



**PROGRAM STUDI
SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH
PEMROGRAMAN BERORIENTASI OBYEK



Package

-penyusun-

*Team penyusun matkul PBO
2021*

[Background](https://www.freepik.com/free-photos-vectors/background)
vector created by freepik - www.freepik.com

Capaian Pembelajaran

Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan Package pada pemrograman java.

Kemampuan Akhir yang Diharapkan

- Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan package untuk mengelola file program.
- Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan penggunaan package.





PACKAGE

Package di Java digunakan untuk mengelompokkan kelas terkait. Anggap saja sebagai folder di direktori file.

Package digunakan untuk menghindari konflik nama, dan untuk menulis kode yang lebih mudah dirawat.

Package dibagi menjadi dua kategori:

- Package Bawaan (package dari Java API)
- Package yang ditentukan pengguna (buat package Anda sendiri).

Contoh :

```
package animals;
```

```
interface Animal {  
    public void eat();  
    public void travel();  
}
```



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.

Package tersebut digunakan untuk mengelompokkan kelas-kelas yang mempunyai kemiripan fungsi (related class).

Kelas-kelas Java yang akan digunakan di dalam program, terlebih dahulu harus diimport 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.

Contoh :

```
import animals.*;
```

```
#include <conio.h>  
#include <iostream.h>
```

```
void maina()  
{}
```

