

PANDUAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN - TK



Disusun Oleh :

LN Harnaningrum

STMIK AKAKOM
Yogyakarta
2014

KATA PENGANTAR

Puji Syukur kepada Tuhan 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	3
Pertemuan ke 3	7
Pertemuan ke 4	15
Pertemuan ke 5	20
Pertemuan ke 6	24
Pertemuan ke 7	26
Pertemuan ke 8	30
Pertemuan ke 9	33
Pertemuan ke 10	35
Pertemuan ke 11	36
Pertemuan ke 12	38
Pertemuan ke 13	41
Pertemuan ke 14	43

PERTEMUAN KE – 1 PENGANTAR ALGORITMA

A. TUJUAN :

- Memahami tentang algoritma
- Menyelesaikan masalah algoritma

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.

C. PRAKTIK :

1. Buat algoritma untuk berangkat kuliah dengan menggunakan olah kata 9MS-Word)
2. Buat algoritma untuk pergi naik pesawat.
3. Siapkan software Netbeans, Textpad, Java.

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

PERTEMUAN KE – 2

Diagram aktifitas dan dasar pemrograman

A. TUJUAN :

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

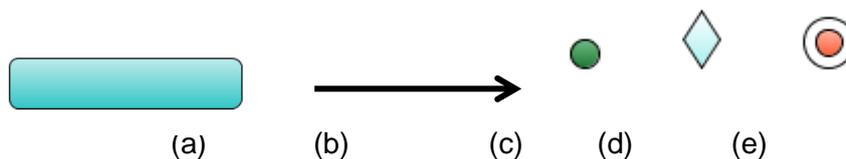
B. TEORI SINGKAT

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.

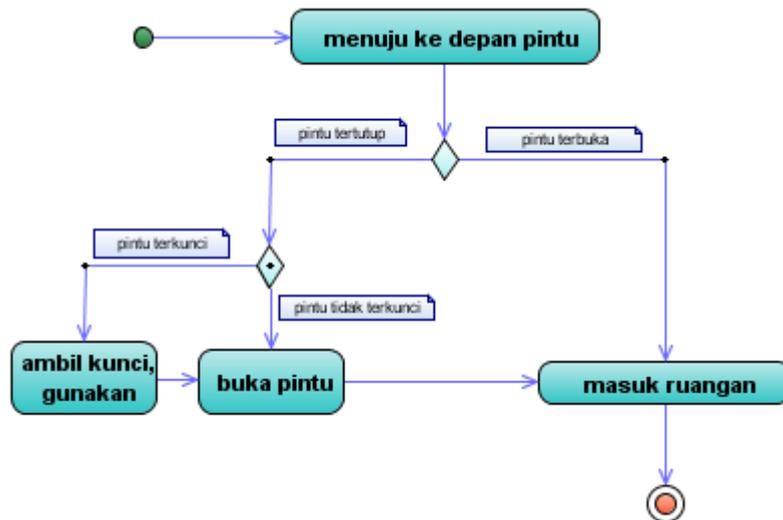


Gambar 1. Simbol *Activity Diagram*

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

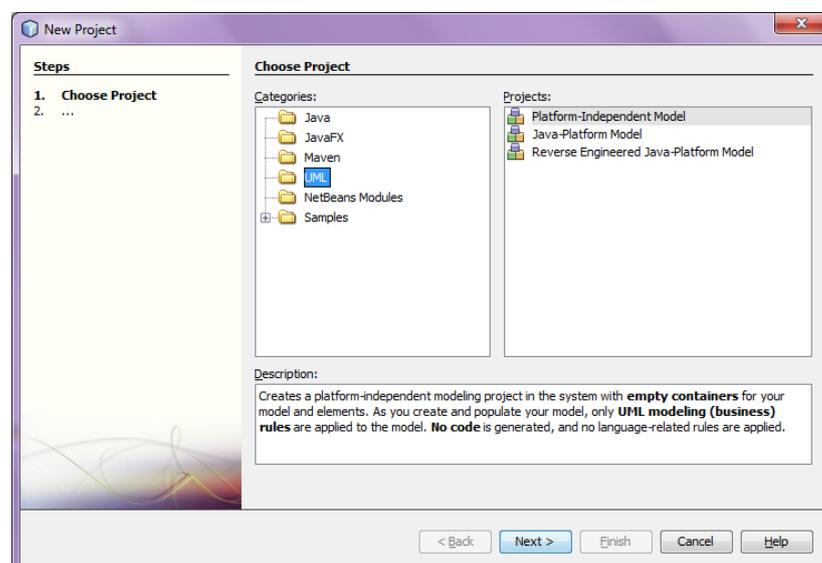
kali yang dikerjakan adalah menuju ke pintu. Kemudian melihat apakah pintu dalam keadaan terbuka atau tertutup. Jika dalam keadaan terbuka, dia langsung masuk. Jika pintu dalam keadaan tertutup, maka orang tersebut akan mengecek apakah pintu dalam keadaan terkunci atau tidak. Kalau terkunci, maka orang 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.



