

PANDUAN PRAKTIKUM  
**ALGORITMA DAN PEMROGRAMAN**  
**MI – TK – KA**



Disusun Oleh :

*Femi Dwi Astuti*

**STMIK AKAKOM**  
Yogyakarta  
2014

## KATA PENGANTAR

Puji Syukur kepada Allah SWT karena modul Algoritma dan Pemrograman ini dapat terselesaikan. Modul ini dibuat karena tuntutan kurikulum baru atau disebut sebagai kurikulum 2014 STMIK AKAKOM yang mulai diberlakukan pada tahun akademik 2014-2015.

Modul ini membahas tentang dasar-dasar pemrograman dengan menggunakan bahasa Pemrograman Java. Mulai dari proses sekuensial, tipe data, seleksi, perulangan, method, larik sampai kepada penerapannya dengan menggunakan variasi bentuk perintah yang ada.

Tentu saja dengan segala keterbatasan yang ada, modul ini jauh dari sempurna. Untuk itu masukan dan saran untuk perkembangan modul ini sangat diharapkan.

Akhirnya semoga modul ini berguna dalam membantu mahasiswa untuk mempelajari tentang Algoritma dan Pemrograman.

Yogyakarta, September 2014.

Penulis

## DAFTAR ISI

Judul.....	i
Kata Pengantar.....	ii
Daftar Isi.....	iii
Pertemuan ke 1.....	1
Pertemuan ke 2.....	4
Pertemuan ke 3.....	9
Pertemuan ke 4.....	15
Pertemuan ke 5.....	20
Pertemuan ke 6.....	24
Pertemuan ke 7.....	28
Pertemuan ke 8.....	31
Pertemuan ke 9.....	33
Pertemuan ke 10.....	36
Pertemuan ke 11.....	39
Pertemuan ke 12.....	41
Pertemuan ke 13.....	43
Pertemuan ke 14.....	46

## PERTEMUAN KE – 1 PENGANTAR ALGORITMA

### A. TUJUAN

Mahasiswa dapat menyusun algoritma untuk menyelesaikan suatu permasalahan

### B. TEORI SINGKAT

#### **Pengenalan Algoritma**

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Kata logis disini berarti benar sesuai dengan logika manusia. Untuk menjadi sebuah algoritma, urutan langkah yang ditempuh untuk menyelesaikan masalah harus memberikan hasil yang benar.

Misalkan saja “algoritma aktifitas pagi hari sebelum berangkat kerja” yang dikerjakan oleh seorang eksekutif junior untuk turun dari tempat tidur dan bekerja: (1) Turun dari tempat tidur; (2) melepas piyama; (3) mandi; (4) berpakaian; (5) makan pagi; (6) baca koran; (7) pergi kerja. Rutin ini membuat eksekutif junior bekerja dengan persiapan yang baik untuk membuat keputusan-keputusan kritis. Andaikan langkah yang sama dilakukan dengan urutan yang sedikit berbeda, misalnya: (1) Turun dari tempat tidur; (2) melepas piyama; (3) berpakaian; (4) mandi; (5) baca koran; (6) makan pagi; (7) berangkat kerja. Di dalam kasus ini eksekutif junior tersebut berangkat ke tempat kerja dalam keadaan basah kuyub.

#### **Sifat algoritma :**

1. Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
2. Tidak tergantung pada suatu bahasa pemrograman.
3. Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.

### C. PRAKTIK :

#### **PRAKTIK 1**

Buat algoritma dengan menggunakan olah kata (MS-Word) untuk kasus-kasus berikut:

##### **1. Algoritma memasak mie instan.**

1. Rebus air hingga mendidih.
2. Masukkan mie instan ke dalam air mendidih tersebut.
3. Tunggu 3 menit
4. Angkat dan tiriskan.
5. Campurkan bumbu-bumbu
6. Aduk hingga rata
7. Mie siap disajikan/dimakan

##### **2. Algoritma menghitung luas persegi panjang.**

1. Masukkan panjang
2. Masukkan lebar
3. Nilai luas adalah panjang dikalikan(\*) lebar
4. Tampilkan luas

### 3. Algoritma menghitung Luas Segi Tiga

1. Masukkan alas
2. Masukkan tinggi
3. Hitung luas dengan cara kalikan alas dengan tinggi lalu kalikan dengan 0.5  
( $0.5 \cdot \text{alas} \cdot \text{tinggi}$ )
4. Cetak luas--

### 4. Algoritma pendaftaran Calon Mahasiswa Baru di STMIK AKAKOM.

1. Membeli Formulir di Bagian Penerimaan mahasiswa baru.
2. Mengisi dan menyerahkan formulir ke Bagian Penerimaan mahasiswa baru
3. Mendapatkan No test bagi yang lewat jalur test.
4. Melakukan Test
5. Jika Test sudah selesai, mendapatkan hasil pengumuman test.
6. Jika mahasiswa dinyatakan diterima diberi surat diterima

## PRAKTIK 2

Buatlah algoritma untuk menyelesaikan permasalahan berikut

1. Bidang Akademik  
Akan dihitung total nilai dari seorang mahasiswa dimana total nilai diperoleh dari 30% nilai UTS, 30% nilai tugas dan 40% nilai UAS.
  - a. Masukkan nilai UTS
  - b. Masukkan nilai TUGAS
  - c. Masukkan nilai UAS
  - d. Hitung total nilai  $= (30\% \cdot \text{UTS}) + (30\% \cdot \text{TUGAS}) + (40\% \cdot \text{UAS})$
  - e. Tampilkan total nilai
2. Bidang Akuntansi  
Hitung jumlah yang harus dibayar oleh seorang pembeli yang membeli satu(1) atau beberapa barang dimana setiap barang yang dibeli memiliki harga sesuai dengan item barangnya, sehingga dihasilkan hasil akhir berupa TotalBayar yang harus dibayarkan oleh pembeli
  - a. Masukkan harga barang
  - b. Masukkan jumlah barang
  - c. Hitung sub total dengan cara mengalikan harga barang dengan jumlah barang
  - d. Jika barang yang dibeli lebih dari satu, ulangi langkah a sampai c hingga tidak ada lagi barang yang dibeli
  - e. Hitung total bayar dengan menambahkan masing-masing sub total
3. Bidang Matematika  
Hitung sisi miring sebuah segitiga siku-siku jika diketahui rumusnya  $c^2 = a^2 + b^2$ 
  - a. Masukkan a
  - b. Masukkan b
  - c. Kuadratkan sisi pertama

- d. Kuadratkan sisi kedua
  - e. Hitung sisi miring kuadrat dengan cara Jumlahkan kuadrat sisi pertama dengan kuadrat sisi kedua
  - f. Hitung sisi miring dengan cara mencari akar dari sisi miring kuadrat
4. Mengambil sejumlah uang di ATM

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS

- a) Buat algoritma untuk menentukan kriteria kelulusan administrasi jika disyaratkan seseorang dinyatakan LENGKAP administrasi jika telah melengkapi data-data sebagai berikut :
  - 1) Akte Kelahiran
  - 2) STTB
  - 3) Nilai UAN > 5

Jika salah satu atau ke-3 data tersebut tidak terpenuhi maka kelengkapan Administrasi dinyatakan KURANG LENGKAP.

- b) Dalam perekrutan anggota kepolisian disyaratkan beberapa hal sebagai berikut :
  1. Umur Minimal 19 Tahun, Maksimal 20 Tahun
  2. Tinggi Badan Minimal 170 cm
  3. Berat Badan Ideal (Berat Badan Ideal =  $(\text{Tinggi Badan} - 100) - 10\% (\text{Tinggi Badan} - 100)$ )
  4. Hasil Akhir berupa LULUS jika ke-3 persyaratan tersebut terpenuhi, jika salah satu persyaratan tidak terpenuhi maka dinyatakan GAGAL

## PERTEMUAN KE – 2 DIAGRAM AKTIFITAS DAN DASAR PEMROGRAMAN

### A. TUJUAN

- Memahami tentang penggunaan simbol-simbol *activity diagram*
- Menggunakan *activity diagram* untuk menyelesaikan masalah algoritma

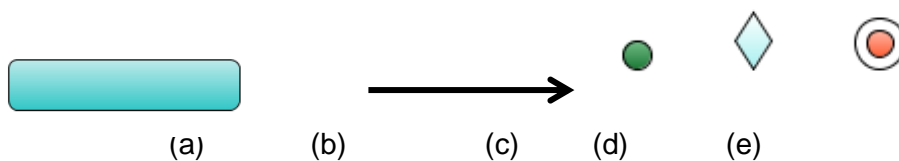
### B. TEORI SINGKAT

#### Pengenalan UML

*Unified Modeling Language* (UML) belakangan ini merupakan skema representasi grafis yang banyak digunakan secara luas untuk pemodelan sistem berorientasi object. UML ini telah menyatukan berbagai skema notasi populer bersama-sama. Banyak yang merancang sistem menggunakan bahasa ini (dalam bentuk diagram) untuk memodelkan sistem mereka.

UML adalah bahasa grafis yang kompleks dan kaya dengan fitur. Salah satu model diagramnya adalah *Activity Diagram*. Dalam pembahasan algoritma ini, digram UML yang akan digunakan adalah *activity diagram*.

Sebuah *activity diagram* memodelkan aspek dari tingkah laku sistem. *Activity diagram* memodel aliran kerja obyek (urutan aktivitas) selama eksekusi program. *Activity diagram* adalah flowchart yang memodel aksi yang akan dikerjakan oleh obyek beserta dengan ordenya.

