

Acara 9

Pokok Bahasan : Percabangan 1

Acara Praktikum/Praktek: Minggu 3/1

Tempat : Politeknik Negeri Jember

Alokasi Waktu : 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menggunakan Pencabangan (IF, IF-ELSE, IF-ELSE-IF-ELSE, switch case).
2. Mahasiswa mengimplementasikan Pencabangan (IF, IF-ELSE, IF-ELSE-IF-ELSE, switch case) dalam bahasa pemrograman Java.

b. Indikator

Kemampuan mahasiswa dalam menjelaskan dan mengimplementasikan Pencabangan (IF, IF-ELSE, IF-ELSE-IF-ELSE, switch case).

c. Dasar Teori

1) Alur Program

Alur jalannya program dapat berubah karena salah satu sebab berikut:

- i. **Pemanggilan** metode (invoke a method). Dengan terjadinya kasus ini, alur program akan berubah dan mengalir ke metode yang dipanggil.
- ii. **Pencabangan** (decision making). Terjadi ketika program dihadapkan pada pilihan dan harus memilih salah satu. Pencabangan dalam bahasa Java diaplikasikan melalui if/else dan switch.
- iii. **Perulangan** (repetition). Perulangan terjadi ketika ada perintah yang harus dikerjakan sampai beberapa kali. Perulangan dalam bahasa Java dapat dilakukan melalui for, while, dan do/while

2) Logika Boolean

Pencabangan dilakukan menggunakan bantuan logika Boolean. Variabel tipe Boolean hanya mempunyai dua kemungkinan nilai: true (benar) atau false (salah). Dalam alur program, nilai Boolean digunakan setelah mengevaluasi kondisi. Evaluasi kondisi dilakukan menggunakan operator perbandingan dan operator Boolean.

3) Pencabangan

Komputer sering kali harus memutuskan yang berarti memilih satu dari beberapa aksi, tergantung dari kondisi yang ada. Inilah yang disebut dengan pencabangan.

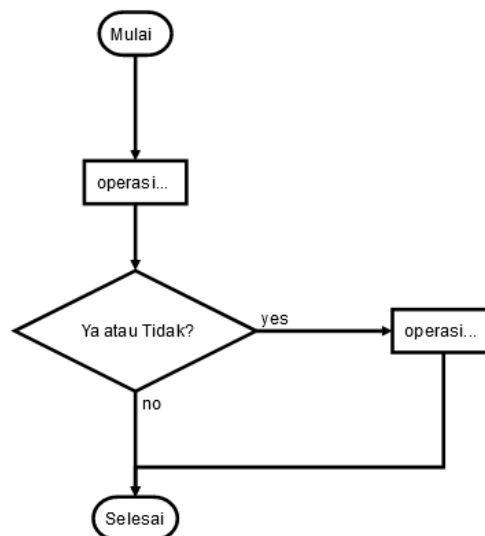
1. Sintaks Percabangan If

If merupakan perintah percabangan yang paling sederhana. If membuat statement atau kumpulan statement dijalankan, jika kondisi bernilai benar (true).

Bentuk umum

```
if (kondisi) {  
    pernyataan;  
    pernyataan;  
    ...  
}
```

Bentuk flowchart:



- Apabila kondisi bernilai benar, maka pernyataan akan dilaksanakan.
- Apabila kondisi bernilai salah, maka pernyataan tidak akan dilaksanakan.

Pada sintaks pemilihan 1 ini, kita akan menggunakan operator hubungan (*relational operator*). Berikut ini operator hubungan dalam bahasa pemrograman Java:

Simbol Operator	Keterangan
==	Sama dengan (Equal to)
>	Sama dengan (Greater than)
<	Sama dengan (Less than)
>=	Sama dengan (Greater than or Equal to)
<=	Sama dengan (Less than or Equal to)
!=	Tidak sama dengan (Not Equal to)

Implementasi atau penggunaan operator hubungan pada sintaks pemilihan adalah sebagai berikut

```
if (condition) { statement(s); } statement;
```

Contoh program percabangan if:

```
import java.util.Scanner;

public class PemrogramanJava {

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

        int angka;

        System.out.print("Masukkan angka: ");
        angka = input.nextInt();

        if (angka > 70) {
            System.out.println("Selamat anda dinyatakan diterima");
        }
    }
}
```

Hasil jika di masukan nilai 88

```
Output - Pemrograman Java (run)

run:
Masukkan angka: 88
Selamat anda dinyatakan diterima
BUILD SUCCESSFUL (total time: 18 seconds)
```

Hal ini disebabkan karena terdapat sebuah kondisi yang menyatakan bahwa jika nilai lebih dari 70 ($\text{angka} > 70$) maka akan tampil **Selamat anda dinyatakan diterima**, sedangkan jika dimasukkan angka 70 atau kurang dari 70 maka tidak akan terdapat tampilan apapun.

```
Output - Pemrograman Java (run)

run:
Masukkan angka: 55
BUILD SUCCESSFUL (total time: 2 seconds)
```

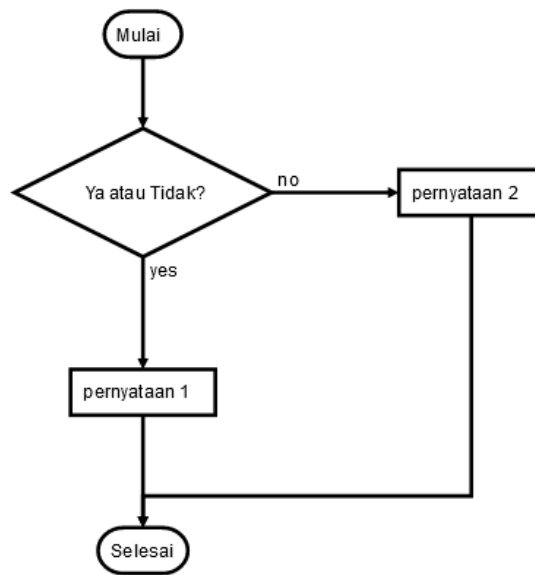
2. Sintaks Percabangan if else

Pencabangan dengan if else membuat statement pertama dijalankan jika kondisinya benar. Namun, jika kondisi bernilai salah, statement setelah else akan dijalankan. Jika statement yang dipakai banyak, digunakan kurung kurawal. Dengan demikian, kumpulan statement di dalam blok kurung kurawal akan dianggap satu statement.