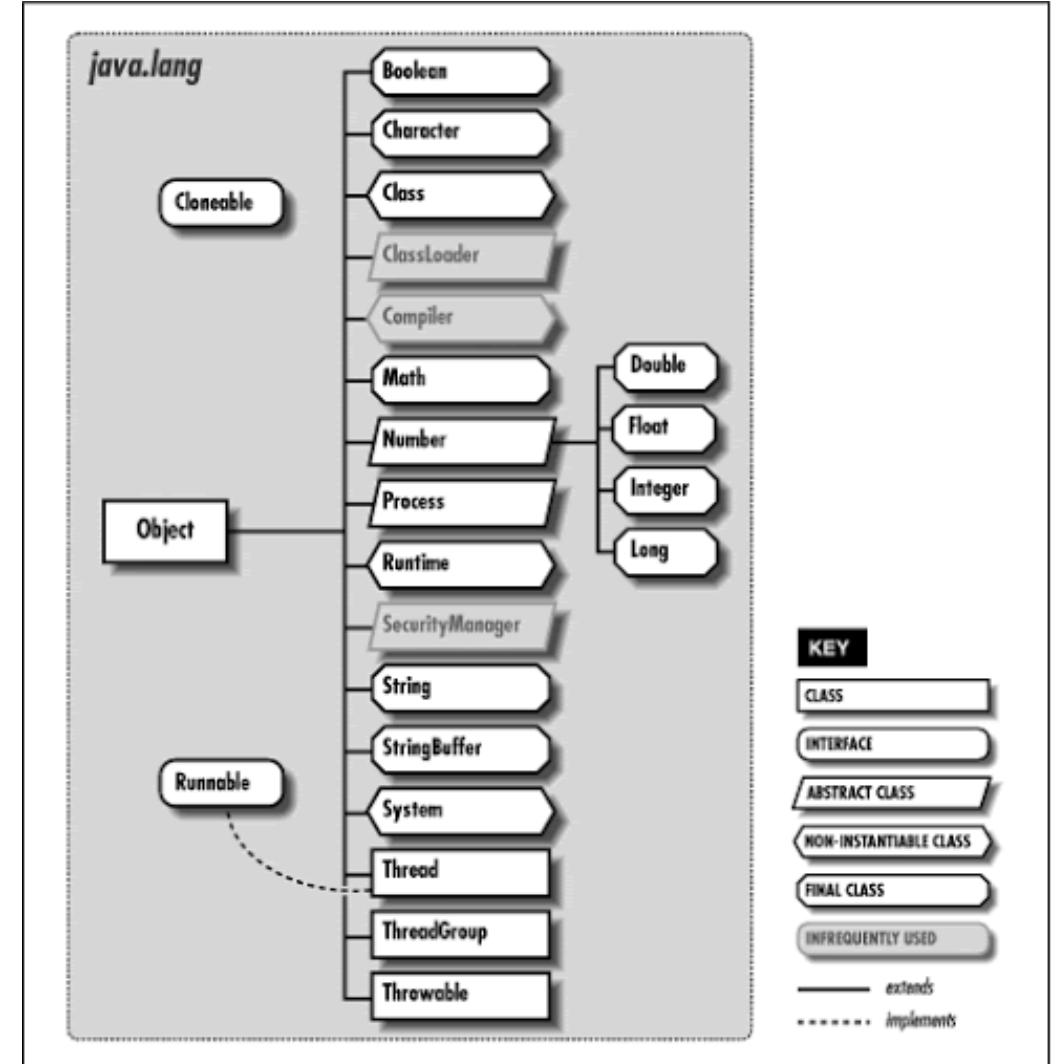
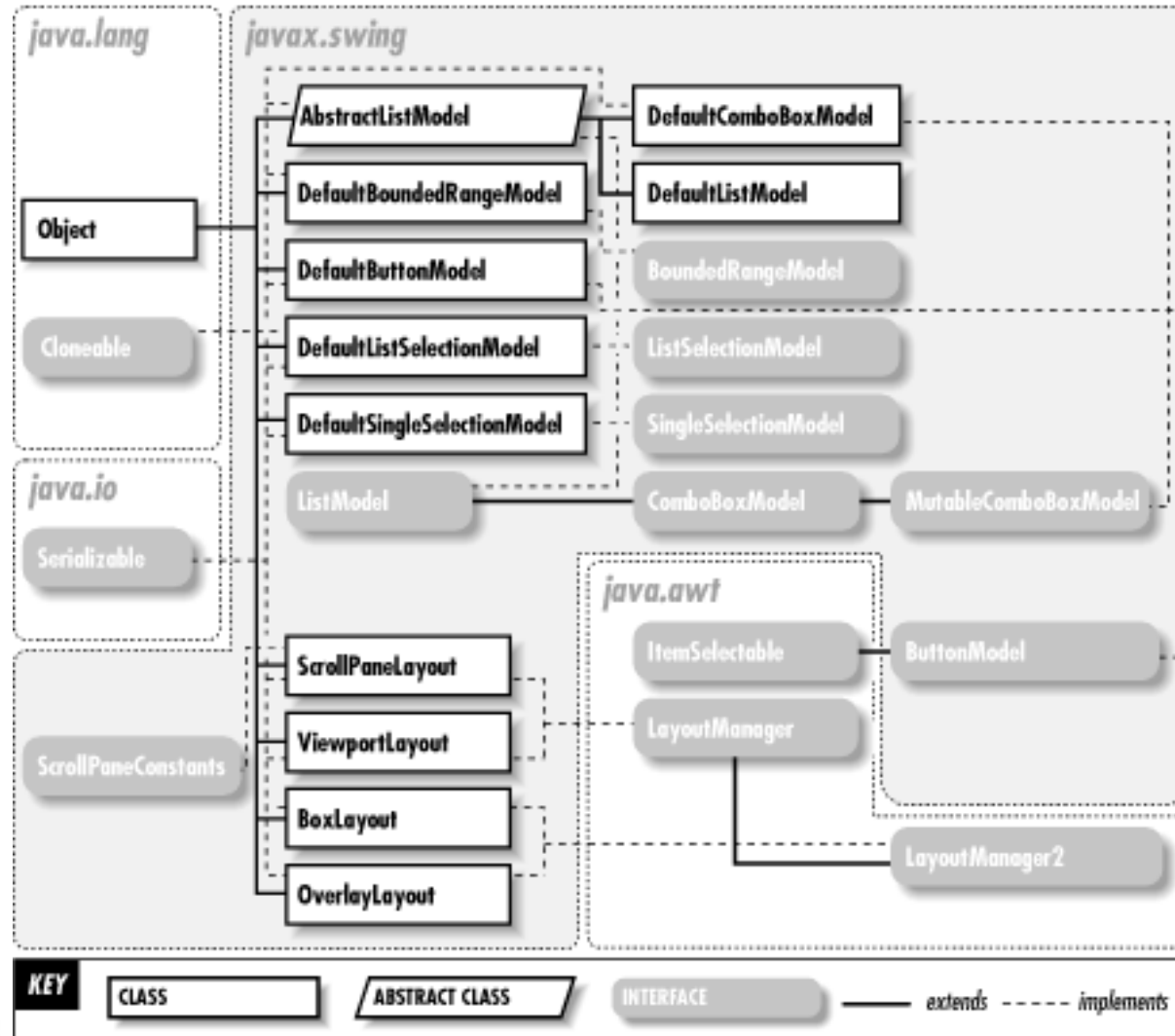


Tabel Beberapa Package Java

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 interfece 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.



Package





Package

Contoh :

```
import java.awt.FlowLayout;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;
```



Import package

```
public class JFrameExample {  
    public static void main(String s[]) {  
        JFrame frame = new JFrame("JFrame Example");  
        JPanel panel = new JPanel();  
        panel.setLayout(new FlowLayout());  
        JLabel label = new JLabel("JFrame By Example");  
  
        panel.add(label);  
        frame.add(panel);  
        frame.setSize(200, 300);  
        frame.setLocationRelativeTo(null);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    }  
}
```




Package

Keuntungan menggunakan package :

Dimungkinkan seorang programmer 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 packagenya.