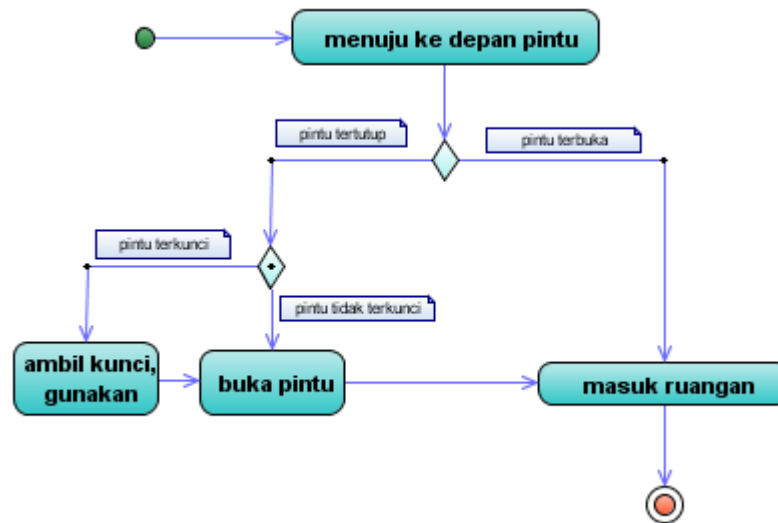


Gambar 1. Simbol *Activity Diagram*

UML merepresentasikan aktivitas dengan simbol sebuah oval (Gambar 1.a.) di dalam *activity diagram*. Nama aktivitas diletakkan di dalam oval. Sebuah anak panah (Gambar 1.b.) menghubungkan dua aktivitas yang menunjukkan orde dimana aktivitas dikerjakan. Lingkaran padat (Gambar 1.c.) menunjukkan dimulainya aktivitas. Percabangan ditunjukkan dengan sebuah jajaran genjang (Gambar 1.d.) dan lingkaran padat dan ditambah dengan lingkaran di luarnya menandakan akhir dari aktifitas (Gambar 1.e.).

Sekarang, mari kita lihat contoh *activity diagram*. Misalkan kita akan membuat *activity diagram* seseorang yang akan masuk ke dalam sebuah ruangan yang berpintu. Pertama kali yang dikerjakan adalah menuju ke pintu. Kemudian melihat apakah pintu dalam keadaan terbuka atau tertutup. Jika dalam keadaan terbuka, dia langsung masuk. Jika pintu dalam keadaan tertutup, maka orang tersebut akan mengecek apakah pintu dalam keadaan terkunci atau tidak. Kalau terkunci, maka orang tersebut akan mengambil kunci dan membuka pakai kunci, jika tidak terkunci, dia akan langsung membuka pintu. Dengan orang tersebut sudah bisa

masuk ke ruangan, maka aktifitas masuk ruangan selesai. *Activity diagram* untuk proses tersebut diperlihatkan pada Gambar 1.2.



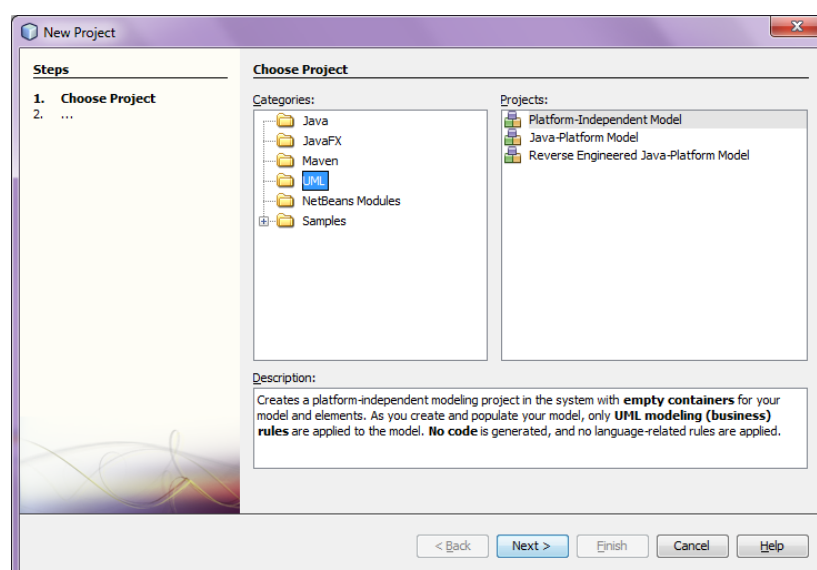
Gambar 2. *Activity diagram* untuk masuk ruangan

Java adalah bahasa pemrograman multi platform. Java tidak menyediakan IDE khusus seperti halnya bahasa pemrograman yang lain. Pemrogram bisa menggunakan IDE yang *support* ke Java, misalnya Netbeans, Eclips, TexPad, dan lain-lain. Editor teks bisa digunakan semisal Notepad. Jika editor yang digunakan tidak support Java, kita tinggal menyimpannya dalam ekstensi .java kemudian kompilasi dan menjalankannya menggunakan command prompt.

### C. PRAKTIK :

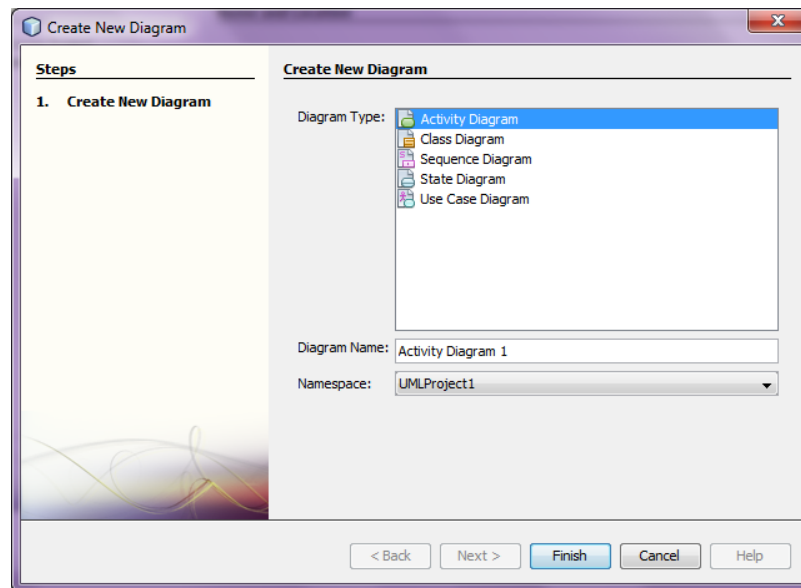
#### PRAKTIK 1

1. Buka Netbeans.
2. Pilih New Project
3. Pilih UML, Platform-Independent Model

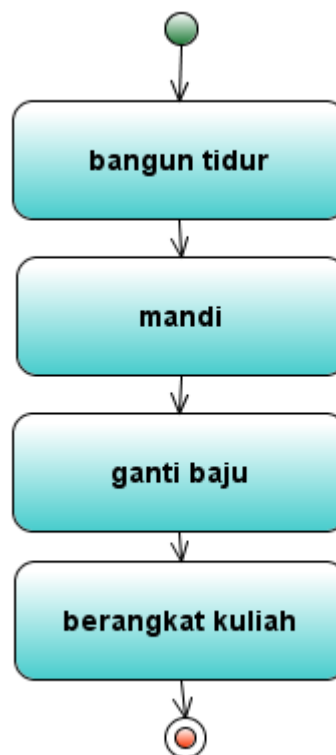




4. Beri nama dan pilih lokasi
5. Pilih Activity Diagram



6. Buat Activity Diagram untuk algoritma bangun tidur, seperti berikut

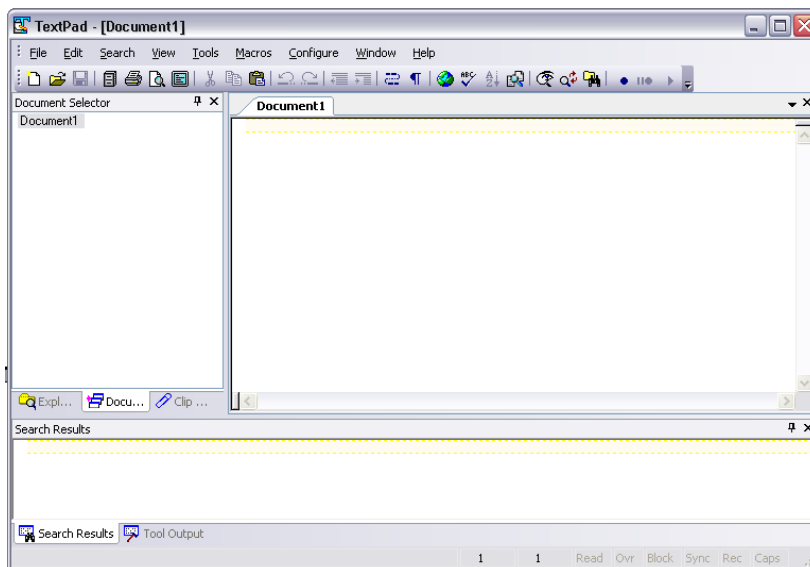


7. Buatlah activity diagram untuk praktik 1 nomor 1-3 pada pertemuan 1

## PRAKTIK 2

Langkah-langkah menggunakan TextPad :

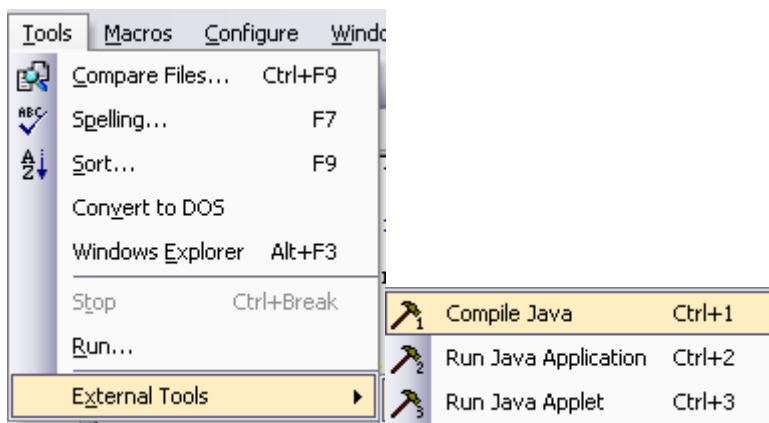
1. Pilih Menu Start
2. Pilih Programming
3. Pilih TextPad, maka akan muncul tampilan awal TextPad sebagai berikut :



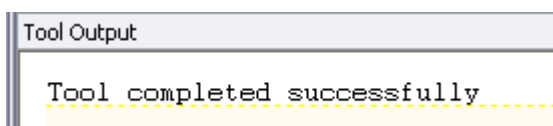
4. Ketik program sederhana berikut ini :

```
public class Pertama{
public static void main(String args[]){
    System.out.println("Selamat Datang di STMIK AKAKOM");
}
}
```