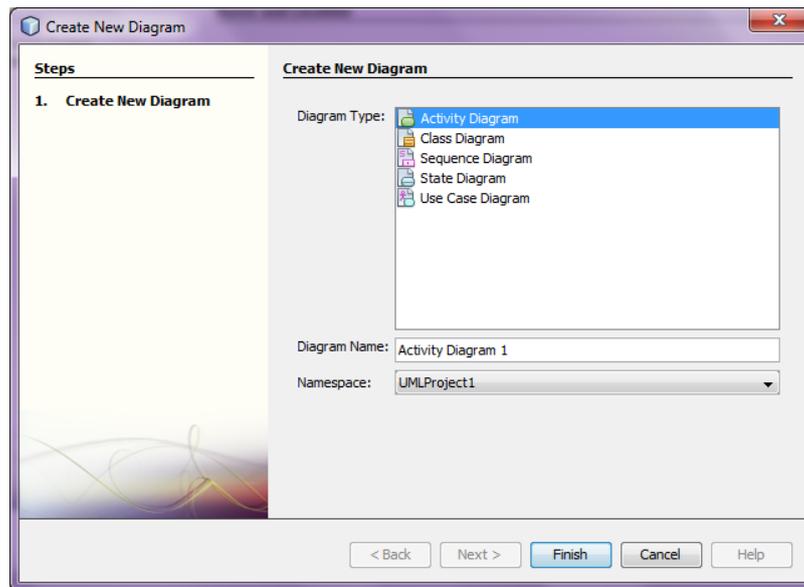
Gambar 2. *Activity diagram* untuk masuk ruangan

C. PRAKTIK :

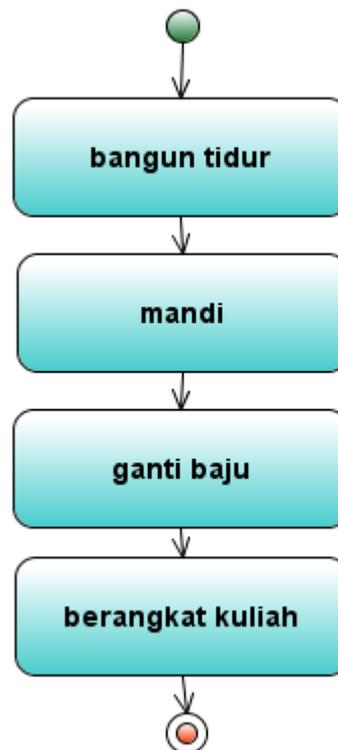
1. Buka Netbeans.
2. Pilih New Project
3. Pilih UML, Platform-Independent Model



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



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



7. Buat juga activity diagram untuk berangkat kuliah.

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 jangkauan 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

- **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.
- **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 menyimpam 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 *presedence* yang lebih tinggi. Operator di dalam tabel 2.1. dituliskan sesuai dengan *presedence* ordernya. Semakin ke bawah, maka *presedence*-nya lebih rendah. Operator dengan *presedence* yang lebih tinggi dikerjakan lebih dulu dari pada operator dengan *presedence* yang lebih rendah. Operator yang ada di dalam baris yang sama mempunyai *presedence* yang sama. Pada saat operator dari *presedence* 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>Ternary</i>	? :
Pemberian	= += -= *= /= %= &= ^= = <<= >>= >>>=

