

# LATIHAN MANDIRI 01

PEMROGRAMAN BERORIENTASI OBYEK

Disusun oleh: Yuan Lukito, S.Kom., M.Cs

## LATIHAN 1

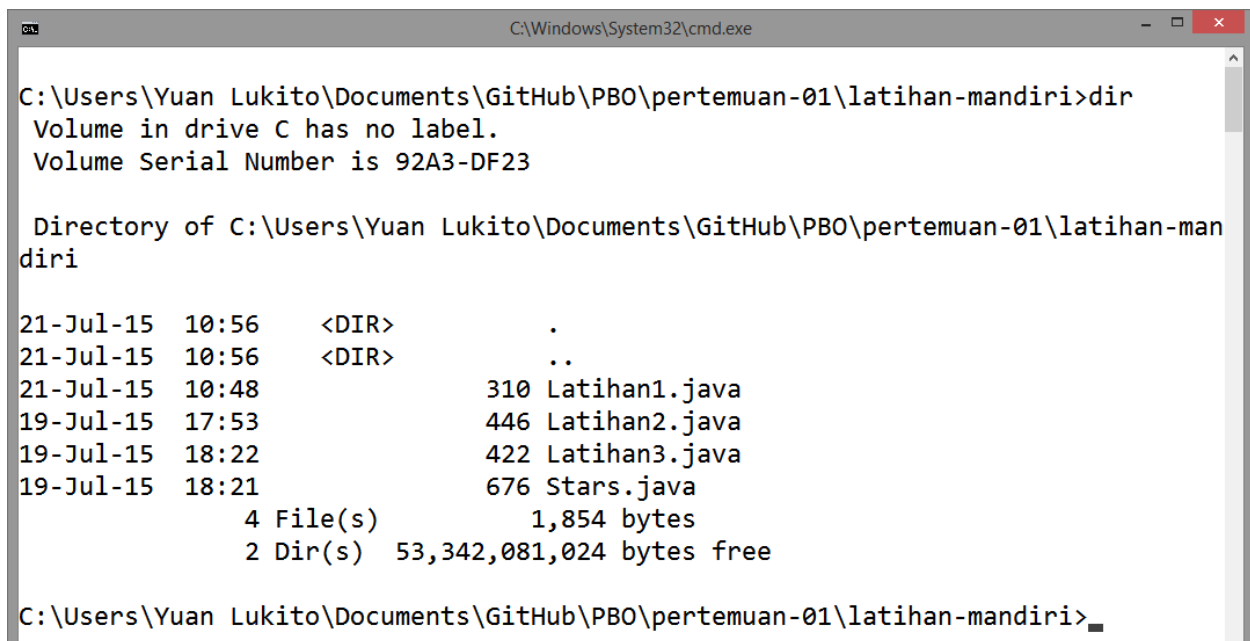
Untuk mengkompilasi program yang ditulis dengan Java, diperlukan Java Software Development Kit (SDK). Install terlebih dahulu Java SDK, kemudian atur path (sesuai dengan demo di kelas) sehingga dapat memanggil **javac** dan **java** pada Command Prompt.

Anda diberi sebuah file bernama Latihan1.java yang berisi definisi dari Class Latihan1 seperti pada Gambar 1.

```
public class Latihan1 {  
    public static void main(String[] args) {  
        System.out.println("Selamat, anda berhasil meng-compile");  
        System.out.println("dan menjalankan program ini!");  
    }  
}
```

Gambar 1: Source code Latihan1.java (PBO\pertemuan-01\latihan-mandiri\Latihan1.java)