5. Simpan dengan nama **Pertama**, harus sama dengan nama class yang ada pada program
6. Jalankan program tersebut dengan memilih menu sbb:

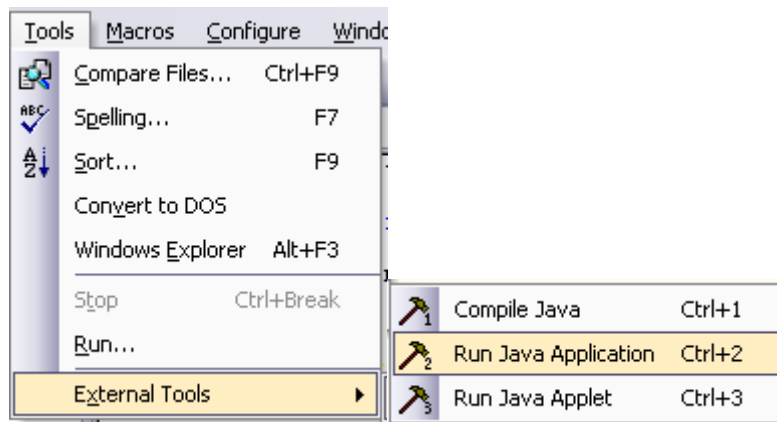


7. Setelah tidak ada kesalahan dan pada Tools Output muncul tulisan sebagai berikut :



Jika tidak ada pesan kesalahan, berarti kompilasi berhasil. Ini ditandai dengan adanya tambahan satu file yaitu **Pertama.class**.

8. Lakukan perintah berikut :



#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

## PERTEMUAN KE – 3 TIPE DATA, OPERATOR, EKSPRESI, PROSES SEKUENSIAL

### A. TUJUAN

- Memahami tentang tipe data dan dapat menggunakannya.
- Memahami tentang penggunaan operator dan ekspresi dalam program.

### B. TEORI SINGKAT

#### 1. Tipe Data Primitif

Bahasa Pemrograman Java adalah bahasa pemrograman yang selalu menggunakan tipe data untuk setiap variabelnya. Itu berarti bahwa semua variabel harus dideklarasikan terlebih dahulu sebelum mereka digunakan.

Misalnya

```
int nilai = 1;
```

- **byte**: Tipe data byte adalah 8-bit integer bertanda *two's complement*. Tipe ini mempunyai nilai minimum -128 dan nilai maksimumnya adalah 127.
- **short**: Tipe data short merupakan integer 16 bit *two's complement* yang mempunyai nilai minimum -32,768 dan nilai maksimum 32,767.
- **int**: Tipe data int adalah integer 32 bit *two's complement*. Dia mempunyai nilai minimum -2,147,483,648 sedangkan nilai maksimumnya adalah 2,147,483,647 (inclusive).
- **long**: Tipe data long adalah integer 64 bit *two's complement*. Nilai minimumnya adalah -9,223,372,036,854,775,808 sedangkan nilai maksimumnya adalah 9,223,372,036,854,775,807 (inclusive). Gunakan tipe data ini pada saat anda memerlukan jangkau nilai yang lebih besar daripada yang bisa disajikan oleh int.
- **float**: Tipe data float merupakan *single-precision 32-bit IEEE 754 floating point*.
- **double**: Tipe data double adalah *double-precision 64-bit IEEE 754 floating point*.

Tabel 1.1. Batas nilai floating point

Parameter	float	double
N	24	53
K	8	11
$E_{max}$	+127	+1023
$E_{min}$	-126	-1022

2. **boolean**: Tipe data boolean hanya mempunyai dua nilai yang mungkin, yaitu true dan false. Gunakan tipe data ini untuk flag-flag sederhana untuk menjejak/menelusuri kondisi true atau false. Tipe data ini merepresentasikan satu bit informasi, tetapi ukurannya tidak didefinisikan dengan tepat.
3. **char**: Tipe data char adalah karakter Unicode 16 bit. Tipe data ini mempunyai nilai minimum '\u0000' (atau 0) dan nilai maksimum '\uffff' (atau 65,535).

#### 2. Variabel dan Konstanta

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah selama eksekusi berlangsung.

Konstanta adalah nilai yang tetap.

Contoh :

```
float jumlah;  
jumlah = 10;
```

### 3. Konstanta bernama

Hal ini dapat dilakukan dengan menggunakan kata kunci `const`.

Contoh :

```
const float PHI = 3.14;
```

## PERNYATAAN MASUKAN DAN KELUARAN DALAM JAVA

Dalam bahasa pemrograman Java, kita akan menggunakan pernyataan masukan dan keluaran. Pernyataan masukan adalah pernyataan untuk mendapatkan masukan dari keyboard. Sedangkan pernyataan keluaran adalah pernyataan untuk menampilkan sesuatu nilai ke layar.

### 1. Komponen keluaran

Untuk menampilkan ke layar secara tekstual, java mempunyai fasilitas output, yaitu dengan perintah

```
System.out.print( hal yang akan ditampilkan, bisa lebih dari satu);
```

untuk menampilkan ke layar dan setelah selesai tidak berpindah baris. Akibatnya perintah keluaran berikutnya akan ditampilkan pada baris yang sama. Dan perintah

```
System.out.println(hal yang akan ditampilkan, bisa lebih dari satu);
```

untuk menampilkan ke layar dan setelah selesai berpindah baris. Akibatnya perintah keluaran setelah itu akan ditampilkan pada baris berikutnya.

```
System.out.println("Hasil penjumlahan "+hasil);
```

```
//menampilkan isi variabel hasil
```

### 2. Komponen masukan

Untuk memasukkan sebuah nilai ke variabel yang sudah didefinisikan digunakan kelas `Scanner` (kelas ini disediakan mulai Java versi 1.5).

#### a. Input data bertipe Integer

Untuk menginputkan data dengan tipe integer digunakan method `nextInt` di dalam kelas `Scanner`.

#### b. Input data bertipe String

Untuk menginputkan data dengan tipe integer digunakan method `nextInt` di dalam kelas `Scanner`.

#### c. Input tipe yang lain

Untuk input data dari keyboard dengan tipe yang lain dan disediakan oleh kelas `Scanner` adalah sebagai berikut

- `nextBoolean` : input tipe boolean (true atau false saja)
- `nextShort` : input tipe short integer

- nextLong : input tipe long integer
- nextFloat : input tipe float
- nextDouble : input tipe double

Operator adalah simbol khusus yang menyajikan operasi khusus pada satu, dua, atau tiga operand dan kemudian mengembalikan hasilnya. Operator-operator tersebut digunakan untuk membentuk ekspresi. Sebuah konstanta atau entitas dengan sendirinya merupakan ekspresi, demikian juga kombinasi dari konstanta dan variabel dengan operator. Suatu ekspresi yang diikuti dengan sebuah titik koma adalah sebuah pernyataan. Operator mempunyai sifat- sifat

Sifat	Keterangan	Contoh
Unary	Operator yang hanya melibatkan 1 operand	-1
Binary	Operator yang hanya melibatkan 2 operand	1 + 2
Tenary	Operator yang hanya melibatkan 3 operand	1 + 2 * 2

Pada saat kita akan menggunakan operator-operator dari bahasa pemrograman Java, kita harus mengetahui terlebih dahulu operator yang mana yang mempunyai *precedence* yang lebih tinggi. Operator di dalam tabel 2.1. dituliskan sesuai dengan *precedence* ordernya. Semakin ke bawah, maka *precedence*-nya lebih rendah. Operator dengan *precedence* yang lebih tinggi dikerjakan lebih dulu dari pada operator dengan *precedence* yang lebih rendah. Operator yang ada di dalam baris yang sama mempunyai *precedence* yang sama. Pada saat operator dari *precedence* yang sama muncul di dalam ekspresi yang sama, harus diatur yang mana yang harus dikerjakan lebih dulu. Semua operator biner kecuali untuk operator pemberian dikerjakan dari kiri ke kanan. Operator pemberian dikerjakan dari kanan ke kiri.

**Tabel 2.1. Precedence Operator**

Operator	Precedence
<i>Postfix</i>	<i>expr++ expr--</i>
<i>Unary</i>	<i>++expr --expr +expr -expr ~ !</i>
Multiplikasi	* / %
Aditif	+ -
Pergeseran	<< >> >>>
Relasional	< > <= >= instanceof
Persamaan	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
Logika AND	&&
Logika OR	
<i>Tenary</i>	? :
Pemberian	= += -= *= /= %= &= ^=  = <<= >>= >>>=

## Operator Aritmatika

Operator	Keterangan
+	Penjumlahan (tanda plus)
-	Pengurangan (tanda minus)
*	Perkalian
/	Pembagian
%	Sisa Pembagian

Operator \*, / dan % mempunyai prioritas yang sama, tetapi lebih tinggi daripada + atau -.

