

# Object-Oriented Programming with Java



Ajib Susanto

[ajibsusanto@gmail.com](mailto:ajibsusanto@gmail.com)

[ajib.susanto@dsn.dinus.ac.id](mailto:ajib.susanto@dsn.dinus.ac.id)

<http://ajibsusanto.net>

@ajibsusanto / 085876247118



# Materi Diskusi

---

1. **Konsep Pemrograman dan Paradigmanya**  
Tingkat Bahasa Pemrograman, Paradigma Pemrograman, Tool Pemrograman
2. **Konsep Dasar Pemrograman Berorientasi Obyek:**  
Kelas (Class), Obyek (Object), Pengkapsulan (Encapsulation), Pewarisan (Inheritance), Polymorphisme
3. **Konsep Lanjut Pemrograman Berorientasi Obyek:**  
Constructor, Destructor, Overloading, Overriding, Keyword This, Multithreading, Package, Interface, Exception



# Konsep Pemrograman dan Paradigmanya

---



# Bahasa Pemrograman?

---

- Komputer bekerja seperti **switching** dan hanya mengenali 0 dan 1
- Manusia **tidak (paham) berbicara** dengan bahasa 0 dan 1
- Perlu bahasa pemrograman yang dapat menjadi **perantara percakapan** antara komputer dan manusia
- Bahasa pemrograman diubah ke dalam bahasa yang dipahami oleh komputer dengan menggunakan **interpreter** atau **kompiler**



# Compiler or Interpreter?

---

- **Compiler:**

Mengkompilasi source code menjadi bentuk **file yang bisa dieksekusi**

- **Interpreter:**

Mengkompilasi dan menjalankan source code **secara langsung**



# Tingkat Bahasa Pemrograman

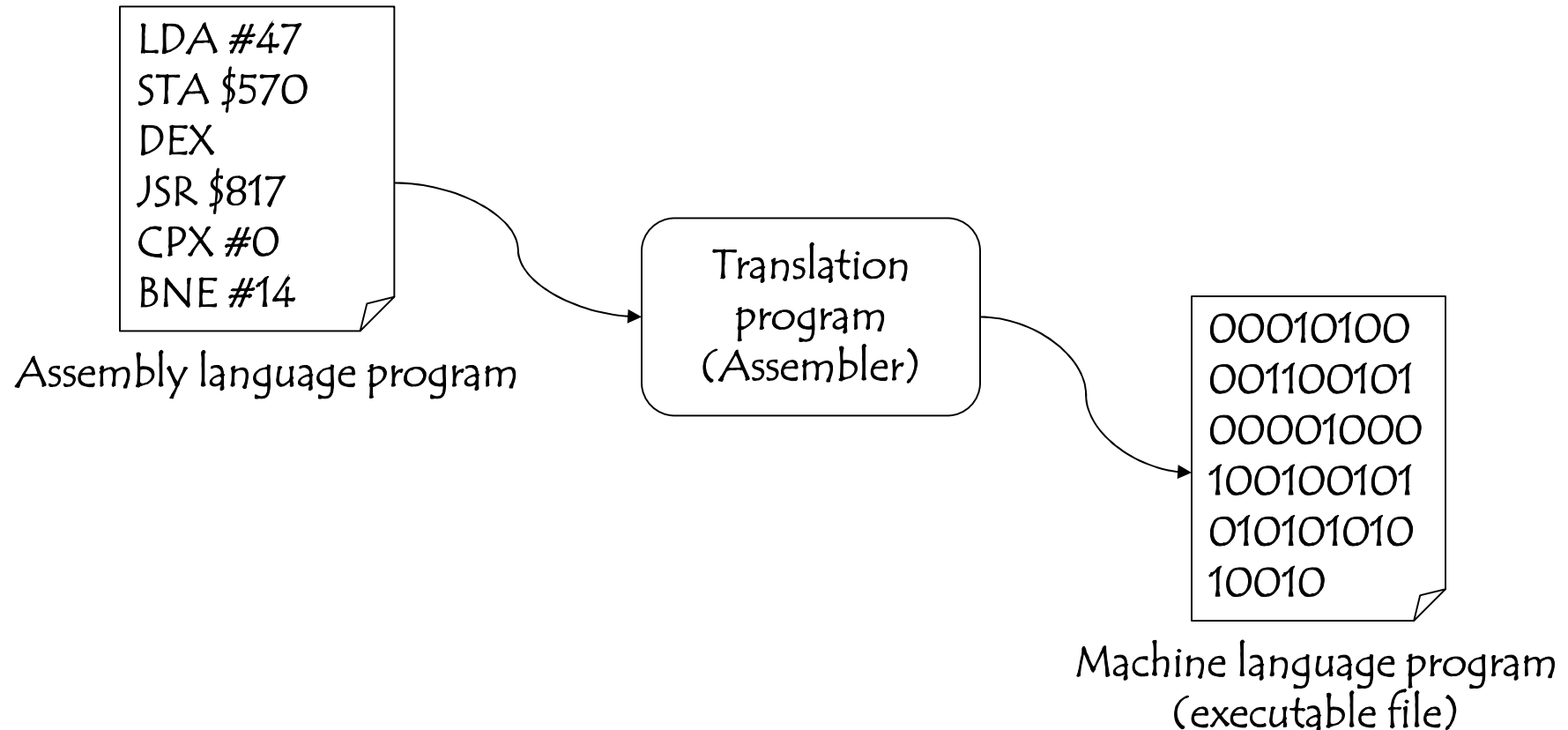
---

- Bahasa Pemrograman Tingkat Rendah (Assembler)
- Bahasa Pemrograman Tingkat Sedang (C, Pascal, Fortran)
- Bahasa Pemrograman Tingkat Tinggi (Java, C++, C#)



