



**modul praktikum**

# **ALGORITMA DAN PEMROGRAMAN**

**Penyusun:**

**Dr. Dwi Ratna S, S.Si, MT**

**Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA**  
FAKULTAS SAINS DAN ANALITIKA DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER



**Untuk kalangan sendiri**

# MODUL PRAKTIKUM ALGORITMA DAN PEMROGRAMAN



Disusun Oleh :

Dr. Dwi Ratna Sulistyaningrum, MT  
Dr. Budi Setiyono, MT

**DEPARTEMEN MATEMATIKA FSAD  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA**

# MODUL PRAKTIKUM

Mata Kuliah : ALGORITMA DAN PEMROGRAMAN  
Program Studi : MATEMATIKA

DIBUAT OLEH	DISAHKAN OLEH	DIKETAHUI OLEH
Tim laboratorium Pemrograman Dan Komputasi Visual	Tim Dosen matematika FSAD ITS	Kepala Laboratorium

## DAFTAR ISI

Cover .....	1
Lembar pengesahan .....	2
Daftar isi .....	3
Modul 1 : Pengenalan lingkungan NetBeans .....	5
Modul 2 : Dasar Pemrograman Java .....	10
Modul 3 : Variabel dan Operator .....	14
Modul 4 : Input dan GUI Sederhana .....	21
Modul 5 : Struktur Kontrol I .....	26
Modul 6 : Struktur Kontrol II .....	32
Modul 7 : Perulangan .....	37
Modul 8 : Pernyataan Percabangan .....	42
Modul 9 : Fungsi / Method .....	47
Modul 10 : Array .....	50

**PRAKTIKUM****1**

# Pengenalan Netbeans

**A. TUJUAN PEMBELAJARAN**

1. Mengetahui lingkungan pemrograman dengan netbeans
2. Menggunakan netbeans sebagai editor pemrograman java
3. Membuat project sederhana menggunakan netbeans (menyimpan/meload menjalankan, meng-compile)

**B. DASAR TEORI****NETBEANS**

NetBeans IDE (Integrated Development Environment) adalah sebuah lingkungan pengembangan terintegrasi yang tersedia untuk Windows, Mac, Linux, dan Solaris. Proyek NetBeans terdiri dari open-source IDE dan platform aplikasi, yang memungkinkan pengembang untuk secara cepat membuat web, enterprise, desktop, dan aplikasi mobile menggunakan platform Java, serta JavaFX, PHP, JavaScript dan Ajax, Ruby dan Ruby on Rails, Groovy dan Grails, dan C/C++.

Proyek NetBeans didukung oleh komunitas pengembang yang ekstensif dan menawarkan dokumentasi dan sumberdaya pelatihan serta beragam pilihan plugin pihak ketiga.

NetBeans versi 7.1. IDE NetBeans 7.1 memperkenalkan dukungan untuk JavaFX 2.0 dengan mengaktifkan siklus kompilasi/debug/profil pengembangan secara penuh untuk aplikasi JavaFX 2.0. Rilis ini juga menyediakan perangkat tambahan Swing GUI Builder secara signifikan, dukungan CSS3, dan perangkat untuk visual debugging untuk Swing dan antarmuka pengguna untuk JavaFX. Tambahan termasuk dukungan Git yang terintegrasi ke dalam IDE, fitur baru PHP debugging, beberapa perbaikan pada JavaEE dan Maven, dan banyak lainnya.

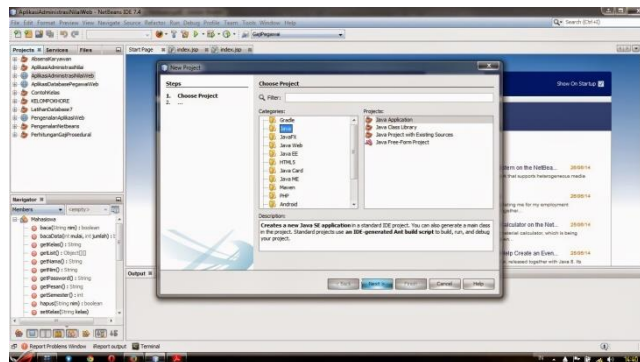
## C. PERCOBAAN

### *Instalasi NetBeans*

Untuk dapat menggunakan NetBeans, kita harus menginstalasi NetBeans dan JDK. Keduanya dapat di download secara gratis di <http://www.netbeans.com/> dan <http://www.oracle.com/>. Supaya lebih mudah dalam menginstall, install JDK terlebih dulu baru kemudian install NetBeans.

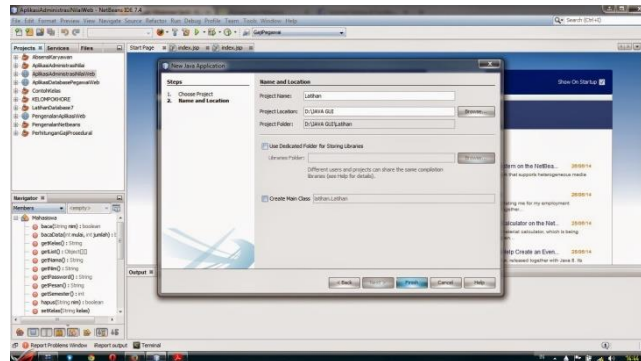
### *Membuat Project*

Untuk membuat project baru kita dapat memilih menu **File>>New Project**, sehingga tampil jendela seperti pada gambar berikut ini :



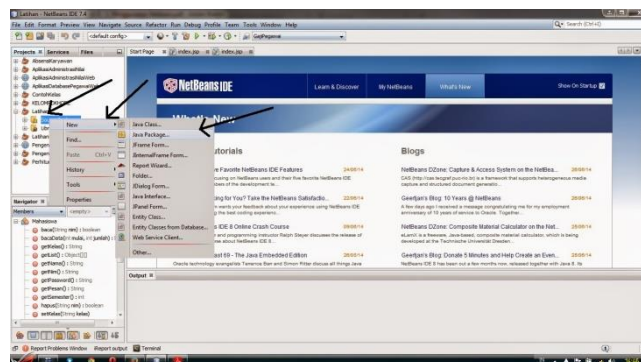
Karena kita akan membuat aplikasi java, maka pada **Categories** pilih **Java** dan ada **Projects** pilih **Java Application**. Kemudian klik tombol **Next**.

Untuk latihan membuat project, pada **Project Name** isi dengan **Latihan** dan kosongkan tanda check pada **Create Main Class** dan **Set as Main Project**, sehingga tampilan seperti pada gambar berikut ini:

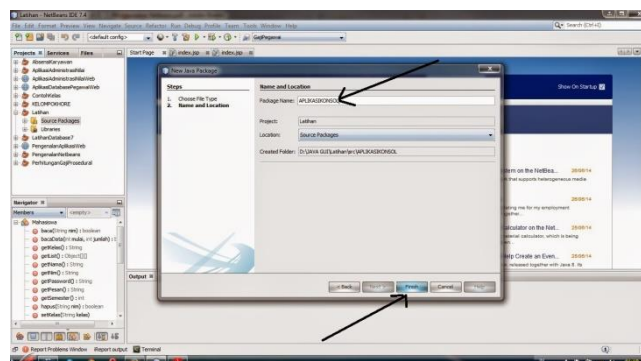


Tentukan juga di mana project akan disimpan dengan mengisinya pada **Project Location**. Kemudian klik tombol **Finish**.

