses pemilihan yang mana eksekusinya tergantung pada pemilihan yang ditentukan sebelumnya (Wahana Komputer, 2012).

Contoh dalam program :

```
import java.util.*; //import package java util
public class MainClass {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in); // buat objek dari class Scanner
        System.out.println("Masukkan nama Anda : ");
        String nama = input.nextLine(); //untuk mengambil input nama
        System.out.println("Masukkan 1 jika umur anda kurang dari 17 tahun"+"\\n"
            + "Masukkan 2 jika umur anda adalah 17 tahun"+"\\n"
            + "Masukkan 3 jika umur anda lebih dari 17 tahun ");/*memberikan pilihan kepada user'

        int pilihUmur = input.nextInt(); //pilihan umur dengan memasukkan angka
        switch(pilihUmur){ //switch case pada variabel pilihUmur
            case 1:
                System.out.println(nama+" "+"memiliki umur kurang dari 17 tahun");
                break;
            case 2:
                System.out.println(nama+" "+"memiliki umur adalah 17 tahun");
                break;
            case 3:
                System.out.println(nama+" "+"memiliki umur lebih dari 17 tahun");
                break;
            default:
                System.out.println("Masukkan angka yang tersedia");
        }
    }
}
```

Running program-1 :

```
Masukkan nama Anda :
BenNita
Masukkan 1 jika umur anda kurang dari 17 tahun
Masukkan 2 jika umur anda adalah 17 tahun
Masukkan 3 jika umur anda lebih dari 17 tahun
3
BenNita memiliki umur lebih dari 17 tahun
```