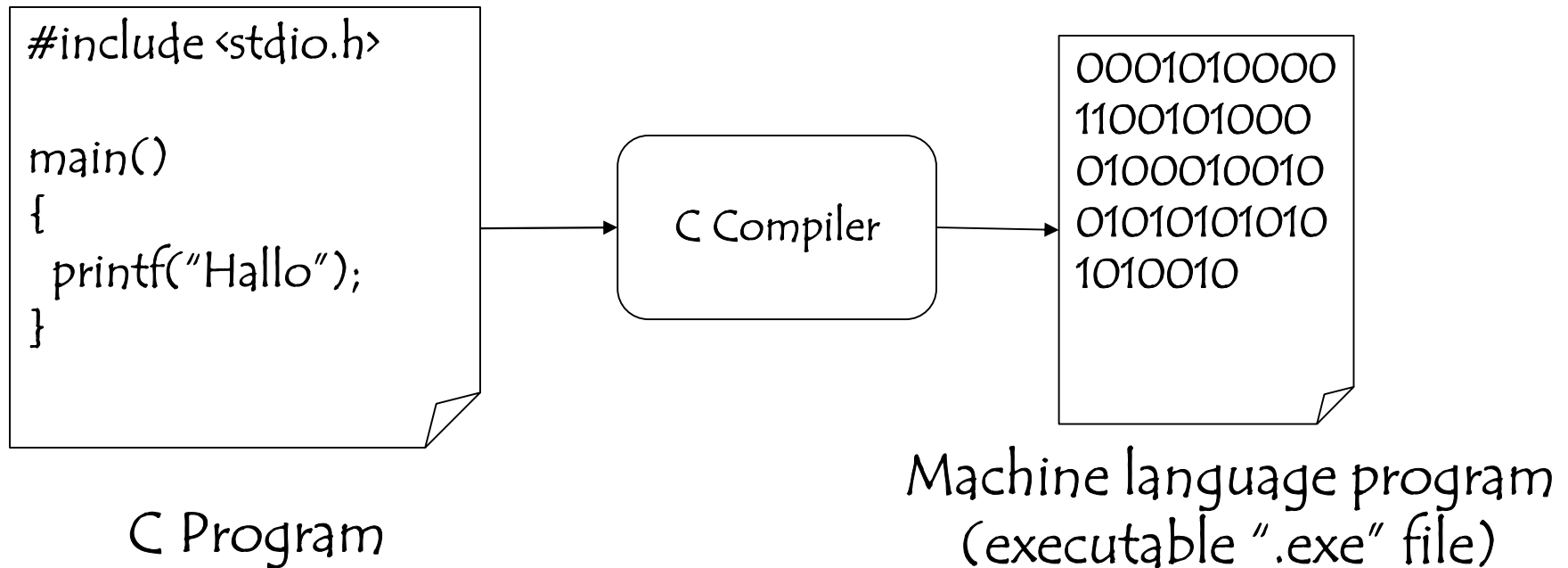
# Assembly Language

---



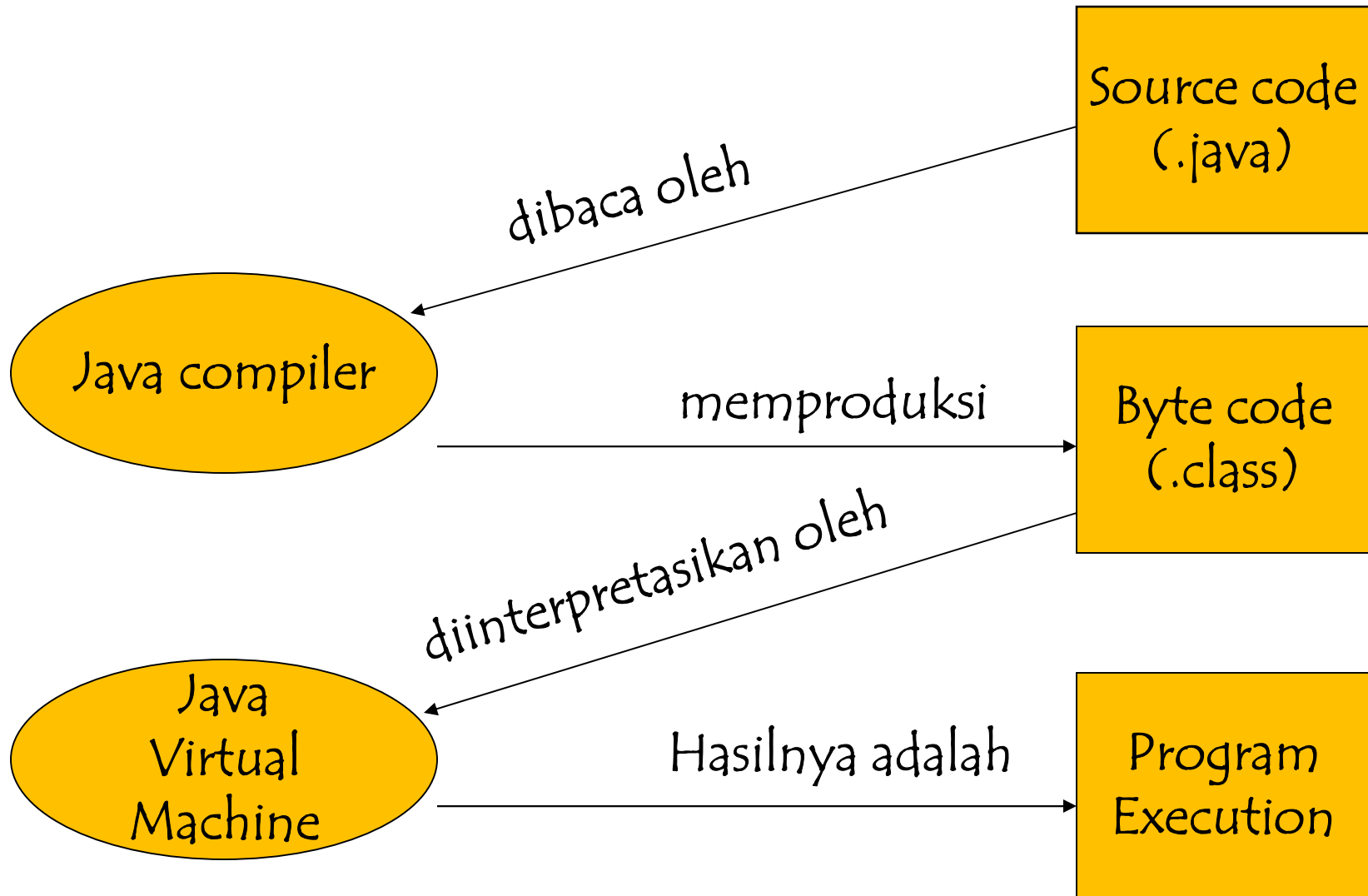
# C Language

---





# Java Language



# Paradigma Pemrograman

---

*Sudut pandang dan style pemrograman* berhubungan dengan bagaimana sebuah masalah diformulasikan dalam bahasa pemrograman