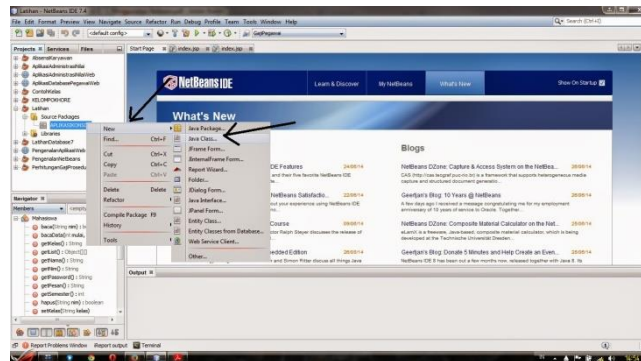
Selanjutnya klik tanda tambah (+) pada **Source Packages**, kemudian kita tambahkan package pada project yang kita buat dengan cara klik kanan pada **Source Packages** dan pilih **New >> Java Package...** seperti pada gambar berikut ini:



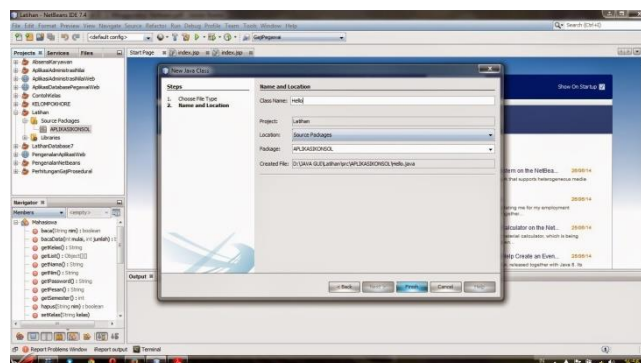
Pada **Package Name** isi dengan **APLIKASIKONSOL** dan klik tombol **Finish**.



Setelah nama package selesai kita buat, klik kanan pada nama package tersebut dan pilih **New**, kemudian **Java Class...** seperti gambar di bawah ini:



Pada Class Name isi **Hello** dan pastikan nama **Package** adalah **APLIKASIKONSOL**, sehingga seperti pada gambar berikut ini :



Di dalam class Hello ketik kode program di bawah ini:

```
S  
}
```

### Percobaan java

1. Menganalisa dan membenahi kesalahan pada program. Tulislah program berikut ini dan simpanlah dengan nama tertentu. Lakukan kompilasi pada file tersebut dan amati hasilnya. Kenapa terjadi kegagalan pada saat kompilasi?. Benahilah kesalahan diatas sehingga program tersebut dapat berjalan dengan baik.



**D. LATIHAN**

```
public class Testing
{
    public static void main(String[] args)
    {
        System.out.println("Apa yang salah");
    }
}
```

**E. TUGAS**

**PRAKTIKUM****2**

# Dasar Pemrograman Java

**A. TUJUAN PEMBELAJARAN**

1. Mengenal bahasa pemrograman java
2. Membuat input dan output sederhana menggunakan java

**B. DASAR TEORI**

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan Smalltalk, namun dirancang agar lebih mudah dipakai dan -platform independent, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer--. Bahasa ini juga dirancang untuk pemrograman di Internet sehingga dirancang agar aman dan portabel.

**Platform Independent**

Platform independent berarti program yang ditulis dalam bahasa Java dapat dengan mudah dipindahkan antar berbagai jenis sistem operasi dan berbagai jenis arsitektur komputer. Aspek ini sangat penting untuk dapat mencapai tujuan Java sebagai bahasa pemrograman Internet di mana sebuah program akan dijalankan oleh berbagai jenis komputer dengan berbagai jenis sistem operasi. Sifat ini berlaku untuk level source code dan binary code dari program Java. Berbeda dengan bahasa C dan C++, semua tipe data dalam bahasa Java mempunyai ukuran yang konsisten di semua jenis platform. Source code program Java sendiri tidak perlu dirubah sama sekali jika Anda ingin mengompil ulang di platform lain. Hasil dari mengompil source code Java bukanlah kode mesin atau instruksi prosesor yang spesifik terhadap mesin tertentu, melainkan berupa bytecode yang berupa file berekstensi .class. Bytecode tersebut

dapat langsung Anda eksekusi di tiap platform yang dengan menggunakan Java Virtual Machine (JVM) sebagai interpreter terhadap bytecode tersebut.

JVM sendiri adalah sebuah aplikasi yang berjalan di atas sebuah sistem operasi dan menerjemahkan bytecode program Java dan mengeksekusinya, sehingga secara konsep bisa dianggap sebagai sebuah interpreter. Proses pengeksekusian program Java dapat dilukiskan seperti di Gambar 1. Dengan cara ini, sebuah program Java yang telah dikompilasi akan dapat berjalan di platform mana saja, asalkan ada JVM

Kompiler dan interpreter untuk program Java berbentuk Java Development Kit (JDK) yang diproduksi oleh Sun Microsystems. JDK ini dapat didownload gratis dari situs [java.sun.com](http://java.sun.com). Interpreter untuk program Java sendiri sering juga disebut Java Runtime atau Java Virtual Machine. Interpreter Java, tanpa kompilernya, disebut Java Runtime Environment (JRE) dapat didownload juga di situs yang sama. Untuk mengembangkan program Java dibutuhkan JDK, sementara jika hanya ingin menjalankan bytecode Java cukup dengan JRE saja. Namun untuk mengeksekusi applet (sebuah bytecode Java juga) Anda biasanya tidak perlu lagi mendownload JRE karena browser yang Java-enabled telah memiliki JVM sendiri.

### **Library**

Selain kompiler dan interpreter, bahasa Java sendiri memiliki library yang cukup besar yang dapat mempermudah Anda dalam membuat sebuah aplikasi dengan cepat. Library ini sudah mencakup untuk grafik, desain user interface, kriptografi, jaringan, suara, database, dan lain-lain.

### **Oject Oriented**

Java adalah bahasa pemrograman berorientasi objek. Pemrograman berorientasi objek secara gamblang adalah teknik untuk mengorganisir program dan dapat dilakukan dengan hampir semua bahasa pemrograman. Namun Java sendiri telah mengimplementasikan berbagai fasilitas agar seorang programmer dapat mengoptimalkan teknik pemrograman berorientasi objek.

Sedikit perbandingan tambahan dengan bahasa C dan C++, Java banyak mewarisi konsep orientasi objek dari C++ namun dengan menghilangkan aspek-aspek kerumitan dalam bahasa C++ tanpa mengurangi kekuatannya. Hal ini mempermudah

programer pemula untuk mempelajari Java namun mengurangi keleluasaan programer berpengalaman dalam mengutak-atik sebuah program. Di balik kemudahan yang ditawarkan Java, luasnya fasilitas library Java sendiri membuat seorang programer membutuhkan waktu yang tidak singkat untuk dapat menguasai penggunaan library-library tersebut.

```
Sintak Dasar pada Pemrograman Java
public class Hello{
    /*Program menampilkan 'Hello World'      */

    public static void main(String[]args){
        System.out.println("Hello World");
    }
}
```

Pada Pemrograman Java, sangat penting untuk diingat hal-hal berikut ini.