Operator Aritmatika

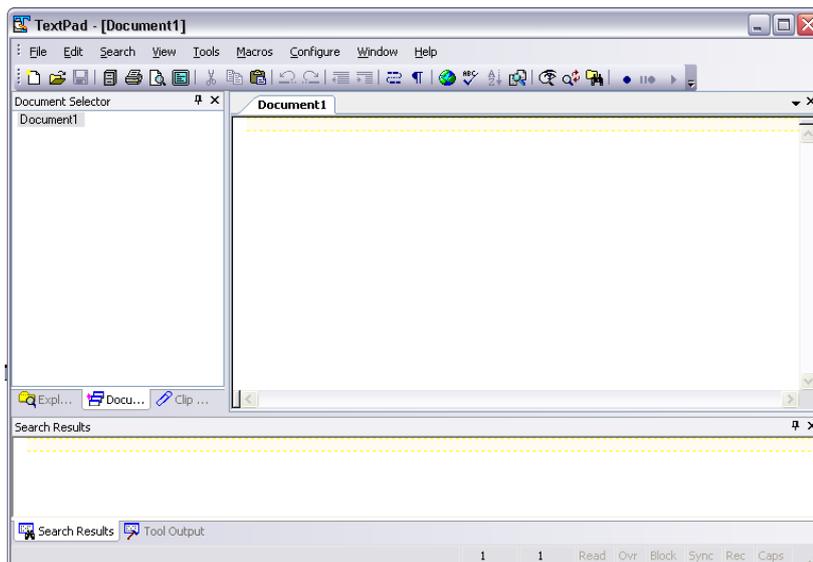
Operator	Keterangan
+	Penjumlahan (tanda plus)
-	Pengurangan (tanda minus)
*	Perkalian
/	Pembagian
%	Sisa Pembagian

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

C. PRAKTIK

Kita akan menuliskan program dengan menggunakan TextPad. TextPad adalah salah satu editor yang dapat digunakan untuk pemrograman Java. Langkah-langkah menggunakan TextPad :

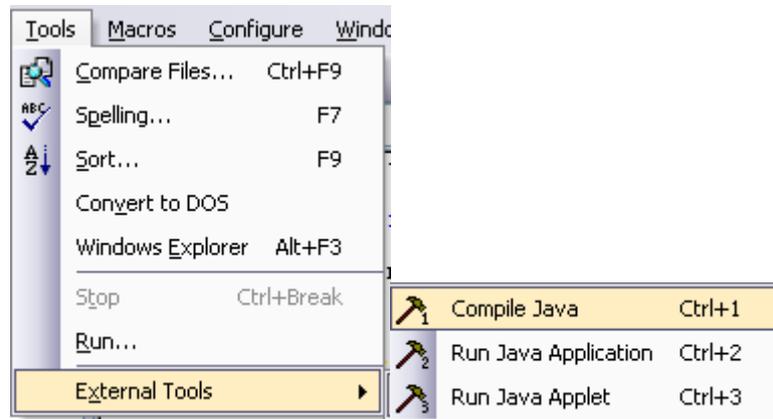
1. Pilih Menu Start
2. Pilih Programming
3. Pilih TextPad, maka akan muncul tampilan awal TextPad sebagai berikut :



4. Ketik program sederhana berikut ini :

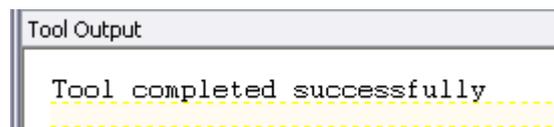
```
public class Pertama{
public static void main(String args[]){
    System.out.println("Selamat Datang di STMIK
AKAKOM");
}
}
```

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

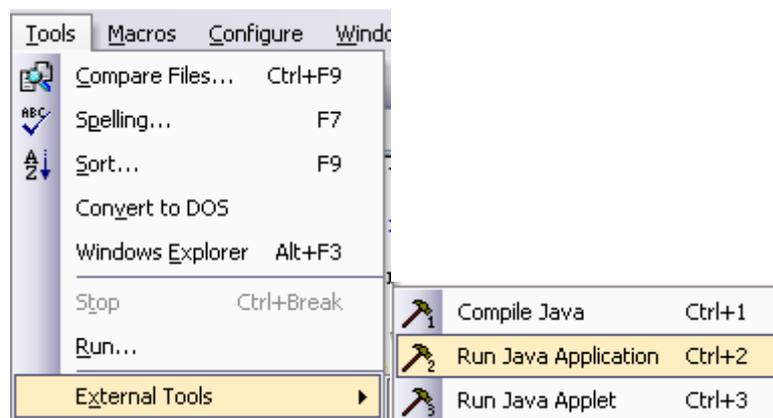


7. Setelah tidak ada kesalahan dan pada Tools Output muncul tulisan sebagai berikut :

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



8. Lakukan perintah berikut :



9. 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);
}
}

```

10. 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));
    }
}

