

PEMROGRAMAN JAVA

- Pengenalan Java
- Kompilasi Java
- Program Java Sederhana

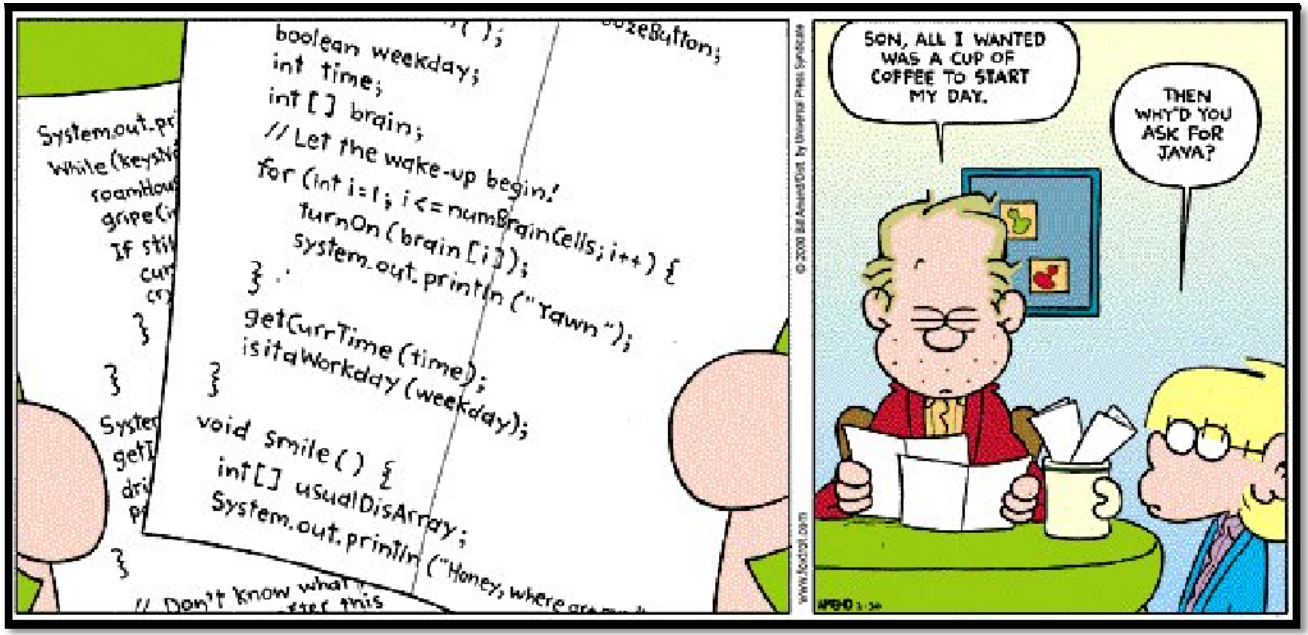
Yoannita

Sun Microsystem

- Java dikembangkan oleh Sun Microsystem
- Ditujukan untuk mengatasi perbedaan pada aneka platform yang ada saat ini.
- Slogan java : *"Write once, run anywhere."*

- Teknologi Java adalah sebuah bahasa pemrograman dan juga merupakan sebuah platform.

Mengapa dinamakan Java?



Sejarah Singkat Java

- Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumer 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”.**
- Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap *platform* mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan *intermediate code untuk mesin hipotesis. Mesin ini sering disebut dengan mesin* maya *(virtual machine).*
- Kode ini kemudian dapat digunakan di sembarang mesin yang memiliki *interpreter. Proyek Green menggunakan mesin maya untuk mengatasi* isu utama tentang netral terhadap arsitektur mesin.
- Karena orang–orang di proyek Green berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama **”Oak”** oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA.
- Nama JAVA sendiri terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal bijih kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java.

Mengapa Mempelajari Java? (1)

Berdasarkan *white paper resmi dari SUN, Java memiliki karakteristik berikut :*

1. **Sederhana**

   Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namunsintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance. Java juga* menggunakan *automatic memory allocation dan memory garbage collection.*

2. **Berorientasi objek (*Object Oriented)***

   Java mengunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrogramanberorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

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). Hal ini menyebabkan *source code Java yang telah dikompilasi menjadi* Java *bytecodes dapat dijalankan pada platform yang berbeda-beda.*

5. **Robust**

   Java mempuyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling untuk* membantu mengatasi error pada pemrograman.