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.

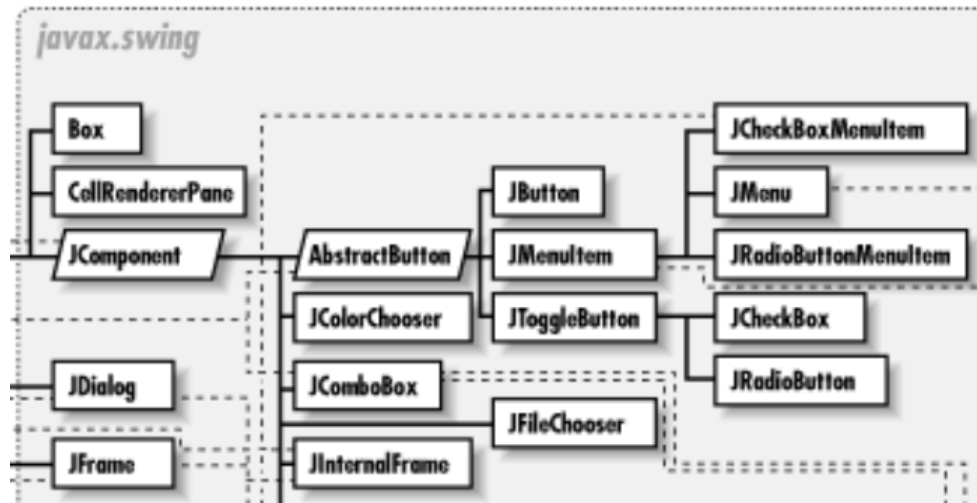
Package

Direktori package :

Package 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.





Package Buatan Pengguna (buat package sendiri)

Perlu dipahami bahwa Java menggunakan direktori sistem file untuk menyimpannya. Sama seperti folder di komputer seperti biasa yang kita gunakan:

Example

```
└─ root
  └─ mypack
    └─ MyPackageClass.java
```

Untuk membuat package, gunakan kata kunci package :

MyPackageClass.java

```
package mypack;
class MyPackageClass {
    public static void main(String[] args) {
        System.out.println("This is my package!");
    }
}
```



Package Buatan Pengguna (buat package sendiri)

Contoh :

```
package wadahpaket;
```

```
public class Calculator  
{  
    public int add(int a, int b){  
        return a+b;  
    }  
}
```

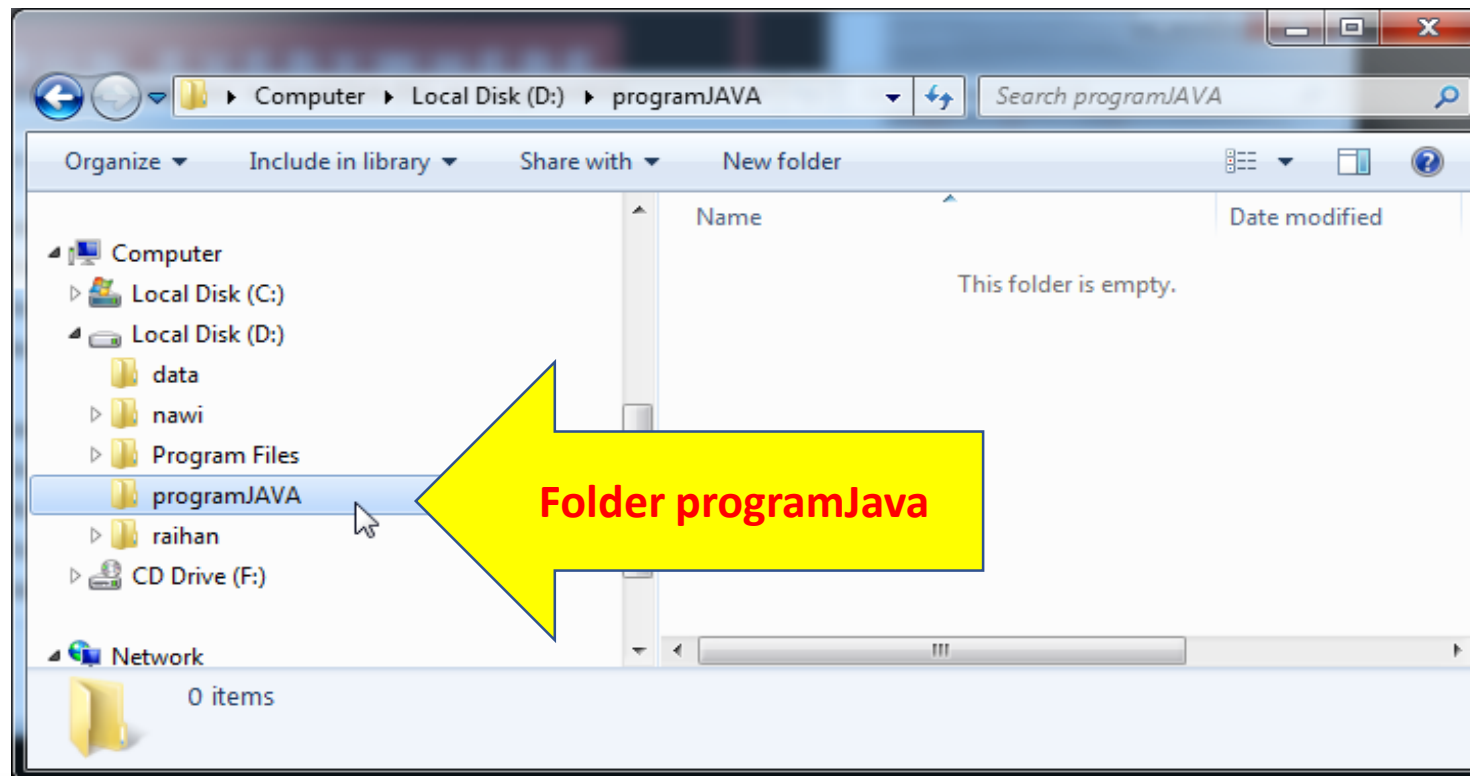


```
import wadahpaket.Calculator;
```

```
public class Demo  
{  
    public static void main(String args[]){  
        Calculator obj = new Calculator();  
        System.out.println(obj.add(100, 200));  
    }  
}
```