### C. PRAKTIK

1. Buatlah program seperti berikut :

```
public class Dua
{
    public static void main(String args[])
    {
        System.out.println("          BIODATA PRIBADI          ");
        System.out.println("=====");
        System.out.println("Nama      : Femi Dwi Astuti");
        System.out.println("NoMhs    : 091159");
        System.out.println("Jurusan: Teknik Informatika");
    }
}
```

Pada program di atas tambahkan data Jenis\_Kelamin, Tgl\_Lahir, Alamat

2. Buatlah program seperti berikut :

```
import java.util.Scanner;
public class Menu
{
    public static void main(String args[])
    {
        Scanner masuk = new Scanner(System.in);
        int pil;
        System.out.println("*****");
        System.out.println("Silahkan pilih menu");
        System.out.println("*****");
        System.out.println("1. Paket lele");
        System.out.println("2. Paket ayam");
        System.out.println("3. Paket nila");
        System.out.print("Masukkan pilihan (1,2,3) = ");
        Pil = masuk.nextInt();
        System.out.println("Paket yang anda pilih adalah paket "+pil);
    }
}
```

3. Buatlah program seperti berikut :

```
import java.util.Scanner;
public class Jumlah{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int A,B, jumlah;
        System.out.print("Bilangan 1 : ");
        A = masuk.nextInt();
        System.out.print("Bilangan 2 : ");
        B = masuk.nextInt();
    }
}
```

```

    jumlah = A + B;
    System.out.println("Jumlah = " + jumlah);
}
}

```

4. Modifikasi program nomor 3 menjadi seperti berikut :

```

import java.util.Scanner;
public class Kombinasi
{
    public static void main(String args[])
    {
        Scanner masuk = new Scanner(System.in);
        int A,B;
        float jumlah,kurang,kali,bagi,sisa;
        System.out.print("Bilangan 1 : ");
        A = masuk.nextInt();
        System.out.print("Bilangan 2 : ");
        B = masuk.nextInt();
        jumlah = A + B;
        kurang=A-B;
        kali=A*B;
        bagi=A/B;
        sisa=A%B;
        System.out.println("Hasil operator aritmatika");
        System.out.println("*****");
        System.out.println("Jumlah = " + jumlah);
        System.out.println("Kurang = " + kurang);
        System.out.println("Kali = " + kali);
        System.out.println("Bagi = " + bagi);
        System.out.println("Sisa = " + sisa);
    }
}

```

Amati hasilnya, ubahlah tipe data pada variabel A dan B menjadi float, bandingkan hasilnya

5. Buat program dengan menggunakan operator aritmatika, seperti berikut ini :

```

public class Aritmatika{
    public static void main(String args[]){
        System.out.println("1 + 2 = " + (1 + 2));
        System.out.println("3 * 2 = " + (3 * 2));
        System.out.println("7 / 3 = " + (7 / 3));
        System.out.println("3 - 2 = " + (3 - 2));
        System.out.println("13% 5 = " + (13 % 5));
        System.out.println("");
        System.out.println("1 + 2 * 3 = " + (1 + 2 * 3));
        System.out.println("1 + 2 / 2 = " + (1 + 2 / 2));
        System.out.println("9 - 2 % 2 = " + (9 - 2 % 2));
    }
}

```

6. Buat program berikut, amati hasilnya

```

import java.util.Scanner;
public class Konversi
{
    public static void main(String args[])
    {
        Scanner masuk = new Scanner(System.in);
        float m, cm, inci;
        System.out.print("Masukan ukuran dalam Meter:");
        m = masuk.nextFloat();
        cm = m * 100;
    }
}

```



```

inci = m * 100 / 2.54f;
System.out.println("Ukuran dalam CM = " + cm);
System.out.println("Ukuran dalam Inchi = " + inci);
}
}

```

7. Buat program berikut, amati hasilnya

```

import java.util.Scanner;
public class KonversiSuhu
{
    public static void main(String args[])
    {
        Scanner masuk = new Scanner(System.in);
        int celcius;
        float fahrenheit;
        System.out.print("Masukkan derajat Celcius : ");
        celcius=masuk.nextInt();
        fahrenheit = 9f/5f * celcius + 32f;
        System.out.println("Fahrenheit : "+fahrenheit);
    }
}

```

Tambahkan program diatas untuk menghitung konversi dari celcius ke Reamur dan Kelvin

Reamur =  $4/5 * \text{Celcius}$

Kelvin =  $\text{Celcius} + 273$ .

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS

Buatlah program persamaan linier yang menerima inputan x dan menghasilkan nilai y dengan persamaan  $y = 3x + 4$

## PERTEMUAN KE – 4 METHOD

### A. TUJUAN :

- Memahami tentang method dan dapat menggunakannya.
- Menyelesaikan permasalahan dengan menggunakan method.

### B. TEORI SINGKAT

Method (atau dalam beberapa bahasa pemrograman sering disebut fungsi atau prosedur) adalah sub program yang membiarkan seorang programmer untuk membagi program dengan membagi masalah ke dalam beberapa sub masalah yang bisa diselesaikan secara modular. Dengan cara demikian, maka pembuatan program bisa lebih dimanajemen.

Kelas (*class*) adalah program java yang akan dieksekusi. Method ada di dalam kelas. Java mempunyai kumpulan kelas yang sudah dimiliki yang tersimpan di dalam paket-paket. Kumpulan kelas tersebut ada di dalam *Java Application Interface* (Java API) atau *Java class libraries* dan beberapa *libraries* lainnya.

### FORMAT METHOD SECARA UMUM

```
tipe_return-value nama_method(parameter1, parameter2, ..., parameterN)
{
    deklarasi dan pernyataan;
}
```

Elemen yang diperlukan dari deklarasi method adalah tipe kembalian method, nama, kurung buka dan tutup ( ) dan isi method yang diawali dan diakhiri dengan kurung kurawal buka dan tutup { }. Secara umum, deklarasi method mempunyai 6 komponen, yaitu

1. Modifier - seperti public, private, dan yang lain yang akan kita pelajari kemudian.
2. Tipe kembalian (*return type*)—tipe data dari nilai yang dikembalikan oleh method, atau void jika method tidak mempunyai nilai kembalian.
3. Nama method—aturan untuk penamaan field diterapkan untuk nama method tetapi kesepakatannya adalah sedikit berbeda.
4. Daftar parameter – pemisah antar parameter input adalah koma, diawali oleh tipe datanya, yang diletakkan diantara tkita kurung ( ...daftar parameter.... ). Jika tidak ada parameter, harus menggunakan kurung buka tutup saja ( ).
5. Daftar exception—tidak akan masuk dalam pembahasan di sini
6. Isi method, diletakkan di antara kurung kurawal buka dan tutup { }—kode-kode method, termasuk deklarasi variabel lokal ada di sini.

Method ada yang mempunyai parameter. Ada 2 buah parameter yaitu

- parameter formal adalah parameter yang tertulis dalam definisi method

- Parameter aktual parameter yang berada pada inputan langsung pada saat penggunaan method tersebut.

Parameter bisa lebih dari satu dengan dipisahkan tanda koma,. Yang perlu diperhatikan pada saat pemanggilan method adalah jumlah, urutan dan tipe parameter aktual harus sesuai dengan jumlah urutan dan tipe parameter formal.

#### Pemberian Variabel Dalam Method

Ada dua tipe data variable passing pada method, yaitu *pass-by-value* dan *pass-by-reference*.

##### ***Pass-by-value***

Ketika *pass-by-value* terjadi, method membuat sebuah salinan dari nilai variable yang dikirimkan ke method. Walaupun demikian method tidak dapat secara langsung memodifikasi nilai variable pengirimnya meskipun parameter salinannya sudah dimodifikasi nilainya di dalam method.

##### ***Pass-by-reference***

Ketika sebuah *pass-by-reference* terjadi, alamat memori dari nilai pada sebuah variable dilewatkan pada saat pemanggilan method. Ini tidak seperti pada *pass-by-value*, method dapat memodifikasi variable asli dengan menggunakan alamat memori tersebut, meskipun berbeda nama variable yang digunakan dalam method dengan variable aslinya, kedua variable ini menunjukkan lokasi dari data yang sama.

### C. PRAKTIK

1. Untuk membuat sebuah method, yang pertama kali perlu diperhatikan adalah nama method mempunyai aturan yang sama dengan penamaan variabel.

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

2. Jalankan dan amati hasilnya.  
Method diatas bersifat static dan bertipe void. Method yang bersifat static bisa langsung dipanggil dengan nama methodnya saja. Sebuah method juga bisa dipanggil lebih dari satu kali.
3. Tuliskan program berikut :

```

public class Fungsi2{
public static void kalimat()
{
    System.out.println("Di dalam method kalimat");
}

public static void main(String args[]){
    kalimat();
    System.out.println("Di dalam main");
    kalimat();
}
}

```

Jalankan dan amati hasilnya

4. Tuliskan program berikut :

```

public class Fungsi3{
public static int jumlah(){
    int a = 7, b = 15;
    return (a + b);
}

public static void main(String args[]){
    System.out.println("Hasil pemanggilan method jumlah");
    System.out.println(jumlah());
}
}

```

Jalankan dan amati hasilnya.

Penggunaan sifat static pada method menyebabkan method tidak perlu dipanggil dengan penciptaan obyeknya. Method main dideklarasikan static agar pada saat penggunaan tidak perlu menciptakan obyeknya. Sebagian method tidak dideklarasikan static. Method yang demikian, bisa dipanggil dengan cara menciptakan obyek class dari method tersebut.

```

public class Fungsi3a{
public int jumlah(){
    int a = 7, b = 15;
    return (a + b);
}

public static void main(String args[]){
    Fungsi3a obyek = new Fungsi3a();
    System.out.println("Hasil pemanggilan method jumlah");
    System.out.println(obyek.jumlah());
}
}

```

5. Tuliskan dan jalankan program berikut ini

```

public class FungsiParameter{
    public static int jumlah(int a){
        return a;
    }
}

public static void main(String args[]){
    System.out.println("Hasil pemanggilan metode ");
    System.out.println(jumlah(5));
}

```

```
}
}
```

Parameter pada baris kedua disebut sebagai parameter formal, dan pada baris ke 8 disebut parameter aktual.

6. Tuliskan program berikut :

```
public class Fungsi4a{

public static int jumlah(int a){
    return (a + a);
}

public static void main (String args[]){
    System.out.println("Panggil method jumlah dengan parameter 5");
    System.out.println(jumlah(5));
    System.out.println("Panggil method jumlah dengan parameter 15");
    System.out.println(jumlah(15));
}
}
```

7. Tulis dan jalankan program berikut

```
public class TestPassByValue{
public static void main(String [] args ){
    int i =10;
    System.out.println(i);
    test(i);
    System.out.println(i);
}
public static void test(int j){
    j=33;
}
}
```

Pada program diatas kita memanggil method tes dan melewati nilai variable i sebagai parameter. Nilai pada i disalinkan ke variable j pada method. Pada kondisi ini variable j adalah merupakan variable pengganti pada method test, jika nilai j berubah maka nilai pada variable i yang terletak pada main tidak akan ikut berubah walaupun awalnya variable j merupakan salinan dari variable i.