Syntax if else :

```
if (kondisi) {
    pernyataan1;
} else {
    pernyataan2;
}
```

Flowchart:



Contoh program if else

```
import java.util.Scanner;
public class PemrogramanJava {

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

        int angka;

        System.out.print("Masukkan angka: ");
        angka = input.nextInt();

        if (angka > 70) {
            System.out.println("Selamat anda dinyatakan diterima");
        } else {
            System.out.println("Silahkan coba tes lagi tahun depan");
        }
    }
}
```

Pada program di atas ditambahkan kode program

```
else {
    System.out.println("Silahkan coba tes lagi tahun depan");
}
```

Hasilnya jika dimasukkan nilai dibawah 70 yaitu 55.

```
Output - Pemrograman Java (run)

run:
Masukkan angka: 55
Silahkan coba tes lagi tahun depan
BUILD SUCCESSFUL (total time: 2 seconds)
```

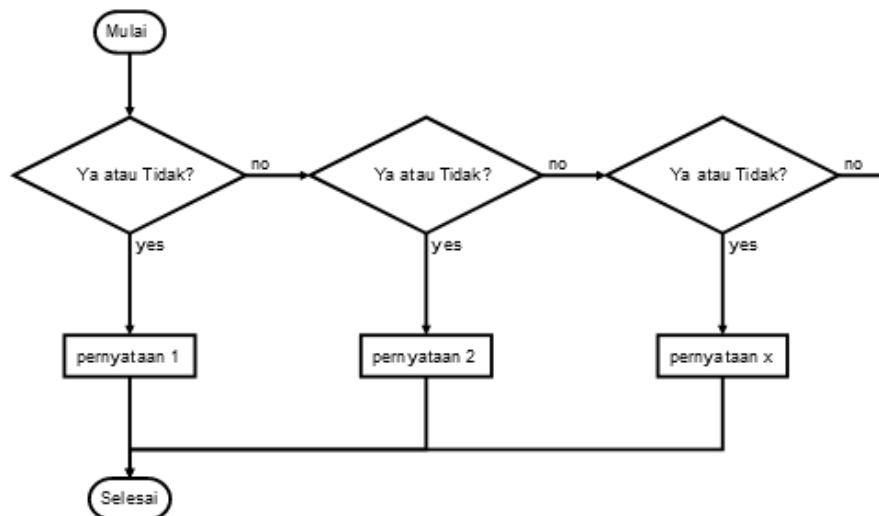
Hal ini disebabkan karena terdapat sebuah kondisi yang menyatakan bahwa jika nilai lebih dari 70 (angka > 70) maka akan tampil **Selamat anda dinyatakan diterima**, sedangkan jika dimasukkan angka 70 atau kurang dari 70 maka akan tampil **Silahkan coba tes lagi tahun depan**.

Sintaks Percabangan if else if else

Bentuk umum:

```
if (kondisi1) {
    pernyataan-1;
} else if (kondisi2) {
    pernyataan-2;
} else if (kondisi3) {
    pernyataan-3;
} else {
    pernyataan-else;
```

Bentuk Flowchart



Pada bentuk **if else if else** di atas, pernyataan 1 akan dijalankan apabila **kondisi1** bernilai benar. Jika **kondisi1** bernilai salah, maka akan dicek **kondisi2**. Jika **kondisi2** benar maka akan dijalankan **pernyataan2**, dan seterusnya. Apabila tidak ada satupun syarat yang terpenuhi, barulah pernyataan-else akan dikerjakan

Contoh Program:

```
import java.util.Scanner;
public class PemrogramanJava {

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

        int bayar;

        System.out.print("Masukkan total belanja anda: ");
        bayar = input.nextInt();

        if (bayar >= 2000000) {
            System.out.println("Selamat anda mendapatkan hadiah kompor gas");
        } else if (bayar >= 1000000) {
            System.out.println("Selamat anda mendapatkan hadiah teflon");
        } else if (bayar >= 500000) {
            System.out.println("Selamat anda mendapatkan hadiah piring");
        } else {
            System.out.println("Maaf anda belum beruntung, tingkatkan belanja anda!");
        }
    }
}
```

Hasil program

```
Output - Pemrograman Java (run)

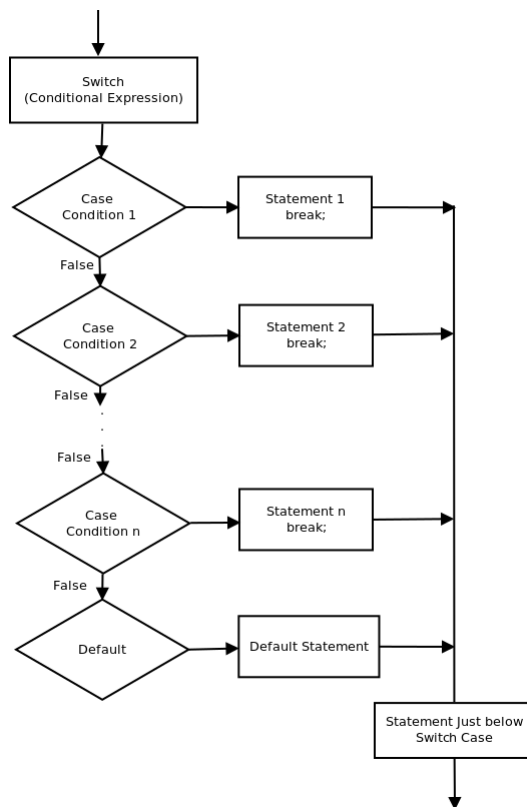
run:
Masukkan total belanja anda: 1000000
Selamat anda mendapatkan hadiah teflon
BUILD SUCCESSFUL (total time: 18 seconds)
```

Sintaks pemilihan switch case

Jika pada pencabangan terdapat pilihan yang banyak, perintah switch lebih dianjurkan daripada menggunakan if. Switch hampir mirip seperti else if. Switch lebih sering dipakai pada beberapa pilihan menggunakan satu variabel sebagai penentu pencabangan.