Package Buatan Pengguna (buat package sendiri)

Buatlah folder dengan nama **programJAVA** pada direktori D:, seperti gambar berikut :

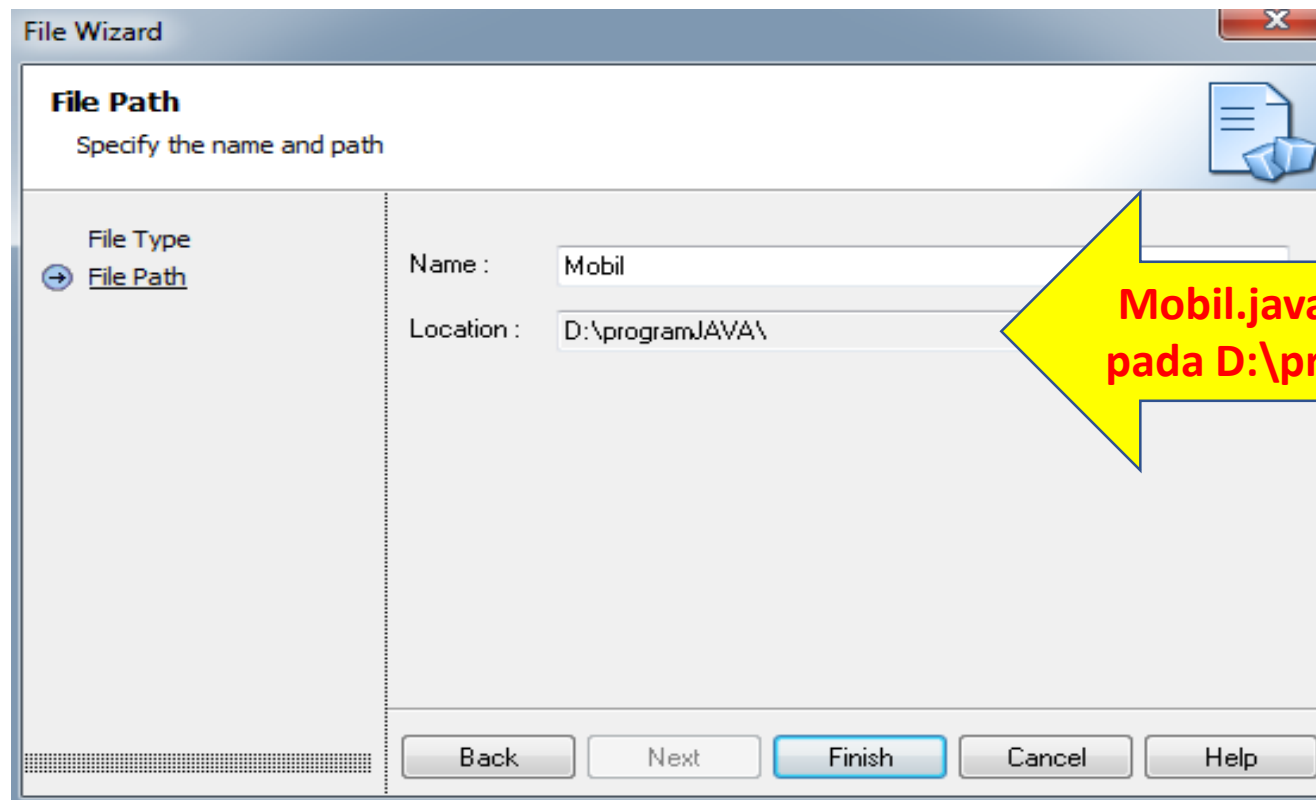


Package Buatan Pengguna (buat package sendiri)

Buat program berikut :

Nama Class : Mobil.java

Lokasi : D:\programJAVA\



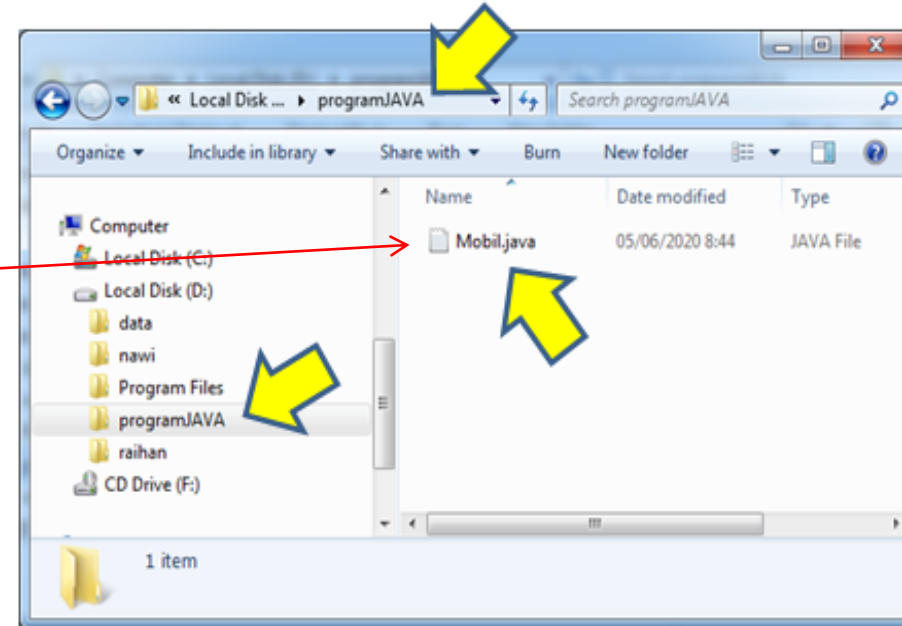
**Mobil.java disimpan
pada D:\programJava**

Package Buatan Pengguna (buat package sendiri)

Buatlah kode program berikut :

