

ELEMEN BAHASA JAVA



Akmal, S.Si, MT

Mata Kuliah : Pemrograman Berorientasi Objek

Tujuan

- ❑ Mahasiswa akan dapat menjelaskan mengenai struktur bahasa java yang digunakan dan tata bahasanya
- ❑ Mahasiswa mengerti bagaimana membuat program sederhana dengan bahasa java
- ❑ Mahasiswa memahami alur runtunan dan bisa menggunakan dasar pemrograman untuk operasi input dan output

Pokok Bahasan

- ❑ Sejarah Bahasa Java
- ❑ Aplikasi Java
- ❑ Struktur program
- ❑ Meng-compile program
- ❑ Keyword
- ❑ Tipe data
- ❑ Operator
- ❑ Alur Input-Proses-Output

Sejarah Bahasa JAVA

- ❑ Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumen seperti *cable TV Box*.
- ❑ Dikarenakan perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat.
- ❑ Juga karena manufaktur-manufaktur berbeda memilih *processor* yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun.
- ❑ Proyek diberi nama kode "Green".

Sejarah Bahasa JAVA

- ❑ Mulanya bahasa yang diciptakan diberi nama "Oak" oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada di seberang kantornya,
- ❑ Namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA.

Fitur Penting

Java telah mengakomodasi hampir seluruh fitur penting bahasa-bahasa pemrograman yang ada semenjak perkembangan komputasi modern manusia :

1. Dari SIMULA, bahasa pada tahun 65-an, bahasa yang paling mempengaruhi Java sekaligus C++. Dari bahasa ini diadopsi bentukan-bentukan dasar dari pemrograman berorientasi objek.
2. Dari LISP – bahasa tahun 55-an. Diadopsi fasilitas *garbage collection*, serta kemampuan untuk meniru *generic list processing*, meski fasilitas ini jarang yang memanfaatkannya.
3. Dari Algol – bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
4. Dari C++, diadopsi sintaks, sebagian semantiks dan *exception handling*
5. Dari bahasa Ada, diambil *strongly type*, dan *exception handling*.
6. Dari Objective C, diambil fasilitas interface.
7. Dari bahasa SmallTalk, diambil pendekatan *single-root class hiérarchie*, dimana object adalah satu kesatuan hirarki pewarisan
8. Dari bahasa Eiffel, fasilitas *assertion* yang mulai diterapkan di sebagian JDK 1.4



Karakteristik JAVA

1. Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.

3. Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries* networking yang terintegrasi pada Java.

4. Interpreter

Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM).

5. Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain.

6. Aman

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

Karakteristik JAVA

7. Architecture Neutral

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan *Java Virtual Machine*.

8. Portabel

Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

9. Performance

Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers* (JIT).

10. Multithreaded

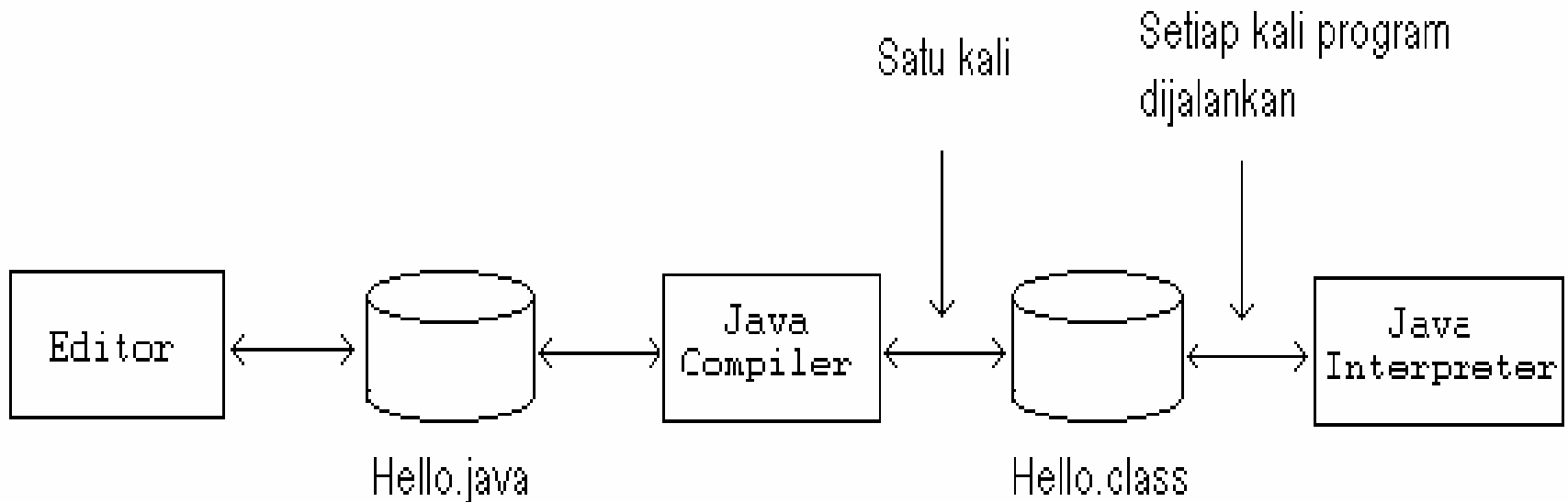
Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