8. Tulis dan jalankan program berikut

```
class TestPassByReference{
public static void main( String [] args){
//membuat array integer
int [] ages ={10,11,12};

//mencetak nilai array
for (int i=0; i<ages.length; i++){
    System.out.println(ages[i]);
}

test(ages);
for (int i=0; i<ages.length; i++){
    System.out.println(ages[i]);
}
}
```

```
public static void test(int[] arr){
//merubah nilai array
for (int i=0; i<arr.length; i++){
arr[i] = i+50;
}
}
}
```

9. Kerjakan Program berikut :

```
public class Fungsi4c {
    public float jumlah(int a, float b) { //tanpa static
        return (a+b);
    }

    public static void main(String args[]){
        Fungsi4c obyek=new Fungsi4c();
        System.out.print("panggil method jumlah dengan parameter 5
                        dan 1.5, hasilnya = ");
        System.out.println(obyek.jumlah(5, 1.5f));
        System.out.print("panggil method jumlah dengan parameter
                        10 dan 2.2, hasilnya = ");
        System.out.println(obyek.jumlah(10, 2.2f));
    }
}
```

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS

- Akan disampaikan oleh Dosen Pengampu saat praktikum

**PERTEMUAN KE – 5**  
**ALGORITMA DAN IMPLEMENTASI SELEKSI**  
**(SESI 1)**

**A. TUJUAN**

- Memahami tentang algoritma seleksi IF ELSE dan dapat menggunakannya.
- Menyelesaikan masalah dengan algoritma seleksi.

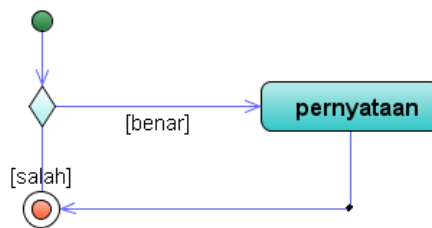
**B. TEORI SINGKAT**

**Pernyataan if**

Pernyataan seleksi dengan IF akan mempunyai beberapa bentuk. Bentuk yang pertama adalah IF dengan satu pilihan. Bentuk umumnya adalah sebagai berikut.

```
if (kondisi)
    Pernyataan1;
```

Activity diagramnya adalah seperti gambar berikut ini.



Gambar 1. Activity Diagram untuk pernyataan IF

Keterangan :

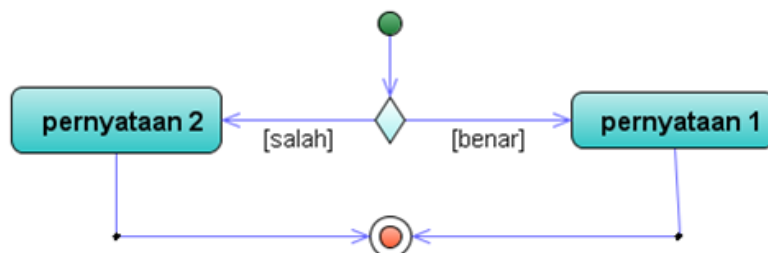
- Kondisi digunakan untuk menentukan pengambilan keputusan. Jika kondisi bernilai benar, maka pernyataan dikerjakan
- Pernyataan, berisi perintah-perintah dan akan dijalankan jika kondisi bernilai benar. Pernyataan disini bisa berupa pernyataan tunggal maupun majemuk.

**Pernyataan if else**

Pernyataan ini akan mempunyai dua pilihan. Jika kondisi bernilai benar, dikerjakan pernyataan 1 dan jika bernilai salah dikerjakan pernyataan 2. Pernyataan 1 dan 2 juga bisa merupakan pernyataan majemuk. Bentuk umumnya adalah sebagai berikut.

```
if (kondisi)
    Pernyataan1;
else
    Pernyataan2;
```

Activity diagramnya adalah seperti gambar berikut ini.



Gambar 2. Activity Diagram untuk pernyataan IF-ELSE

Keterangan :

- Kondisi digunakan untuk menentukan pengambilan keputusan. Jika kondisi bernilai benar, maka pernyataan1 dikerjakan, jika salah, maka Pernyataan2 yang akan dikerjakan.

### Pernyataan if elseif

Untuk membuat seleksi persyaratan yang lebih kompleks. Bentuk umumnya adalah sebagai berikut :

```
if(kondisi)
    Pernyataan1;
else if(kondisi2)
    Pernyataan2;
...
...
else
    Pernyataan3;
```

## C. PRAKTIK

1. Dengan menggunakan TextPad ketikkan program program – program berikut

```
import java.util.Scanner;
public class IfSatuPilihan{
public static void main(String args[]){
    Scanner masuk = new Scanner(System.in);
    int bil;
    System.out.print("Masukkan bilangan : ");
    bil=masuk.nextInt();
    if (bil==0)
        System.out.println("Bilangan Nol");
    }
}
```

2. masukkan angka 7 dan apa hasilnya?
3. Modifikasi program diatas sehingga menjadi seperti berikut :

```
import java.util.Scanner;
public class IfDuaPilihan{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        System.out.print("Masukkan bilangan : ");
        bil=masuk.nextInt();
        if (bil==0)
            System.out.println("Bilangan Nol");
        else
            System.out.println("Bilangan Bukan Nol");
    }
}
```

4. masukkan angka 7 dan jelaskan hasilnya!
5. Tuliskan program berikut dan amati hasilnya

```
import java.util.Scanner;
public class ifbool {
    public static void main(String arg[])
    {
        Scanner input = new Scanner(System.in);
        int a,b;
        boolean benar;
        System.out.print("Masukkan nilai a : ");
```



```

        a = input.nextInt();
        System.out.print("Masukkan nilai b : ");
        b = input.nextInt();
        benar=a>b;
        if(benar==true)
        {
            System.out.println(a+" lebih besar
dari "+b);
        }
        else
        {
            System.out.println(a+" lebih kecil
dari "+b);
        }
    }
}

```

6. Tuliskan program berikut

```

import java.util.Scanner;
public class IfJurusan{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int pil;
        System.out.print("Masukkan pilihan : ");
        pil = masuk.nextInt();
        if (pil==1)
            System.out.println("Jurusan MI/D3");
        else if(pil==2)
            System.out.println("Jurusan TK/D3");
        else if(pil==3)
            System.out.println("Jurusan KA/D3");
        else if(pil==4)
            System.out.println("Jurusan TI/S1");
        else if(pil==5)
            System.out.println("Jurusan SI/S1");
        else
            System.out.println("Pilihan Salah!!!");
    }
}

```

6. Buat program seperti di bawah

```

import java.util.Scanner;
public class Grade
{
    public static void main(String args[])
    {
        Scanner masuk = new Scanner(System.in);
        int bil;
        System.out.print("Masukkan bilangan : ");
        bil=masuk.nextInt();
        if (bil%2==0)
            System.out.println("Bilangan Genap");
        else if (bil%2==1)
            System.out.println("Bilangan Ganjil");
        else
            System.out.print("Salah masukkan bilangan");
    }
}

```

Masukkan bilangan 111, amati hasilnya. Ulangi dengan memasukkan angka 22 amati hasilnya.

7. Buat program seperti di bawah

```
import java.util.Scanner;
public class Grade {
    public static void main( String[] args )
    {
        Scanner masuk = new Scanner(System.in);
        int grade;
        System.out.print("Masukkan grade : ");
        grade = masuk.nextInt();
        if( grade >= 90 )
        {
            System.out.println( "Excellent!" );
        }
        else if( (grade < 90) && (grade >= 80))
        {
            System.out.println("Bagus!" );
        }
        else if( (grade < 80) && (grade >= 60))
        {
            System.out.println("Belajar lagi!" );
        }
        else
        {
            System.out.println("Maaf, Anda gagal.");
        }
    }
}
```

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

**PERTEMUAN KE – 6**  
**ALGORITMA DAN IMPLEMENTASI SELEKSI**  
**(SESI 2)**

**A. TUJUAN :**

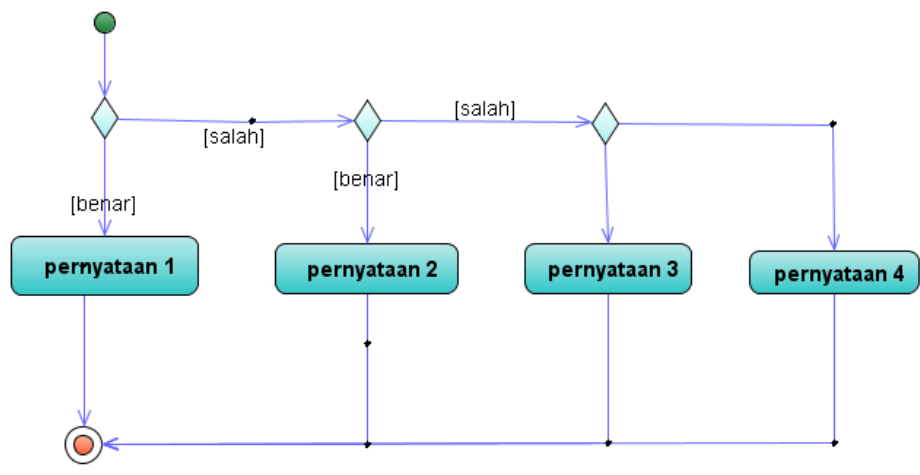
- Memahami tentang algoritma seleksi SWITCH dan dapat menggunakannya.
- Menyelesaikan masalah dengan algoritma seleksi

**B. TEORI SINGKAT**

Pernyataan switch adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. Setiap ungkapan diungkapkan dengan sebuah nilai integral konstan, seperti sebuah nilai dengan tipe byte, short, int atau char.

Bentuknya :

```
switch (ungkapan)
{
    case ungkapan1:
        pernyataan1;
        break;
    case ungkapan2:
        pernyataan2;
        break;
    .....
    default:
        pernyataan_x;
}
```



Keterangan :

- ungkapan1, ungkapan2 dan seterusnya dilakukan secara berurutan dimulai dari yang pertama, sekiranya cocok pernyataan yang mengikuti **case** dijalankan.
- **break** ditemukan dari eksekusi pernyataan **switch** berakhir
- **default** hanya akan dijalankan jika ungkapan pada bagian case tidak ada yang cocok.

