

# Program Java Pertama

Mari kita lihat kode sederhana yang akan mencetak kata-kata *Hello World* .

## Contoh

```
public class MyFirstJavaProgram {  
  
    /* This is my first java program.  
     * This will print 'Hello World' as the output  
     */  
  
    public static void main(String []args) {  
        System.out.println("Hello World"); // prints Hello World  
    }  
}
```

Mari kita lihat bagaimana cara menyimpan file, mengkompilasi, dan menjalankan program. Silakan ikuti langkah-langkah berikut

- Buka notepad dan tambahkan kode seperti di atas.
- Simpan file sebagai: MyFirstJavaProgram.java.
- Buka jendela prompt perintah dan masuk ke direktori tempat Anda menyimpan kelas. Asumsikan itu C: \.
- Ketik 'javac MyFirstJavaProgram.java' dan tekan enter untuk menyusun kode Anda. Jika tidak ada kesalahan dalam kode Anda, prompt perintah akan membawa Anda ke baris berikutnya (Asumsi: Variabel jalur disetel).
- Sekarang, ketik 'java MyFirstJavaProgram' untuk menjalankan program Anda.
- Anda akan dapat melihat 'Hello World' tercetak di jendela.

## Keluaran

```
C:\> javac MyFirstJavaProgram.java  
C:\> java MyFirstJavaProgram  
Hello World
```

## Sintaks Dasar

Tentang program Java, sangat penting untuk memperhatikan poin-poin berikut.

- Sensitivitas Huruf - Java bersifat case sensitive, yang berarti pengenalan Hello dan hello akan memiliki arti yang berbeda di Java.
- Nama Kelas - Untuk semua nama kelas, huruf pertama harus dalam Huruf Besar. Jika beberapa kata digunakan untuk membentuk nama kelas, huruf

pertama setiap kata dalam harus menggunakan Huruf Besar.

Contoh: `class MyFirstJavaClass`

- Nama Metode - Semua nama metode harus dimulai dengan huruf Kecil. Jika beberapa kata digunakan untuk membentuk nama metode, maka huruf pertama setiap kata dalam harus menggunakan Huruf Besar.

Contoh: `public void myMethodName ()`

- Nama File Program - Nama file program harus sama persis dengan nama kelas.

Saat menyimpan file, Anda harus menyimpannya menggunakan nama kelas (Ingat Java case sensitive) dan tambahkan '.java' di akhir nama (jika nama file dan nama kelas tidak cocok, program Anda tidak akan dikompilasi ).

Tetapi harap dicatat bahwa jika Anda tidak memiliki kelas publik yang ada di file, maka nama file bisa berbeda dari nama kelas. Juga tidak wajib memiliki kelas publik di file tersebut.

Contoh: Asumsikan 'MyFirstJavaProgram' adalah nama kelas. Maka file tersebut harus disimpan sebagai '*MyFirstJavaProgram.java*'

- `public static void main (String args [])` - Pemrosesan program Java dimulai dari method main() yang merupakan bagian wajib dari setiap program Java.

## Pengenal/Identifier Java

Semua komponen Java membutuhkan nama. Nama digunakan untuk menyebutkan kelas, variabel, dan metode disebut identifier / pengidentifikasi / pengenal.

Di Java, ada beberapa hal yang perlu diingat tentang pengenal. Mereka adalah sebagai berikut :

- Semua pengidentifikasi harus dimulai dengan huruf (A sampai Z atau a sampai z), karakter mata uang (\$) atau garis bawah (\_).
- Setelah karakter pertama, pengenal dapat memiliki kombinasi karakter apa pun.
- Kata kunci tidak dapat digunakan sebagai pengenal.
- Yang terpenting, pengenal peka huruf besar / kecil.
- Contoh pengenal resmi: usia, \$gaji, \_value, \_\_1\_value.
- Contoh pengenal ilegal: 123abc, -salary.

