



PENGENALAN BAHASA PBO

Disusun oleh
Mardiyyah Hasnawi



Topik Bahasan

- *Pengenalan Bahasa PBO : Java*
- *Menulis Program Java*

BAHASA PEMROGRAMAN JAVA



Pengenalan Teknologi Java

- ❑ Java adalah suatu bahasa pemrograman, dan sekaligus suatu platform.
 - Bahasa Pemrograman yang fiturnya mendukung paradigma pemrograman berbasis objek, prosedur, dan fungsional
- ❑ Java dikenal sebagai bahasa pemrograman tingkat tinggi.
- ❑ Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++.
- ❑ Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan.
- ❑ Teknologi java menyediakan: kompiler, interpreter, documentation generator, class file packaging.

Lanjut...

- ❑ Java dirancang agar dapat dijalankan di semua platform.
- ❑ Dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java.
- ❑ Java bersifat *neutral architecture*, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai Java Bytecode.

Fitur dari Java Technology

- Java Virtual Machine (JVM)
- Garbage collection
- Sekuritas kode

Sejarah Singkat

- 1991 : the green project
- 1992 : bernama OAK
- 1993 : the Green Project menjadi First Person
- 1994 : webRunner dikeluarkan - pengimnas-web pertama yang dapat menyokong objek-objek bergerak dan pelaksanaan dinamik
- 1995 : teknologi java dikeluarkan kepada kumpulan terpilih di website dan diubah ke Java
- 1996 : Jdk 1.0 rilis
- 1997 : lebih dari 220000 rilis
- 1999 : platform java 2 rilis dan j2ee
- 2000 : 400 user mnggunakan java
- 2001-2002 : >1 jt pengguna j2ee sdk dan pengembangannya
- 2003 : teknologi java basis desktop
- 2004 : java2 edisi standar 5 rilis
- 2005 – 2006 : teknologi standar dan mobile berkembang

Sejarah Singkat - lanjut

- 2001-2002 : >1 jt pengguna j2ee sdk dan pengembangannya
- 2003 : teknologi java basis desktop
- 2004 : java2 edisi standar 5 rilis
- 2005 – 2006 : teknologi standar dan mobile berkembang

Karakteristik

- Simplicity/kemudahan
- Berbagai macam lingkungan pemakai
- Robustness

Kelebihan dan Kekurangan





















































































Kelebihan

- ☐ Sederhana
- ☐ Kemudahan dalam pengembangan dan penyusunan script
- ☐ Usability
- ☐ Dinamis
- ☐ Handal
- ☐ Full-modularity
- ☐ Multiplatform

“Once write run anywhere....”

Kekurangan

- ☐ Penggunaan memori cukup tinggi
- ☐ Mudah didekompilasi

	Java	SmallTalk	TCL	Perl	Shells	C	C++
Simple							
Object Oriented							
Robust							
Secure							
Interpreted							
Dynamic							
Portable							
Neutral							
Threads							
Garbage Collection							
Exceptions							
Performance	High 	Medium 	Low 	Medium 	Low 	High 	High 

Arsitektur / Framework

- Java terdiri dari 2 bagian utama:

a. Java Virtual Machine (JVM) → (interpreter)

b. Java Application Programming Interface (Java API):

- Applet (browser) berjalan dinamis

- Networking → TCP, UDP, URL dan

IP ADDRESS

- JDBC

- Security

- 2D/3D

- Swing/AWT

- RMI/RPC

- Java Server Pages

- JNI

- Java Sound

- Java IDL+CORBA

- JAVA CARD

- JTAPI

JVM

- Menyediakan spesifikasi hardware platform
- Membaca byte code yang sudah terkompilasi yang mendukung multi-platform
- Diimplementasikan sebagai software atau hardware
- Diimplementasikan di Java technology development tool atau web browser

The Java Virtual Machine

- JVM provides definitions for the:
 - ▼ Instruction set (central processing unit [CPU])
 - ▼ Register set
 - ▼ Class file format
 - ▼ Stack
 - ▼ Garbage-collected heap
 - ▼ Memory area