Sintaks Switch bentuk umum:

```
switch (kondisi) {
case konstanta-1:
    pernyataan-1;
    break;
case konstanta-2:
    pernyataan-2;
    break;
...
...
case konstanta-x:
    pernyataan-x;
    break;
default:
    pernyataan;
}
```



```

import java.util.Scanner;
public class PemrogramanJava {

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

        int angka;

        System.out.print("Masukkan kode kelas anda: ");
        angka = input.nextInt();

        switch(angka) {
            case 1:
                System.out.println("Golongan A");
                break;
            case 2:
                System.out.println("Golongan B");
                break;
            case 3:
                System.out.println("Golongan C");
                break;
            case 4:
                System.out.println("Golongan D");
                break;
            default:
                System.out.println("Kode yang anda masukkan salah");
        }
    }
}

```

```
Output - Pemrograman Java (run)

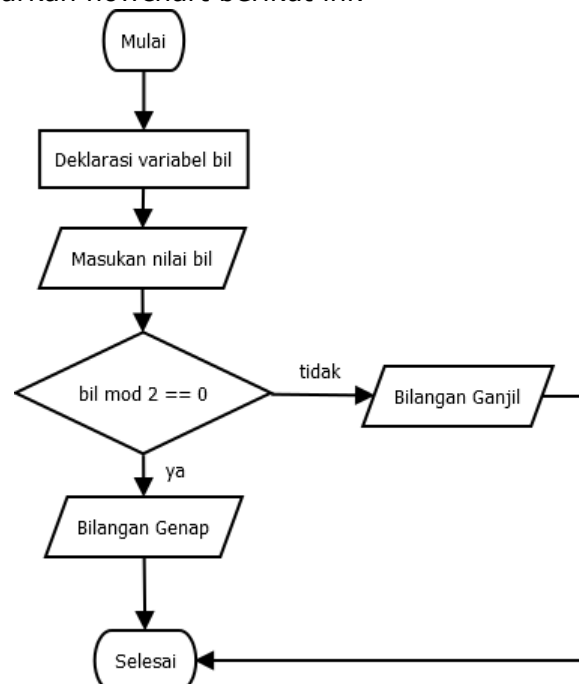
run:
Masukkan kode kelas anda: 1
Golongan A
BUILD SUCCESSFUL (total time: 2 seconds)
|
```

TUGAS PRAKTIKUM

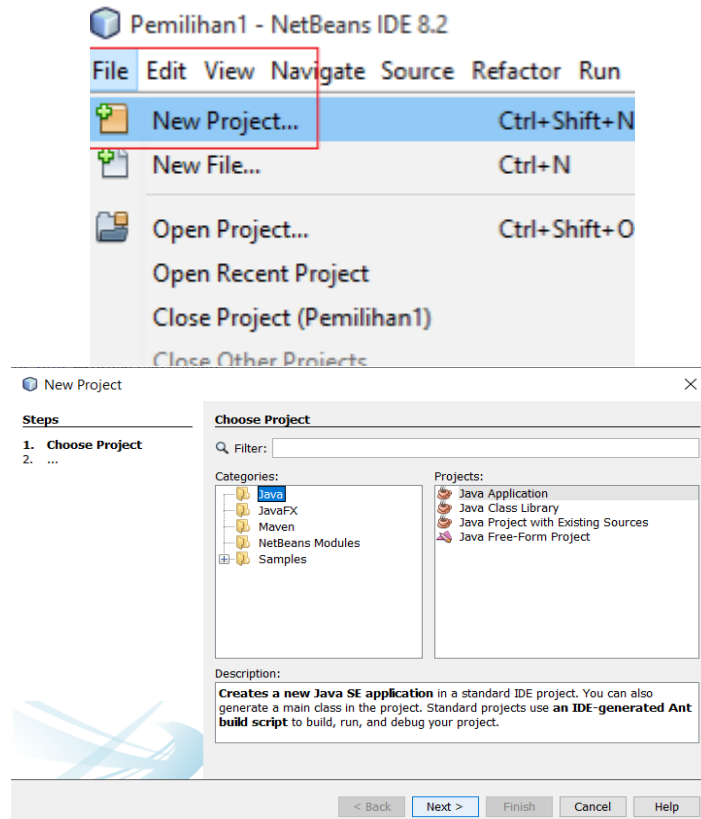
Buatlah laporan dari tugas-tugas berikut disertai dengan penjelasan fungsi kode program.

TUGAS 1

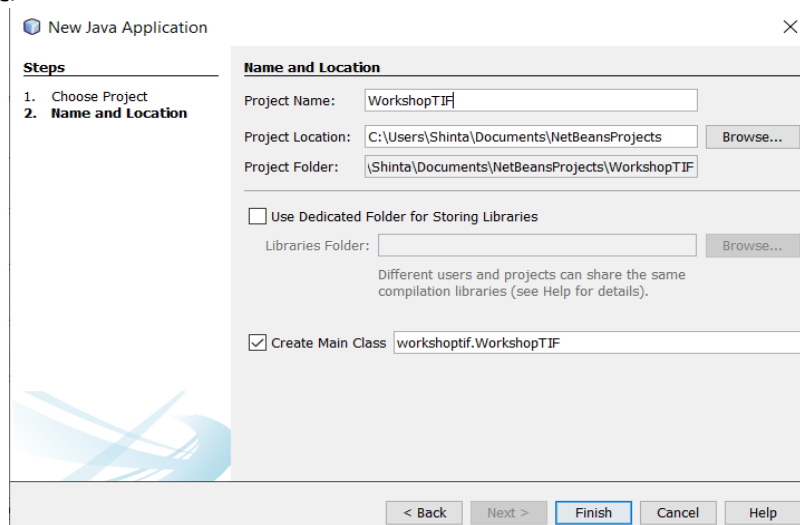
Buatlah program berdasarkan flowchart berikut ini:



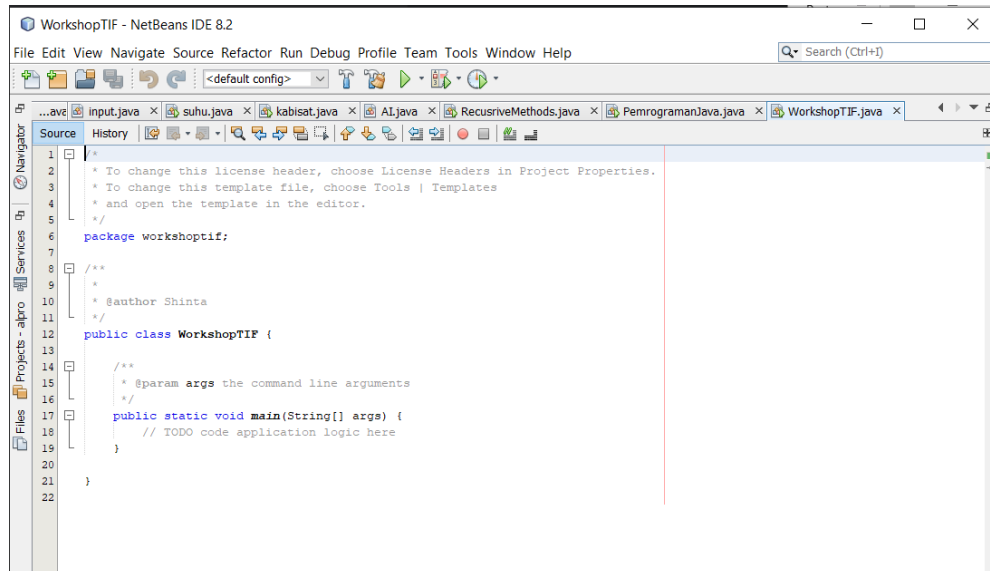
- Buka Netbeans yang sudah anda install
- Buat project baru dengan nama "WorkshopTIF" dengan cara
File -> New Project -> Java Application ->Next ->WorkshopTIF ->Finish