ELEMEN BAHASA JAVA

Langkah-langkah membuat sebuah program Java :



▣ **Compile** : `javac Hello.java`

▣ **Run** : `java Hello`

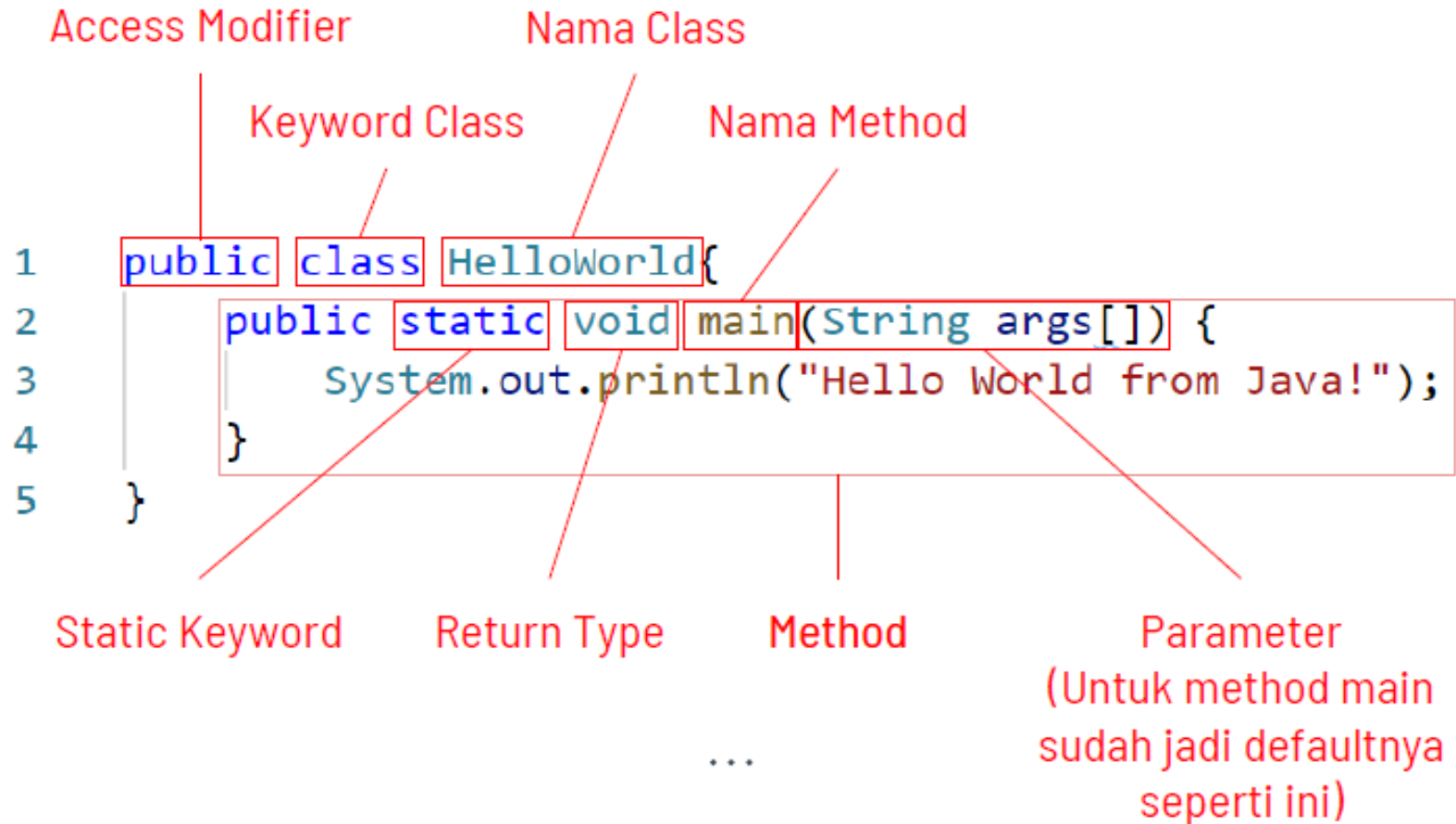
▣ -----

■ **Buat j.bat (batch file)**

`javac %1.java`

`java %1`

Struktur Bahasa Java



The diagram illustrates the structure of a Java class and its main method. The code is as follows:

```
1 public class HelloWorld{  
2     public static void main(String args[]) {  
3         System.out.println("Hello World from Java!");  
4     }  
5 }
```

Annotations and their corresponding code elements:

- Access Modifier**: points to `public` on line 1.
- Keyword Class**: points to `class` on line 1.
- Nama Class**: points to `HelloWorld` on line 1.
- Static Keyword**: points to `static` on line 2.
- Return Type**: points to `void` on line 2.
- Method**: points to `main` on line 2.
- Parameter** (Untuk method main sudah jadi defaultnya seperti ini): points to `(String args[])` on line 2.

Ellipses (`...`) are shown below the parameter annotation.

Style 1 penulisan program Java

```
/* Nama program   : HelloWorld.java
   Nama           : Akmal
   NPM            :
   Tanggal buat   :
   Deskripsi      :

*****/

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    } //End of main
} //End of HelloWorld Class
```

- ❑ Program Java yang dibuat harus selalu diakhiri dengan ekstensi file **.java**.
- ❑ Nama File harus sesuai/sama dengan nama class public nya. Sebagai contoh, jika nama class public adalah **HelloWorld**, maka harus disimpan file tersebut dengan nama **HelloWorld.java**.

Style 2 penulisan program Java

```
/* Nama program   :  
   Nama           : Akmal  
   NPM            :  
   Tanggal buat   :  
   Deskripsi       :  
*****/  
  
public class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello World");  
    } //End of main  
} //End of HelloWorld Class
```

Komentar

