

PACKAGE

Pengertian PACKAGE

- Package adalah sarana/cara pengelompokkan dan pengorganisasian kelas-kelas dan interface yang sekelompok menjadi suatu unit tunggal dalam library.
- Package juga mempengaruhi mekanisme hak akses ke kelas-kelas di dalamnya.

Fungsi PACKAGE

mengelompokkan file kelas yang terkait (karena jenisnya, fungsinya atau karena alasan lainnya) pada direktori yang sama, dimana di dalam setiap kelasnya terdapat directive (statement java dalam code yang digunakan untuk membuat kelas) package yang mengacu pada direktori tersebut.

Pengaruh Package terhadap Method main()

- Kelas yang mengandung method main() memiliki syarat tidak berada dalam suatu package, dan hirarki posisi foldernya di atas package yang diimport.

Membuat Package

1. Mendeklarasikan dan memberi nama package.
2. Membuat struktur dan nama direktori yang sesuai dengan struktur dan nama package.
3. Mengkompilasi kelas-kelas sesuai dengan packagenya masing-masing.

Mendeklarasikan Package

- diletakkan pada bagian paling awal (sebelum deklarasi import) dari source code setiap kelas yang dibungkus package tersebut.
- Bentuk umum deklarasi package :
package namaPackage;
- Deklarasi tersebut akan memberitahukan kompilator, ke library manakah suatu kelas dikompilasi dan dirujuk.

Memberi Nama Package

- Diawali huruf kecil
- Menggambarkan kelas-kelas yang dibungkusnya
- Harus unik (berbeda dengan nama package standard)
- Merepresentasikan path dari package tersebut.
- Harus sama dengan nama direktorinya.

Contoh package standard :

- java.lang (berisi kelas-kelas fundamental yang sering digunakan).
- java.awt dan javax.swing (berisi kelas-kelas untuk membangun aplikasi GUI)
- java.io (berisi kelas-kelas untuk proses input output)

Membuat Struktur Direktori

- Package dapat bersarang di package lain, sehingga dapat dibuat hirarki package.
- Bentuk umum pernyataan package multilevel :
package namaPackage1[.namaPackage2[.namaPackage3]];
- Contoh hirarki package di JDK :
package java.awt.image;

Compile dan Run Kelas dari suatu Package

- Selanjutnya masing-masing kelas tersebut dalam package tersebut dikompilasi menjadi byte code (*.class).
- Artinya package tersebut siap digunakan.

Menggunakan Package

- Kelas yang menggunakan berada **dalam direktori (package) yang sama** dengan kelas-kelas yang digunakan. Maka tidak diperlukan import.
- Kelas yang menggunakan berada **dalam direktori (package) yang berbeda** dengan kelas-kelas yang digunakan. Maka pada awal source code di kelas pengguna harus mencantumkan :

```
import namaPackage>NamaKelas; atau  
import namaPackage.*;
```

Contoh :

```
import java.text.DecimalFormat;  
import javax.swing.*;
```

Setting Classpath

- Path hirarki package, didaftarkan sebagai salah satu nilai variabel lingkungan yang bernama Classpath.
- Classpath diset dengan aturan : berawal dari drive (C:\ atau D:\) sampai dengan satu tingkat sebelum kita mendeklarasikan package.