- Klik **Next>**



- Beri nama project WorkshopTIF & klik **Finish**
- Maka akan muncul tampilan awal berikut ini



- Tambahkan import library Scanner.
- Deklarasikan Scanner:

```
Scanner input = new Scanner(System.in);
```

- Buatlah variabel bertipe int dengan nama bil

```
int bil;
```

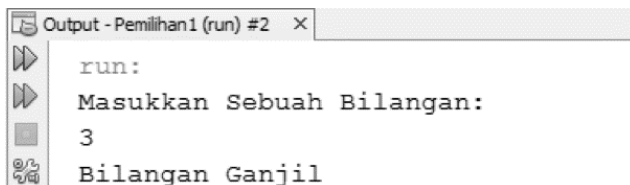
- Buatlah struktur kondisi untuk mengecek apakah bilangan tersebut merupakan bilangan genap atau ganjil

```

if (bil%2==0){
    System.out.println("Bilangan Genap");
}
else{
    System.out.println("Bilangan Ganjil");
}

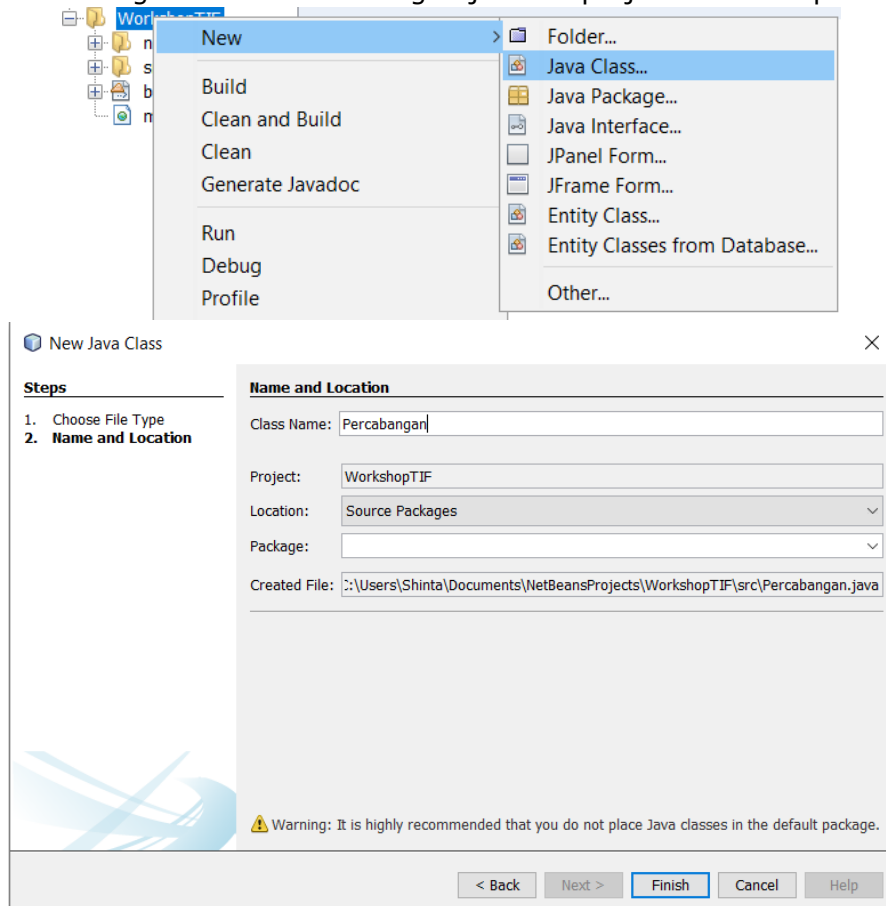
```

- Jalankan program, jika berhasil maka outputnya adalah sebagai berikut:



TUGAS 2

1. Buat Class baru dengan nama "Percabangan.java" di project "WorkshopTIF".



2. Tambahkan import library Scanner.
3. Deklarasikan Scanner:

```
Scanner input = new Scanner(System.in);
```
4. Buatlah variabel bertipe int dengan nama bil

```
int nilai;
```
5. Tambahkan kode program

```
System.out.println("Masukkan Sebuah Bilangan: ");
nilai=input.nextInt();

if (nilai>=100){
    nilai+=10;
}
else{
    nilai-=10;
}

System.out.println("Hasil Nilai Akhir adalah " +nilai);
```

6. Jalankan program



d. Hasil dan Pembahasan

1. Laporan hasil praktikum
2. Laporan berupa file word dikumpulkan pada <http://jti.poliije.ac.id/elearning>

e. Kesimpulan

Mahasiswa memahami dan mampu mengimplementasikan percabangan IF, IF-ELSE, IF-ELSE-IF-ELSE, switch case.

f. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambatan dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					

Acara 10

Pokok Bahasan : Percabangan 2

Acara Praktikum/Praktek: Minggu 3/2

Tempat : Politeknik Negeri Jember

Alokasi Waktu : 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menggunakan Pencabangan (If Else Nested).
2. Mahasiswa mengimplementasikan Pencabangan (If Else Nested) dalam bahasa pemrograman Java.

b. Indikator

Kemampuan mahasiswa dalam menjelaskan dan mengimplementasikan Pencabangan (If Else Nested).

c. Dasar Teori

Pada pertemuan sebelumnya telah kita pelajari bersama tentang penggunaan percabangan IF, kemudian IF-ELSE, IF-ELSE-IF ELSE dan switch case. Kali ini kita mempelajari Nested If Else.