- **Functional Programming:** Urutan fungsi secara sekuensial (Scheme, Lisp)
- **Procedural Programming:** Pemecahan masalah berdasarkan prosedural kerja. Fungsi adalah unit programming (C, Pascal, Fortran)
- **Object-Oriented Programming:** Koleksi object yang saling berinteraksi . Class adalah unit programming (Java, C#, C++)



# Tool Pemrograman Java

---

1. **Text Editor + Compiler (Interpreter):**  
TextPad, Notepad++, Java Standard Edition, Vs Code
2. **Integrated Development Environment (IDE):**  
Eclipse, Netbeans, Jcreator



# Install, Compile and Running Java

---



# Java Family Suite

- ❑ Java Standard Edition  
(Java SE)

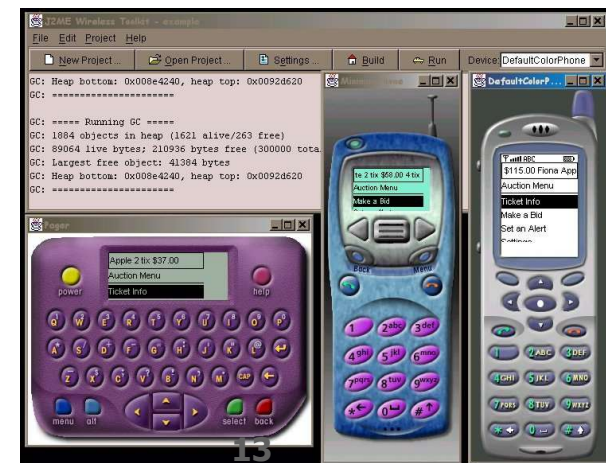
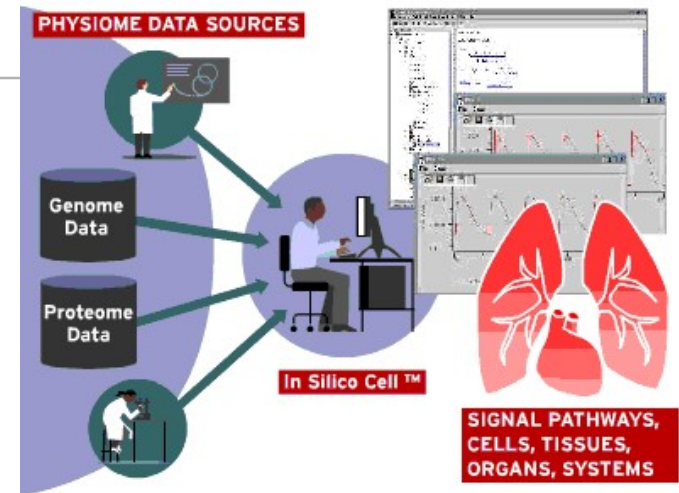
- ❑ For desktop, client/server application

- ❑ Java Enterprise Edition  
(Java EE)

- ❑ For e-bussiness, e-commerce web based application

- ❑ Java Micro Edition  
(Java ME)

- ❑ For small devices, like palm,handphone, etc



# Instalasi Java Standard Edition

---

1. Download Java Standard Edition dari:  
<https://www.oracle.com/java/technologies/java-downloads.html>
2. Instal Java Standard Edition dengan mengklik:  
`jdk-13.0.2_windows-x64_bin.exe`
3. Klik `Start` → `Control Panel` → `System` → `Advanced` → `Environment Variables` dan set PATH:  
`;C:\Program Files\Java\jdk-13.0.2\bin`
4. Instal text editor untuk editing code:  
`textpad`, `notepad++`, `Vs Code`



# Menulis Java Program

---

- Program harus punya method di public class:  
`public static void main(String[] args)`  
→ *Program Java mulai dari sini!*
- Bentuk program:
  1. **GUI application** → public class biasanya menggunakan JFrame
  2. **"Text based" Console Application** → public class adalah class yang sederhana



# Menulis Program "Hello World!"

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

```
PS D:\data\JavaMagelang> javac Hello.java  
PS D:\data\JavaMagelang> java Hello  
Hello Java  
PS D:\data\JavaMagelang> java -cp . Hello  
Hello Java  
PS D:\data\JavaMagelang>
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINNT\System...'. The command prompt shows the command 'X>java -cp . Hello' being entered, followed by the output 'Hello World!'. The prompt 'X>' is visible on the next line. The window has standard Windows controls (minimize, maximize, close) and a scroll bar on the right.

```
C:\WINNT\System...  
X>java -cp . Hello  
Hello World!  
X>
```





# GUI Version “Hello World!”

---

```
// HeloGUI.java
// Simple GUI Hello World Program

import javax.swing.*;