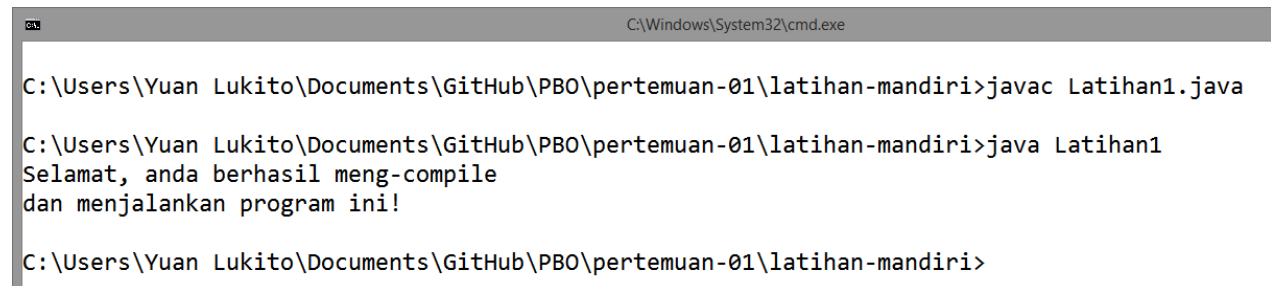
Jalankan Command Prompt, masuk ke dalam folder tempat Latihan1.java berada. Sebagai panduan, lihat Gambar 2.



```
C:\Windows\System32\cmd.exe  
  
C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>dir  
Volume in drive C has no label.  
Volume Serial Number is 92A3-DF23  
  
Directory of C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri  
  
21-Jul-15  10:56    <DIR>          .  
21-Jul-15  10:56    <DIR>          ..  
21-Jul-15  10:48                310 Latihan1.java  
19-Jul-15  17:53                446 Latihan2.java  
19-Jul-15  18:22                422 Latihan3.java  
19-Jul-15  18:21                676 Stars.java  
                4 File(s)              1,854 bytes  
                2 Dir(s)  53,342,081,024 bytes free  
  
C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>
```

Gambar 2: Tampilan Command Prompt saat berada di folder tempat Latihan1.java berada

Lakukan kompilasi pada file Latihan1.java dengan menggunakan perintah **javac Latihan1.java**, kemudian lanjutkan dengan menjalankannya dengan menggunakan perintah **java Latihan1**. Hasilnya akan seperti pada Gambar 3.



```
C:\Windows\System32\cmd.exe

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>javac Latihan1.java

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>java Latihan1
Selamat, anda berhasil meng-compile
dan menjalankan program ini!

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>
```

Gambar 3: Kompilasi dan menjalankan program yang ditulis dengan Java

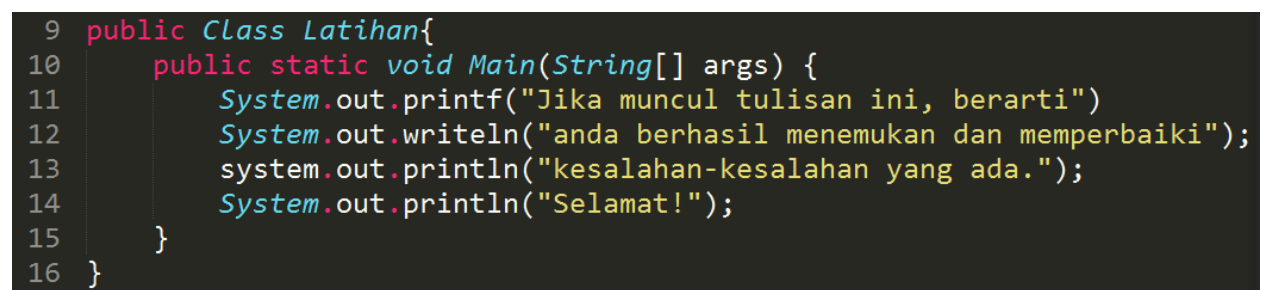
Beberapa hal yang perlu diperhatikan:

1. Anda dapat meletakkan file Latihan1.java di mana saja, tidak perlu persis sama seperti yang ditunjukkan pada Gambar 2 dan Gambar 3.
2. Untuk dapat memanggil perintah **javac** dan **java**, perlu melakukan pengaturan path terlebih dahulu.
3. Kompilasi dilakukan dengan menggunakan perintah **javac**, yang diikuti dengan nama file Java yang akan dikompilasi. Misalnya: **javac Latihan1.java**. Jika kompilasi berhasil, tidak akan muncul pesan kesalahan. Jika proses kompilasi gagal, akan muncul beberapa pesan kesalahan. Kesalahan-kesalahan tersebut harus diperbaiki terlebih dahulu sebelum bisa menjalankan program tersebut.
4. Hasil kompilasi berupa file **.class**, yang merupakan format bytecode dari Java. Untuk menjalankannya, gunakan perintah **java Latihan1**. Ingat, saat menjalankan tidak perlu menggunakan nama file lengkap, seperti: **java Latihan1.class** (salah!).

Jika anda berhasil mengkompilasi dan menjalankan Latihan1.java, selamat! Silahkan lanjut ke latihan-latihan berikutnya.