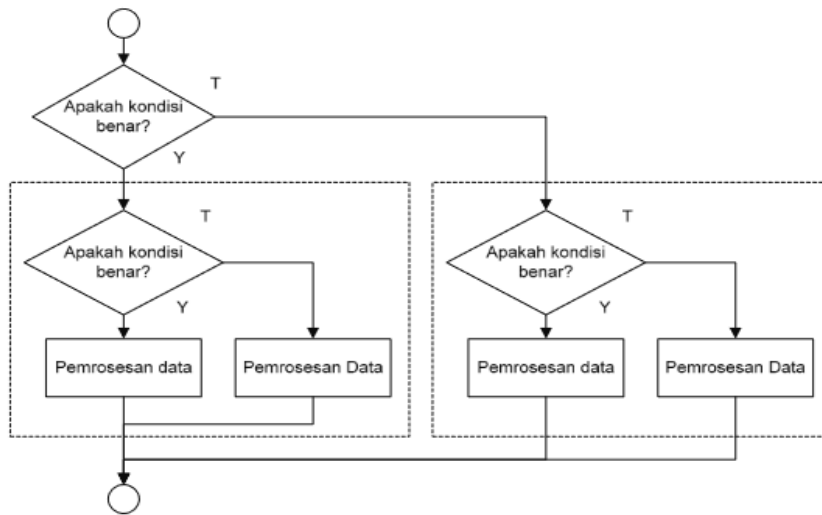
1) Nested If Else

If Else Nested terjadi jika ada pencabangan di dalam pencabangan. Maksudnya, terdapat if else di dalam if atau else yang lebih tinggi.

Sintaks umum:

```
if (condition)
if (condition) statement;
else statement;
else
if (condition) statement;
else statement;
```

Flowchart:

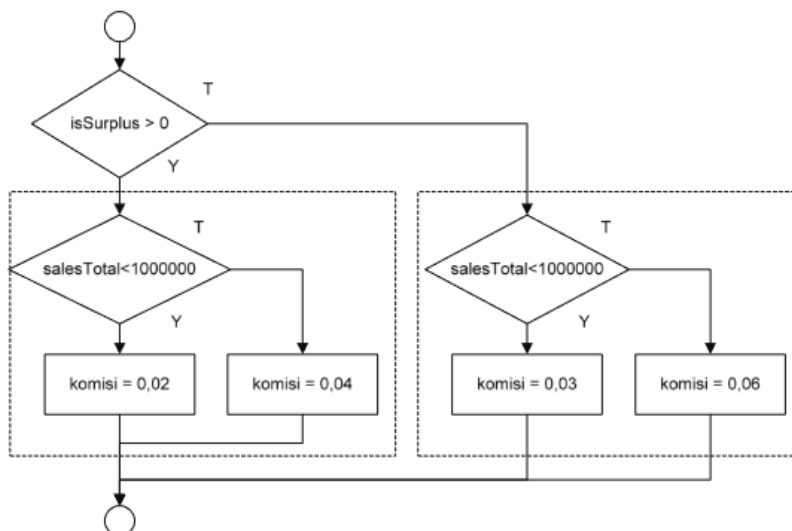


Gambar *Flowchart Nested If Else*

Berikut contoh listing program dengan nested if else.

```

if (isSurplus == 0)
  if (salesTotal < 10000000)
    komisi = 0.02;
  else
    komisi = 0.04;
else
  if (salesTotal < 10000000)
    komisi = 0.03;
  else
    komisi = 0.06;
  
```



Tugas Praktikum

Buatlah laporan disertai penjelasan fungsi dari sintaks pada program kasir dengan menggunakan NESTED IF.

Berikut merupakan alur program kasir. Ketika seseorang akan melakukan pembayaran di kasir, kasir akan memberikan pertanyaan sebagai berikut:

1. Apakah pelanggan mempunyai kartu anggota?

TRUE: Pelanggan mempunyai kartu anggota

2. Apakah total harga barang belanjaan lebih dari Rp 500.000?

TRUE: Total harga barang belanjaan lebih dari Rp 500.000, pelanggan mendapatkan diskon Rp 50.000

FALSE: Total harga barang belanjaan tidak lebih dari Rp 500.000, pelanggan mendapatkan diskon Rp 25.000

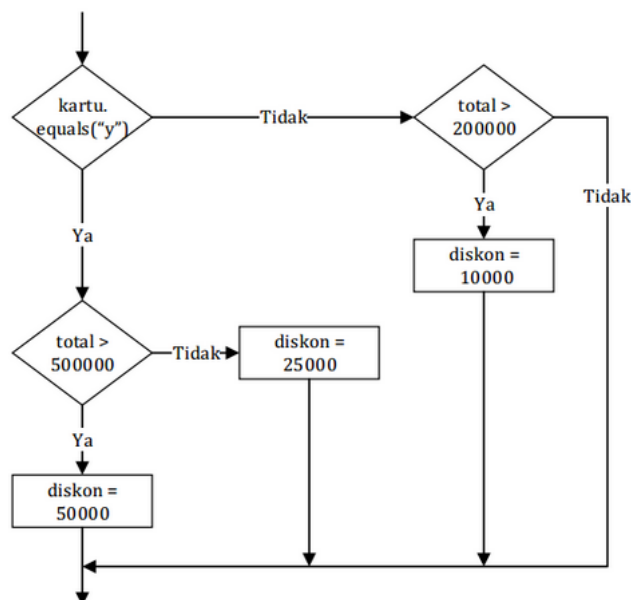
FALSE: Pelanggan tidak mempunyai kartu anggota

3. Apakah total harga barang belanjaan lebih dari Rp 200.000?

TRUE: Total harga barang belanjaan lebih dari Rp 200.000, pelanggan mendapatkan diskon Rp 10.000

FALSE: Total harga barang belanjaan tidak lebih dari Rp 200.000, pelanggan tidak mendapatkan diskon

4. Untuk lebih memperjelas alur percabangan pada contoh kasus tersebut, perhatikan flowchart berikut:



5. Buatlah Class kasir seperti pertemuan sebelumnya?

6. Jalankan program berikut

```
import java.util.Scanner;
public class kasir {
    public static void main(String[] args) {
        int total, diskon, bayar;
        String kartu;
        Scanner sc = new Scanner (System.in);
        System.out.print("Apakah pelanggan mempunyai kartu anggota (y atau t)? ");
        kartu = sc.nextLine();
        System.out.print("Berapa total harga barang belanjaan? Rp ");
        total = sc.nextInt();
        if (kartu.equals("y")) {
            if (total > 500000) {
                diskon = 50000;
            } else {
                diskon = 25000;
            }
        } else {
            if (total > 200000) {
                diskon = 10000;
            } else {
                diskon = 0;
            }
        }
        bayar = total - diskon;
        System.out.println("Total yang harus dibayar: Rp " + bayar);
    }
}
```

Pada kode ini tersebut, kasir diminta untuk memasukkan input, apakah pelanggan mempunyai kartu anggota atau tidak. Selanjutnya kasir juga perlu memasukkan total harga barang belanjaan.