- ❑ Komentar adalah catatan yang ditulis pada kode dengan tujuan sebagai bahan dokumentasi. Teks tersebut bukan bagian dari program dan tidak mempengaruhi jalannya program.
- ❑ Java mendukung jenis komentar :
 - C++ style komentar satu baris,
Awalan : `//` komentar
 - C style beberapa baris
Pasangan : `/*` komentar `*/`

Identifier (Pengenal)

- ❑ Java Identifier adalah suatu tanda yang mewakili nama-nama variabel, method, class, dsb. Contoh dari Identifier adalah : Hello, main, System, out.
- ❑ Pendeklarasian Java adalah case-sensitive. Hal ini berarti bahwa Identifier : **H**ello tidak sama dengan **h**ello. Identifier harus dimulai dengan salah satu huruf, underscore “_”, atau tanda dollar “\$”.

Petunjuk :

- ❑ Untuk pemberian nama dari class Java, diberikan huruf kapital untuk huruf pertama pada nama class. Untuk nama method dan variabel, huruf pertama dari kata harus dimulai dengan huruf kecil.

Sebagai contoh:

ThisIsAnExampleOfClassName

thisIsAnExampleOfMethodName

- ❑ Pada kasus untuk identifier lebih dari satu kata, menggunakan huruf kapital untuk mengindikasikan awal dari kata kecuali kata pertama. Sebagai contoh, charArray, fileNumber, ClassName.
- ❑ Hindari menggunakan underscores pada awal identifier seperti _read atau _write.

GAYA PENULISAN

- ❑ Gaya penulisan case (case style) yang digunakan oleh Java adalah: **camelCase**, **PascalCase**, dan **ALL UPPER**.
 - *camelCase* digunakan pada nama variabel, nama objek, dan nama method.
 - *PascalCase* digunakan pada penulisan nama class
 - *ALL UPPER* atau semua huruf kapital digunakan pada pembuatan nama konstanta. Untuk penulisan dua suku kata atau lebih, *ALL UPPER* dipisah dengan garis bawah atau *underscore* ()

Keyword (Kata kunci)

- Kata yang bermakna khusus yang tidak bisa digunakan sbg pengenal

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

* not used

** added in 1.2

*** added in 1.4

**** added in 5.0

Type Data

- Java mendefinisikan delapan tipe data primitive :