public class HelloGUI{
    public static void main(String[] args){
        JOptionPane.showMessageDialog(null, "Hello World");
    }
}
```



# How Java Works?

Jalankan dengan:

1. C:\>java HelloWorld (Application)
2. Web Browser (Applet)

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

**HelloWorld.java**

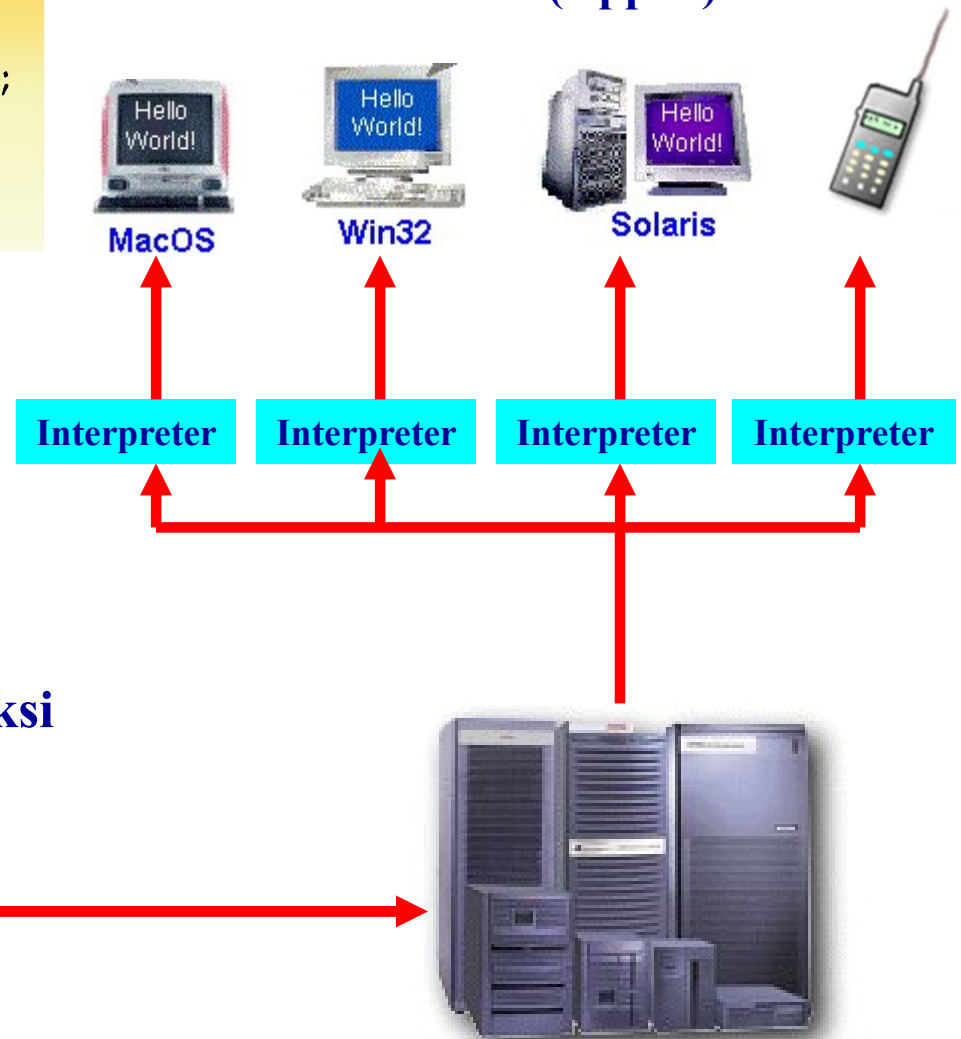


**javac compiler**



**Kompiler memproduksi  
Bytecode (Class)**

**Java bytecode  
(HelloWorld.class)**



**Write Once Run Everywhere !**

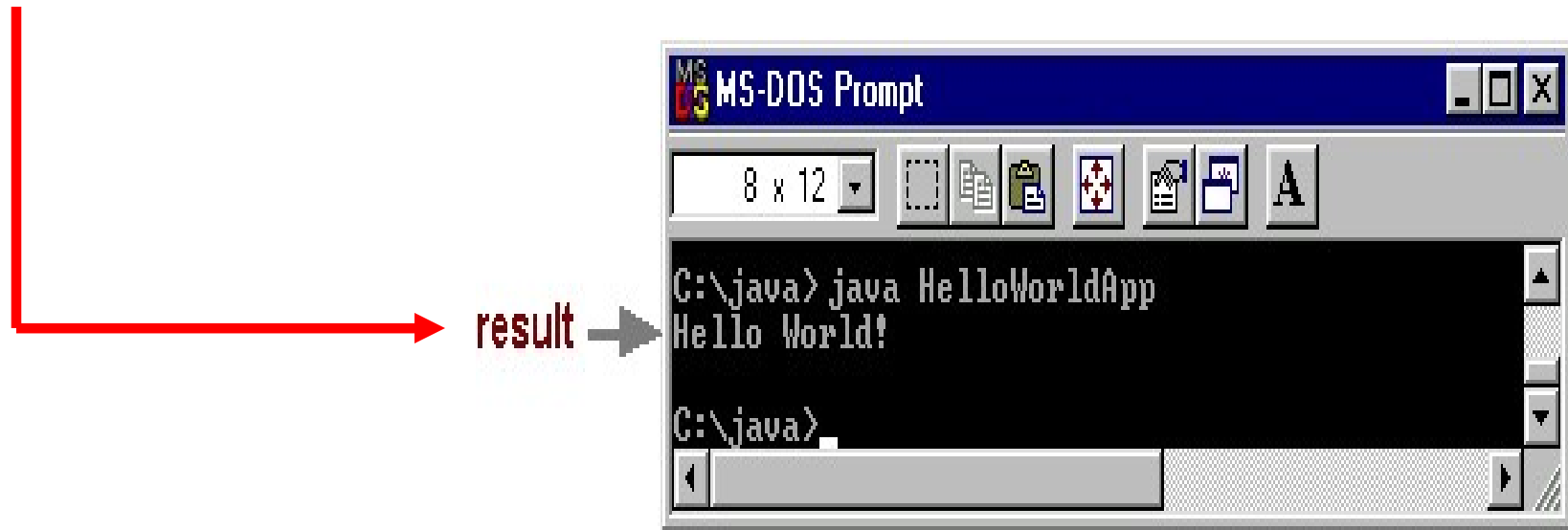
Web Server  
ajib fik udinus



# Compile and Run Java Application

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

**C:\javac HelloWorldApp.java**



# Compile and Run Java Applet

```
import java.applet.*;
import java.awt.*;

/**
 * The HelloWorld class implements an applet that
 * simply displays "Hello World!".
 */
public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        // Display "Hello World!"
        g.drawString("Hello world!", 50, 25);
    }
}
```

C:\javac HelloWorld.java



C:\appletviewer Hello.html

```
<HTML>
<HEAD>
<TITLE>A Simple Program</TITLE>
</HEAD>
<BODY>
Here is the output of my program:
<APPLET CODE="HelloWorld.class"
WIDTH=150 HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```



# NetBeans IDE

JavaApplication1 - NetBeans IDE 8.2

File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

JavaApplication1

- Source Packages
  - javaapplication1
    - JavaApplication1.java
- Libraries
  - JDK 1.8 (Default)
- Pengujian
  - Source Packages
    - pengujian
      - Kalkulator.java
      - MainTest.java
  - Test Packages
  - Libraries
    - Test Libraries
  - SeleniumWeb
  - Source Packages

Navigator

Members

JavaApplication1

- main(String[] args)

Source

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package javaapplication1;
7
8  /**
9   *
10  * @author User
11  */
12  public class JavaApplication1 {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Hello World!!!");
20      }
21
22  }
```

JavaApplication1.java - Properties

Properties

Name	JavaApplication1
Extension	java
File Size	504
Modification Time	Apr 1, 2019 1:31:02 PM
All Files	C:\Users\User\Documents\W...
Classpaths	
Compile Classpath	
Runtime Classpath	C:\Users\User\Documents\W...
Boot Classpath	C:\Program Files\Java\jdk1.8...

JavaApplication1.java

Output - JavaApplication1 (run)

run:

```
Hello World!!!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Finished building JavaApplication1 (run).