Kondisi yang akan diseleksi pertama kali adalah nilai dari variabel "kartu". Jika pengguna memasukkan input "y", maka seleksi kondisi ini bernilai benar, dan selanjutnya dilakukan penyeleksian total harga barang belanjaan untuk menentukan diskon yang diperoleh.

d. Hasil dan Pembahasan

Mahasiswa memahami dan mampu mengimplementasikan percabangan If Else Nested

e. Kesimpulan

Mahasiswa mengimplemetasikan percabangan if else secara tepat

f. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambat dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					

Acara 11

Pokok Bahasan : Perulangan 1

Acara Praktikum/Praktek: Minggu 3 / 3

Tempat : Politeknik Negeri Jember

Alokasi Waktu : 100 Menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menggunakan Perulangan (For Loop, While Loop, Do While Loop).
2. Mahasiswa mengimplementasikan Perulangan (For Loop, While Loop, Do While Loop) dalam bahasa pemrograman Java.

b. Indikator

Kemampuan mahasiswa dalam menjelaskan dan mengimplementasikan Perulangan (For Loop, While Loop, Do While Loop).

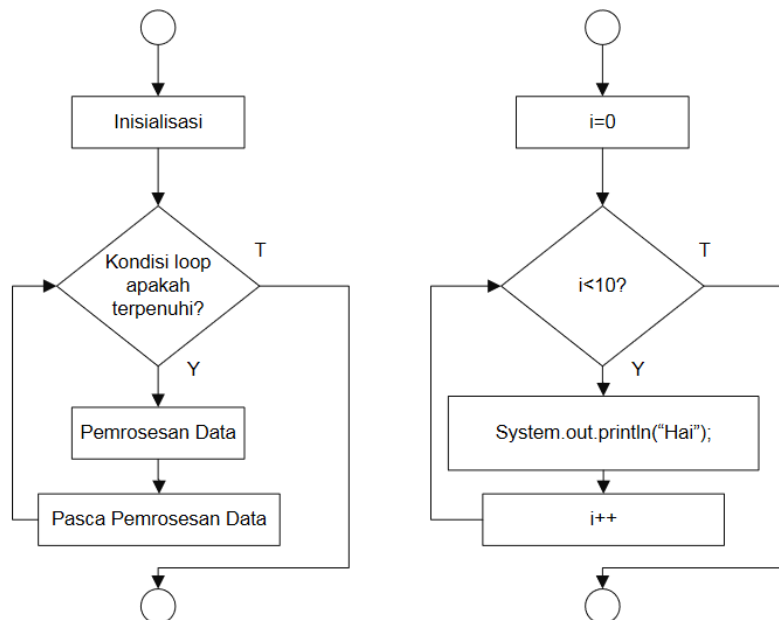
c. Dasar Teori

Perulangan

Ada beberapa perintah yang digunakan untuk menerapkan perulangan dalam Java, antara lain: For loop, While loop dan Do while loop

1. FOR LOOP

`for` adalah kode yang digunakan untuk menjalankan serangkaian kode secara berulang-ulang. Pada kode `for` ini terdapat beberapa komponen yang dicantumkan, antara lain: (1) inisialisasi, (2) kondisi, (3) perubahan nilai, (4) statement yang diulang.



Berikut ini format sintaks untuk kode for.

```
for(inisialisasi; kondisi; perubahan_nilai){  
    statement;  
    ...  
}
```

Berikut keterangan tentang sintaks for di atas:

- Inisialisasi: Nilai awal (inisialisasi) pada variabel yang digunakan untuk looping
- Kondisi: Kondisi yang akan dicek untuk menentukan apakah looping akan dijalankan atau tidak. Jika kondisi bernilai benar, looping dilanjutkan. Namun, jika tidak, looping akan berhenti.
- perubahan_nilai: ekspresi yang dijalankan di akhir setiap statement.
- Statement(s): merupakan perintah dalam for loop body.

```
import java.util.Scanner;  
public class PemrogramanJava {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        for (int a = 0; a < 10; a++) {  
            System.out.println("Hello JTI");  
        }  
    }  
}
```

Kode program di atas akan menyebabkan tulisan "Hello JTI" muncul se-banyak sepuluh kali. Dimulai saat $i = 0$ hingga $i = 9$.

```
Output - Pemrograman Java (run)  
  
run:  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
Hello JTI  
BUILD SUCCESSFUL (total time: 0 seconds)
```

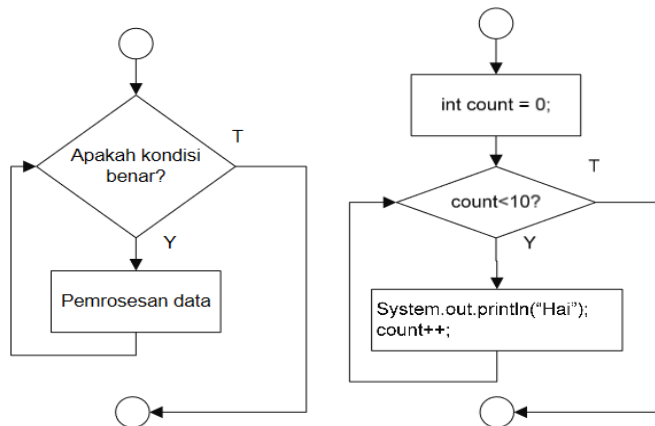
2. While loop

Perulangan while loop dimulai dengan mengevaluasi kondisi apakah benar atau tidak. Jika benar, statement akan dijalankan dan program kembali lagi melakukan evaluasi kondisi apakah benar atau tidak. Demikian seterusnya hingga kondisi bernilai salah. Jika salah, statement akan dilewati dan evaluasi kondisi tidak akan dilakukan lagi. Dengan kata lain, perintah pada statements akan terus dieksekusi selama kondisi pada condition bernilai benar (true).