Garbage collection

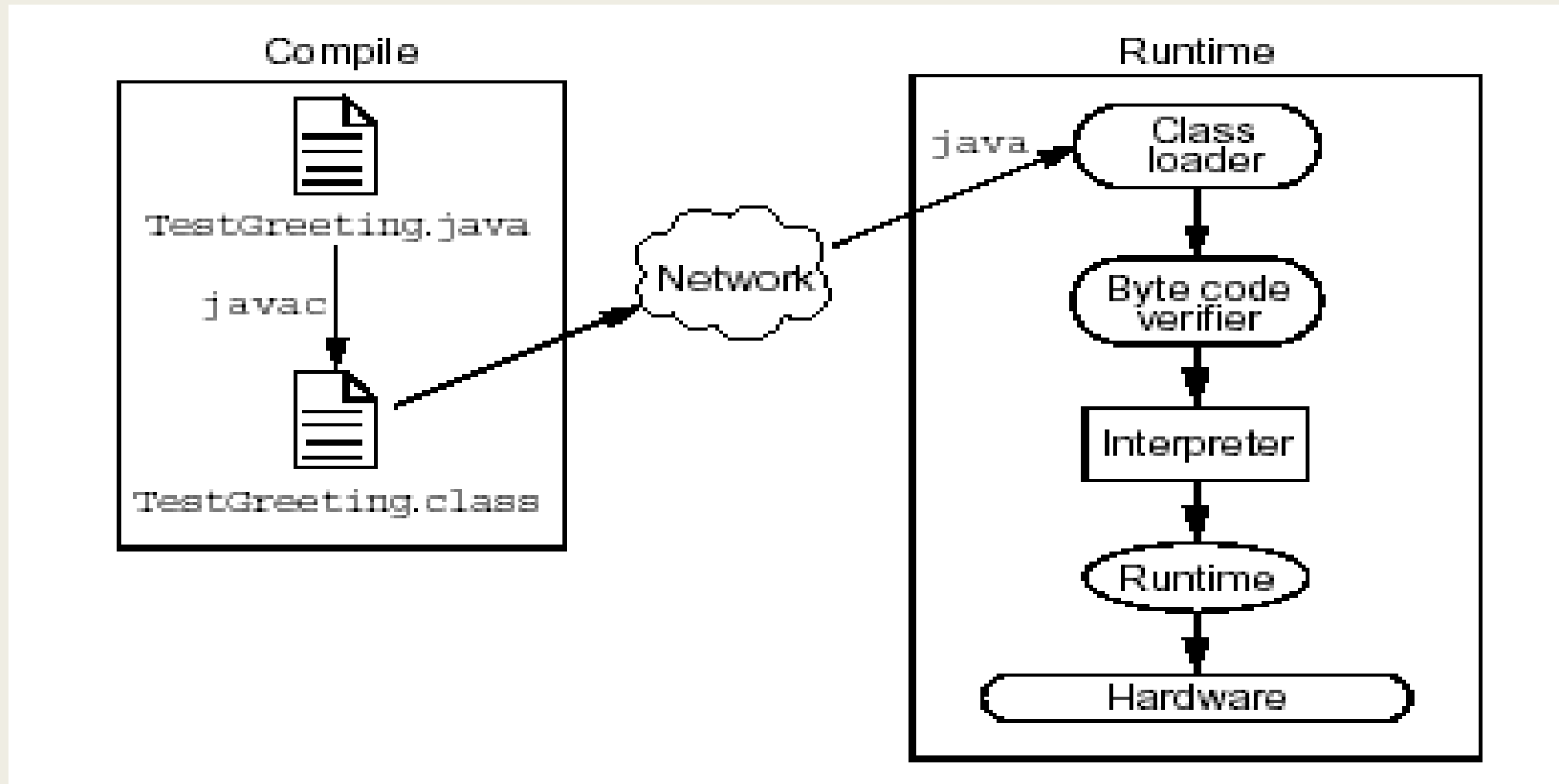
- Mengecek dan membersihkan memori yang tidak lagi terpakai
- Dilakukan otomatis
- Berjalan sesuai dengan implementasi JVM