```

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

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 mthod. 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 Fungsi1{  
  
    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();  
    }  
}
```

4. Jalankan dan amati hasilnya

```
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());
    }
}

```

5. 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());
    }
}

```

6. 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.

7. 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));
}
}
```

8. 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.

9. 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;
}
}
}

```

10. 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 :

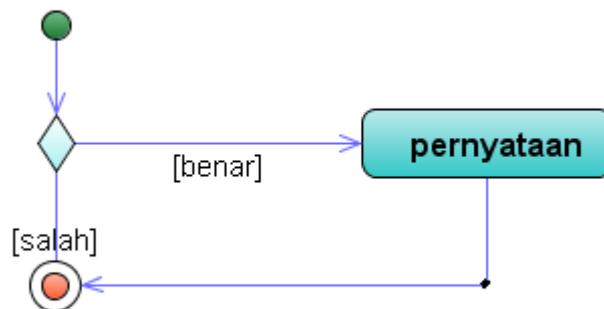
- Memahami tentang algoritma seleksi IF ELSE dan dapat menggunakannya.
- Menyelesaikan masalah dengan algoritma seleksi.

B. TEORI SINGKAT

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

if (kondisi)
 pernyataan;

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.

B. TEORI SINGKAT

1. Dengan menggunakan TextPad ketikkan program program – program berikut

Program 1

```

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. Buat *activity diagramnya*
4. 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");
    }
}

```

1. masukkan angka 7 dan jelaskan hasilnya!
2. Buat *activity diagramnya*.
3. 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!!!");
    }
}
```

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 :

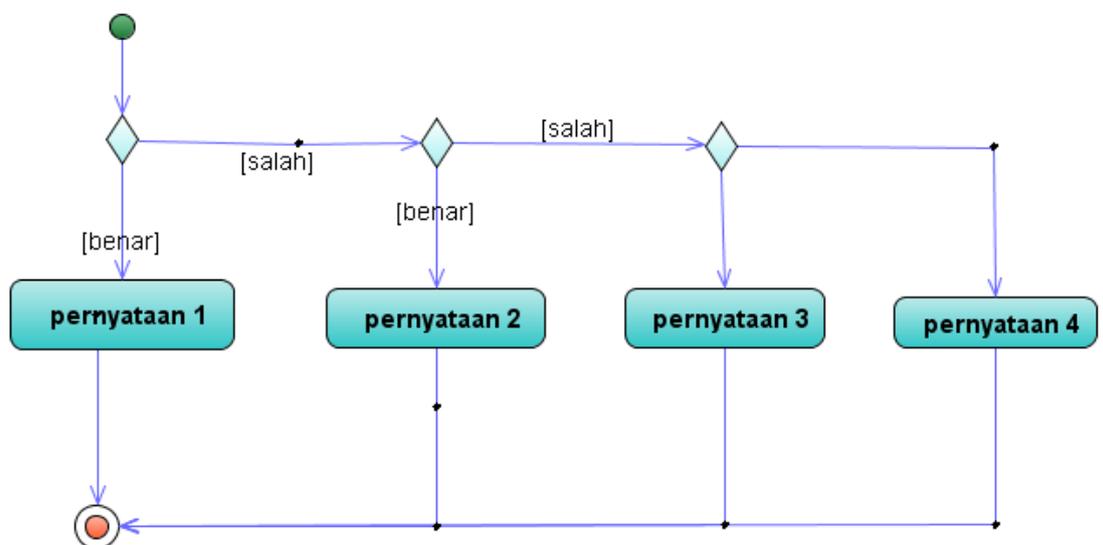
- Memahami tentang algoritma seleksi SWITCH dan dapat menggunakannya.
- Menyelesaikan masalah dengan algoritma seleksi.

B. TEORI SINGKAT

Pernyataan switch adalah pernyataan yang digunakan untuk menjalankann 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 memasukan 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 memasukan beberapa kemungkinan.

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

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.

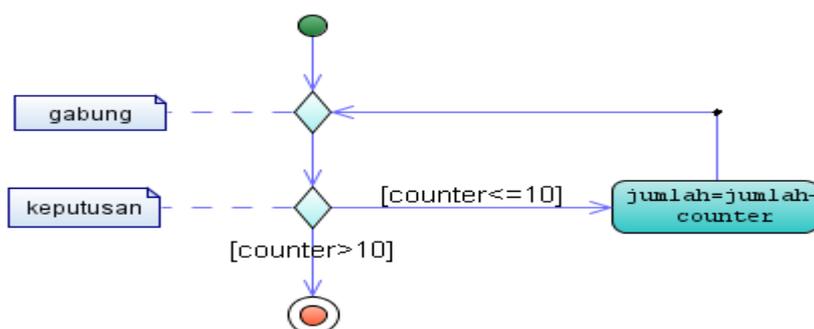
Bentuknya :

while (ungkapan)
Pernyataan;

Keterangan :

- bagian pernyataan akan dieksekusi selama ungkapan dalam **while** bernilai benar.
- Pengujian terhadap ungkapan 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 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++;
        }
    }
}
```