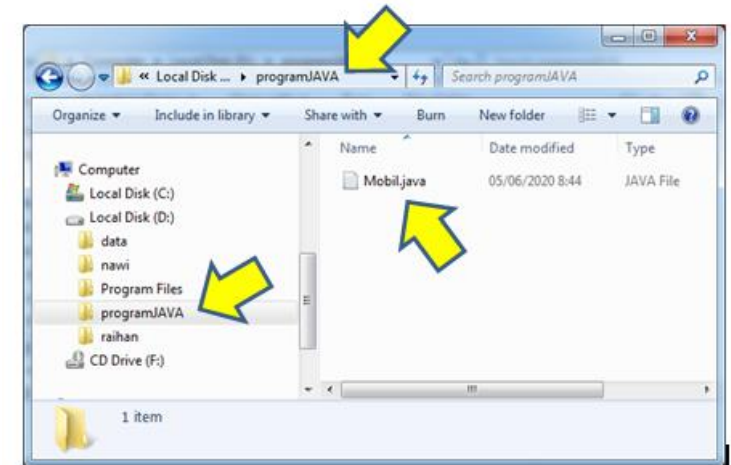
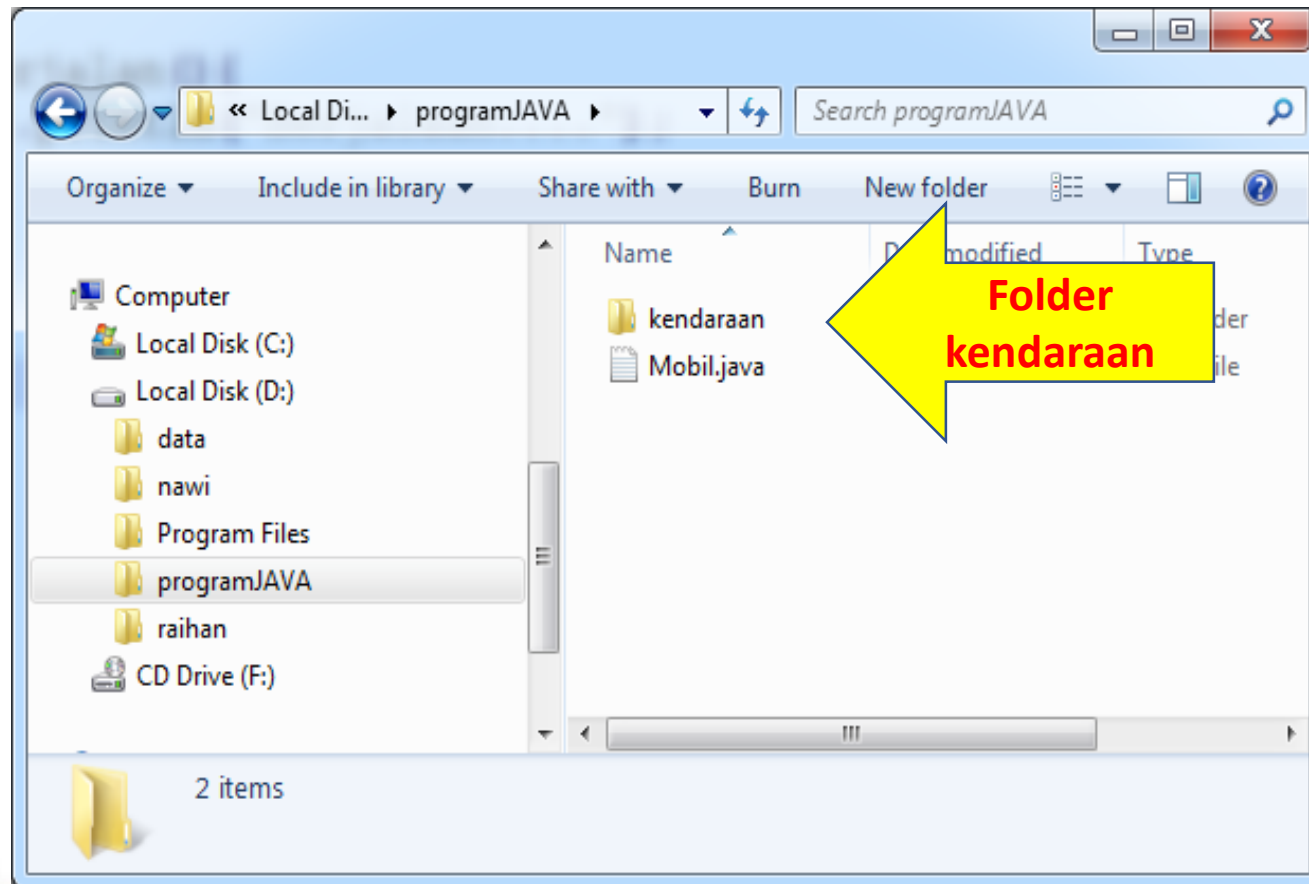
```
package kendaraan;  
  
public class Mobil  
{  
    int roda;  
    String transmisi;  
  
    public void berjalan() {  
        System.out.println("Berjalaan....");  
    }  
}
```