Java Runtime Environment

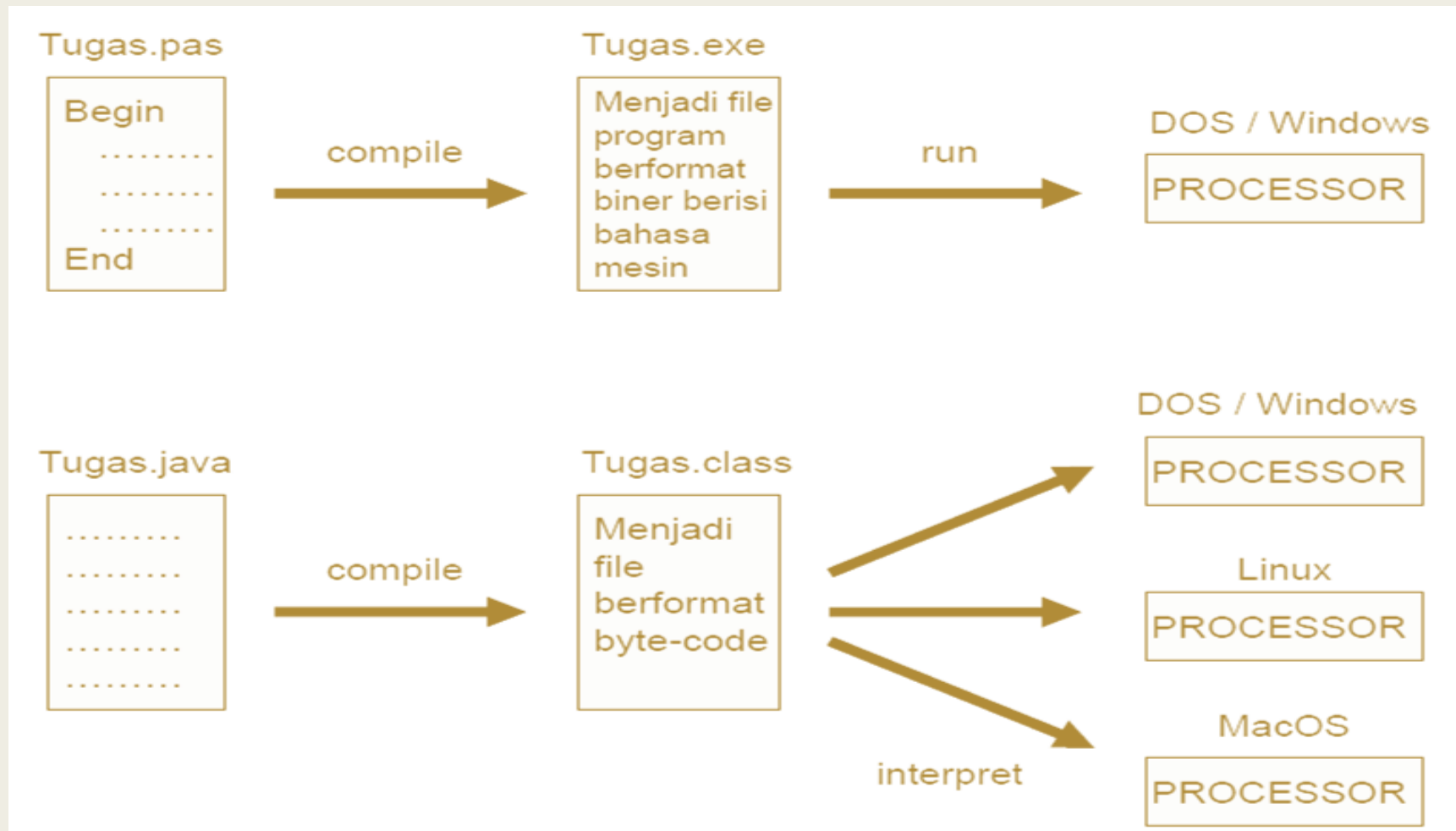
Menangani 3 tugas utama:

- *Melakukan pemanggilan kode*
- *Verifikasi kode*
- *Mengeksekusi kode*

Operation of the JRE

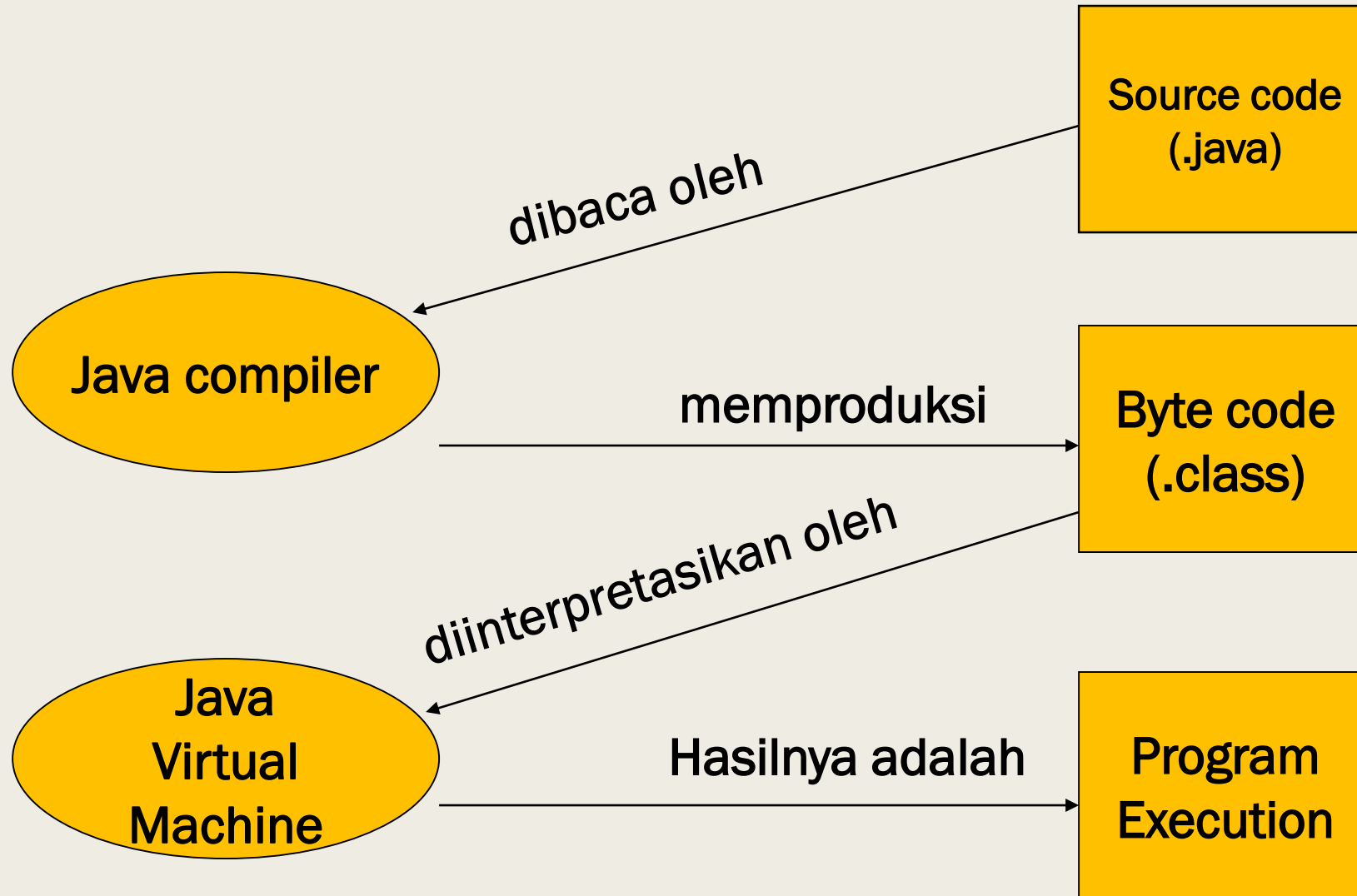


Compiler & Interpreter



- **Compiler :**
menerjemahkan menjadi byte code (class/ bahasa yang bias dipahami komputer)
- **Interpreter :**
menerjemahkan menjadi bahasa yang mudah dipahami oleh manusia