## Modifier / Pengubah Java

Seperti bahasa lain, dimungkinkan untuk memodifikasi kelas, metode, dll., Dengan menggunakan modifier / pengubah. Ada dua kategori modifier :

- Modifier Akses - default, public, protected, private
- Modifier Non-akses - final, abstract, strictfp

# Variabel Java

Berikut adalah jenis-jenis variabel di Java -

- Variabel Lokal
- Variabel Kelas (Variabel Statis)
- Variabel Instance (Variabel Non-statis)

## Array Java

Array adalah objek yang menyimpan banyak variabel dengan tipe yang sama. Namun, array itu sendiri adalah objek di heap. Kita akan melihat bagaimana mendeklarasikan, membangun, dan menginisialisasi di bab-bab selanjutnya.

## Java Enums

Enum diperkenalkan di Java 5.0. Enum membatasi variabel untuk memiliki salah satu dari sedikit nilai yang telah ditentukan. Nilai-nilai dalam daftar yang disebutkan dinamakan enum.

Dengan penggunaan enum, dimungkinkan untuk mengurangi jumlah bug dalam kode Anda.

Misalnya, jika kita mempertimbangkan aplikasi untuk toko jus segar, akan memungkinkan untuk membatasi ukuran gelas menjadi kecil, sedang, dan besar. Ini akan memastikan bahwa itu tidak akan memungkinkan siapa pun untuk memesan ukuran apa pun selain kecil, sedang, atau besar.

### Contoh

```
class FreshJuice {
    enum FreshJuiceSize{ SMALL, MEDIUM, LARGE }
    FreshJuiceSize size;
}

public class FreshJuiceTest {

    public static void main(String args[]) {
        FreshJuice juice = new FreshJuice();
        juice.size = FreshJuice.FreshJuiceSize.MEDIUM ;
        System.out.println("Size: " + juice.size);
    }
}
```

Contoh di atas akan menghasilkan hasil sebagai berikut -

## Keluaran

Size: MEDIUM

Catatan - Enum dapat dideklarasikan sendiri atau di dalam kelas. Metode, variabel, konstruktor dapat didefinisikan di dalam enum juga.

## Kata Kunci Java

Daftar berikut menunjukkan kata-kata keyword di Java. Kata-kata khusus ini tidak boleh digunakan sebagai konstanta atau variabel atau nama pengenalan lainnya.

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
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		

## Komentar di Java

Java mendukung komentar satu baris dan banyak baris yang sangat mirip dengan C dan C ++. Semua karakter yang tersedia di dalam komentar apa pun diabaikan oleh kompiler Java.

### Contoh

```
public class MyFirstJavaProgram {

    /* This is my first java program.
     * This will print 'Hello World' as the output
     * This is an example of multi-line comments.
     */

    public static void main(String []args) {
        // This is an example of single line comment
        /* This is also an example of single line comment. */
        System.out.println("Hello World");
    }
}
```

### Keluaran

Hello World

## Menggunakan Garis Kosong

Sebuah baris yang hanya berisi spasi, atau kemungkinan dengan komentar, disebut sebagai baris kosong, dan Java mengabaikannya sama sekali.

## Warisan / Inheritance

Di Java, kelas bisa diturunkan dari kelas lain. Pada dasarnya, jika Anda perlu membuat kelas baru dan di sini sudah ada kelas yang memiliki beberapa kode yang Anda perlukan, maka dimungkinkan untuk mendapatkan kelas baru Anda dari kode yang sudah ada.

Konsep ini memungkinkan Anda untuk menggunakan kembali field dan method kelas yang ada tanpa harus menulis ulang kode di kelas baru. Dalam skenario ini, kelas induk disebut superclass dan kelas turunannya disebut subclass .

## Interface

Dalam bahasa Java, antarmuka dapat didefinisikan sebagai kontrak antar objek tentang cara berkomunikasi satu sama lain. Interface memainkan peran penting dalam hal konsep warisan.

Sebuah interface mendefinisikan method, kelas turunan (subclass) harus mengimplementasikan method tersebut. Tetapi penerapan method ini sepenuhnya tergantung pada subclass.