Syntax While loop

```
while (condition) { statement(s); }
```

Berikut contoh flowchart perintah while



Berikut iterasi menggunakan perintah while.

```
package pemrograman.java;

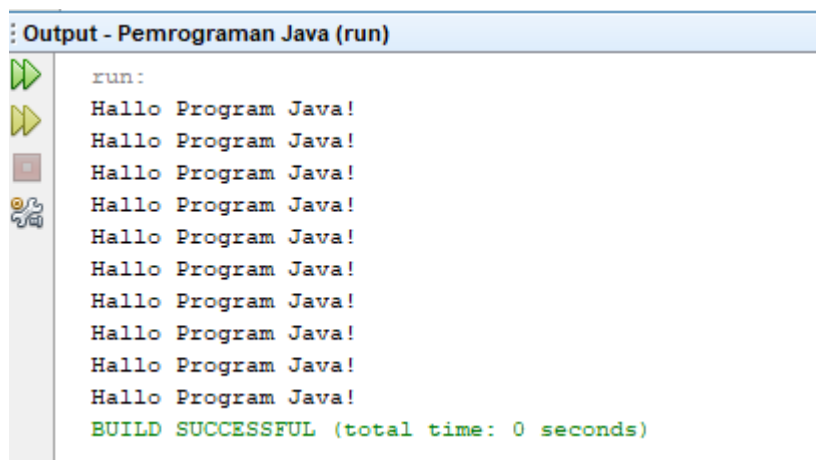
/**
 *
 * @author Shinta
 */
import java.util.Scanner;
public class PemrogramanJava {

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

        int count = 0; while (count < 10) {
            System.out.println("Hallo Program Java!");
            count++;
        }
    }
}
```

Keterangan contoh kode program dengan While Loop diatas:

- Variabel count mempunyai nilai awal 0.
- Perintah untuk menampilkan tulisan "Halo Program Java" akan terus dieksekusi selama nilai count kurang dari 10.
- Setelah menampilkan tulisan "Halo Program Java", nilai pada variabel count ditambah dengan satu.
- Dengan demikian, tulisan "Halo Program Java" akan ditampilkan sebanyak 10 kali. Mulai dari saat nilai count=0 hingga nilai count=9. Seperti dibawah ini.



```
run:
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
Halo Program Java!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Kesalahan penghitungan dapat menimbulkan infinite loop di mana perulangan terjadi terus dan tidak bisa berhenti. Kasus ini terjadi jika kita salah menuliskan kode sehingga kondisi pada while loop selalu benar. Untuk menghentikan infinite loop, tutup program aplikasi dan perbaiki kode agar tidak melakukan infinite loop.

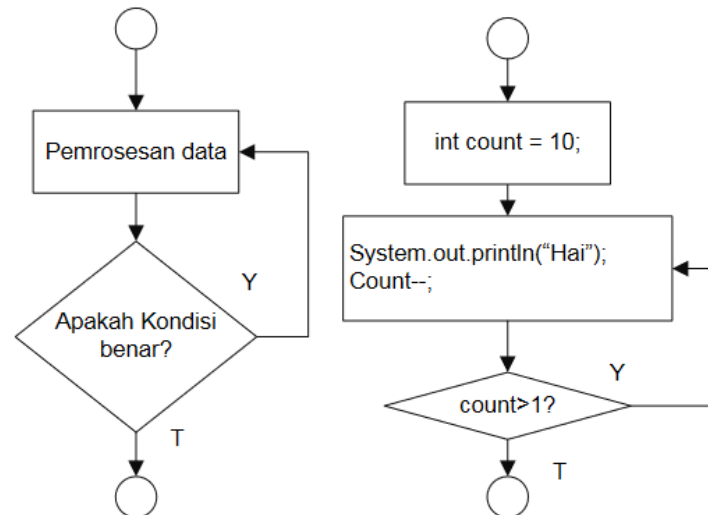
3. Do While Loop

Perulangan dengan *do while loop* hampir sama dengan *while loop*. Bedanya adalah perintah ini pasti menjalankan statement minimal satu kali. Jadi, meskipun kondisi tidak memenuhi, perintah dalam statement tetap dijalankan satu kali. Berikut contoh kode program untuk do while loop.

Sintaks secara umum:

```
do {
    statement(s);
} while (condition);
```

Perulangan dengan do while loop hampir sama dengan while loop. Bedanya adalah perintah ini pasti menjalankan statement minimal satu kali. Jadi, meskipun kondisi tidak memenuhi, perintah dalam statement tetap dijalankan satu kali.



Berikut contoh kode program untuk *do while loop*.

```

import java.util.Scanner;
public class PemrogramanJava {

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

        int count = 10;
        do{
            System.out.println("Hai");
            count--;
        } while (count > 1);
    }
}

```

Dari perintah di atas, tulisan "Hai" akan muncul sebanyak sembilan kali. Mulai dari saat nilai variabel count=10 hingga menjadi 2.

Hasil program

```

: Output - Pemrograman Java (run)

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

```

Tugas Praktikum

Silahkan buat flowchart dari program berikut dan tulis ulang code program berikut ini, kemudian jalankan!

1. perulangan dengan **for**.

```
public static void main(String[] args) {  
    int angka, fac, i;  
    System.out.println ("====PROGRAM MENGHITUNG NILAI FAKTORIAL DENGAN FOR====");  
    System.out.print("Masukkan bilangan : ");  
  
    Scanner input = new Scanner (System.in);  
    angka = input.nextInt();  
    fac =1;  
    for (i=1; i<=angka;i++)  
    {  
        fac = fac*i;  
    }  
    System.out.printf("Nilai faktorial bilangan tersebut adalah : %d \n", fac);  
}
```

2. Perulangan dengan **while**

```
public static void main(String[] args) {  
    Scanner input = new Scanner (System.in);  
    int angka, fac, i;  
    System.out.println ("====PROGRAM MENGHITUNG NILAI FAKTORIAL DENGAN WHILE====");  
    System.out.print("Masukkan bilangan : ");  
  
    angka = input.nextInt();  
    fac =1;  
    i = 1;  
    while (i<=angka)  
    {  
        fac = fac*i;  
        i++;  
    }  
    System.out.printf("Nilai faktorial bilangan tersebut adalah : %d \n", fac);  
}
```

3. Perulangan dengan **do-while** dan **break**