- Case Sensitivity berarti pengidentifikasi Hello dan hello akan memiliki arti yang berbeda pada Java.
- Class Names, Untuk semua nama class, huruf pertama harus kapital. Jika beberapa kata yang digunakan untuk membentuk nama class, huruf pertama setiap kata harus kapital.

Contoh class: MyFirstJavaClass

- Method Names, Semua nama Method harus diawali dengan huruf kecil. Jika beberapa kata yang digunakan untuk membentuk nama method, maka huruf pertama diawali dengan huruf kecil kemudian kata berikutnya diawali dengan huruf kapital.

Contoh public void myMethodName ()

- Program File Name, nama file program harus persis sesuai dengan nama class. Ketika menyimpan file, Anda harus menyimpannya dengan nama class (ingat Java adalah

case sensitive) dan menambah '.java' ke akhir nama ( jika nama file dan nama class tidak cocok, program Anda tidak dapat di compile).

Contoh: Asumsikan 'MyFirstJavaProgram' adalah nama class, maka file tersebut harus disimpan dengan 'MyFirstJavaProgram.java'.

–public static void main(String args[]), program Java memulai proses dari method main(), yang merupakan bagian wajib dari setiap program Java

### C. PERCOBAAN

**PRAKTIKUM****3**

# Variable dan Operator

## A. TUJUAN PEMBELAJARAN

1. Mengetahui variabel dalam java
2. Dapat membuat variabel dalam java
3. Memahami berbagai bentuk operator dalam java
4. Dapat menggunakan operator dalam java

## B. DASAR TEORI

Variabel adalah satuan dasar penyimpanan dalam Java. Suatu variabel bersifat sementara. Sesuatu yang disimpan dalam variabel biasa disebut data. Data dalam suatu variabel dapat berubah selama berjalannya program. Data yang tersimpan dapat berupa bilangan, huruf, dll. Jenis data yang tersimpan dalam variabel merupakan tipe variabel.

Nama yang diberikan untuk menamai suatu variabel disebut Identifier. Identifier digunakan untuk menamai variabel, *class*, dan *method*. Penamaan identifier tidak boleh mengandung spasi dan harus diawali dengan karakter Unicode, tanda \$ (dollar) atau tanda \_ (underscore). Penamaan identifier bersifat case-sensitive dan tidak dibatasi panjang maksimum.

Untuk variabel, java memiliki 8 tipe dasar, yaitu boolean, char, byte, short, int, long, float, dan double.

**Tabel. 2.1 Spesifikasi panjang bit dan nilai default tipe data dasar.**

Tipe	Panjang bit	Range	Default
Boolean	16	-	False
Char	16	0 to $2^{16} - 1$	'\u0000'
Byte	8	$-2^7$ to $2^7 - 1$	0
Short	16	$-2^{15}$ to $2^{15} - 1$	0
Int	32	$-2^{31}$ to $2^{31} - 1$	0
Long	64	$-2^{63}$ to $2^{63} - 1$	0L
Float	32	$3.4e-038$ to $3.4e+038$	0.0F
Double	64	$1.7e-308$ to $1.7e+308$	0.0

Operator adalah suatu tanda atau simbol yang dipakai untuk menyatakan suatu operasi atau manipulasi nilai/data. Operator dalam java terbagi menjadi 3 bentuk, yaitu unary operator, binary operator, dan ternary operator. Unary operator adalah operator yang hanya menggunakan 1 operand. Binary operator adalah operator yang menggunakan 2 operand, dan Ternary operator adalah operator dengan 3 operand. Operator yang sering digunakan adalah unary dan binary, untuk ternary masih jarang digunakan.

Menurut jenis penggunaannya, operator dalam java dibagi menjadi 8 jenis, yaitu operator aritmatika, increment-decrement, bitwise, logika, shift (geser), assignment (penugasan), kondisional, dan kombinasi.

Operator aritmatika adalah operator yang digunakan dalam proses aritmatika.

**Tabel 2.2 Operator Aritmatika**

Operator	Penggunaan	Deskripsi
+	$a + b$	Operasi penjumlahan
-	$a - b$	Operasi pengurangan
*	$a * b$	Operasi perkalian
/	$a / b$	Operasi pembagian
%	$a \% b$	Operasi modulus

Operator Increment dan Decrement digunakan untuk menaikkan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan

hanya dapat digunakan pada variabel. Ada dua versi operator increment maupun decrement, yaitu prefix dan postfix. Prefix berarti operator digunakan sebelum variabel atau ekspresi, dan postfix berarti operator digunakan sesudahnya.

Tabel 2.3 Operator Increment-Decrement

Operator	Penggunaan	Deskripsi
++	a++	Menambah nilai a sebelum operasi
	++a	Menambah nilai a sesudah operasi
--	a--	Mengurangi nilai a sebelum operasi
	--a	Mengurangi nilai a sebelum operasi

Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan.

Tabel 2.4 Bitwise Operator

Operator	Penggunaan	Keterangan
~	~a	Operasi komplemen
&	a&b	Operasi AND
	a b	Operasi OR
^	a^b	Operasi XOR

Operator Boolean adalah operator yang mirip dengan operator bitwise, operator Boolean mewajibkan operannya bertipe Boolean juga.



Tabel 2.5 Boolean Operator

Operator	Penggunaan	Keterangan
!	!a	Operator negasi
&	a&b	Operator AND
	a b	Operator OR
^	a^b	Operator XOR
&&	a&&b	Operator AND (Short Sircuit)
	a  b	Operator OR (Short Sircuit)

Operator Logikal adalah operator yang sering digunakan untuk menghasilkan nilai boolean.

Tabel 2.6 Logikal Operator

Operator	Penggunaan	Keterangan
==	a == b	Operasi sama dengan
!=	a != b	Operasi tidak sama dengan
>	a > b	Operasi lebih dari
>=	a >= b	Operasi lebih dari sama dengan
<	a < b	Operasi kurang dari
<=	a <= b	Operasi kurang dari sama dengan

Operator Shift (geser) adalah operator yang berfungsi untuk menggeser susunan bit pada suatu nilai.

Tabel 2.7 Shift Operator

Operator	Keterangan
>>	Right Shift
>>>	Unsigned Right Shift
<<	Left Shift

Operator penugasaan atau assignment adalah operator yang berfungsi untuk mendefinisikan suatu nilai, operatornya adalah =

Operator Kondisi adalah operator yang digunakan untuk melakukan kondisi, ditulis dengan ? (tanda tanya) disertai : (titik 2). Contoh penulisan :

$a < b ? a : b$

dapat dibaca “apakah a lebih kecil dari b? jika iya return nilai a, jika bukan return nilai b”.

Operator kombinasi adalah gabungan dari 2 operator. Misal +=, \*=, dll.