Mengapa Mempelajari Java? (2)

Berdasarkan *white paper resmi dari SUN, Java memiliki karakteristik berikut :*

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.

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

10. **Multithreaded**

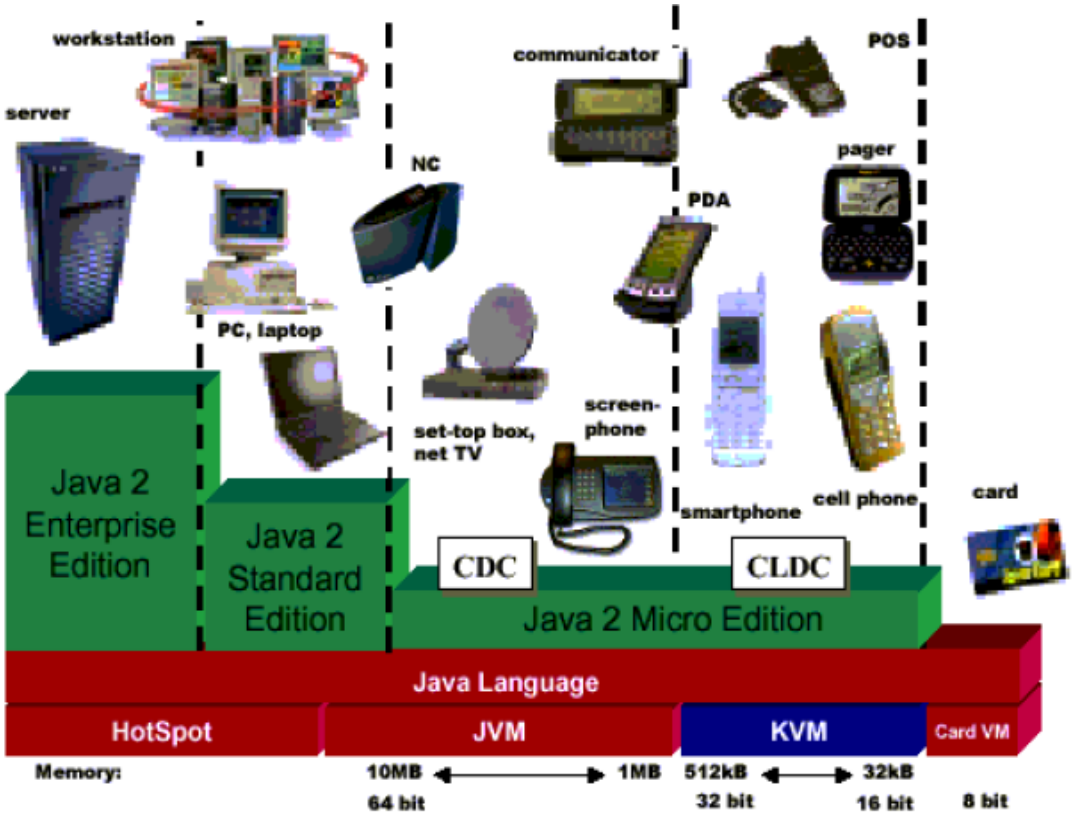
    Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. **Dinamis**

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

Pembagian Java

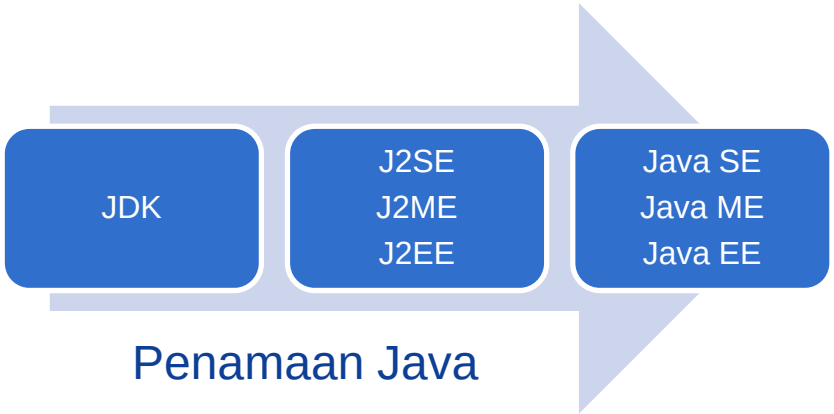
- Java Standard Edition (Java SE)
  - “desktop Java”
- Java Micro Edition (Java ME)
  - “wireless Java”
- Java Enterprise Edition (Java EE)
  - “server-side Java”
- Java Card
  - difokuskan ke aplikasi *smart card.*
  - *Java Card* khusus dikembangkan untuk membuat aplikasi-aplikasi pada *smart card,* misalnya aplikasi kartu telepon CHIP, kartu VISA, kartu SIM pada ponsel dan aplikasi *mobile banking* BCA yang saat ini sudah umum digunakan.