Tipe data	Byte	Batasan
char	1	Karakter ASCII antara 0 s.d. 255
byte	1	Karakter ASCII antara - 128 s.d. 127 Bilangan bulat antara -128 s.d. 127
short	2	Bilangan bulat antara -32.768 s.d. 32.767 (-2^{15} s.d $2^{15}-1$)
int	4	Bilangan bulat antara -2.147.483.648 s.d. 2.147.483.647 (-2^{31} s.d $2^{31}-1$)
long	8	-2^{63} s.d 2^{63}
float	4	Bilangan real antara - 3.4 E+38 s.d. 3.4E+38 (7 digit presisi)
double	8	Bilangan real antara -1.7E+308 s.d. 1.7E+308 (15 digit presisi)
boolean	1	true dan false

Variabel

- ❑ **Variabel** adalah item yang digunakan data untuk menyimpan pernyataan objek dengan nilai yang bisa berubah.
- ❑ Variabel memiliki **tipe data** dan **nama**. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri.
- ❑ **Nama variabel** harus mengikuti aturan untuk identifier.

deklarasi

type namaVar [=nilaiAwal], ...

Contoh :

double nilai=0.0;

*// **jangan** double a=0.0;*

int num = 10;

String name = "Hello"

Petunjuk:

1. *Sangat baik untuk menginisialisasi variabel yang dibuat*
2. *Gunakan nama yang bersifat menggambarkan deskriptif untuk variabel yang dibuat, Misalkan jika ingin mempunyai variabel yang terdiri atas nilai siswa, beri nama dengan nama **grade / nilai** dan jangan hanya beberapa huruf random*

Konstanta

final type nama = nilai_konst;

Contoh :

final double phi=3.1415;

Jenis Konstanta	Contoh
Konstanta desimal	x = 10;
Konstanta hexadesimal	x = 0x1B; x = 0x10;
Konstanta Octal	x = 015;
Konstanta Float	x = 20e6; x = 6.5536E-6;
Konstanta Long	x = 30000L; x = 30000l;
Konstanta karakter	x = 'c';

Operator Aritmatika & Assignment

Simbol	Fungsi	Contoh Penggunaan
-	Pengurangan	x = x - 10;
+	Penjumlahan	x = x + 10;
/	Pembagian	x = x / 10;
*	Perkalian	x = x * 10;
%	Modulo	x = 11 % 2;
++	Increment	x++
--	Decrement	x--

```
System.out.println("13 % 5 = "+(13%5));
```

Simbol	Keterangan
-=	x = x - y dapat ditulis x -= y
+=	x = x + y dapat ditulis x += y
/=	x = x / y dapat ditulis x /= y
*=	x = x * y dapat ditulis x *= y
%=	x = x % y dapat ditulis x %= y

```
int nilai=50;
nilai +=20;
System.out.println("nilai = "+nilai);
// hasil nilai=70
```

Operator Relasi dan Logika

Simbol	Keterangan
==	Equal (sama dengan)
!=	Not Equal (tidak sama dengan)
<	Less than (lebih kecil)
<=	Less than or equal (lebih kecil atau sama dengan)
>	Greater than (lebih besar)
>=	Greater than or equal (lebih besar atau sama)

```
int nilai = 100;  
if (nilai==100)  
    System.out.println("nilai sempurna ");
```

Simbol	Keterangan
&&	AND
	OR
!	Not

```
int nilai = 70;  
if (nilai>=60 && nilai <=100)  
    System.out.println("Lulus ");
```

Operator Bitwise dan Shift

Simbol	Keterangan
&	AND
	OR
^	XOR
~	Complement
>>	Shift Right
<<	Shift Left

```
int a = 13;           // biner : 1101
int b = 6;            // biner : 0110
System.out.println("a & b =" + (a & b)); // biner : 0100 = 4
```

Bentuk umum dari operator shift :

Shift Right (Geser Kanan) :

variabel >> nomor posisi bit

Shift Left (Geser Kiri):

variable << nomor posisi bit

```
int a = 13;           // biner : 1101
System.out.println("a << 1 =" + (a << 1)); // biner : 11010 = 26
```

Operator Kondisi

Kondisi ? ekspresi1 : ekspresi2;

Jika kondisi dipenuhi atau true, maka ekspresi1 akan dijalankan, tetapi bila sebaliknya maka akan dijalankan ekspresi2.