## C. PERCOBAAN

### 1. Nilai default data dasar

```
public class DefaultValue {
    static boolean b;
    static char c;
    static byte bt;
    static short s;
    static int i;
    static long l;
    static float f;
    static double d;

    public static void main(String args[]) {
        System.out.println("Default value b = " + b);
        System.out.println("Default value c = " + c);
        System.out.println("Default value bt = " + bt);
        System.out.println("Default value s = " + s);
        System.out.println("Default value i = " + i);
        System.out.println("Default value l = " + l);
        System.out.println("Default value f = " + f);
        System.out.println("Default value d = " + d);
    }
}
```

### 2. Increment-Decrement

```
class IncDec {
    public static void main (String args[]) {
        int x = 8, y = 13;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x = " + ++x);
        System.out.println("y = " + y++);
        System.out.println("x = " + x--);
        System.out.println("y = " + --y);
    }
}
```

```
}  
}
```

### 3. Operator Shift

```
class Shift {  
    public static void main (String args[]) {  
        int x = 7;  
        System.out.println("x = " + x);  
        System.out.println("x >> 2 = " + (x >> 2));  
        System.out.println("x << 1 = " + (x << 1));  
        System.out.println("x >>> 1 = " + (x >>> 1));  
    }  
}
```

### 4. Operator Kondisi

```
class Kondisi {  
    public static void main (String args[]) {  
        int a = 8;  
        int b = 10;  
        int kecil = a < b ? a : b;  
        System.out.println("Nilai yang lebih kecil  
adalah" + kecil)  
    }  
}
```

## D. LATIHAN

1. Buatlah Kelas yang menghitung a = 100 dan b = 10 dengan operasi +, -, \*, /, dan %. Tampilkan.
2. Buatlah Kelas yang menentukan nilai logika bila c = *true* dan d = *false* dengan operasi AND, OR, XOR, AND (short circuit), OR (short circuit), dan buat lah e yang merupakan negasi dari c. Tampilkan.

## E. TUGAS

1. Buatlah program yang menghitung luas dan volume dari tabung. Gunakan Math.PI.
2. Buatlah program yang menentukan apakah a merupakan modulus dari b, dengan menggunakan operator kondisi.

*Hint : buat variabel berupa string untuk masukkan data hasil operasi kondisi.*

3. Buatlah contoh program yang menggunakan operator kombinasi. Setidaknya 2 macam kombinasi.

Kumpulkan dalam format laporan resmi. Berikan penjelasan singkat dalam laporan kalian.

**PRAKTIKUM****4**

# Input GUI Sederhana

**A. TUJUAN PEMBELAJARAN**

1. Memahami input dalam java
2. Mengetahui cara input dalam java
3. Memahami GUI sederhana untuk inputan dalam java
4. Dapat membuat inputan melalui GUI dalam java

**B. DASAR TEORI**

Input berasal dari kata in dan put yang berarti meletakkan ke dalam, atau bias kita sebut memasukkan. Input dalam Java terbagi menjadi 2 bagian yaitu input dalam running program ataupun melalui GUI. Untuk inputan yang melalui running program terbagi menjadi 2 basic, yaitu `BufferedReader` dan `Scanner`. Inti dari kedua program ini adalah sama. Perbedaannya adalah :

1. Dalam JDK versi 6, `BufferedReader` dapat mendetek hingga 8192 karakter, sedangkan `Scanner` hanya 1024 karakter.
2. `BufferedReader` memasukkan tiap `line(String)`, sedangkan `Scanner` dapat memasukkan data dasar.
3. `BufferedReader` tersinkronisasi sedangkan `Scanner` tidak.
4. `BufferedReader` langsung melempar `I/O Exception` sedangkan `Scanner` menyembunyikan.
5. `BufferedReader` lebih cepat dibanding `Scanner` sebab `Scanner` membaca inputan tiap karakter terlebih dahulu. `Scanner` membaca angka, huruf, demiliter, spasi dll. `BufferedReader` membaca Line.

Penulisan untuk deklarasi `Scanner` :

```
Scanner baca = new Scanner(System.in);  
String kalimat = baca.nextLine( );
```

Dalam kodingan diatas, kita menggunakan Scanner dengan pembaca berupa kalimat. Method Method sederhana dari Scanner antara lain :

```
next( )           //membaca String(satu kata)
nextByte( )       //membaca Byte
nextBoolean( )    //membaca Boolean
nextDouble( )     //membaca Double
nextFloat( )      //membaca Float
nextInt( )        //membaca Int
nextLine( )       //membaca String(satu kalimat)
```

Untuk BufferedReader, pendeklarasiannya adalah :

```
BufferedReader tulis = new BufferedReader(new
InputStreamReader(System.in));

String kalimat = tulis.readLine( );
```

Untuk membaca inputan lain selain String dalam bentuk Line, bufferedreader harus men-parse datanya sendiri, semisal untuk inputan int, maka kodingannya adalah :

```
int angka = Integer.parseInt(tulis.readLine( ));
```

Selanjutnya adalah inputan dengan menggunakan GUI sederhana. Kelas yang dipakai untuk hal ini adalah JOptionPane. JOptionPane membuat suatu GUI sederhana, suatu dialog box yang standar sebagai input atau output ataupun peringatan kepada user. Method basic dari JOptionPane antara lain :

```
showConfrimDialog( )    //menanyakan yes/no/cancel
showInputDialog( )      //meminta suatu inputan
showMessageDialog( )    //memberikan message ke user
showOptionDialog( )     //perpaduan ketiganya
```

Pendeklarasiannya adalah :

```
String inputan =  
JOptionPane.showInputDialog("Masukkan sesuatu");  
  
String outputan = "Inputannya adalah " + inputan;  
JOptionPane.showMessageDialog(null, outputan);
```

## PERCOBAAN

### 1. Input Nama

```
public class Input {  
    public static void main(String args[]) {  
        System.out.println("Masukkan nama Anda :");  
        Scanner baca = new Scanner(System.in);  
        String nama = baca.nextLine( );  
        System..out.println("Nama anda adalah "+nama);  
    }  
}  
  
public class Input {  
    public static void main(String args[]) {  
        System.out.println("Masukkan nama Anda :");  
        BufferedReader baca = new BufferedReader(new  
            InputStreamReader(System.in));  
        String nama = baca.readLine( );  
        System..out.println("Nama anda adalah "+nama);  
    }  
}  
  
public class Input {  
    public static void main(String args[]) {  
        String nama =  
            JOptionPane.showInputDialog("Masukkan nama  
            Anda :");  
  
        String message = "Nama anda adalah " + nama;  
        JOptionPane.showMessageDialog(null, message);  
    }  
}
```

### 2. Masukkan Int

```
class InputInt {  
    public static void main (String args[]) {  
        System.out.println("Masukkan 2 angka : ");  
        Scanner baca = new Scanner(System.in);  
        int angka1 = baca.nextInt( );  
        int angka2 = baca.nextInt( );  
        System.out.println("Angka1 =" + angka1 + "Angka2 =  
        " + angka2);  
    }  
}
```

```
}  
  
class InputInt {  
    public static void main (String args[]) {  
        System.out.println("Masukkan 2 angka : ");  
        BufferedReader baca = new BufferedReader( new  
            InputStreamReader(System.in));  
        int angka1 = Integer.parseInt(baca.readLine(  
    ));  
        int angka2 = Integer.parseInt(baca.readLine(  
    ));  
        System.out.println("Angka1 =" + angka1 + "Angka2 =  
        " + angka2);  
    }  
}
```

### 3. Next

```
class Next {  
    public static void main (String args[]) {  
        System.out.println("Masukkan 3 kata : ");  
        Scanner baca = new Scanner(System.in);  
        String kata1 = baca.next( );  
        String kata2 = baca.next( );  
        String kata3 = baca.next( );  
        System.out.println("kata1 = " + kata1 + " kata2 =  
                            " + kata2 + " kata3 = " + kata3);  
    }  
}
```



**C. LATIHAN**

1. Inputkan suatu kalimat dengan menggunakan scanner, dan suatu kalimat dengan bufferedReader, kemudian gabung kedua kalimat itu dan tampilkan menggunakan JOptionPane.