Penamaan Java oleh Sun Microsystem

- 2 versi penamaan : penamaan versi dari **divisi engineer** dan versi dari **divisi marketing.**
- versi 1.0 hingga 1.4 ini penamaan dari sisi engineer, namun saat versi berikutnya SUN lebih memilih penamaan divisi marketing yaitu cukup dengan menyebutkan **jdk** lalu diikuti versi releasenya seperti jdk 5.0.
- Ketika **Java 2** diperkenalkan tahun 1999, platform Java terbagi menjadi 3 variant :
  - **Java 2 Standard Edition (J2SE),**
  - **Java 2 Micro Edition (J2ME),**
  - **Java 2 Enterprise Edition (J2EE).**
- angka 2 setelah huruf J (J2SE, J2EE, J2ME) bukanlah merupakan no versi releasenya akan tetapi merupakan trademark dari SUN untuk memberitahukan bahwa setelah java berada di versi 1.2, java telah mengalami perubahan dan peningkatan besar-besaran.

- Tahun 2005, Dari java versi 6 yang sedang dikembangkan saat itu, SUN telah menghilangkan istilah J2SE, J2EE, dan J2ME tsb, dengan menyebutkan java dengan Java SE, Java EE, dan Java ME diikuti nomor releasenya dari penamaan divisi marketing.
  - Java Standard Edition (Java SE)
  - Java Micro Edition (Java ME)
  - Java Enterprise Edition (Java EE)



Penamaan Java oleh Sun Microsystem

What's in a name?

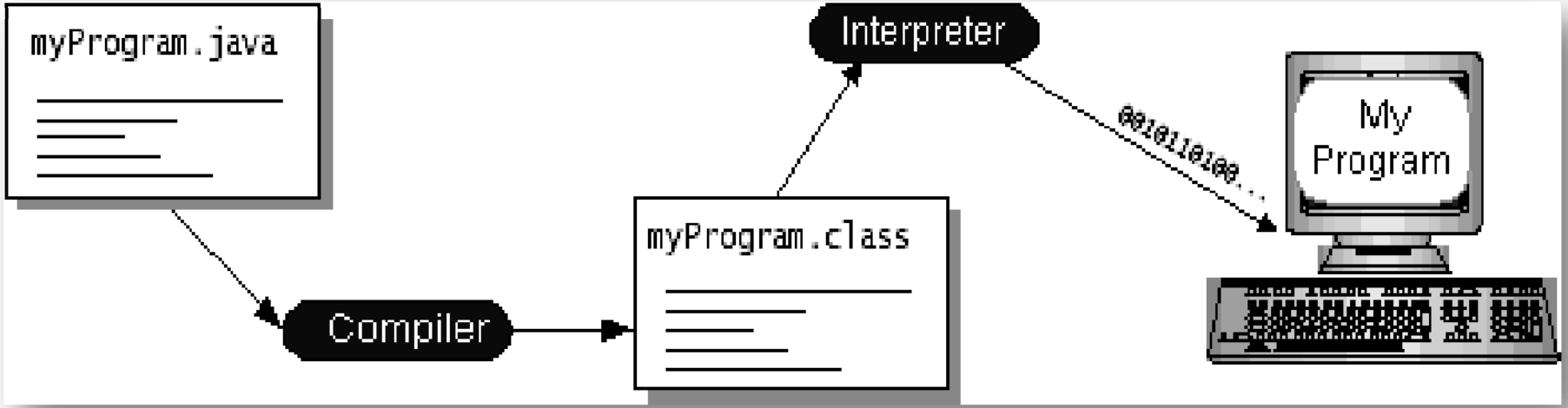
The numbering of Java's SDK versions is really confusing. Instead of "Java 1," "Java 2," and "Java 3," the numbering of Java versions winds through an obstacle course. Here's how it works:

- Java JDK 1.0 (1996)
- Java JDK 1.1 (1997)
- Java 2 SDK, 1.2 (1998): Sun Microsystems added an additional "2" and changed "JDK" (Java Development Kit) to "SDK" (Software Development Kit)
- Java 2 SDK, 1.3 (2000)
- Java 2 SDK, 1.4 (2002)
- Java 2 JDK, 5.0 (2004): Sun reverts to "JDK" and gives up on the silly 1.x numbering scheme.
- Java 6 JDK (2006): Sun drops the unnecessary *2* and gets rid of the *.0* too.