1:1 INS

# Tipe Data, Variabel & operator

---

- Java memiliki tipe data yang dapat dikategorikan menjadi dua kelompok, yaitu tipe data primitif dan referensi.
- Tipe Data Primitif :
  - Integral : merepresentasikan nilai-nilai bilangan bulat (tidak memiliki elemen pecahan desimal).
  - Floating Point : merepresentasikan nilai-nilai bilangan real ( memiliki elemen pecahan desimal )
  - Tekstual : merepresentasikan nilai-nilai berupa alphabet.
  - Logika : merepresentasikan nilai-nilai logika ( hanya bernilai *true* dan *false* ).



# Tipe Data Primitif

## Integer ( Bilangan Bulat )

Tipe Data	Panjang	Rentang Nilai	Contoh Penulisan Nilai yang Diperbolehkan
byte	8 bit	$-2^7$ sampai $2^7$ (-128 sampai 127 ) (256 kemungkinan nilai)	5 -126
short	16 bit	$-2^{15}$ sampai $2^{15}$ (-32.768 sampai 32.767) (65.535 kemungkinan nilai)	9 -23659
int	32 bit	$-2^{31}$ sampai $2^{31}$ ( -2.147.483.648 sampai 2.147.483.647) (4.294.967.296 kemungkinan nilai)	2067456397 -1456398567
long	64 bit	$-2^{63}$ sampai $2^{63}$ ( -9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807) (18.446.744.073.709.551.616 kemungkinan nilai)	3L -2147483648L 67L



# Tipe Data Primitif

---

- Tipe Data Floating Point

Tipe Data	Panjang	Contoh Penulisan Nilai yang Diperbolehkan
float	32 bit	78F -34736.86F 6.4E4F ( sama dengan $6,4 \times 10^4$ )
double	64 bit	-2356 3.5E7 67564788965.567





# Tipe Data Primitif

---

- Tipe Data Tekstual : char
  - Contoh penggunaan :

```
public char alphabet = 'A';  
public char ascii = '\111'; // jika dicetak, akan  
                           //menghasilkan  
                           // huruf 'I';
```
- Tipe Data Logika : boolean
  - Contoh penggunaan :

```
public boolean status = true;  
public boolean check = 10 < 5 ; // nilai check menjadi  
                           //false  
public boolean hasil = (10<5) && (var==3);
```



# Type Data Primitif

---

- Memilih Tipe Data :
  - Perhatikan Jenis Data ( bilangan bulat / bilangan real / logika / karakter )
  - Terdapat operasi pembagian ? → disarankan menggunakan tipe floating point
  - Jika program memperhatikan ukuran data dalam memory, gunakan tipe-tipe data dengan representasi bit kecil.



# Type Data Reference

---

- Kelebihan pemrograman berorientasi objek adalah dapat mendefinisikan tipe data baru yang merupakan objek dari class tertentu.
- Tipe data ini digunakan untuk mereferensikan objek atau class tertentu, seperti String.

```
String namaMhs;
```

```
Segitiga s = new Segitiga();
```

```
Object o;
```

```
Cat myNewCatReferenceVariable;
```

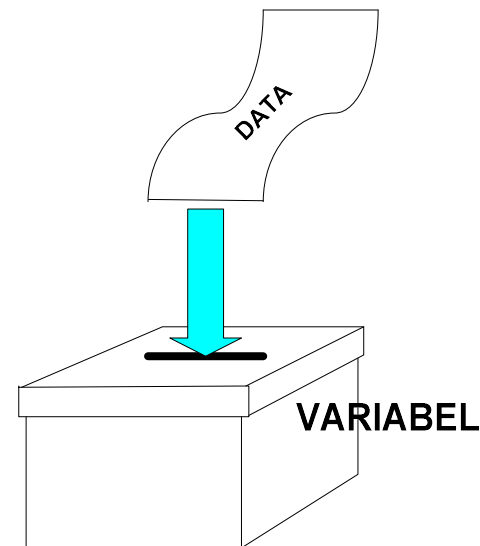
```
String s1, s2, s3;
```



# Variabel

---

- Mendeklarasikan Variabel
  - Variabel : entitas penyimpanan data yang paling elementer
  - Variabel lebih mengacu ke alokasi memory daripada nilai data



# Variabel

- Penamaan Variabel :
  - Identifier variabel harus dimulai dengan alfabet huruf besar, huruf kecil, tanda dollar ( \$ ) atau *underscore* ( \_ ). Setelah karakter pertama, dapat diikuti dengan angka.
  - Identifier variabel tidak boleh mengandung *punctuation*, spasi, atau *dashes* ( - )
  - Diawali dengan huruf kecil pada kata pertama dan huruf besar pada kata kedua dan selanjutnya.
  - Kata kunci pada teknologi Java , tidak dapat dijadikan nama identifier variabel.

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



# Variabel

---

- Penamaan Variabel :

- Contoh nama variabel yang valid :

- @2var
    - \_status
    - tanggal
    - jumlahBarang
    - nama\_kecil
    - final\_test
    - int\_float

- Melewatkan Nilai ke Variabel

- Contoh :

```
1  ...  
2      public String noPol = "D 234 LE";  
3      public int harga = 70000000;  
4  
5  ...
```



# Sun's Java Code Conventions

---

- **Classes and interfaces** The first letter should be capitalized, and if several words are linked together to form the name, the first letter of the inner words should be uppercase (a format that's sometimes called "camelCase"). For classes, the names should typically be nouns. For example:

Dog

Account

PrintWriter

For interfaces, the names should typically be adjectives like

Runnable

Serializable

- **Methods** The first letter should be lowercase, and then normal camelCase rules should be used. In addition, the names should typically be verb-noun pairs. For example:

getBalance

doCalculation

setCustomerName



# Sun's Java Code Conventions

---

- Variables Like methods, the camelCase format should be used, starting with a lowercase letter. Sun recommends short, meaningful names, which sounds good to us. Some examples:  
    buttonWidth  
    accountBalance  
    myString
- Constants Java constants are created by marking variables static and final. They should be named using uppercase letters with underscore characters as separators:  
    MIN\_HEIGHT





# Access to Member Class

Visibility	Public	Protected	Default	Private
From the same class	Yes	Yes	Yes	Yes
From any class in the same package	Yes	Yes	Yes	No
From a subclass in the same package	Yes	Yes	Yes	No
From a subclass outside the same package	Yes	Yes, <i>through inheritance</i>	No	No
From any non-subclass class outside the package	Yes	No	No	No