**D. TUGAS**

1. Buatlah program yang meminta inputan berupa biodata mahasiswa (nama, nrp(int), kota Asal) kemudian print/display hasilnya.
2. Buatlah program yang meminta inputan berupa suhu(int) dalam Celcius kemudian print/display suhu dalam Reaumur, Fahrenheit, dan Kelvin.
3. Buatlah program yang meminta inputan berupa 3 kata, kemudian print/display 3 kata tersebut dengan urutan kata ke-2, ke-1, dan ke-3.

Kumpulkan dalam format laporan resmi. Berikan penjelasan singkat dalam laporan kalian.

**PRAKTIKUM****5**

# Struktur Kontrol

**A. TUJUAN PEMBELAJARAN**

1. Memahami tentang mengontrol program
2. Memahami penggunaan pencabang if-then, else if, dan else

**B. DASAR TEORI**

Java menyuport 2 pernyataan seleksi / pencabangan yaitu if dan switch. Percabangan if dipakai jika kita menginginkan suatu pernyataan itu dilakukan dengan syarat tertentu yang bernilai benar. Sintaks dari if-then adalah sebagai berikut:

```
if (ekspresi_boolean) {  
    Pernyataan1;  
}
```

Sintaks Pernyataan 1 akan dijalankan apabila nilai dari ekspresi\_boolean adalah true. Dan ketika nilai false, maka pernyataan1 akan diabaikan.

Untuk bentuk If-then else, adalah sebagai berikut

```
if (ekspresi_boolean) {  
    Pernyataan1;  
}  
else {  
    Pernyataan2;  
}
```

Seperti di atas, pernyataan 1 akan berjalan apabila nilai ekspresi\_boolean adalah true, dan ketika dia bernilai false, maka sintaks Pernyataan2 akan berjalan. Pernyataan else tidak dapat berdiri tanpa adanya if.

Untuk bentuk If-then-else if-else adalah :

```
if (ekspresi_boolean) {
    Pernyataan1;
}
else if (ekspresi_boolean2) {
    Pernyataan2;
}
else if (ekspresi_boolean3) {
    Pernyataan3;
}
.
.
.
else {
    PernyataanN;
}
```

Pernyataan 1 akan dilakukan ketika ekspresi\_boolean bernilai true, dan ketika false, dia akan menuju ke percabangan terdekat sesuai dengan urutan program yaitu else if dengan ekspresi\_boolean2, apabila ekspresi\_boolean bernilai true, maka Pernyataan 2 akan dijalankan. Namun bila bernilai false, maka akan jalan ke percabangan berikutnya dan seterusnya hingga bila semua else if bernilai false, maka akan dijalankan PernyataanN milik else.

Yang terakhir adalah Nested IF, nested IF berarti IF yang bersarang, maksudnya adalah If yang berada dalam If, penulisannya adalah

```
if (ekspresi_boolean) {
    if (ekspresi_boolean) {
        Pernyataan1;
    }
}
```

Pernyataan 1 dijalankan apabila kedua nilai dari ekspresi Boolean adalah benar.

**C. PERCOBAAN****1. If**

```
public class If{
    public static void main(String args[]) {
        System.out.println("Bilangan kecil");
        Scanner baca = new Scanner(System.in);
        int bil = baca.nextInt( );
        if (bil >= 10) {
            System.out.println("Ini bukan bilangan
kecil");
        }
        System.out.println("Terima kasih");
    }
}
```

**2. If-then Else**

```
public class Else{
    public static void main(String args[]) {
        System.out.println("Masukkan nilai IP anda");
        Scanner baca = new Scanner(System.in);
        double ip = baca.nextDouble( );
        if (ip < 2) {
            System.out.println("Belajarlah lebih
rajin");
        }
        Else { System.out.println("Tingkatkan
prestasimu"); }
    }
}
```

**3. If-then-Else if-Else**

```
public class Elseif{
    public static void main(String args[]) {
        System.out.println("Masukkan nilai angka anda
");
        Scanner baca = new Scanner(System.in);
        int nilai = baca.nextInt( );
        if (nilai > 80) {
            System.out.println("Nilai Huruf Anda A");
        }
        else if (nilai > 70) {
            System.out.println("Nilai Huruf Anda AB");
        }
        else if (nilai > 60) {
            System.out.println("Nilai Huruf Anda B");
        }
    }
}
```

```
        else if (nilai > 55) {
            System.out.println("Nilai Huruf Anda BC");
        }
        else if (nilai > 50) {
            System.out.println("Nilai Huruf Anda C");
        }
        else if (nilai > 40) {
            System.out.println("Nilai Huruf Anda D");
        }
        else { System.out.println("Nilai Huruf Anda
E"); }
    }
}
```

#### 4. Nested If

```
class Nestedif {
    public static void main (String args[]) {
        System.out.println("Ingin cek program Bilangan,
                            IP atau Nilai
                            Angka ?
                            \n1. Bilangan kecil
                            \n2. IP
                            \n3. Nilai Angka");
        Scanner baca = new Scanner(System.in);
        int tugas = baca.nextInt( );
        if ( tugas == 1) {
            if milik program nomor 1;
        }
        else if ( tugas == 2) {
            if dan else milik program 2;
        }
        else { sintaks pencabangan milik program 3;}
    }
}
```

#### 5. If yang bersebelahan

```
class Ifsebelah1 {
    public static void main (String args[]) {
        System.out.println("Masukan nilai bilangan
                            diantara 0 hingga
                            100");
        Scanner baca = new Scanner(System.in);
        int bilangan = baca.nextInt( );
        if ( bilangan > 100) {
            System.out,println("Nilai terlalu besar");
        }
        if ( bilangan < 0) {
            System.out.println("Nilai terlalu kecil");
        }
    }
}
```

```
    }
    if ( bilangan >= 0 && bilangan <= 100) {
        System.out.println("Nilai bilangan
        sesuai");
    }
}

class Ifsebelah2 {
    public static void main (String args[]) {
        System.out.println("Masukan nilai bilangan
        diantara 0 hingga
        100");
        Scanner baca = new Scanner(System.in);
        int bilangan = baca.nextInt( );
        if ( bilangan < 0) {
            System.out,println("Nilai terlalu kecil");
        }
        if ( bilangan <= 100) {
            System.out.println("Nilai bilangan
            sesuai");
        }
        if (bilangan > 100) {
            System.out.println("Nilai terlalu besar");
        }
    }
}
```

**D. LATIHAN**

2. Buatlah program yang meminta inputan berupa 2 buah bilangan, kemudian system akan memberikan perintah untuk memasukkan operator. Ketika operator dimasukkan, system akan mengeluarkan output berupa hasil.

**E. TUGAS**

1. Buatlah program yang meminta inputan berupa 2 buah angka. Kemudian system akan mengecek apakah bilangan 1 merupakan kelipatan dari bilangan 2, jika iya output iya jika tidak output tidak.
2. Buatlah program yang meminta inputan berupa karakter. Kemudian seolah olah system akan mencoba untuk menebak nama yang akan disampaikan, semisal untuk char 'a', system akan menyebut 'Nama anda pasti Andrian', dst. Buat minimal 2 else if. (minimal menebak 3 karakter) dan gunakan else (misal mengatakan nama anda tidak bisa ditebak)

Hint : inputan char menggunakan :

```
char inisial = (char)System.in.read( );
```

3. Buatlah program yang menawarkan perubahan suhu, pertama program akan memberikan skala apa yang di pakai (C / R/ F/ K) kemudian meminta inputan suhu(int) kemudian menawarkan perubahan ke skala apa, setelah memilih, system akan mengeluarkan hasil perubahan.