To make things worse, the file-numbering scheme isn't consistent with the product numbering scheme. So when you install Java 6 JDK, you get a directory named `jdk1.6.0`. The *1.x* numbering returns to haunt you, and so does the *.0* decimal point business.

You can find the old versions of Java, along with all the intermediate releases not listed here, at `java.sun.com/products/archive`.

Kompilasi



- *Compile* ➔ menerjemahkan program ke bentuk kode yang dapat dimengerti oleh mesin (komputer).
- Agar sebuah program java dapat dijalankan, maka file dengan extensi **`.java`** harus dikompilasi menjadi file bytecode.

- *Compiler* akan mengecek *syntax* lalu *mengubah program ke kode dalam bahasa mesin*. Kode dalam bahasa mesin inilah yang akan dieksekusi oleh komputer.

Java Script = Java ?

- Java Script merupakan ***scripting language*** yang digabung dengan HTML sehingga memungkinkan suatu web page mampu berinteraksi lebih baik lagi dengan penggunanya.
- Javascript tidak memerlukan kompilasi, hanya diinterpretasikan dari *web browser*.

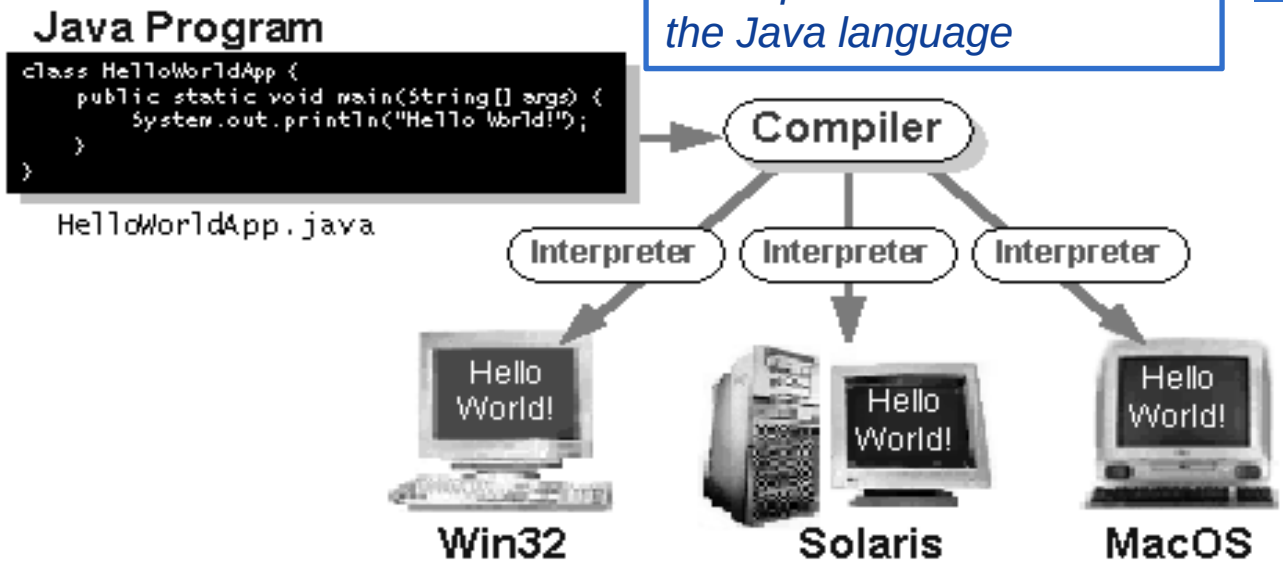
Sedangkan,

- Java merupakan ***full programming language*** yang dieksekusi oleh JVM (*Java Virtual Machine*)

Cross-platform

"Write once, run anywhere." cross-platform benefits of the Java language

Apa yang membuat Java begitu menarik? Jawabnya terletak pada kemampuannya untuk menghasilkan program yang mampu berjalan di atas segala jenis platform. Sekali kita menulis program dalam bahasa Java, maka ia akan siap bekerja di segala jenis platform tanpa perlu modifikasi kode.