### C. PRAKTIK

1. Buatlah program berikut ini :

```
import java.util.Scanner;
public class CaseJurusan{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int pil;
        System.out.print("Masukkan pilihan : ");
        pil = masuk.nextInt();
        switch (pil) {
            case 1: System.out.println("Jurusan MI/D3");break;
            case 2: System.out.println("Jurusan TK/D3");break;
            case 3: System.out.println("Jurusan KA/D3");break;
            case 4: System.out.println("Jurusan TI/S1");break;
            case 5: System.out.println("Jurusan SI/S1");break;
            default: System.out.println("Pilihan Salah!!!");
                    break;
        }
    }
}
```

2. Uji program di atas dengan memasukkan angka 4 dan 6
3. Buat program dengan menggunakan pernyataan switch, seperti berikut ini :

```
import java.util.Scanner;
public class CaseTV{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int pil;
        System.out.print("Masukkan pilihan : ");
        pil = masuk.nextInt();
        switch (pil) {
            case 1: System.out.println("RCTI");break;
            case 2: System.out.println("SCTV");break;
            case 3: System.out.println("TPI");break;
            case 4: System.out.println("INDOSIAR");break;
            case 5: System.out.println("TRANS7");break;
            case 6: System.out.println("TVONE");break;
            case 7: System.out.println("METRO");break;
            case 8: System.out.println("GLOBAL");break;
            case 9: System.out.println("TRANS7");break;
            case 10: System.out.println("TVRI");break;
            default: System.out.println("Salah masukkan pilihan");
                    break;
        }
    }
}
```

4. Uji program di atas dengan memasukkan beberapa kemungkinan.
5. Hilangkan break yang ada pada case 1, jalankan dengan memasukkan pilihan 1, amati hasilnya
6. Buat program seperti dibawah, masukkan berbagai kemungkinan kode\_bulan

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

```

int kode_bulan;
System.out.print("Masukkan kode bulan : ");
kode_bulan = masuk.nextInt();
switch (kode_bulan) {
case 1:case 3:case 5: case 7: case 8:case 10:case 12:
    System.out.println("Jumlah hari = 31 hari");
    break;
case 4:case 6: case 9: case 11:
    System.out.println("Jumlah hari = 30 hari");
    break;
case 2:
    System.out.println("Jumlah hari = 28 atau 29 hari");
    break;
default:
    System.out.println("Salah masukan kode bulan");
    break;
}
}
}

```

7. Buat program seperti dibawah, masukkan berbagai kemungkinan kode\_bulan

```

import java.util.Scanner;
public class CaseHari{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int nohari;
        System.out.print("Masukkan nomor hari (1...7) : ");
        nohari=masuk.nextInt();
        switch(nohari)
        {
        case 1:
            System.out.println("Hari ke-"+nohari+" : adalah Minggu");
            break;
        case 2:
            System.out.println("Hari ke-"+nohari+" : adalah Senin");
            break;
        case 3:
            System.out.println("Hari ke-"+nohari+" : adalah Selasa");
            break;
        case 4:
            System.out.println("Hari ke-"+nohari+" : adalah Rabu");
            break;
        case 5:
            System.out.println("Hari ke-"+nohari+" : adalah Kamis");
            break;
        case 6:
            System.out.println("Hari ke-"+nohari+" : adalah Jumat");
            break;
        case 7:
            System.out.println("Hari ke-"+nohari+" : adalah Sabtu");
            break;
        default:
            System.out.println("Tidak terdapat nama hari ke-"+nohari);
        }
    }
}
}

```

#### D. LATIHAN

Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS :

1. Ubahlah praktik diatas menjadi bentuk if
2. Setelah anda melakukan praktikum dengan menggunakan pernyataan if dan switch jelaskan kekurangan dan kelebihan dari kedua pernyataan tersebut

**PERTEMUAN KE – 7**  
**ALGORITMA DAN IMPLEMENTASI PERULANGAN**  
**(SESI 1)**

**A. TUJUAN :**

- Memahami tentang perulangan dengan WHILE dapat menggunakannya.
- Menyelesaikan permasalahan dengan menggunakan WHILE.

**B. TEORI SINGKAT**

Pernyataan ini berguna untuk memproses suatu pernyataan atau beberapa pernyataan beberapa kali. Selama ungkapan bernilai benar, pernyataan akan selalu dikerjakan.

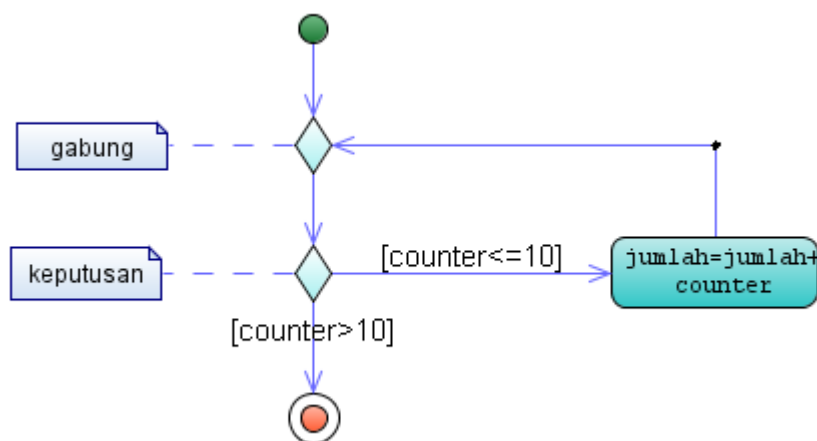
Bentuknya :

```
Ungkapan1;  
while (Ungkapan2)  
{  
    Pernyataan;  
    Ungkapan 3;  
}
```

Keterangan :

- bagian pernyataan akan dieksekusi selama ungkapan2 dalam **while** bernilai benar.
- Pengujian terhadap ungkapan2 pada **while** dilakukan sebelum bagian pernyataan.
- Kemungkinan pernyataan pada **while** tidak dijalankan sama sekali, jika ketemu kondisi yang pertama kali bernilai salah.

Activity diagramnya adalah seperti gambar berikut :



Catatan :

Pernyataan perulangan dengan while akan selalu dikerjakan jika ungkapan selalu benar. Oleh karena itu, kita harus membuat kondisi suatu saat ungkapan bernilai salah agar perulangan berakhir.

**C. PRAKTIK**

1. Buatlah program seperti berikut ini :

```

import java.util.Scanner;
public class Ulang{
    public static void main(String arg[])
    {
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
        System.out.println("STMIK AKAKOM Yogyakarta ");
    }
}

```

2. Buatlah program seperti berikut ini :

```

import java.util.Scanner;
public class Ulangwhile2{
    public static void main(String arg[])
    {
        for(int u=1;u<=10;u++)
        {
            System.out.println("STMIK AKAKOM Yogyakarta ");
        }
    }
}

```

Amati hasilnya, bandingkan dengan output praktik nomor 1

3. Buatlah program seperti berikut ini :

```

import java.util.Scanner;
public class UlangWhile1{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        bil=1;
        while (bil<=5) {
            System.out.println(bil);
            bil++;
        }
    }
}

```

4. Ubah pernyataan **bil=1** menjadi **bil=5**, pernyataan while (**bil<=5**) dengan while(**bil>=1**) dan **bil++** menjadi **bil--**, amati hasil outputnya.

5. Dengan while, buatlah program untuk mencetak bilangan genap dari 0 sampai dengan 10.

```

import java.util.Scanner;
public class UlangWhile3{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        bil=2;
        while (bil<=10) {
            System.out.println(bil);
            bil+=2;
        }
    }
}

```



```
    }  
  }  
}
```

6. Ubah program di atas untuk menampilkan bilangan ganjil saja yaitu 1, 3, 5, 7, 9
7. Buat program seperti di bawah

```
import java.util.Scanner;  
public class UlangWhile4{  
  public static void main(String args[]){  
    char A ='a';  
    int B = 0;  
    int C=1;  
    while (A<='e')  
    {  
      System.out.println("Nilai A="+A);  
      System.out.println("Nilai B="+B);  
      System.out.println("Nilai C="+C);  
      System.out.println(" ");  
      A++;  
      B=B+5;  
      C=C*10;  
    }  
  }  
}
```

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

**PERTEMUAN KE – 8**  
**ALGORITMA DAN IMPLEMENTASI PERULANGAN**  
**(SESI 2)**

A. TUJUAN :

- Memahami tentang perulangan dengan DO WHILE dapat menggunakannya.
- Menyelesaikan permasalahan dengan menggunakan DO WHILE

B. TEORI SINGKAT

Seperti halnya perulangan dengan while, perulangan dengan do ... while ini juga digunakan untuk mengerjakan sebuah atau sekelompok pernyataan berulang-ulang. Bedanya dengan while adalah pernyataan do ... while akan mengecek kondisi di belakang, sementara while cek kondisi ada di depan.

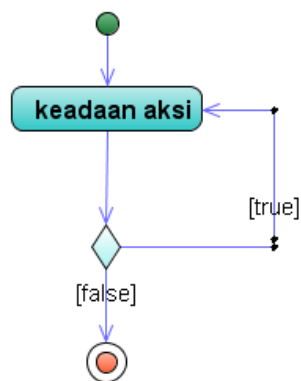
Bentuknya :

```
Ungkapan 1;  
do  
{  
    Pernyataan;  
    Ungkapan 3;  
}  
while(Ungkapan2);
```

Keterangan :

- Bagian pernyataan dijalankan secara berulang sampai ungkapan2 bernilai salah.
- Pengujian ungkapan2 dilakukan setelah bagian pernyataan, maka pada pernyataan **do ... while** minimal akan dijalankan sekali, karena begitu masuk ke blok perulangan, tidak ada cek kondisi tetapi langsung mengerjakan pernyataan.