## LATIHAN 2

Anda diberi source code Java dalam file Latihan2.java, yang isinya seperti pada Gambar 4.



```
9 public Class Latihan{
10     public static void Main(String[] args) {
11         System.out.printf("Jika muncul tulisan ini, berarti")
12         System.out.writeln("anda berhasil menemukan dan memperbaiki");
13         system.out.println("kesalahan-kesalahan yang ada.");
14         System.out.println("Selamat!");
15     }
16 }
```

Gambar 4: Source code Latihan2.java

Jika anda mencoba kompilasi dengan perintah **javac Latihan2.java**, akan muncul beberapa kesalahan seperti terlihat pada Gambar 5.

```

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>javac Latihan2.java
Latihan2.java:9: error: class, interface, or enum expected
public Class Latihan{
    ^
Latihan2.java:10: error: class, interface, or enum expected
    public static void Main(String[] args) {
                ^
Latihan2.java:13: error: class, interface, or enum expected
        system.out.println("kesalahan-kesalahan yang ada.");
                ^
Latihan2.java:14: error: class, interface, or enum expected
        System.out.println("Selamat!");
                ^
Latihan2.java:15: error: class, interface, or enum expected
    }
    ^
5 errors

```

*Gambar 5: Ditemukan 5 kesalahan saat mengkompilasi Latihan2.java*

Jika suatu saat nanti anda mengkompilasi suatu source code Java dan menemukan banyak kesalahan, tidak perlu panik apalagi putus asa. Secara umum kompiler Java memberikan informasi-informasi terkait kesalahan-kesalahan yang ditemukan saat kompilasi. Yang perlu anda ingat, **telusuri kesalahan-kesalahan dan perbaiki mulai dari kesalahan pertama**, jangan mulai dari kesalahan terakhir. Dalam beberapa kasus, kesalahan di baris-baris awal program akan menyebabkan baris-baris berikutnya menjadi salah.

Lakukan perbaikan kesalahan-kesalahan pada Latihan2.java, sehingga saat dikompilasi dan dijalankan akan menampilkan output seperti pada Gambar 6.

```

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>javac Latihan2.java

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>java Latihan2
Jika muncul tulisan ini, berarti
anda berhasil menemukan dan memperbaiki
kesalahan-kesalahan yang ada.
Selamat!

C:\Users\Yuan Lukito\Documents\GitHub\PBO\pertemuan-01\latihan-mandiri>

```

*Gambar 6: Hasil kompilasi dan output yang dihasilkan setelah Latihan2.java berhasil diperbaiki.*

Beberapa hal yang perlu diperhatikan:

1. Nama file .java harus sama dengan nama class-nya. Nama class selalu diawali dengan huruf besar, usahakan nama file source code Java anda juga disimpan dengan nama yang diawali dengan huruf besar. Selain itu, nama class juga tidak boleh mengandung spasi.
2. Java merupakan bahasa pemrograman yang case-sensitive, artinya huruf besar dan huruf kecil dianggap berbeda. **System** dan **system** adalah dua hal yang berbeda.
3. Lakukan perbaikan mulai dari kesalahan pertama yang ditemukan (paling atas), jangan mulai dari bawah.

Jika anda berhasil mengkompilasi dan menjalankan Latihan2.java, selamat! Silahkan melanjutkan ke latihan 3.

## LATIHAN 3

Anda diberi dua file source Java, yaitu Latihan3.java (Gambar 7) dan Stars.java.

```
public class Latihan3{
    public static void main(String[] args) {
        //definisikan object bernama star dari class Stars

        //panggil method print dari star

        //ubah property lineCount menjadi 15

        //panggil method print sekali lagi

    }
}
```

Gambar 7: Source code Latihan3.java

Anda diharapkan dapat menggunakan class Stars dari method main di class Latihan3. Ada 4 baris kode program yang perlu anda lengkapi di Latihan3.java (anda tidak perlu mengubah Stars.java):

1. Definisikan object bernama star dari class Stars. Pada langkah ini anda membuat object dari class Stars. Gunakan keyword **new** untuk membuat object (istilah lain: instansiasi).
2. Panggil method print dari object star. Gunakan **.** (titik) untuk mengakses method atau property dari suatu object.
3. Setiap object dari class Stars memiliki property lineCount yang dapat diubah. Ubah nilainya menjadi 15 dengan mengakses method setLineCount dan gunakan 15 sebagai parameternya. Method yang mengeset nilai suatu property biasa dikenal sebagai setter/mutator.
4. Sama seperti langkah 2, panggil sekali lagi method print. Setelah nilai lineCount berubah, hasil yang didapatkan juga akan berubah.

Jika anda melengkapi Latihan3.java dengan benar, saat dikompilasi dan dijalankan hasilnya akan seperti pada Gambar 8. Selamat!

