# Package

Overriding class dari Interface

MouseListener

Package pada Java digunakan untuk mengelompokkan kelas-kelas yang terkait (hasil compilasi program).

Bisa dipahami sebagai folder dalam direktori file.

Package digunakan untuk menghindari konflik nama, dan untuk menulis kode yang dapat dikelola dengan lebih baik.

**Paket dibagi menjadi dua kategori:**

* Paket Built-in (paket dari Java API)
* Paket Buatan Pengguna (buat paket Anda sendiri)

**Paket Built-in (paket dari Java API)**

Java class library (Java API) memiliki banyak sekali kelas yang sudah terdefinisi (Java predefined classes) yang diletakkan di dalam paket (package) untuk dapat digunakan kembali di program Java. Paket tersebut digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class).

Kelas-kelas Java yang akan digunakan di dalam program, terlebih dahulu harus diimpor beserta dengan nama paket di mana kelas tersebut berada, kecuali untuk kelas Java yang berada di paket java.lang seperti kelas JOptionPane yang berada di paket javax.swing dan kelas DecimalFormat yang berada di paket java.text.

Tabel Beberapa Paket Java

| **Paket** | **Keterangan** |
| --- | --- |
| javax.swing | Paket ini berisi kelas-kelas dan interface untuk komponen GUI (Graphical User Interface) swing Java yang menyediakan dukungan untuk tampilan grafis yang portabel. |
| javax.swing.event | Paket ini berisi kelas-kelas dan interface yang memperbolehkan penanganan event untuk komponent grafis yang terletak di paket javax.swing. |
| java.lang | Paket ini berisi kelas-kelas dan interface yang diperlukan oleh banyak program Java. Paket ini diimpor oleh kompiler ke semua program Java secara otomatis. Jadi Anda tidak perlu mengimpor lagi untuk menggunakan kelas dan interface di paket ini. |
| java.text | Paket ini berisi kelas-kelas dan interfes yang memperbolehkan program Java untuk memanipulasi angka, tanggal, karakter dan juga string. |
| java.net | Paket ini berisi kelas-kelas yang memperbolehkan program untuk berkomunikasi melalui jaringan. |

Contoh penggunaan :

File class yang akan digunakan dihubungkan terlebih dahulu dengan menggunakan perintah impor, seperti contoh berikut :

Import javax.swing



### 1. Keuntungan menggunakan paket.

Dimungkinkan seorang programer Java membuat kelas sendiri yang mempunyai nama sama seperti kelas yang sudah ada di Java API tanpa disengaja. Untuk mengatasi masalah terkait penamaan kelas, Anda dapat meletakkan kelas yang mempunyai nama sama ke dalam package yang berbeda dan mengakses kelas tersebut beserta dengan nama package nya.

Kelas-kelas Java yang mempunyai kemiripan fungsi seharusnya diletakkan di paket yang sama sehingga mempermudah penempatan dan pendistribusian. Paket juga berfungsi memberi proteksi pada kelas dan interface yang ada di dalamnya. Sebagai contoh, kelas yang dideklarasikan dengan kata kunci private, hanya dapat diakses oleh kelas-kelas lain yang berada dalam paket yang sama.

### 2. Direktori paket.

Paket sebenarnya adalah direktori untuk menyimpan file bytecode (file berekstensi .class). Anda bisa mempunyai paket yang berada di dalam paket lain. Sebagai contoh, javax.swing.JOptionPane menunjukkan bahwa kelas JOptionPane berada di paket swing dan paket swing sendiri berada di paket javax. Level jenjang paket digunakan untuk memastikan keunikan nama paket.

**Paket Buatan Pengguna (buat paket Anda sendiri)**

Perhatikan folder programJAVA pada direktori D:, seperti gambar berikut :





Buat program berikut :

Nama Class : Mobil.java

Lokasi : D:\programJAVA\



Pilih finish, kemudian tuliskan kode program berikut :



Perhatikan file mobil Mobil.java tersimpan pada direktori : D:\programJAVA\



Lakukan proses kompilasi, perhatikan di D:\programJAVA\ terdapat forlder baru dengan nama kendaraan (gambar berikut )



Berikutnya, buka folder kendaraan, maka didalamnya terdapat file Mobil.class hasil kompilasi dari file java : Mobil.java. (perhatikan gambar berikut) :



Sehingga pada saat dilihat pada folder D:\programJAVA\ terdapat folder kendaraan dan file Mobil.java



Sehingga dapat digambarkan sebagai berikut :



Bentuk di atas dari perintah :



Untuk menggunakan class mobil yang ada di package kendaraan dengan cara menggunakan perintah import seperti berikut :



Output program sebagai berikut :



Struktur file menjadi :



Program java yg mengimport Mobil.class

Latihan :

Gambarkan susunan file dari proses program berikut : (folder letak file .java di D:\programJAVA\)



**Access Modifier**

Access Modifier adalah sebuah “hak akses” yang diberikan kepada sebuah variabel/method/class dengan tujuan untuk menjaga integritas dari data tersebut ketika ingin diakses object lain. Hak akses sendiri diberikan oleh pembuat program sebagai salah satu bentuk implementasi dari OOP itu sendiri. Dengan adanya Access Modifier, kita dapat membatasi resource-resource mana saja yang dapat diakses oleh object tertentu, turunannya, ataupun oleh method tertentu.

Di bahasa JAVA sendiri ada 4 jenis access modifier yang dapat digunakan :



### 1. Public

Access modifier public mempunyai hak akses paling luas dibanding yang lainnya. Karena aksesnya sangat luas, maka access modifier ini biasanya digunakan untuk method setter getter sesuai konsep OOP.

Contoh :



### 2. Protected

Access modifier protected biasanya digunakan untuk mewariskan variabel yang ada di super class terhadap child class.

Contoh :



### 3. No Access Modifier

Sesuai namanya, hak akses yang satu ini tidak perlu dituliskan di method/variabelnya. Dengan hak akses ini, variabel/method dapat diakses dari class lain asalkan masih dalam satu package yang sama.

Contoh :



### 4. Private

Access modifier private bersifat tertutup. Sesuai dengan konsep OOP Encapsulation, maka setiap variabel wajib untuk dilindungi hak aksesnya secara langsung dari luar. Oleh karena itu, variabel diberikan hak akses private dan untuk melakukan pengaksesan/perubahan data digunakan setter getter.

Contoh :



Latihan :

Buatlah contoh program interface dan class turunannya dengan topik : **Kendaraan**