Activity diagramnya :



C. PRAKTIK

1. Tuliskan program berikut ini :

```
import java.util.Scanner;  
public class UlangDo1{  
    public static void main(String args[]){  
        Scanner masuk = new Scanner(System.in);
```

```

        int bil;
        bil=1;
        do
        {
        System.out.println(bil);
        bil+=2;
        }
        while (bil<=10);
    }
}

```

- Ubah program di atas agar mendapat hasil output yang bilangan genap saja (0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20)
- Buatlah program mencetak konversi suhu dari celcius ke fahrenheit mulai dari 1 sampai 10 dengan membuat tabel.

```

public class UlangDo2 {
    public static void main(String args[]){
        int c;
        double f;
        System.out.println("-----");
        System.out.println("CELCIUS          FAHREINHEIT");
        System.out.println("-----");
        c=1;
        do {
            f=1.8 * c + 32;
            System.out.println("Celcius:"+c+"Fahrenheit: "+f);
            c++;
        } while (c<=10);
        System.out.println("-----");
    }
}

```

- Buatlah program seperti di bawah

```

import java.util.Scanner;
public class cobacoba{
    public static void main(String args[]){
        char A ='a';
        int B = 0;
        int C=1;
        do
        {
        System.out.println("Nilai A="+A);
        System.out.println("Nilai B="+B);
        System.out.println("Nilai C="+C);
        System.out.println(" ");
        A++;
        B=B+5;
        C=C*10;
        }while (A<='e');
    }
}

```

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum



**PERTEMUAN KE – 9**  
**ALGORITMA DAN IMPLEMENTASI PERULANGAN**  
**(SESI 3)**

A. TUJUAN :

- Memahami tentang perulangan dengan FOR dapat menggunakannya.
- Menyelesaikan permasalahan dengan menggunakan FOR.

B. TEORI SINGKAT

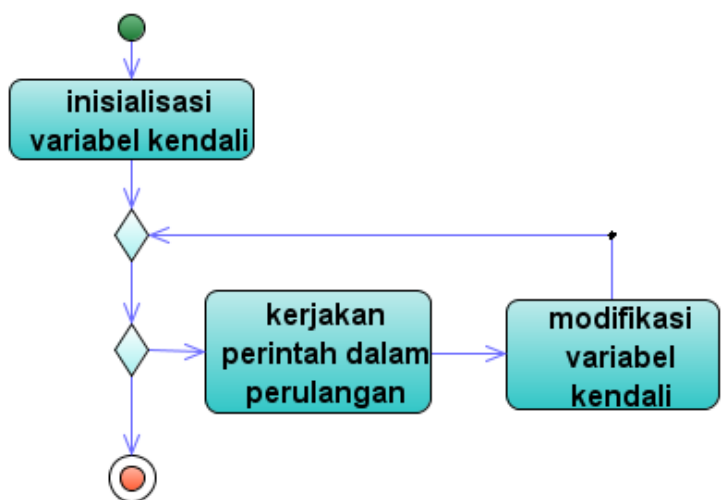
Sama seperti pernyataan perulangan while dan do...while, pernyataan for juga digunakan untuk mengerjakan pernyataan atau sekelompok pernyataan secara berulang. Bedanya adalah dengan pernyataan for perulangan akan dikerjakan dalam hitungan yang sudah pasti, sementara while dan do...while tidak.

Bentuknya :

```
for (ungkapan1;ungkapan2;ungkapan3)
    Pernyataan;
```

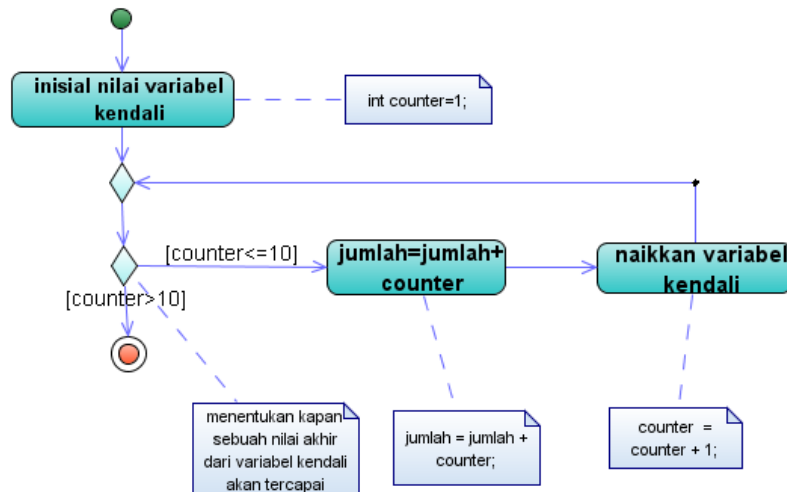
Keterangan :

- ungkapan1 merupakan pernyataan inisialisasi
- ungkapan2 sebagai kondisi yang menentukan pengulangan terhadap pernyataan atau tidak
- ungkapan3 digunakan sebagai pengatur variabel yang digunakan didalam ungkapan1



Gambar 4.15. Activity Diagram untuk perulangan dengan FOR

Contoh activity diagram untuk perulangan dengan for.



Gambar 9.1. Activity Diagram untuk perulangan dengan FOR

### C. PRAKTIK

1. Ketikkan program program – program berikut

```

public class UlangFor {
    public static void main (String args[]){
        int bil;
        for (bil=1;bil<=5;bil++)
            System.out.println(bil);
    }
}
  
```

2. Ketikkan program program – program berikut

```

public class UlangFor {
    public static void main (String args[]){
        int bil;
        for (bil=5;bil>=1;bil--)
            System.out.println(bil);
    }
}
  
```

3. Buatlah program untuk menampilkan bilangan kelipatan 5 antara 125 sampai dengan 200

4. Buat program seperti di bawah

```

import java.util.Scanner;
public class UlangFor {
public static void main (String [] args) {
    int maks,min,nilai;
    int maksiawal = 1;
    System.out.println("Program deret bilangan ganjil FOR Java");
    Scanner input = new Scanner(System.in);
    System.out.print("Masukkan angka minimal : ");
    min = input.nextInt();
    System.out.print("Masukkan angka maksimal : ");
    maks = input.nextInt();
    for(nilai=min; nilai<maks; nilai+=4)
        System.out.println(" "+nilai);
    }
}
  
```

5. Program untuk menghitung jumlah dan rata – rata

```
import java.util.Scanner;
public class UlangFor6{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int i;
        float n, jum, x, rata;
        System.out.print("Banyaknya data : ");
        n = masuk.nextFloat();
        jum=0;
        for (i=1;i<=n;i++){
            System.out.print("Data ke-"+i+" : ");
            x=masuk.nextFloat();
            jum += x;
        }
        System.out.println("Jumlah : "+jum);
    }
}
```

6. Modifikasi program diatas untuk menghitung rata – rata dari sejumlah data yang dimasukkan!

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

**PERTEMUAN KE – 10**  
**ALGORITMA DAN IMPLEMENTASI PERULANGAN**  
**(SESI 4)**

**A. TUJUAN**

- Mengubah dari bentuk perulangan satu ke yang lain.
- Menyelesaikan permasalahan dengan menggunakan WHILE, DO WHILE dan FOR.

**B. TEORI SINGKAT**

Perulangan mempunyai 3 bentuk. Kasus perulangan dapat diubah dari bentuk satu ke bentuk yang lainnya. Bentuk umum dari masing-masing perulangan

<b>While</b>	<b>Do-While</b>	<b>For</b>
<pre>Ungkapan1; while (Ungkapan2) { Pernyataan; Ungkapan 3; }</pre>	<pre>Ungkapan 1; do { Pernyataan; Ungkapan 3; } while (Ungkapan2);</pre>	<pre>for (Ungkapan1;Ungkapan2;Ungkapan3) { Pernyataan; }</pre>

**C. PRAKTIK**

1. Tuliskan program berikut

**Bentuk while**

```
import java.util.Scanner;
public class Ulang{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        bil=1;
        while (bil<=20) {
            System.out.println(bil);
            bil+=2;
        }
    }
}
```

**Bentuk Do-While**

```
import java.util.Scanner;
public class Ulang{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        bil=1;
        do
        {
            System.out.println(bil);
            bil+=2;
        }
        while (bil<=20) ;
    }
}
```



```
}
```

### Bentuk For

```
import java.util.Scanner;
public class Ulang{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        for(bil=1;bil<=20;bil+=2)
        {
            System.out.println(bil);
        }
    }
}
```

- Ubah kode bil=1 menjadi bil=22, jalankan, amati perbedaan hasilnya antara ketiga bentuk perulangan tersebut
- Buat program untuk menampilkan deret berikut dengan menggunakan WHILE, FOR, dan DO-WHILE

1 1 2 3 5 8 13..

- Buatlah program seperti di bawah

```
public class forbersarang {
    public static void main(String arg[])
    {
        int b,k;
        for(b=1;b<=3;b=b+1)
        {
            for(k=1;k<=3;k=k+1)
            {
                System.out.print(b+" ");
            }
            System.out.println("");
        }
    }
}
```

- Buatlah program seperti di bawah

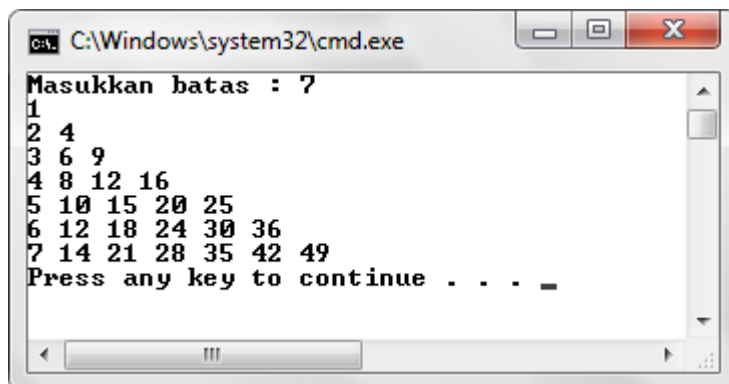
```
public class bintang {
    public static void main (String [] args) {
        int azim, abidin;
        for (azim=1; azim<=4; azim++) {
            for (abidin=1; abidin<=azim; abidin++)
                System.out.print(" * " );
            System.out.println();
        }
        for (azim=5; azim>0; azim--) {
            for (abidin=0; abidin<azim; abidin++)
                System.out.print(" * " );
            System.out.println();
        }
    }
}
```

### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Buatlah program untuk menampilkan hasil seperti di bawah, batas dimasukkan melalui keyboard



```
C:\Windows\system32\cmd.exe
Masukkan batas : 7
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
Press any key to continue . . . _
```