Java Language

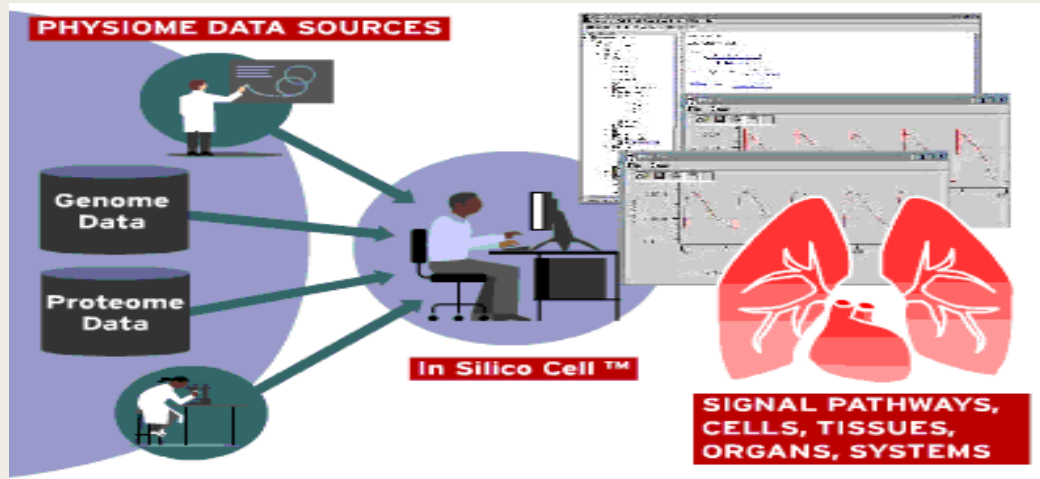


Arsitektur / Framework

1. *Java 2 Standard Edition (J2SE)* untuk aplikasi standar termasuk desktop.
2. *Java 2 Enterprise Edition (J2EE)* untuk aplikasi web, aplikasi system tersebar dengan beraneka ragam klien dan kompleksitas tinggi serta superset dari standar java
3. *Java 2 Micro Edition (J2ME)* merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device

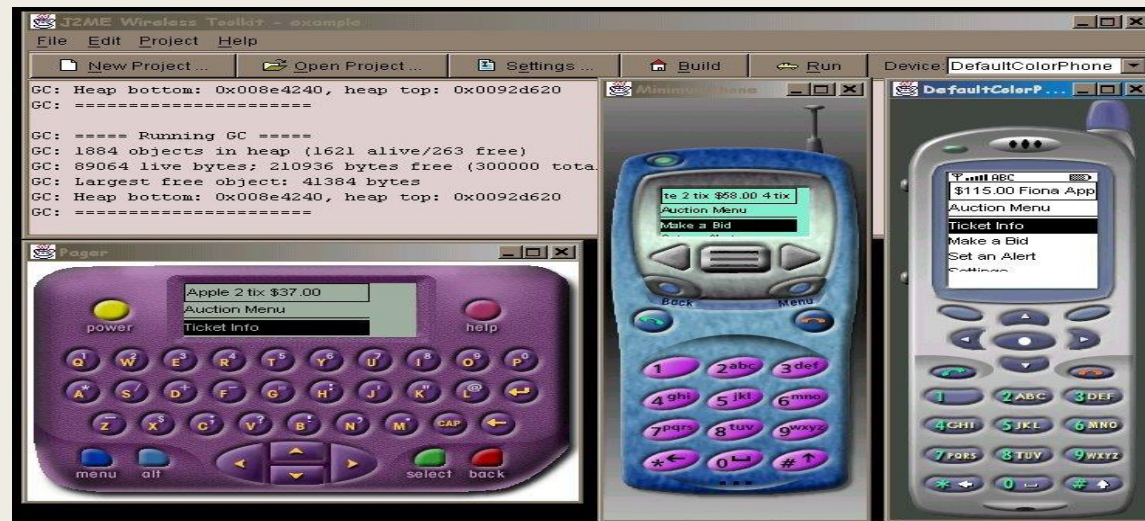
Lanjut..