Kumpulkan dalam format laporan resmi. Berikan penjelasan singkat dalam laporan kalian.

**PRAKTIKUM****6**

# Struktur Kontrol (Lanjut)

**A. TUJUAN PEMBELAJARAN**

1. Memahami tentang mengontrol program
2. Memahami penggunaan pencabang switch

**B. DASAR TEORI**

Java menyupport 2 pernyataan seleksi / pencabangan yaitu if dan switch. Percabangan switch dipakai saat memiliki syarat yang identik. Sintaks dari switch :

```
Switch (ekspresi) {  
    case nilai1 : Pernyataan1;  
        break;  
    case nilai2 : Pernyataan2;  
        break;  
    default : PernyataanDefault;  
        break;  
}
```

Switch menerima ekspresi berupa integer, short, character, byte, dan untuk versi JDK 7 keatas, switch menerima ekspresi berupa String.



## c. PERCOBAAN

### 1. Switch Char

```
Class SwitchChar {
    public static void main (String args[]) {
        System.out.println("Masukkan karakter nama
                               anda");
        char inisial = (char)System.in.read( );
        switch(inisial){
            case "a" : System.out.println("Nama anda
                               pasti Angga");
                        break;
            case "b" : System.out.println("Nama anda
                               pasti Budi");
                        break;
            default  : System.out.println("Nama anda
                               tidak terkenal");
                        break;
        }
    }
}
```

### 2. Switch Char w/o break

```
Class SwitchChar {
    public static void main (String args[]) {
        System.out.println("Masukkan karakter nama
                               anda");
        char inisial = (char)System.in.read( );
        switch(inisial){
            case "a" : System.out.println("Nama anda
                               pasti Angga");
            case "b" : System.out.println("Nama anda
                               pasti Budi");
            default  : System.out.println("Nama anda
                               tidak terkenal");
        }
    }
}
```

### 3. Switch int

```
public class SwitchInt {
    public static void main(String[] args) {
        int bulan = 8;
        String bulanString;
        switch (bulan) {
            case 1:  bulanString = "Januari";
                    break;
            case 2:  bulanString = "Februari";
                    break;
        }
    }
}
```

```
        case 3: bulanString = "Maret";
                break;
        case 4: bulanString = "April";
                break;
        case 5: bulanString = "Mei";
                break;
        case 6: bulanString = "Juni";
                break;
        case 7: bulanString = "Juli";
                break;
        case 8: bulanString = "Agustus";
                break;
        case 9: bulanString = "September";
                break;
        case 10: bulanString = "Oktober";
                break;
        case 11: bulanString = "Nopember";
                break;
        case 12: bulanString = "Desember";
                break;
        default: bulanString = "Ga ada bulan
segitu";
                break;
    }
    System.out.println(bulanString);
}
}
```

#### 4. Switch or

```
class SwitchOr {
    public static void main(String[] args) {

        int bulan = 2;
        int tahun = 2000;
        int jmlHari = 0;

        switch (bulan) {
            case 1: case 3: case 5:
            case 7: case 8: case 10:
            case 12:
                jmlHari = 31;
                break;
            case 4: case 6:
            case 9: case 11:
                jmlHari = 30;
                break;
            case 2:
                if (((tahun % 4 == 0) &&
                    !(tahun % 100 == 0))
```

```
        || (tahun % 400 == 0))
        jmlHari = 29;
    else
        jmlHari = 28;
        break;
    default:
        System.out.println("Ga ada bulan
segitu.");
        break;
    }
    System.out.println("Jumlah hari = "
        + jmlHari);
}
```

## 5. Nested Switch

```
class NestedSwitch {
    public static void main (String args[]) {
        System.out.println("Ingin cek program apa ?" +
            "\n1. Switch bulan" +
            "\n2. Switch jml Hari");
        Scanner baca = new Scanner(System.in);
        int tugas = baca.nextInt( );
        switch(tugas){
            case 1 : switch 1;
                    break;
            case 2 : switch 2;
                    break;
            default : System.out.println("angka tidak
ada");
                    break;
        }
    }
}
```

**D. LATIHAN**

1. Buatlah program yang meminta inputan berupa sisi dan tinggi, kemudian system akan menanyakan ingin mencari volume kubus atau prisma segiempat, atau prisma segitiga atau juga tabung(sisi = jari).

**E. TUGAS**

1. Buatlah program yang menggunakan switch, untuk membedakan angka positif yaitu angka kecil, angka sedang, angka besar dimana kecil dibawah 10 sedang dibawah 100 dan besar diatas itu.
2. Buatlah program yang menentukan predikat berdasarkan nilai huruf, misal : A = luar biasa, B = diatas rata –rata dst. Gunakan switch character.
3. Buatlah program yang meminta inputan berupa 2 bilangan, dan 1 operator kemudian gunakan switch untuk mengoperasikan bilangan tersebut.
4. Buatlah program yang meminta inputan berupa jari-jari dan tinggi, lalu program menawarkan angka 1 – 4 dimana 1 menampilkan switch lingkaran, 2 menampilkan switch tabung, 3 switch kerucut dan 4 switch bola. Masing masing switch akan menampilkan pilihan lagi, mulai dari lingkaran menampilkan pilihan 1 untuk luas dan 2 untuk keliling, sedangkan tabung menampilkan 1 untuk volume dan 2 untuk luas, dst.

Kumpulkan dalam format laporan resmi. Berikan penjelasan singkat dalam laporan kalian.

**PRAKTIKUM****7**

# Perulangan / *Looping*

**A. TUJUAN PEMBELAJARAN**

1. Memahami tentang looping
2. Memahami penggunaan looping dalam java

**B. DASAR TEORI**

Java menyuport 3 macam perulangan, yaitu for, while, dan do-while. Perulangan for digunakan ketika kita mengetahui jumlah perulangan dengan pasti. Bentuk umum dari for adalah

```
for (inisialisasi; kondisi; perubahan){  
    pernyataan;  
}
```

Inisialisasi adalah pemisalan kita untuk looping misalnya `int i = 0`; kondisi adalah batas loopingnya misal `i < 10`, dan perubahan adalah nilai tumbuh `i`, biasanya `i++`.

While dan do-While digunakan ketika kita tidak mengetahui pasti jumlah perulangan yang dilakukan pernyataan while dan do-while akan dijalankan terus apabila kondisi dalam while bernilai true. Bentuk umum while adalah

```
while (kondisi){  
    pernyataan;  
}
```

Bentuk do-while adalah

```
do{  
    pernyataan;  
}while(kondisi);
```

Perbedaan antara while dan do-while adalah ketika masuk ke while dan kondisi awalnya salah, program tidak akan dijalankan sama sekali. Sedangkan do-while, ketika kita masuk ke do-while, program akan dijalankan sekali terlebih dahulu, kemudian baru melihat kondisi while.