## PERTEMUAN KE – 11 GABUNGAN SELEKSI, ITERASI DAN METHOD

### A. TUJUAN

- Memahami tentang perintah-perintah yang akan digunakan yaitu seleksi, iterasi dan method
- Menyelesaikan permasalahan dengan menggunakan beberapa gabungan perintah tersebut

### B. TEORI SINGKAT

Iterasi adalah

- Dapat digunakan pada struktur kontrol pengulangan
- Proses diakhiri ketika kondisi loop gagal dilaksanakan
- Lebih cepat

Rekursif

- Memanggil suatu method secara berulang-ulang
- Menggambarkan sebuah masalah kedalam lingkup yang lebih kecil dari pengulangan itu sendiri
- proses yang berakhir dengan kondisi tertentu disebut
- permasalahan dasar yang telah tercukupi oleh suatu
- pembatasan kondisi
- Mendorong proses pelatihan pemrograman yang baik

### C. PRAKTIK

#### 1. Tuliskan program berikut, jalankan

```
class FactorialIter {
    static int factorial(int n) {
        int result = 1;
        for (int i = n; i > 1; i--) {
            result *= i;
        }
        return result;
    }
    public static void main(String args[]){
        System.out.println(factorial(3));
    }
}
```

#### 2. Tuliskan program berikut, jalankan

```
class FactorialRecur {
    static int factorial(int n) {
        if (n == 1) {
            return 1;
        }
        return factorial(n-1)*n;
    }
    public static void main(String args[]) {
        System.out.println(factorial(3));
    }
}
```

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS

- Akan disampaikan oleh Dosen Pengampu saat praktikum

## PERTEMUAN KE – 12 ARRAY SATU DIMENSI

### A. TUJUAN :

- Memahami tentang array (larik) satu dimensi dan dapat menggunakannya.
- Menyelesaikan permasalahan dengan menggunakan Array.

### B. TEORI SINGKAT

Larik adalah sebuah struktur data yang terdiri dari data yang bertipe sama. Ukuran larik bersifat tetap, larik akan mempunyai ukuran yang sama pada saat sekali dibuat. Larik dalam Java adalah obyek, disebut juga sebagai tipe referensi. Sedangkan elemen dalam larik Java bisa primitif atau referensi. Posisi dari larik biasa disebut sebagai elemen. Elemen larik dimulai dari 0 (nol). Penyebutan larik diberikan dengan cara menyebutkan nama lariknya dan diikuti dengan indeksinya, dimana indeks dituliskan diantara tanda kurung siku. Gambar 1. memperlihatkan gambaran larik dengan 10 elemen, dimana setiap elemennya bertipe integer, dengan nama A.

Nama	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
Isi larik	12	-56	23	45	-16	-2	85	41	15	20

### DEKLARASI DAN MENCIPTAKAN LARIK

Sebagai sebuah obyek, larik harus diciptakan dengan menggunakan kata cadang `new`. Deklarasi dan penciptaan variabel larik gambar 1 adalah sebagai berikut.

```
int A[] = new int[10];
```

larik dideklarasikan dan langsung diciptakan . Atau

```
int A[];
```

```
A = new int[10];
```

larik dideklarasikan, baru pada pernyataan berikutnya larik diciptakan.

### C. PRAKTIK

#### 1. Tuliskan program berikut

```
import java.util.Scanner;
public class Larik1{
    public static void main (String args[]) {
        Scanner masuk=new Scanner(System.in);
        float nilai[]=new float[5];
        System.out.println("masukkan 5 buah data nilai");
        for(int i=0;i<5;i++)
        {
            System.out.print("Data ke"+(i+1)+"": ");
            nilai[i]=masuk.nextFloat();
        }
        System.out.println("data nilai yang dimasukkan");
        for(int i=0;i<5;i++)
            System.out.println(nilai[i]);
    }
}
```

```
}  
}
```

2. Tuliskan program untuk menampilkan bilangan dari 1 sampai 10 dengan pangkatnya masing – masing berikut:

```
public class Larik4{  
public static void main(String args[]){  
  
int kuadrat[];  
kuadrat = new int[10];  
for (int i=0;i<10;i++){  
    kuadrat[i]=(i+1)*(i+1);  
    System.out.println("Kuadrat "+(i+1)+" = "+kuadrat[i]);  
}  
}  
}
```

3. Larik juga dapat diberikan nilai awal (diinisialisasi) pada saat pendefinisian. Tuliskan program berikut

```
public class Larik6{  
public static void main(String args[]){  
    int hari []= {31, 28, 31, 30, 31, 30, 30, 31, 30, 31, 30, 31};  
  
    for(int i = 0; i < 12; i++)  
        System.out.println("Bulan " + (i+1) + " = "+hari[i]);  
}  
}
```

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS

- Akan disampaikan oleh Dosen Pengampu saat praktikum

## PERTEMUAN KE – 13

### Array Dua Dimensi

#### A. TUJUAN

- Memahami tentang array (larik) dua dimensi dan dapat menggunakannya
- Menyelesaikan permasalahan dengan menggunakan Array

#### B. TEORI SINGKAT

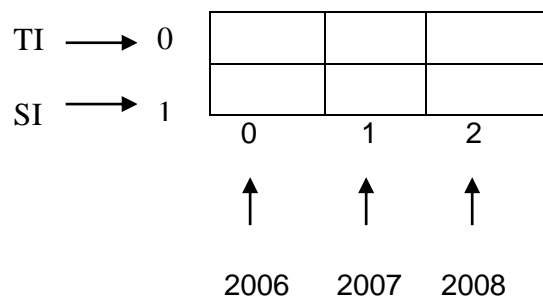
Kita juga bisa membuat variabel larik yang tipe elemennya adalah larik. Dengan cara demikian, kita membuat larik dua dimensi. Dengan larik dua dimensi, maka kita mempunyai elemen yang berindeks tidak hanya satu, tetapi dua. Kita bisa membayangkan larik dua dimensi tersebut seperti sebuah tabel yang berisi baris dan kolom. Penyebutan sel tabel selalu diikuti dengan penyebutan baris berapa dan kolom berapa.

Contoh :

Diberikan data kelulusan mahasiswa sebuah perguruan tinggi sebagai berikut.

Jurusan	2006	2007	2008
Teknik Informatika	110	125	135
Sistem Informasi	56	75	80

```
int data_lulus [2] [3]
```



[0] [0]	[0] [1]	[0] [2]
[1] [0]	[1] [1]	[1] [2]

#### C. PRAKTIK

1. Buatlah program seperti berikut

```
public class ArrayDimensiDua{
    public static void main(String [] args){
        int [][] piksel = new int[2][3];
        // mengisi elemen tertentu
        piksel[0][0] = 70;
        piksel[0][1] = 18;
        piksel[0][2] = 45;
        piksel[1][0] = 75;
        piksel[1][1] = 66;
        piksel[1][2] = 89;
        //menampilkan elemen array
    }
}
```

```

int i,j;
for(i=0;i<2;i++){
    for (j=0; j<3;j++)
        System.out.print(piksel[i][j] + " ");
        System.out.println("");
    }
}
}

```

2. Tuliskan program untuk menampilkan menampilkan matrik yang elemen – elemennya dimasukkan melalui keyboard

```

import java.util.Scanner;
public class ArrayDimensiDua2{
public static void main(String args[]){
    Scanner masuk = new Scanner(System.in);
    int nilai[][]= new int[2][3];
    System.out.println("Masukkan data nilai");
    for (int i = 0; i < 2; i++){
        for (int j = 0; j < 3; j++){
            System.out.print("(" + (i + 1) + " , " + (j+1) + ")"+ " : ");
            nilai[i][j]=masuk.nextInt();
        }
    }

    System.out.println("Data nilai yang dimasukan");
    for (int i = 0; i < 2; i++){
        for (int j = 0; j < 3; j++)
            System.out.print(nilai[i][j]+" ");
        System.out.println();
    }
}
}

```

3. Tuliskan program untuk menjumlahkan matrik berikut ini.

```

import java.util.Scanner;
public class JumlahMatriks{
    Scanner masuk = new Scanner(System.in);
    public void masukData(float data[][]){
        for (int i = 0; i < 3; i++){
            for (int j = 0; j < 3; j++){
                System.out.print( "("+(i + 1) + " , " + (j+1)+")" + " : ");
                data[i][j]=masuk.nextFloat();
            }
        }
    }

    public float[][] tambah(float AA[[[[],
float BB[] []]){
        float hasil[[[[]]= new float[3][3];
        for (int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++)
                hasil[i][j] = AA[i][j] + BB[i][j];
        return hasil;
    }

    public void tampilData(float data[[[[], char nama){
    for (int i=0;i<3;i++){
        for (int j=0;j<3;j++)
            System.out.print(nama+"[" + (i + 1)
            +"] [" + (j+1) + "] = " + data[i][j]+" ");
    }
}
}

```



```

        System.out.println();
    }
}

public static void main(String args[]){
    float A[][] = new float[3][3];
    float B[][] = new float[3][3];
    float C[][] = new float[3][3];
    JumlahMatriks jumlah = new JumlahMatriks();
    System.out.println("Masukkan data matriks A");
    jumlah.masukData(A);System.out.println("Masukkan data matriks
B");
    jumlah.masukData(B);
    C = jumlah.tambah(A,B);
    jumlah.tampilData(C, 'C');
}
}

```

#### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

#### E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

## PERTEMUAN KE – 14 STUDI KASUS

### A. TUJUAN

- Memahami tentang keseluruhan materi yang sudah diberikan
- Menyelesaikan permasalahan dengan menggabungkan

### B. TEORI SINGKAT

Dalam banyak kasus nyata, terkadang permasalahan-permasalahan tidak bisa diselesaikan hanya dengan menggunakan satu bentuk perintah saja, misalnya hanya dengan konsep seleksi saja, perulangan saja, array saja tetapi digabungkan dengan bentuk yang lainnya.

### C. PRAKTIK

- Studi kasus akan disampaikan oleh Dosen Pengampu saat praktikum

### D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

### E. TUGAS

- Akan disampaikan oleh Dosen Pengampu saat praktikum