Java 2 Standard Edition (J2SE)



Java 2 Enterprise Edition (J2EE)

Java 2 Micro Edition



MENULIS PROGRAM JAVA

Bentuk program

1. *"Text based" Console Application* → *public class* adalah class yang sederhana
2. *GVI application* → *public class* biasanya menggunakan *JFrame*

Tool Pemrograman Java

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

Persiapan

- Aplikasi yang diperlukan :
 - a. **Java Development Kit (jdk) / Java Runtime Environment (jre)** = compiler/interpreter sesuai Sistem Operasi yang digunakan
 - b. **Editor** : Notepad atau Jcreator, Netbeans, eclipse dll
 - c. **Command Prompt** : Eksekusi Kode Program → SETTING PATH
- *Pastikan seluruh Aplikasi di atas telah terpasang.*

Instalasi Java Standard Edition

1. Download Java Standard Edition dari:
<https://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Instal Java Standard Edition dengan mengklik:
jdk-11.0.2_windows-x64_bin.exe
3. Instal text editor untuk editing code:
textpad, notepad++

Proses Pembangunan Aplikasi Menggunakan Bahasa Java

- 1) Analisa kasus (algoritma dan logika)
- 2) Penulisan kode program melalui text editor seperti notepad atau IDE.
Kemudian simpan pada folder tertentu. Penulisan Nama file sama dengan nama kelas dengan tipe/ekstensi **__namaklas.java**
3. SETTING PATH untuk kompilasi kode program melalui cmd, dua cara:
 1. Klik **Control Panel\System and Security\System\Advanced system settings→Environment Variables→Edit →PATH:
C:\Program Files\Java\jdk-11.0.2\bin; atau Environment Variables→New→Variable Name= PATH→ Variable Value = alamat jdk**
 2. **Klik cmd → tulis SET PATH=C:\Program Files\Java\jdk-11.0.2\bin;**

Proses Pembangunan Aplikasi Menggunakan Bahasa Java

4. Kompilasi Kode Program

- ✓ Proses kompilasi melalui cmd, konsole atau tool aplikasi
- ✓ Perintah untuk melihat versi java : C:\jakode>**java -version**
- ✓ Perintah kompilasi : C:\jakode>**javac _namaFile.java**
 - ✓ Hasil kompilasi berupa file-file class
- ✓ Perintah interpretasi: C:\jakode>**java _namaFile** (tanpa ekstensi)

5. Eksekusi kode program berdasarkan uji kasus

Struktur Kode Program Bahasa Java

- Sintak : Deklarasi klas beserta anggotanya

```
_ modifier class Filename {  
    // variables and data types  
    // constructor  
    // methods  
    // inner class  
    // etc...  
}
```

atau

```
[modifier] [class] namaclass {  
    //...  
}
```

- Contoh :

```
public class X{  
    public static void main(String [] args) {  
        // Isi  
    }  
}
```

Struktur Program Java

- Program harus punya method di public class:

public static void main(String[] args)

→ *Program Java mulai dari sini!*

Contoh

Tulis Kode Program berikut kemudian simpan di direktori yang anda inginkan dengan namafile = namakelas yaitu Hello.java

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

Compiler & running program

- Eksekusi melalui command prompt dengan mencari file java tersebut di direktori lalu compile dengan perintah:

```
C:\aplikasi> javac Hello.java
```

- Hasil : Hello.class

- Interpreter :

```
C:\aplikasi > java Hello
```

- Hasil : Hello World !

Penjelasan Hello.java

■ Elemen 1

```
public class Hello{  
    ....  
}
```

Pendefinisian class dengan modifier public, agar dapat dikompilasi dan dijalankan oleh Java VM.