```
public static void main(String[] args) {
    Scanner input = new Scanner (System.in);
    int angka, b;
    System.out.println ("=====PROGRAM LOOP DENGAN BREAK=====");
    b=0;
    do
    {
        System.out.print("Masukkan bilangan : ");
        angka = input.nextInt();
        b += angka;
        if (b>50) break;
    }
    while (true);
    System.out.printf("Angka berhenti pada angka : %d \n", b);
}
```

d. Hasil dan Pembahasan

1. Laporan hasil praktikum
2. Laporan berupa file word dikumpulkan pada <http://jti.poliije.ac.id/elearning>

e. Kesimpulan

Mahasiswa memahami dan mampu mengimplementasikan percabangan Perulangan

f. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambatan dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					

Acara 12

Pokok Bahasan : Perulangan 2

Acara Praktikum/Praktek: Minggu 3 / 4

Tempat : Politeknik Negeri Jember

Alokasi Waktu : 100 Menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menggunakan Perulangan (Nested loop).
2. Mahasiswa mengimplementasikan Perulangan (Nested loop). dalam bahasa pemrograman Java.

b. Indikator

Kemampuan mahasiswa dalam menjelaskan dan mengimplementasikan Perulangan (Nested loop).

c. Dasar Teori

Sebelumnya, telah dibahas tentang konsep dasar perulangan. Pada bahasan tersebut disebutkan bahwa logika perulangan digunakan untuk melakukan beberapa proses atau statement program secara berulang-ulang, dengan suatu pola tertentu. Pada perulangan, proses atau statement akan terus dilakukan atau akan terus diulang, selama kondisi perulangan bernilai benar/true. Dan sebaliknya, perulangan akan berhenti dan proses atau statement tidak akan dieksekusi lagi ketika kondisi perulangan bernilai salah/false. Jadi, dalam logika perulangan, suatu kondisi perulangan diperlukan untuk menentukan apakah suatu perulangan masih akan berlangsung lagi atau harus berhenti.

Nested Loop

Nested loop terjadi ketika terdapat loop di dalam loop. Dengan demikian, jumlah perulangan yang terjadi lebih banyak; yaitu perkalian antara loop yang di dalam dengan loop yang di luar. Nested loop banyak dipakai untuk melakukan perulangan bertingkat, misalnya untuk menampilkan daftar perkalian bilangan, faktorial, dan lain-lain.

Salah satu contoh nested loop misalnya pada permasalahan untuk membuat tabel perkalian:

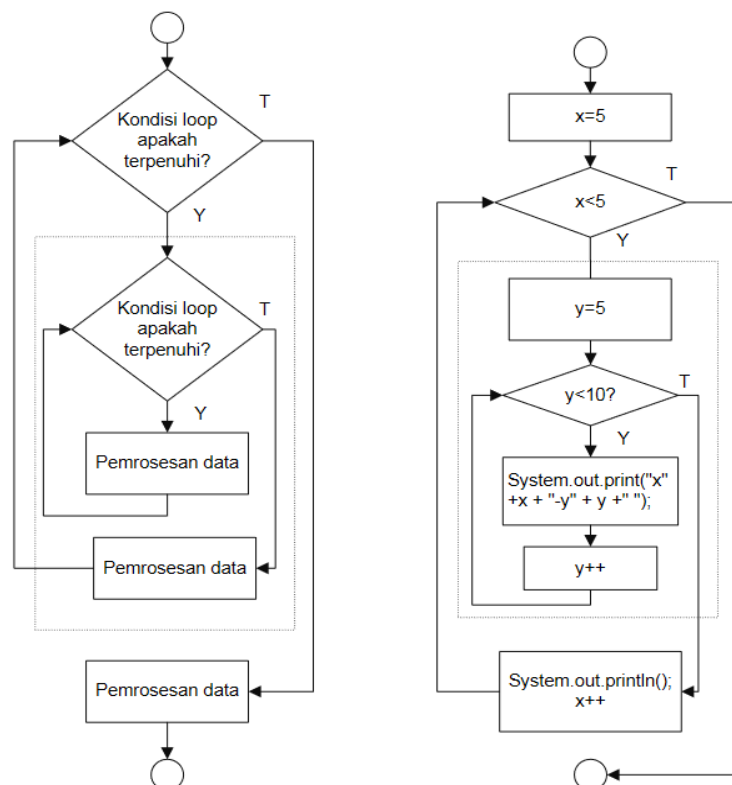
1 2 3 4 5 6 7 8

1	2	3	4	5	6	7	8
2	4	6	8	10	12	14	16
3	6	9	12	15	18	21	24
4	8	12	16	20	24	28	32
5	10	15	20	25	30	35	40
6	12	18	24	30	36	42	48
7	14	21	28	35	42	49	56
8	16	24	32	40	48	56	64

Bentuk Umum

Secara umum dan sederhana flowchart perulangan bersarang ditunjukkan pada gambar dibawah.

1. Pada gambar tersebut terlihat terdapat perulangan dalam yang berada di dalam perulangan luar. Tiap perulangan memiliki kondisi sendiri-sendiri dan statement masing-masing.



Dan apa dituliskan dalam pseudocode yang sangat sederhana

```
import java.util.Scanner;
public class PemrogramanJava {

    public static void main(String[] args){    for(int x = 0; x < 5; x++){
        for (int y = 5; y < 10; y++)
            System.out.print("x"+x + "-y" + y + "");
        System.out.println();
    }
}
```

Output - Pemrograman Java (run)

```
run:
x0-y5x0-y6x0-y7x0-y8x0-y9
x1-y5x1-y6x1-y7x1-y8x1-y9
x2-y5x2-y6x2-y7x2-y8x2-y9
x3-y5x3-y6x3-y7x3-y8x3-y9
x4-y5x4-y6x4-y7x4-y8x4-y9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada contoh ini, kode program ini, akan dibuat program yang menam-pilkan urutan looping. Terlihat bahwa loop dengan variabel y terletak di dalam, sedangkan loop dengan variabel x terletak di luar.

Praktikum

TUGAS 1

Cobalah code program berikut ini, buatlah laporan dan berikan deskripsi:

```
import java.util.Scanner;
public class PemrogramanJava {

    public static void main(String[] args) {    long bilangan = 15L; // Nilai bilangan maksimal
    long faktorial = 1L; // Variabel untuk menyimpan faktorial
    // Looping untuk mencari faktorial pada bilangan
    for (long i = 1L; i <= bilangan; i++)
    {    faktorial = 1L; // inisialisasi faktorial
    // Menghitung nilai faktorial
    for (long faktor = 2; faktor <= i; faktor++) {
        faktorial *= faktor;
    }
    System.out.println("Faktorial dari "+i + " adalah " + faktorial);
    } }
}
```

Jalankan program tersebut apakah hasilnya seperti di bawah ini?

```
Faktorial dari 4 adalah 24
Faktorial dari 5 adalah 120
Faktorial dari 6 adalah 720
Faktorial dari 7 adalah 5040
Faktorial dari 8 adalah 40320
Faktorial dari 9 adalah 362880
Faktorial dari 10 adalah 3628800
Faktorial dari 11 adalah 39916800
Faktorial dari 12 adalah 479001600
Faktorial dari 13 adalah 6227020800
Faktorial dari 14 adalah 87178291200
Faktorial dari 15 adalah 1307674368000
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Alat dan Bahan

1. Komputer
2. Netbeans
3. JDK

e. Hasil dan Pembahasan

1. Laporan hasil praktikum
2. Laporan berupa file word dikumpulkan pada <http://jti.polije.ac.id/elearning>

f. Kesimpulan

Mahasiswa memahami dan mampu mengimplementasikan Perulangan (Nested Loop)

g. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambatan dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					