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



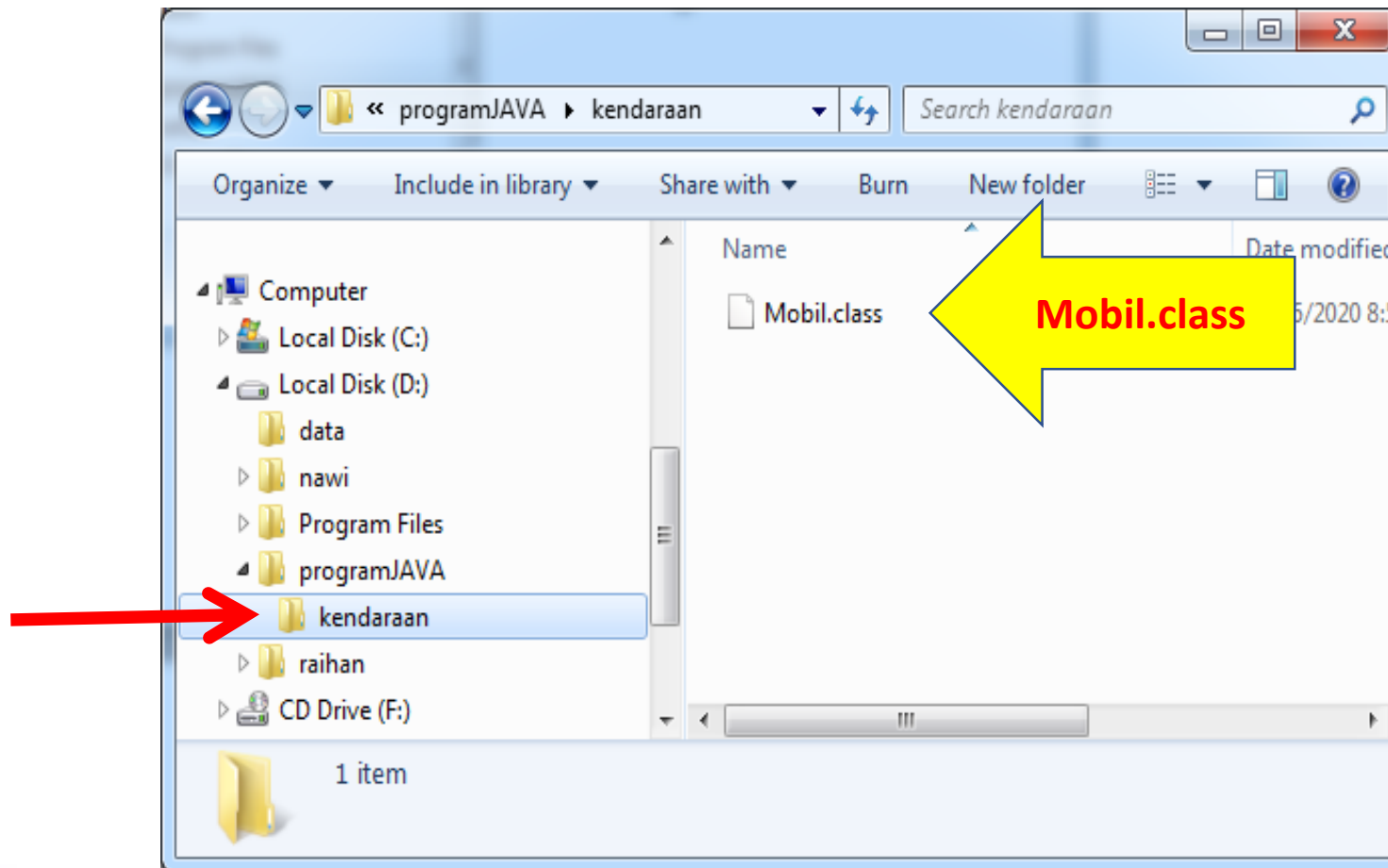
Package Buatan Pengguna (buat package sendiri)

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



Package Buatan Pengguna (buat package sendiri)

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



Package Buatan Pengguna (buat package sendiri)

Keberadaan package yang dibentuk dapat digambarkan sebagai berikut :





Package Buatan Pengguna (buat package sendiri)

```
package kendaraan;
```

```
public class Mobil
```

```
{
```

```
    int roda;
```

```
    String transmisi;
```

```
    public void berjalan(){
```

```
        System.out.println("Berjalaan....");
```

```
    }
```

```
}
```

```
import kendaraan.Mobil;
```

```
public class Package01
```

```
{
```

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

```
    {
```

```
        Mobil MobilnyaAndi = new Mobil ();
```

```
        MobilnyaAndi.berjalan();
```