Penjelasan Hello.java

■ Element 2

```
public static void main (String[] args) {  
    . . . . .  
}
```

Kode program ini adalah bagian utama, karena bagian ini pertama kali dijalankan.

Penjelasan :

- **public** : salah satu macam modifier.
- **Static** : tipe method
- **void** : menunjukkan bahwa method tidak mengembalikan nilai atau objek.
- **main** : nama method utama dari program java.
- **String** : tipe argumen yang diterima untuk parameter dari command java.
- **args** : array argumen yang bisa ditambahkan pada saat menggunakan command java untuk menjalankan program java

Penjelasan Hello.java

■ Element 3

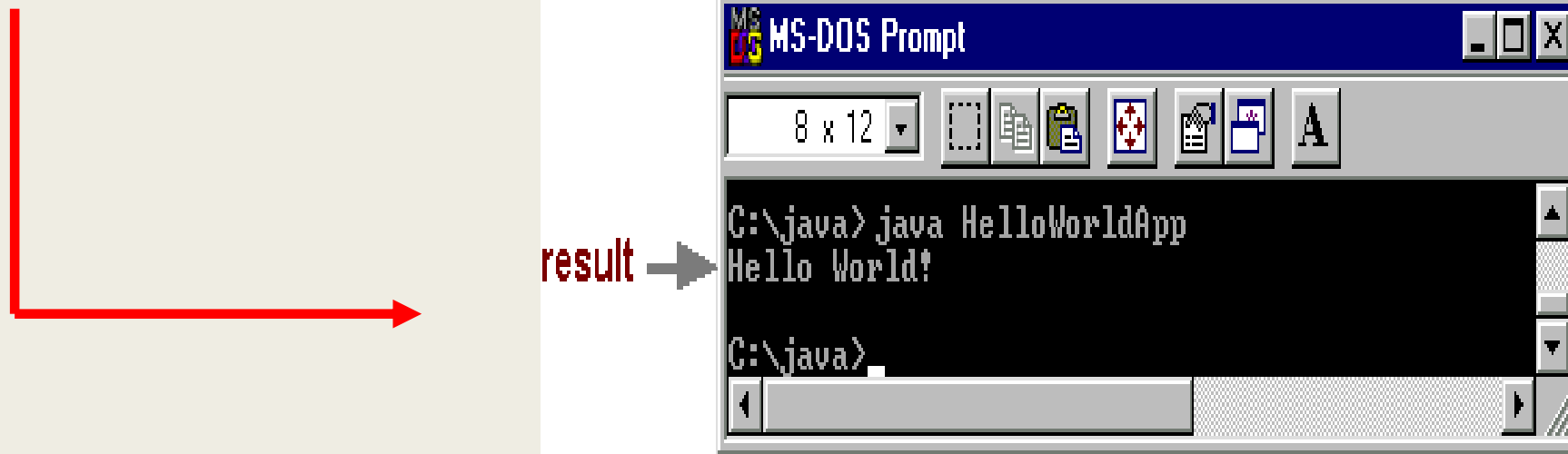
```
System.out.println("Hello Word!");
```

- Fungsi kode tersebut adalah menampilkan pada konsol data yang dimasukkan sebagai parameter.
- Pada kode diatas, tipe datanya String yang di inputkan sebagai parameter pada method println().
- Untuk menginputkan data karakter atau numerik sebagai String maka diawali dan diakhiri dengan tanda " "

Contoh lain

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

C:\javac HelloWorldApp.java



TUGAS 1

- Buat 3 Class dengan ketentuan sebagai berikut :
 - 1) *Buatlah sebuah aplikasi dalam bahasa java yang akan menampilkan teks : "Selamat Belajar"*
 - 2) *Buatlah sebuah aplikasi dalam bahasa java yang akan menampilkan teks : "Belajar Bahasa Pemrograman Java"*
 - 3) *Buatlah sebuah aplikasi dalam bahasa java yang akan menampilkan teks : "Selesai Belajar Pemrograman Java"*
- Upload Tugas Program di akun github masing-masing. Atur data masing-masing github sehingga dapat dicari lebih mudah..
- Share link github masing-masing melalui grup