2. Ubah pernyataan **bil=1** menjadi **bil=5**, pernyataan while (**bil<=5**) dengan **while(bil>=1)** dan **bil++** menjadi **bil--**, amati hasil outputnya.
3. 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;
        }
    }
}
```

4. Ubah program di atas untuk menampilkan bilangan ganjil saja yaitu 1, 3, 5, 7, 9

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.

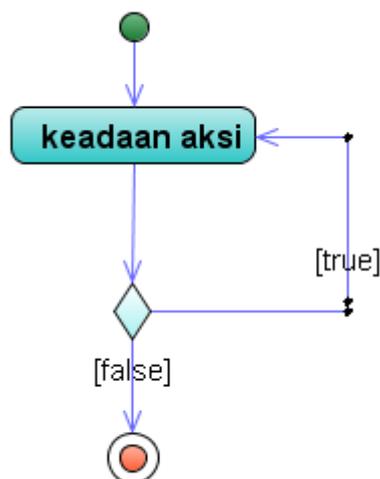
Bentuknya :

```
do
{
    pernyataan1;
    pernyataan2;
    .....
    pernyataan_N;
}
while (ungkapan)
```

Keterangan :

- Bagian pernyataan1 hingga pernyataanN dijalankan secara berulang sampai ungkapan bernilai salah.
- Pengujian ungkapan 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;
        while (bil<=10) {
            System.out.println(bil);
            bil+=2;
        }
    }
}
```