# Operator (1)

- Aritmatik

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Penjumlahan	+	sum=num1 + num2	
Pengurangan	-	diff=num1 - num2	
Perkalian	*	prod=num1 * num2	
Pembagian	/	quot=num1 / num2	jika num1 dan num2 adalah integer, pembagian akan menghasilkan nilai integer tanpa mengikutsertakan sisa, jika terdapat sisa.
Sisa modulus )	%	mod=num1 % num2	Hasil operasi modulus adalah sisa dari operasi num1 / num2. Hasil operasi modulus memiliki tanda ( +/- ) yang sama dengan operand pertama



# Operator (2)

- Aritmatika Unary**

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Pre-Increment	<code>++operand</code>	<code>int i = 8 ;</code> <code>int j = ++i;</code> i bernilai 8, j bernilai 8	
Post-Increment	<code>operand++</code>	<code>int i = 8;</code> <code>int j = i++;</code> i bernilai 9, j bernilai 8	
Pre-Decrement	<code>--operand</code>	<code>int i = 8 ;</code> <code>int j = --i;</code> i bernilai 7 , j bernilai 7	
Post-Decrement	<code>operand--</code>	<code>int i=8;</code> <code>int j = i--;</code> i bernilai 7, j bernilai 8	



# Operator (3)

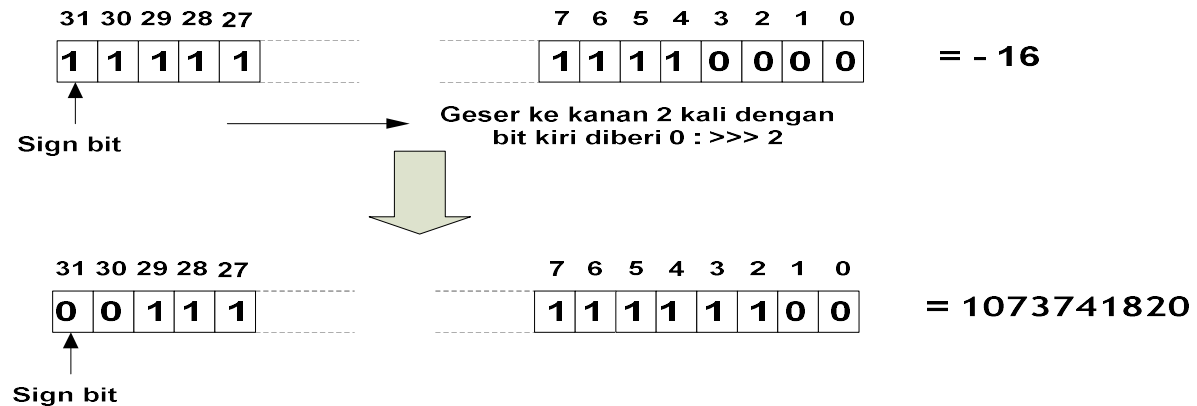
- Bitwise

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Shift Kiri	<<	int b = -16; int c = b<<2; nilai c = -64	
Shift Kanan	>>	int b = -16; int c = b>>2; nilai c = -4	Ketika digeser ke kanan 1 kali, maka bit paling kiri terisi dengan angka yang sama dengan <i>sign</i> bit sebelumnya.
Shift Kanan, dengan pengisian "0" pada bit-bit sebelah kiri	>>>	int b = -16; int c = -16>>>2; nilai c = 1073741820	Ketika digeser ke kanan 1 kali, maka bit paling kiri terisi dengan 0
AND	&	int a = 12; int b = -13; int c = a & b; nilai c = 0	Yang di-AND adalah setiap bit dari a dan b yang menempati posisi bit yang sama, misalnya bit ke-2 variabel a di-AND dengan bit ke-2 variabel b.
OR		int a = 12; int b = -13; int c = a   b; nilai c = -1	Yang di-OR adalah setiap bit dari a dan b yang menempati posisi bit yang sama
Exclusive OR	^	int a = 13; int b = -13; int c = a ^ b; nilai c = -2	Yang di-exclusive OR adalah setiap bit dari a dan b yang menempati posisi bit yang sama
Complement	~	int a = 12; int c = ~a; nilai c = -13	Nilai setiap bit diganti dengan lawannya. Jika bit bernilai 1, maka nilai tersebut akan dirubah menjadi 0

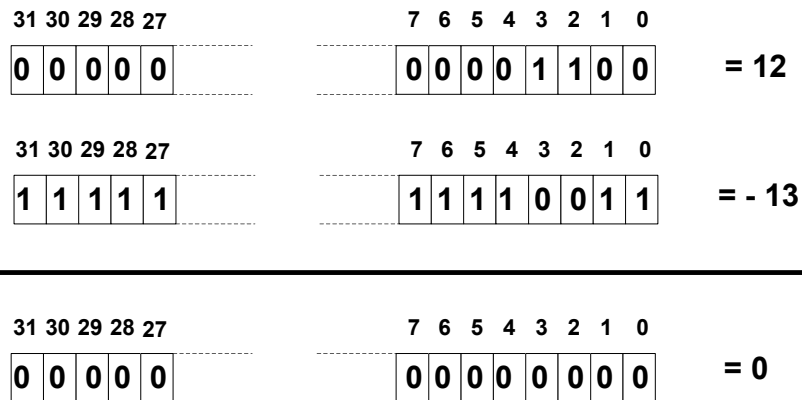


# Operator Bitwise (1)

- Shift Kanan Dengan Penambahan '0' pada Bit-bit Kiri



- Operator '&'



# Operator Bitwise (2)

- Operator '~'

31	30	29	28	27		7	6	5	4	3	2	1	0	
0	0	0	0	0		0	0	0	0	1	1	0	0	= 12

~

31	30	29	28	27		7	6	5	4	3	2	1	0	
1	1	1	1	1		1	1	1	1	0	0	1	1	= -13



# Prioritas Operator (1)

---

- Prioritas dari yang paling dahulu dikerjakan :
  - Operator pada tanda kurung “(...)”
  - Operator-operator Increment dan Decrement
  - Operator Perkalian dan Pembagian
  - Operator Penjumlahan dan Pengurangan
  - Operator Bitwise, dari kiri ke kanan



# Prioritas Operator (2)

---

- Contoh 1:

```
int c = 12 * 3 + 5 / (8 - 3) ;
```

Urutan Operasinya :

```
int c = 12 * 3 + 5 / 5 ;
```

```
int c = 36 + 5 / 5;
```

```
int c = 36 + 1 ;
```

```
int c = 37;
```

- Contoh 2:

```
int c = 3 + 4 >> 1 + 1 << 1;
```

Urutan Operasinya :

```
int c = 7 >> 1 + 1 << 1;
```

```
int c = 7 >> 2 << 1;
```

```
int c = 1 << 1;
```

```
int c = 2;
```





# Promosi dan Type-Casting(1)

---

- Promosi

- proses pengubahan representasi bit variabel primitif dari representasi bit yang lebih rendah ke representasi bit yang lebih tinggi

- Promosi karena *assigning* tipe data dengan representasi bit yang lebih tinggi ke yang lebih rendah

- Contoh :

```
short a = 12;  
int b = a ; // nilai a promosi ke integer
```

- Promosi karena *assigning* tipe data integral ke tipe data floating-point

- Contoh :

```
int a = 30;  
float b = a ;
```



# Promosi dan Type-Casting(2)

---

- Type-Casting
  - proses pengubahan representasi bit variabel primitif dari representasi bit yang lebih tinggi ke representasi bit yang lebih rendah.