Contoh :

```
public class ConditionalOperator{  
    public static void main( String[] args ){  
        String status = "";  
        int grade = 80;  
        //mendapatkan status pelajar  
        status = (grade >= 60)? "Passed" : "Fail";  
        //print status  
        System.out.println( status );  
    }  
}
```

Hasil keluaran dari program ini akan menjadi,
Passed

Type casting (Pengarah type)

Deklarasi

(typeData) var;

Ubah ke yang lebih besar ukurannya

Contoh :

```
int jumlah=10,ndata=3;
```

```
float rata;
```

```
System.out.println((jumlah/ndata);
```

```
// hasil = 3
```

```
rata= (float) jumlah / nData;
```

```
// hasil = 3.333
```

Hati-hati :

•**Widening Casting** (automatically) - converting dari kecil ke besar

byte -> short -> char -> int -> long -> float -> double

•**Narrowing Casting** (manually) - converting dari besar ke yang lebih kecil

double -> float -> long -> int -> char -> short -> byte

INPUT dan OUTPUT

Input dengan BufferedReader

```
public class InputReader {  
    public static void main(String[] args) {  
        BufferedReader dataIn = new BufferedReader(new InputStreamReader( System.in) );  
        String name = "";  
        String input="";  
        int nilai=0;  
        try{  
            System.out.print("Masukkan Nama anda : ");  
            name = dataIn.readLine();  
            System.out.print("Masukkan Nilai    : ");  
            input= dataIn.readLine();  
  
            nilai = Integer.valueOf(input).intValue();  
  
        }  
        catch( IOException e ){  
            System.out.println("Error!");  
        }  
  
        System.out.println("Hello " + name + " !");  
        System.out.println("Nilai = " + nilai + " !");  
    }  
}
```

Input dengan Scanner

```
public class InputScanner {  
    public static void main(String[] args) {  
  
        Scanner input=new Scanner (System.in);  
  
        System.out.println("Selamat Belajar Java, input dari keyboard....");  
  
        System.out.print("Masukkan nama anda (string) : ");  
        String str = input.nextLine();  
  
        System.out.print("Masukkan nilai (integer)   : ");  
        int x = input.nextInt();  
  
        System.out.print("Masukkan nilai IPK (float) : ");  
        float y = input.nextFloat();  
  
        System.out.println("Yang diinput adalah : ");  
  
        System.out.println("Nama   : "+ str);  
        System.out.println("Nilai  : "+ x);  
        System.out.println("IPK    : "+ y);  
  
    }  
}
```

Output (Keluaran)

- Untuk menampilkan output ke layar bisa dengan menggunakan perintah ***System.out.println()*** atau ***System.out.print()***
- Apa yang membedakan diantara perintah `System.out.println()` and `System.out.print()`?
- Yang pertama menambahkan baris baru pada akhir data untuk dikeluarkan, sementara selanjutnya tidak menambahkan baris baru.

Perhatikan pernyataan tersebut,

```
System.out.print("Hello ");  
System.out.print("world!");
```

Pernyataan tersebut akan menghasilkan output berikut ini pada layar,

```
Hello world!
```

Sekarang perhatikan pernyataan berikut,

```
System.out.println("Hello ");  
System.out.println("world!");
```

Pernyataan ini akan menghasilkan output sebagai berikut pada layar,

```
Hello  
world!
```

Latihan 1

a. Hello World!

Menggunakan NetBeans, membuat class dengan nama : [NamaAnda].

Hasil dari program yang harus tampil di layar :

Selamat Datang di Dunia Pemrograman Java [NamaAnda]!!!

b. Mendapatkan nilai rata-rata dari tiga angka

- Buatlah program yang menghasilkan output nilai rata-rata dari tiga angka. Nilai dari masing-masing tiga angka tersebut adalah 10, 20 dan 45. Tampilan Output yang diharapkan adalah,

number 1 = 10

number 2 = 20

number 3 = 45

Rata-rata = 25

c. Menampilkan nilai terbesar

- Diberikan tiga angka, tuliskan program yang menghasilkan output angka dengan nilai terbesar diantara tiga angka tersebut. Gunakan operator kondisi ?: yang telah dipelajari sebelumnya (**PETUNJUK:** Anda akan perlu menggunakan dua set operator ?: untuk memecahkan permasalahan ini).