2. Ubah program di atas agar mendapat hasil output yang bilangan genap saja (0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20)
3. 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("-----");
}
}
```

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

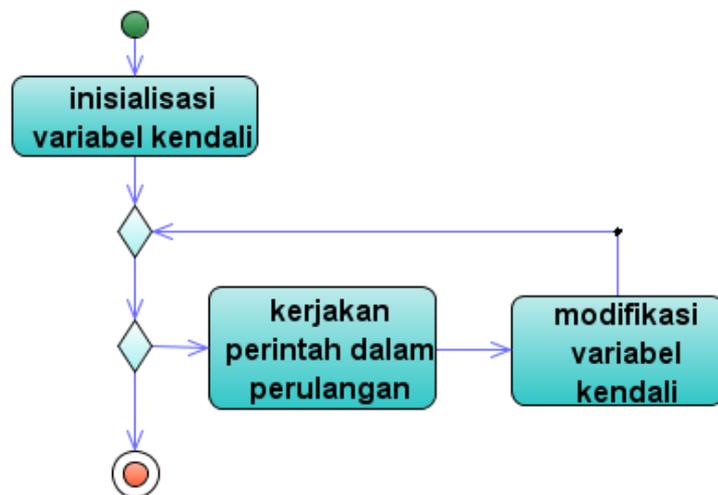
Sama seperti pernyataan perulangan while dan do...while, pernyataan for juga digunakan untuk mengerjakan pernyataan atau sekelompok pernyataan secara berulang. Bedanya adalah dengan pernyataan for perulangan akan dikerjakan dalam hitungan yang sudah pasti, sementara while dan do...while tidak.

Bentuknya :

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

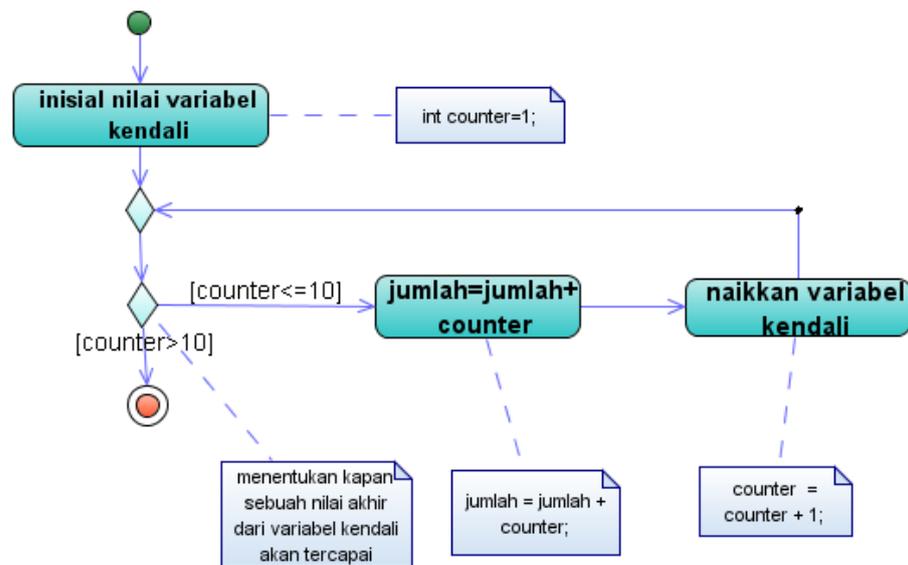
Keterangan :

- 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);
    }
}
  
```

```

public class UlangFor2 {
    public static void main (String args[]){
        int bil;
        for (bil=5;bil>=1;bil--)

            System.out.println(bil);
    }
}
  
```

2. Buatlah program untuk menampilkan bilangan kelipatan 5 antara 125 sampai dengan 200
3. 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);
    }
}
```

4. 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.

C. PRAKTIK

1. Tuliskan program berikut

```
import java.util.Scanner;
public class UlangDo1{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int bil;
        bil=1;
        while (bil<=100) {
            System.out.println(bil);
            bil+=2;
        }
    }
}
```

2. Ubah ke dalam bentuk WHILE dan FOR
3. Buat program untuk menampilkan deret berikut

1 1 2 3 5 8 13 . . .

4. Dengan menggunakan WHILE, FOR, dan Do WHILE

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum

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 perintahtersebut.

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.

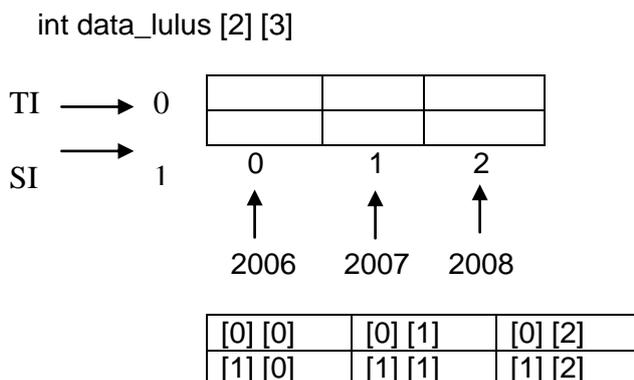
B. TEORI SINGKAT

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



C. PRAKTIK

1. Tuliskan program 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, tidak bisa diselesaikan hanya dengan menggunakan satu bentuk perintah saja, tetapi digabungkan dengan bentuk yang lainnya.

C. PRAKTIK

Akan diberikan kasus oleh dosen pengampu.

D. LATIHAN

- Akan disampaikan oleh Dosen Pengampu saat praktikum

E. TUGAS :

- Akan disampaikan oleh Dosen Pengampu saat praktikum