- Syntax :

`identifier = (target_type) value ;`

- Contoh :

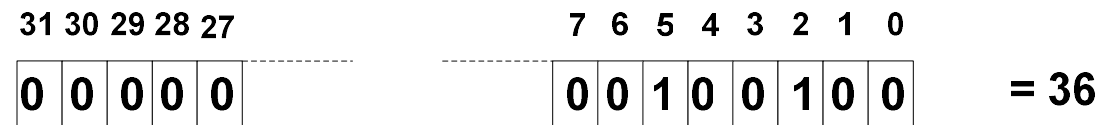
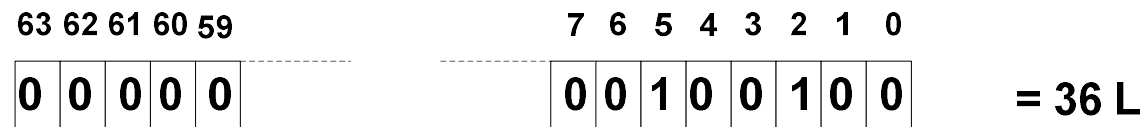
```
int num1 = 34;
int num2 = 45;
short num3 = (short)(num1 + num2 );
```

Keterangan : (num1 + num2) menghasilkan nilai *integer*. Agar nilainya dapat di-*assign* ke num3, dilakukan casting dengan pernyataan :  
(short) (num1+num2)



# Promosi dan Type-Casting(3)

- Type-Casting merupakan proses pemotongan bit
  - Misalnya : Type-casting dari *float* ke *int*



- Contoh :

```
int num1;  
long num2 = 123987654321L;  
num1 = (int)(num2);
```

num1 akan bernilai -566397263



# Beberapa Catatan Promosi & Casting (1)

---

- Operasi Aritmatika Menghasilkan Nilai di Luar Batas Tipe Data

- Contoh :

```
int a = 55555;  
int b = 66666;  
int c = a * b  
System.out.println(c);
```

Output : -591337666 ( != 55555 \* 66666 )

Solusi :

```
int a = 55555;  
long b = 66666;  
long c = a * b;  
System.out.println(c);
```

Output : 3703629630 ( = 55555 \* 66666 )



# Beberapa Catatan Promosi & Casting (2)

---

- Asumsi dasar compiler untuk tipe data integer:
  - Nilai yang di-*assign* tanpa penambahan keterangan apapun, diasumsikan sebagai nilai *integer*
  - Jika pada suatu operasi aritmatika, operand-operand pada ruas kanan berbeda tipe datanya ( semuanya masih termasuk tipe data integral ), dan semua tipe data merupakan tipe data yang representasi bitnya di bawah *integer*, maka hasil operasi aritmatika tersebut akan diasumsikan sebagai integer
  - Jika pada suatu operasi aritmatika, operand-operand pada ruas kanan berbeda tipe datanya ( semuanya masih termasuk tipe data integral ), dan salah satu tipe data merupakan tipe data yang representasi bitnya di atas *integer*, yaitu *long*, maka hasil operasi aritmatika tersebut akan diasumsikan sebagai *long*



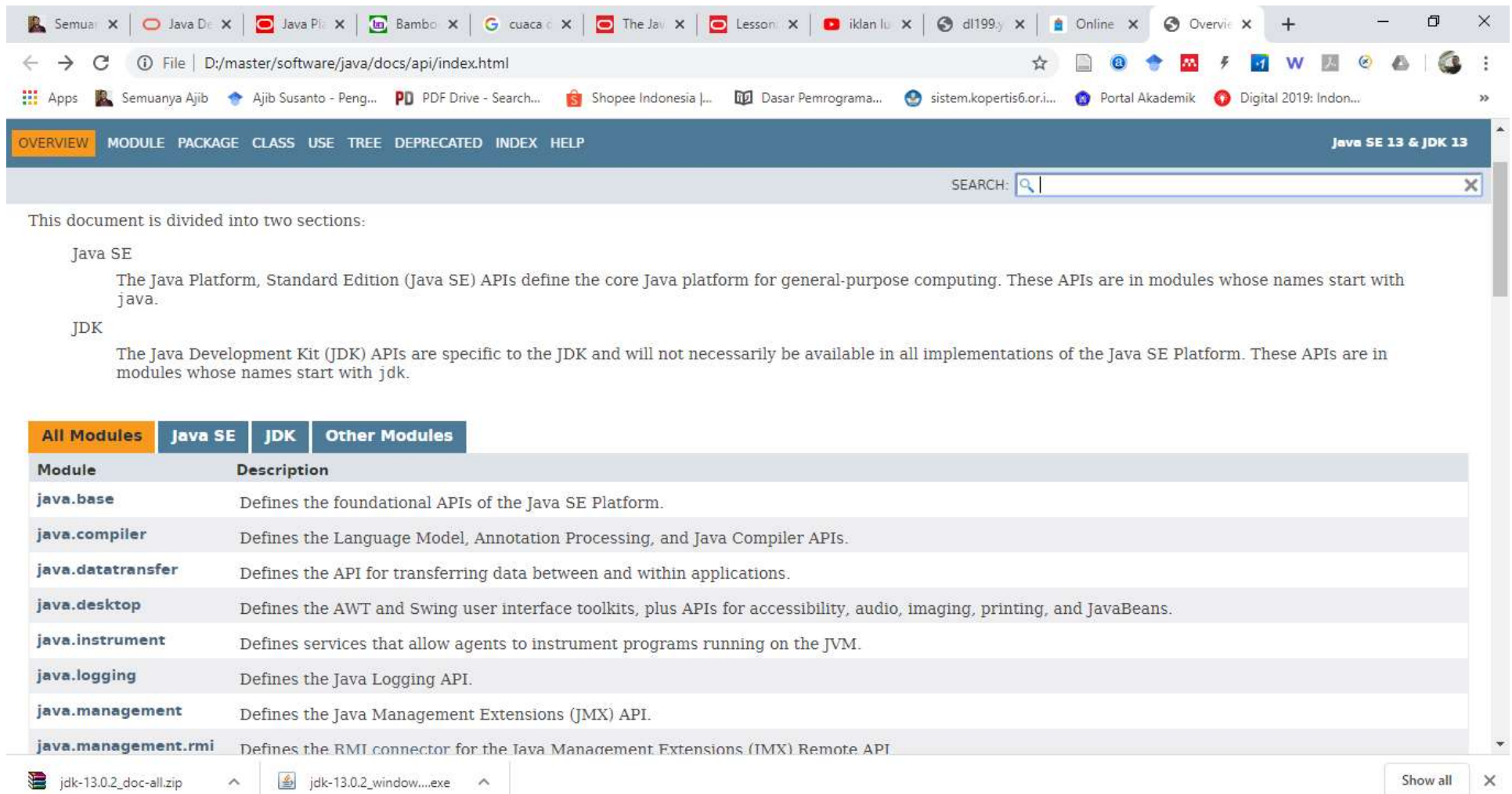
# Beberapa Catatan Promosi & Casting (3)