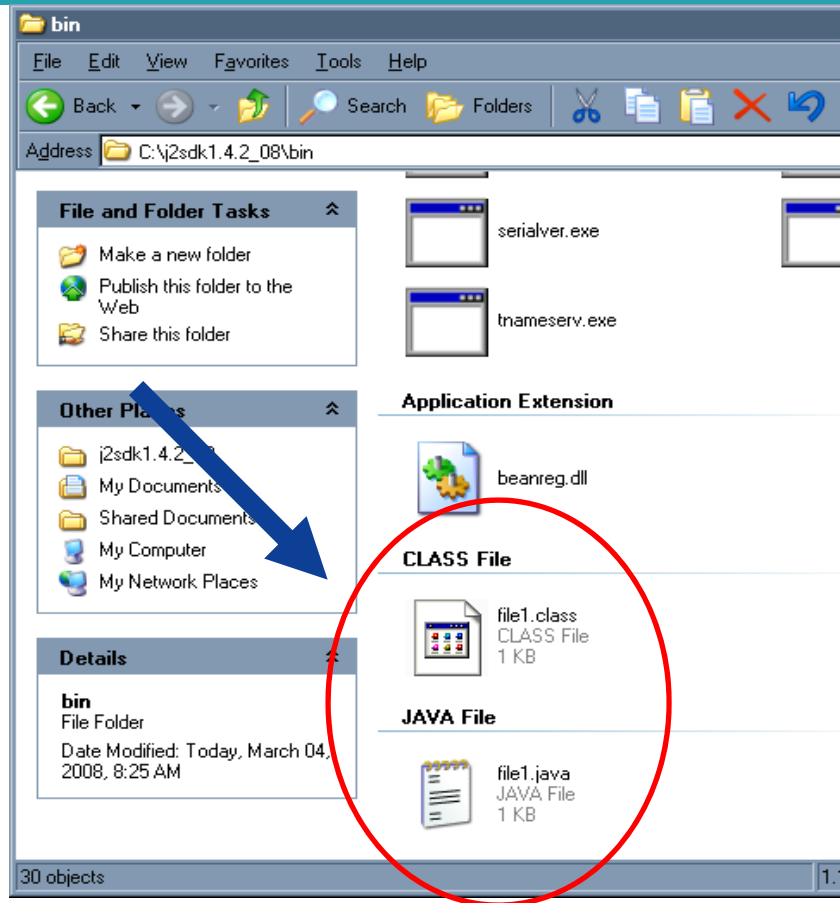


- Java berdiri di atas sebuah mesin interpreter yang diberi nama Java Virtual Machine (JVM). JVM (Java Virtual Machine) adalah mesin untuk menjalankan bytecode pada file kelas Java pada mikroprosesor, baik yang berada pada komputer atau pada piranti elektronik lainnya.
- Bahasa java disebut sebagai bahasa yang *portable (write once run anywhere)* karena dapat dijalankan pada berbagai Sistem Operasi, asalkan pada sistem tersebut terdapat JVM.

Kompilasi

- **Javac <namafile.java>**
  - Javac file1.java
  - ➔ **Membuat file1.class** dari file1.java
- **Java <namaclass>**
  - Java file1
  - ➔ **Mengeksekusi** class file1

```
// nama file :  file1.java
class file1
{
   public static void main (String[] args )
   {
      System.out.println("Hello world");
   }
}
```



Kerangka Pertama Program Java

```
class <nama_class>{
  public static void main (String[] args) {
     // statements
  }
}
```

- <nama class> dapat diganti dengan nama class pilihan anda. Karakter pertama sebaiknya dibuat huruf besar (lihat keterangan identifier)
- *args* boleh diganti
- *public static void main* jangan diganti

Program Java Sederhana

```
class Sederhana {
   public static void main (String[] args) {
       System.out.println (“Hello java…”);
   }
}
```

- class (nama class: Sederhana)

  ➔ sekumpulan *data attributes* dan *method*

- Method : main()

  ➔ sekumpulan *statement*

Method Utama

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

- Baris kode di atas mendeklarasikan suatu method dengan nama *main*.

main

- Merupakan tempat dimulainya program dieksekusi. Jika tidak ada method ini maka akan keluar pesan kesalahan:

  Exception in thread "main".java.lang.NoSuchMethodError: main

- Nama main di sini merupakan suatu keharusan dalam Java karena Java akan mencari method yang bernama main ini sebagai titik awal eksekusi program.

public static void main (String[] args)

Public

- Keyword public merupakan *access specifier* yang menentukan *visibility level* dari method ini.
- *Public* berarti method ini dapat diakses/dipanggil dari luar class dimana ia dideklarasikan.
- selain *public*, jenis access specifier yang lain adalah *private*, *protected*, dan *default*.
- Method main() akan dipanggil dari luar oleh *run-time Java* saat program akan dieksekusi sehingga access specifier yang dimiliki haruslah public.