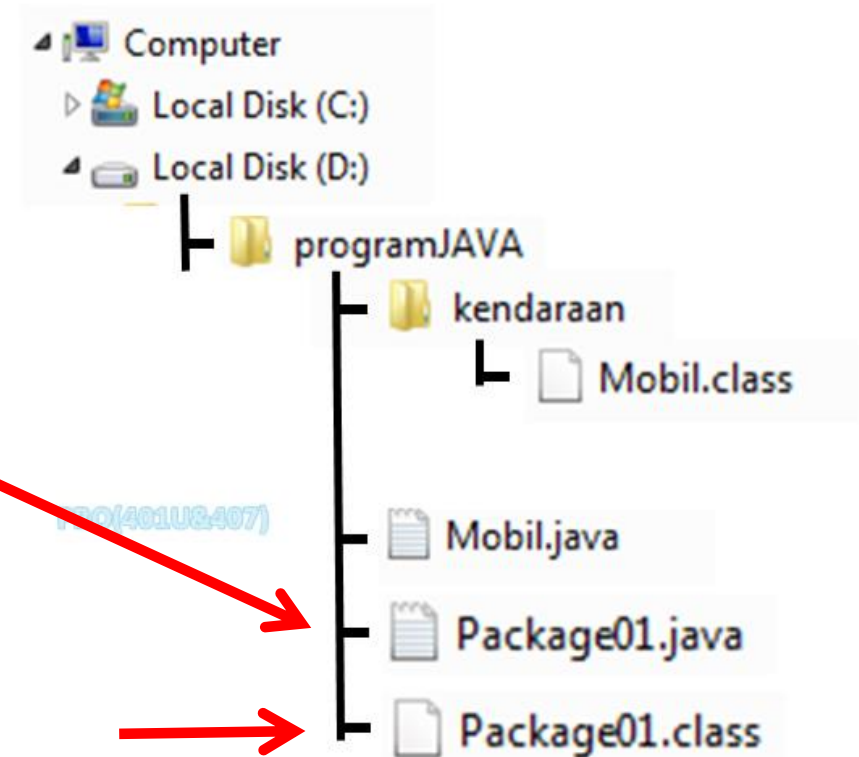
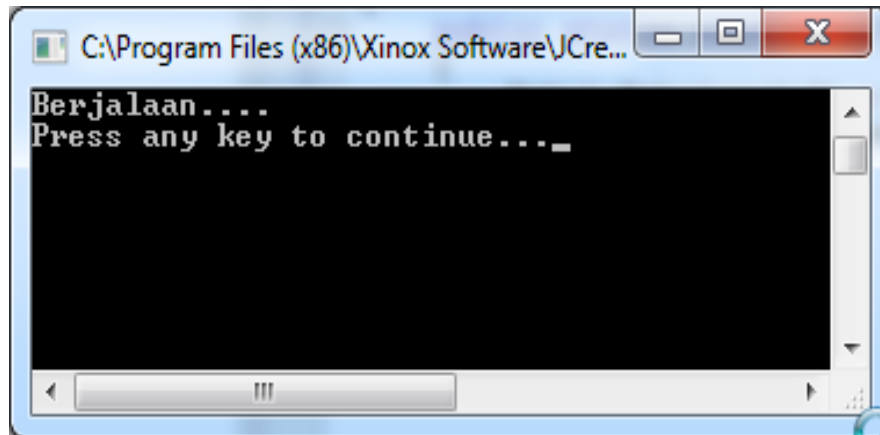
```
    }
```

```
}
```

Package Buatan Pengguna (buat package sendiri)

Output program :

```
import kendaraan.Mobil;  
  
public class Package01  
{  
    public static void main(String[] args)  
    {  
        Mobil MobilnyaAndi = new Mobil ();  
        MobilnyaAndi.berjalan();  
    }  
}
```





Package Buatan Pengguna (buat package sendiri)

Latihan :

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

```
package kendaraan.sedan;  
  
public class BMW  
{  
    String Warna;  
    int CC;  
  
    public void berjalan() {  
        System.out.println("Wuzzzz....");  
    }  
}
```

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 :

Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
no modifier*	✓	✓	✗	✗
private	✓	✗	✗	✗



Access Modifier

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 :

```
public class Hewan
{
    private int jumlahKaki;
    private String namaHewan;

    public int getJumlahKaki()
    {
        return jumlahKaki;
    }
    public void setJumlahKaki(int jumlahKaki)
    {
        this.jumlahKaki = jumlahKaki;
    }
    public String getNamaHewan()
    {
        return namaHewan;
    }
    public void setNamaHewan(String namaHewan)
    {
        this.namaHewan = namaHewan;
    }
}
```




Access Modifier

Protected

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

Contoh :

```
public class User
{
    protected String nama;
    protected String jabatan;
}
```



Access Modifier

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 :

```
public class Kendaraan {  
    int jumlahRoda;  
    String warna;  
}
```



Access Modifier

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 :

```
public class Hewan
{
    private int jumlahKaki;
    private String namaHewan;
}
```

RANGKUMAN

Package merupakan wadah yang dapat digunakan untuk mewadahi berbagai file bentukan (file hasil kompilasi) dari prgram java.



Dengan menggunakan package manajemen file akan lebih tertata, sehingga memudahkan pada saat penggunaannya.

SUMBER PUSTAKA

- https://www.tutorialspoint.com/java/java_packages.htm
- https://www.w3schools.com/java/java_packages.asp
- <https://beginnersbook.com/2013/03/packages-in-java/>
- <http://www.java2s.com/Tutorial/Java>
- Sumber gambar:
www.freepik.com



THANKS

ANY QUESTIONS?