Sebagai contoh, diberikan angka 10, 23 dan 5,

Program akan menghasilkan output :

number 1 = 10

number 2 = 23

number 3 = 5

Nilai tertinggi adalah angka = 23

Latihan 2

- ❑ Buatlah program untuk mencari keliling persegi panjang dengan rumus

- $\text{Keliling} = 2 * \text{panjang} + 2 * \text{lebar}$

Atau

- $\text{Keliling} = 2 * (\text{panjang} + \text{lebar})$

- Gunakan input keyboard (scanner)

- Tampilan :

- ❑ Masukkan panjang : 5
 - ❑ Masukkan lebar : 2
 - ❑ Keliling Persegi panjang : 14

Latihan 3

□ Diketahui $a=53$, $b=19$ (format 8 bit)

Tentukan hasil

$a \& b$,

$a | b$,

$a \wedge b$,

$a \ll 2$,

$b \gg 1$

Pengertian

- Java Card adalah teknologi Java yang digunakan pada peralatan elektronik yang memiliki memori yang sangat terbatas.
- J2ME (Java 2 Platform, Micro Edition) adalah teknologi Java edisi mikro yang digunakan untuk penerapan teknologi Java pada peralatan elektronik.
- J2SE (Java 2 Platform, Standard Edition) adalah teknologi Java edisi standar yang digunakan untuk penerapan teknologi Java pada computer desktop.
- J2EE (Java 2 Platform, Enterprise Edition) adalah teknologi Java edisi enterprise yang digunakan untuk penerapan teknologi Java pada computer server.
- Java API, Java Application Programming Interface adalah yang terdiri atas kumpulan library yang digunakan untuk keperluan pemrograman.
- JRE, Java Run Time Environment adalah lingkungan yang membuat aplikasi Java yang dapat dijalankan. Komponen penting pada JRE adalah JVM atau Java Virtual Machine.
- JVM adalah mesin Virtual berupa aplikasi yang digunakan untuk menjalankan kode Java (Byte Code).
- Integrated Development Environment (IDE) Java adalah aplikasi yang digunakan khusus untuk mengembangkan program Java.
- Java Development Kit (JDK) merupakan seperangkat aplikasi yang digunakan untuk mengembangkan program dengan Bahasa Java.
- NetBeans adalah IDE yang telah banyak dipakai untuk mengembangkan program dalam bahasa Java.
- Applet merupakan aplikasi java yang dapat diintergrasikan dalam halaman Web dan aplikasi ini dapat berjalan diatas aplikasi web browser.

Pengertian

- ❑ Identifier adalah nama yang digunakan untuk mengidentifikasikan elemen program.
- ❑ Keyword atau kata kunci merupakan kata yang mempunyai arti khusus sehingga tidak dapat menggunakan Identifier.
- ❑ Tipe data Primitif adalah tipe data yang mempunyai nilai tertentu bukan referensi class ataupun objek.
- ❑ Tipe data Boolean adalah tipe data yang digunakan untuk menentukan suatu kondisi apakah benar atau salah. Nilai Boolean sering digunakan untuk mengatur alur program terutama pada perulangan dan percabangan.
- ❑ Floating Point adalah bilangan rasional. Pada penulisannya menggunakan tanda titik sebagai tanda decimal atau bisa juga menggunakan tanda Eksponensial e atau E.
- ❑ Variabel digunakan untuk menyimpan data sehingga dapat diolah oleh program.
- ❑ Konstanta digunakan untuk menyimpan data yang tidak akan kita ubah.
- ❑ Type Enumerated adalah tipe data yang didefinisikan sendiri oleh programmer dan juga sering digunakan untuk membuat daftar nilai – nilai dalam jumlah terbatas.
- ❑ Operator digunakan untuk melakukan pengolahan data, biasanya melibatkan konstanta dan variable yang telah kita buat.
- ❑ Literal digunakan untuk memberi nilai pada suatu variable ataupun konstanta, Literal tidak melibatkan Operator.
- ❑ Operator Shorthand adalah operator yang digunakan dengan menyingkat penulisan, selain melakukan perhitungan operator ini juga bertugas melakukan assignment.
- ❑ Operator Perbandingan adalah operator yang digunakan untuk membandingkan antara nilai yang satu dengan nilai lainnya.
- ❑ Scope digunakan untuk membatasi sampai sejauh mana nilai suatu konstanta maupun variable masih berlaku.
- ❑ Operator Bitwise digunakan untuk melakukan operasi pada tingkat digital.
- ❑ Whitespace adalah karakter kosong yang dapat berupa tab, spasi atau enter. Berfungsi untuk merapikan kode program.
- ❑ Branching Statement digunakan untuk menginterupsi proses yang sedang berjalan.
- ❑ Break digunakan untuk menghentikan perulangan.