Sekilas public & private

- Public : siapa saja bisa mengakses member ini :
  - kode yang ada dalam class itu sendiri
  - atau yang berada di luar class

- Private : hanya dapat digunakan oleh internal member dari class tersebut saja
  - tidak ada kode satupun dari luar class tersebut yang diizinkan mengakses / mengubah nilai dari member tersebut

`public static void main (String[] args)`

static

- Keyword static memungkinkan method main() dipanggil tanpa harus terlebih dahulu membuat *instance* dari class file1
- Ini diperlukan karena method main() akan dieksekusi sebelum objek dari class file1 dibuat di memori.

void

- Keyword void berarti bahwa method main() tidak mengembalikan nilai apapun setelah dipanggil/dieksekusi.

args

- Argument **args** `(String[] args)` adalah array objek `string` argument baris-baris perintah.
- variabel **args** bertipe *array* dari `String`

System.out.println ("Hello java…");

System : class

- nama dari salah satu class standar yang dimiliki oleh java.

Out : objek

- anggota dari class System dan juga merupakan objek tersendiri, `out` merupakan objek yang mewakili standard *output stream* yang dalam hal ini adalah layar komputer.
- Seperti halnya *method main*(), objek `out` ini dideklarasikan menggunakan keyword **`static`** di dalam classnya sehingga dapat langsung dipanggil tanpa perlu terlebih dahulu membuat *instance* dari class System.

println : method

- Method yang terdapat pada objek out. Berfungsi untuk mencetak keluaran ke standard output. Method ini juga mencetak karakter pindah baris.
  - println vs print

"Hello java…" : paramenter

- Parameter dari method println() yang diterima oleh internal method ini dan dicetak ke standard output(layar komputer).

Tanda **;**

- Menandakan akhir suatu statement/pernyataan/ perintah.

Contoh (potongan) kode program

System.out.println

```
class DataDiri {
    public static void main
    (String[] args) {
int a = 10;
int b = 1;

System.out.println (a);
System.out.println ();
System.out.println (b);
    }
}

/*
  Hasilnya :
    10

    1
*/
```

System.out.print

```
Note : misalkan :
…
System.out.print (a);
// System.out.print (); -> Akan
     //menghasilkan output error
System.out.print (b);
….

/*
  Hasilnya :
    ab
*/
```

Literal

Kode	Arti
\n	Baris baru
\t	Tab
\b	Backspace
\r	Carriage Return
\f	Formfeed01
\\	Backslash
\’	Tanda kutip tunggal
\”	Tanda kutip ganda
\ddd	Bilangan oktal
\xdd	Bilangan hexadesimal
\udddd	Karakter unicode

Literal Karakter dan artinya

Contoh kode program

Menampilkan lebih dari satu baris

```
// nama file:  DataDiri.java
// deskripsi:  program menampilkan beberapa kalimat
// kompilasi: javac DataDiri.java
// eksekusi:  java DataDiri

class DataDiri {
   public static void main (String[] args) {
     System.out.println (“Nama: MDP”);
     System.out.println (“Alamat: Jln. Rajawali 14”);
     System.out.println (“Tlp:376400\nfax:376360\nwebsite:www.stmik-mdp.net”);
   }
}

/*   \n : newline, juga berfungsi untuk pindah baris    */
```

Case Sensitive

- Huruf kecil dan huruf kapital pada identifier tidak dianggap sama
- Identifier alamat, Alamat, dan ALAMAT menyatakan tiga identifier yang berbeda.

```
class CaseSensitive1
{
  public static void main (String[] args )
  {
            String nama = “Evi";
            String Nama = “Eva";
            String NAMA = “Evo";

            System.out.println("String yang dicetak : " + Nama);
 }
```

- **Apakah hasil output program di atas?**

Latihan

- Buatlah program java untuk menampilkan kalimat berikut:

Hello World!
My Name is 'Hero'
It's been nice knowing you.
"Goodbye World!"

Note:
Tanda ' dan " ditampilkan ke layar