---

- Asumsi dasar compiler untuk tipe data floating-point:
  - Nilai yang di-*assign* tanpa penambahan keterangan apapun akan diasumsikan sebagai *double*
  - Jika pada operasi aritmatika dengan semua operand pada ruas kanan menggunakan tipe data floating-point, maka hasil operasi aritmatika tersebut direpresentasikan dengan tipe data yang mengikuti tipe data dengan representasi tertinggi pada ruas kanan operasi aritmatika tersebut
  - Jika pada operasi aritmatika dengan sebagian operand pada ruas kanan menggunakan tipe data floating-point dan sebagian menggunakan tipe data integral, maka hasil operasi aritmatika tersebut direpresentasikan dengan tipe data floating-point yang mengikuti tipe data floating-point dengan representasi bit tertinggi pada ruas kanan operasi aritmatika tersebut



# Java Doc



Overview of Java SE 13 & JDK 13 API documentation.

This document is divided into two sections:

- Java SE**  
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.
- JDK**  
The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

All Modules	Java SE	JDK	Other Modules
Module	Description		
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.		
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.		
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.		
<code>java.logging</code>	Defines the Java Logging API.		
<code>java.management</code>	Defines the Java Management Extensions (JMX) API.		
<code>java.management.rmi</code>	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		

# Ada pertanyaan

---

?





# Kontrak Nilai

---

NO	Nilai	Prosentase
1	Tugas	30% (16), 1 UTS, 1 UAS, 14 Mingguan
2	UTS	35%, Live coding
3	UAS	35%, Presentasi face to face



# Latihan

---

1. Buatlah program untuk menghitung luas :
  - Segitiga
  - Lingkaran
  - Bujursangkar
  - Empat Persegi Panjang



# Hasil Akhir Kuliah OOP


---

- Dokumentasi Program
- Java Fundamental – Oracle Academy
- Sertifikat Java
  - Bamboomedia.net → Java & Oracle



# Proyek Akhir

### LOGIN



TELEBOT PKKP JATENG

Selamat Datang & Silahkan Login

username

password

**LOGIN** **RESET**

Search

PKKP 2019 Jateng bot

February 24

/hola 7:52 AM ✓

pkkp2020jaten... 7:52 AM

Hey a71b, untuk bahan ujian...

PKKP 2020 7:51 AM

Ahmozy: Hati-hati di jalan te...

PKKP Restu 22.02.20

PKKP Restu joined Telegram

Defri Dinus 17.02.20

Defri Dinus joined Telegram

Tari Fkes 10.02.20

Tari Fkes joined Telegram

Budi Polines 7.02.20

Budi Polines joined Telegram

UPDATE TELEGRAM

Write a message...

### Telegram Bot

#### KELOLA MEMBER

Nomor HP

Username

Chat ID (ASAP)

**DAFTAR** **CLEAN** **HAPUS** **EDIT**

#### DATA MEMBER

Nomor HP	Username	Chat ID
0818455527	a71bsusanto	678279378
085876247118	ajibbsusanto	201575235
687540252	Kevin	687540252
862305413	Nadya	862305413
574290378	Wim	574290378

\*pengisian CHAT ID setelah member terdaftar **REFRESH**

#### BROADCAST PESAN

**BROADCAST**

#### PESAN KELUAR & MASUK

a71bsusanto : /hola

SERVER : Hai 'a71bsusanto' Selamat Datang, ketik /help : untuk bantuan

a71bsusanto : /tes

SERVER : Hai 'a71bsusanto' Selamat Datang, ketik /help : untuk bantuan

SERVER : 1) Tanggal 17 - 19 Februari 2020 : Sebelum ujian cek lokasi ujian  
2) Jam 07.30-08.30 : Registrasi peserta  
3) Jam 08.30-09.00 : Penjelasan ujian tertulis  
4) Jam 09.00-10.00 : Ujian tertulis  
5) Jam 10.00-11.00 : Koreksi lembar jawaban & Pengumuman Hasil  
6) Jam 11.00-selesai : Tes wawancara

Gunakan kata kunci berikut : /start : untuk memulai c...

**KELOLA KATA KUNCI** **DATABASE PESAN** **KELUAR**

# ChatBot





# Rehat Sejenak

---



# Referensi

---

- Object First With Java, Fifth edition, David J. Barnes & Michael Kölling, Prentice Hall / Pearson Education, 2012.
- The Java™ Tutorial,  
<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/>,  
Oracle, 1995–2014.
- Java SE Tutorial,  
<http://www.oracle.com/technetwork/java/javase/downloads/jav-se-7-tutorial-2012-02-28-1536013.html>, Oracle, 2014.
- Java Platform, SE Documentation,  
<https://docs.oracle.com/en/java/javase/index.html>
- SCJP Sun Certified Programmer for Java™ 6 Study Guide Exam (310-065), Kathy Sierra & Bert Bates, Mc Graw Hill, 2008.
- Object Oriented Programming with Java, Romi Satria Wahono, 2008.