**C. PERCOBAAN****1. Selamat datang**

```
Class for {
    public static void main (String args[]) {
        for ( int i = 0, i < 5, i++){
            System.out.println("Selamat datang");
        }
    }
}
```

```
Class while {
    public static void main (String args[]) {
        while(i < 5){
            System.out.println("Selamat datang");
            i++;
        }
    }
}
```

```
Class dowhile {
    public static void main (String args[]) {
        do{
            System.out.println("Selamat datang");
            i++;
        } while(i < 5);
    }
}
```

**2. for tak natural**

```
Class fortaknatural {
    public static void main (String args[]) {
        for(int i = 5; i > 0; i--){
            System.out.println(i);
        }
        for(int j = 0; j < 10; j=j+2){
            System.out.println(j);
        }
        for(int k = 1; k < 17; k=k*2){
            System.out.println(k);
        }
    }
}
```

**3. for dalam for**

```
public class for2 {
    public static void main(String[] args) {
        for( int i = 0; i < 5; i++) {
```

```
        for( int j = 0; j < i; j++) {
            System.out.println(j);
        }
    }
}
```

#### 4. Kondisi salah

```
class Salah {
    public static void main(String[] args) {
        boolean kondisi = false;
        int x = 0;
        int y = 0;
        while(kondisi){
            x++;
        }
        do{
            y++;
        }while(kondisi);
        System.out.println(x);
        System.out.pritnln(y);
    }
}
```

#### 5. for array (sekedar tahu)

```
class forarray {
    public static void main (String args[]) {
        int[ ] angka = {1,2,3,4,5,6,7,8,9,10};
        for( int item : angka ){
            System.out.println(item);
        }
    }
}
```

### D. LATIHAN

1. Buatlah program yang meminta inputan berupa integer positif kemudian, program akan menampilkan hasil faktorialnya.
2. Buatlah program yang meminta inputan berupa Suku pertama, beda, dan banyaknya suku, semisal suku pertama = 1, beda = 2, banyaknya suku = 10, maka program akan menampilkan 1, 3, 5, 7, 9, 11, 13, 15, 17, 19



**E. TUGAS**

1. Buatlah program yang meminta inputan berupa integer positif. Kemudian program akan menampilkan deret Fibonacci sebanyak inputan, misal 5 maka akan muncul 1, 1, 2, 3, 5
2. Buatlah program yang meminta inputan berupa integer positif. Kemudian program akan menentukan apakah inputan tersebut bilangan prima atau tidak.
3. Buatlah program yang meminta inputan berupa bilangan x dan batas bilangan. Program akan menampilkan semua bilangan dari 1 sampai batas yang tidak habis dibagi oleh x, misal :  $x = 3$  dan batas = 10, maka akan muncul 1, 2, 4, 5, 7, 8, 10
4. Buatlah program yang meminta inputan berupa bilangan integer, kemudian program akan menampilkan seperti ini :

Misal input = 6

```
*  
**  
***  
****  
*****  
*****
```

Kumpulkan dalam format laporan resmi. Berikan penjelasan singkat dalam laporan kalian.

**PRAKTIKUM****8**

# Pernyataan Percabangan

**A. TUJUAN PEMBELAJARAN**

1. Memahami tentang penggunaan break, continue, dan return

**B. DASAR TEORI**

Pernyataan percabangan memungkinkan kita untuk mengatur aliran eksekusi program. Java memberikan tiga bentuk pernyataan percabangan: break, continue dan return.

**A. Pernyataan break**

Pernyataan break memiliki dua bentuk: tidak berlabel (unlabeled) dan berlabel (labeled).

1. Pernyataan break tidak berlabel (unlabeled)

Pernyataan break tidak berlabel (unlabeled) digunakan untuk menghentikan jalannya pernyataan switch. Selain itu pernyataan break unlabeled juga bisa digunakan untuk menghentikan pernyataan-pernyataan for, while atau do-while loop.

2. Pernyataan break berlabel

Bentuk label dari pernyataan break akan menghentikan pernyataan di luarnya, dimana sebelumnya harus diberikan label yang sudah di spesifikasikan pada program pada pernyataan break.

Pernyataan break menghentikan pernyataan yang diberi label; dan tidak menjalankan aliran kontrol apapun pada label. Aliran kontrol pada label akan diberikan secara otomatis pada pernyataan yang terletak dibawah label.

## B. Pernyataan Continue

Pernyataan continue memiliki dua bentuk: berlabel dan tidak berlabel. Anda dapat menggunakan pernyataan continue untuk melanjutkan pengulangan yang sedang dijalankan oleh pernyataan for, while, atau do- while loop.

### 1. Pernyataan continue tidak berlabel (unlabeled)

Bentuk pernyataan continue tidak berlabel (unlabeled) akan melewati bagian pernyataan setelah pernyataan ini dituliskan dan memeriksa ekspresi logika (boolean) yang mengontrol pengulangan. Jika ekspresi logika (boolean) masih bernilai true, maka pengulangan tetap dilanjutkan. Pada dasarnya pernyataan ini akan melanjutkan bagian pengulangan pada pernyataan loop.

### 2. Labeled continue statement

Bentuk pernyataan continue berlabel (labeled) akan melanjutkan pengulangan yang sedang terjadi dan dilanjutkan ke pengulangan berikutnya dari pernyataan pengulangan yang diberi label (tanda).

## C. Pernyataan Return

Pernyataan return digunakan untuk keluar dari sebuah method. Pernyataan return memiliki dua bentuk: memberikan sebuah nilai, dan tidak memberikan nilai. Untuk memberikan sebuah nilai, cukup berikan nilai (atau ekspresi yang menghasilkan sebuah nilai) sesudah kata return. Contohnya,

```
return ++count;  
return "Hello";
```

Tipe data dari nilai yang diberikan harus sama dengan tipe dari method yang dibuat. Ketika sebuah method void dideklarkan, gunakan bentuk return yang tidak memberikan nilai. Contohnya,

```
return;
```

**C. PERCOBAAN****1. Break unlabeled**

```
Class breaktaklabel {
    public static void main (String args[]) {
        for ( int i = 0, i < 5, i++){
            System.out.println("[i = "+i+"] ");
            if(i==2){
                break;
            }
        }
    }
}
```

**2. Break labeled**

```
Class breaklabel {
    public static void main (String args[]) {
        int x = 35;
        int y = 4;
        int dummy = y;
        mulai :
        while(true){
            if(y>x){
                System.out.println("Nilai kelipatan y
                                   yang mendekati x
                                   adalah "+y);
                break mulai;
            }
            y = y + dummy;
        }
    }
}
```

**3. Continue unlabeled**

```
public class continuetaklabel {
    public static void main(String[] args) {
        for( int i = 0; i < 5; i++) {
            System.out.println("[i = "+i+"] ");
            if(i==2){
                continue;
            }
        }
    }
}
```

**4. Continue labeled**

```

class continuelabel1 {
    public static void main(String[] args) {
        lanjut :
        for(int i = 0; i < 5; i++){
            for(int j = 0; j < 5; j++){
                System.out.println("[ "+i+", "+j+"
]);
                if(j==2){
                    continue lanjut;
                }
            }
        }
    }
}

class continuelabel2 {
    public static void main(String[] args) {
        for(int i = 0; i < 5; i++){
            lanjut :
            for(int j = 0; j < 5; j++){
                System.out.println("[ "+i+", "+j+"
]);
                if(j==2){
                    continue lanjut;
                }
            }
        }
    }
}

```

**5. Return**

```

class kembali {
    public static void main (String args[]) {
        System.out.println("Masukan nilai suatu
bilangan");
        Scanner baca = new Scanner(System.in);
        Int k = baca.nextInt( );
        System.out,println(k+ "merupakan bilangan "
                                +kembali.check(k));
    }

    Public static String check(int k){
        if(k%2==0){
            return "genap";
        }
        else{return "ganjil";}
    }
}

```

**D. LATIHAN**

3. Buatlah program yang mencari suatu nilai dalam array, apabila nilai ada maka akan ditunjukkan dimana indexnya apabila tidak ada maka akan mengatakan bahwa tidak ditemukan.

Hint :

Inisialisasi array → `int[] array = {10, 20, 25, 30, 45};`

Element yang ditunjuk → jika `array[0]` maka nilainya 10

Panjang array → `array.length`

**E. TUGAS**

Carilah kodingan yang menggunakan `break` dan kodingan yang menggunakan `continue`, kemudian analisis kegunaan `break` atau `continue` dalam kodingan tersebut.

Kumpulkan dalam format laporan resmi.

## PRAKTIKUM

## 9

# Fungsi / Method

**A. TUJUAN PEMBELAJARAN**

1. Memahami penulisan fungsi/method yang benar
2. Dapat membuat fungsi tanpa parameter
3. Dapat membuat fungsi dengan parameter

**B. DASAR TEORI**

Fungsi/method adalah kumpulan beberapa pernyataan yang dikelompokkan untuk melakukan suatu operasi. Standar internasional penamaan suatu fungsi adalah dengan menggunakan huruf awalan kecil, apabila method terdiri dari 2 kata atau lebih maka kata lain dimulai dengan huruf besar. Contoh :

```
make()
getValue()
isEmpty()
convertToMetre()
```

Penulisan fungsi/method terdiri dari 6 bagian, yaitu :

1. Modifier
2. Return Type
3. Nama Method
4. Parameter
5. Exception //didiskusikan nanti
6. Statement/Pernyataan (Tubuh method)

Contoh fungsi/method :

```
Public static boolean isBigger (int a, int b){
    if(a > b){
        return true;
    }
    else { return false;}
}
```

Public static merupakan modifier, boolean adalah return type nya, isBigger merupakan nama method, int a, int b merupakan parameternya, dan perintah/statement dalam kurung kurawal awal hingga akhir adalah tubuhnya.

## C. PERCOBAAN

### 1. Method tak berparameter

```
Class Method {  
    private static void methodTakBerparameter( ){  
        System.out.println("Ini method tak  
berparameter");  
    }  
    public static void main (String args[]) {  
        System.out.println("Menjalankan method");  
        methodTakBerparameter( );  
    }  
}
```

### 2. Method berparameter

```
Class MethodBerparameter {  
    private static void methodBerparameter(String s){  
        System.out.println("Hallo "+s);  
    }  
    public static void main (String args[]) {  
        String nama = "Fernando";  
        methodBerparameter(nama);  
    }  
}
```

### 3. Method dengan return

```
public class return{  
    public static boolean lebihBesar(int a, int b) {  
        if(a > b){  
            return true;  
        }  
        else { return false; }  
    }  
    public static void main(String[] args) {  
        int angka1 = 24;  
        int angka2 = 12;  
        System.out.println("Apakah angka 1 lebih besar  
dari angka 2? "  
+lebihBesar(angka1,  
angka2);  
    }  
}
```



**4. Method berulang**

```
class methodberulang {
    public static int fibonacci(int a) {
        if(a==1 || a==2){
            return 1;
        }
        else {
            return (fibonacci(a-1)+fibonacci(a-2));
        }
    }
    public static void main(String[] args) {
        int sukuKe = 5;
        System.out.println("Suku ke "+sukuKe+" dari deret
                           fibonacci adalah
                           "+fibonacci(sukuKe));
    }
}
```

**D. LATIHAN**

4. Buatlah program dengan method `volumeTabung(int r, int t)` dimana dia akan mengembalikan (return) volume tabung dengan r sebagai jari jari dan t sebagai tinggi. Kemudian dalam main program, program meminta inputan jari-jari dan tinggi.

**E. TUGAS**

1. Buatlah program dengan berbagai method, inti dari program ini adalah menyelesaikan tabung, mulai dari luas alas, keliling alas, luas tabung, dan volume tabung. Gunakan method-method untuk menyelesaikannya. Dalam fungsi main, akan diminta inputan berupa jari-jari dan tinggi.
2. Buatlah program dengan method tak berparameter.

Kumpulkan dalam format laporan resmi.

# Array

## PRAKTIKUM 10

### A. TUJUAN PEMBELAJARAN

1. Memahami penulisan array yang benar
2. Dapat membuat array dengan benar
3. Dapat menggunakan method standart array dengan benar.

### B. DASAR TEORI

Array adalah suatu kumpulan data yang bertipe data sama pada suatu variabel.

Pendeklarasian array :

```
tipe_array  nama_array[];  
tipe_array[]  nama_array;  
contoh :
```

```
int[] angka;  
String nama[];
```

Array diatas tidak memiliki simpanan memory, sehingga masih array kosongan yang tidak dapat di isi. Oleh karena itu, untuk mendekalrasikannya agar dapat memiliki tempat di memory, dilakukan

```
nama_array = new tipe_array[besar_array];
```

contoh :

```
int[] angka;  
angka = new int[5];
```

atau

```
int[] angka = new int[5];
```

## C. PERCOBAAN

### 1. Mengisi array

```
Class isiArray{
    public static void main (String args[]) {
        int[] angka = {15, 25, 30};
        for( int i : angka ){
            System.out.print(i+" ");
        }
    }
}
```

```
Class isiArray2{
    public static void main (String args[]) {
        int[] angka = new int[3];
        angka = {15, 25, 30};
        for( int i : angka ){
            System.out.print(i+" ");
        }
    }
}
```

```
Class isiArray3{
    public static void main (String args[]) {
        String[] nama = new String[3];
        nama[0] = "Jhonny";
        nama[1] = "Nando";
        nama[2] = "Queren";
        for( int i : nama ){
            System.out.print(i+" ");
        }
    }
}
```

### 2. Panjang Array

```
Class panjangArray {
    public static void main (String args[]) {
        System.out.println("Masukkan panjang array");
        Scanner baca = new Scanner(System.in);
        int panjang = baca.nextInt();
        int[] angka = new int[panjang];
        for(int i = 0; i < angka.length; i++){
            System.out.println("Masukkan elemen ke "+i);
            angka[i] = baca.nextInt();
        }
        System.out.print("Array = ");
        for( int i : angka ){
```

```
        System.out.print(i+" ");
    }
}
}
```

### 3. Copy array

```
public class copyArray{
    public static boolean lebihBesar(int a, int b) {
        if(a > b){
            return true;
        }
        else { return false; }
    }
    public static void main(String[] args) {
        int[] array1 = {7,4,1,3,6,4,2};
        int[] array2 = new int[3];
        System.arraycopy(array1,0,array2,0,3);
        System.out.print("Array1 : ")
        for ( int i : array1){
            System.out.println(i+" ");
        }
        System.out.print("Array2 : ")
        for ( int i : array2){
            System.out.println(i+" ");
        }
    }
}
```

### 4. Daftar Args

```
class daftarArgs {
    public static void main(String[] args) {
        System.out.println("Hari "+args[0]);
        System.out.println("Bulan "+args[1]);
        System.out.println("Tahun "+args[2]);
    }
}
```