tapi misal 'package' tidak ditulis bagaimana? apakah masih bisa diakses

Pertanyaan ini **luar biasa tajam** 😄 — karena kamu sekarang menyentuh **perilaku dasar Java** soal *package* dan *default access modifier* yang sering membingungkan pemula (padahal konsepnya penting banget).

Kita bahas pelan-pelan ya 👇

**🧩 1. Kalau package tidak ditulis sama sekali**

Contoh:

public class Mobil {

private String merk = "Toyota";

String getMerk() { // tidak public

return merk;

}

}

dan:

public class Main {

public static void main(String[] args) {

Mobil m = new Mobil();

System.out.println(m.getMerk());

}

}

➡️ **Jawabannya: Tetap bisa diakses.** ✅  
Karena… ketika kamu **tidak menulis deklarasi package sama sekali**,  
Java **secara otomatis** menempatkan file tersebut ke dalam **default package**.

Artinya:

* Kedua file (Mobil.java dan Main.java) **berada di package yang sama** — yaitu *default package*.
* Maka, method getMerk() (yang **default access**, alias tanpa modifier) **boleh diakses** oleh Main.

**📦 2. Apa itu *default package*?**

*Default package* adalah package “tak bernama” yang digunakan oleh class-class yang **tidak menyebutkan package** di awal file.

Contoh:

// Tidak ada deklarasi package

public class Mobil { ... }

dan

// Tidak ada deklarasi package juga

public class Main { ... }

➡️ Maka keduanya **otomatis berada di package yang sama** (default package).  
Jadi method dengan default access (String getMerk()) **tetap bisa diakses.**

**🚫 3. Tapi… default package punya batasan besar!**

Class di *default package*:

* ❌ **Tidak bisa diimport** oleh class yang berada di package bernama.
* ❌ Tidak direkomendasikan untuk program besar.
* ✅ Hanya cocok untuk **program kecil / latihan** yang semua filenya di folder yang sama.

Contoh error jika kamu mencoba mengimport dari default package:

// File di default package

public class Mobil { }

// File di package bernama

package app;

import Mobil; // ❌ ERROR: classes in default package cannot be imported

public class Main {

public static void main(String[] args) {

Mobil m = new Mobil();

}

}

**📚 4. Ringkasan Perilaku Akses jika package Tidak Ditulis**

| **Kondisi** | **Akses Default (String getMerk() {})** | **Keterangan** |
| --- | --- | --- |
| Kelas berada **di file yang sama tanpa package** | ✅ Bisa diakses | Karena keduanya otomatis dalam *default package* |
| Kelas berada **di folder yang sama tanpa package** | ✅ Bisa diakses | Asal dikompilasi bersamaan |
| Kelas berada **di package berbeda** | ❌ Tidak bisa diakses | Karena default access hanya berlaku dalam 1 package |
| Kelas berada **di default package**, sedangkan lainnya punya package | ❌ Tidak bisa diakses | Class di default package tidak bisa diimport |

**💬 5. Kesimpulan**

| **Situasi** | **Bisa diakses?** | **Penjelasan** |
| --- | --- | --- |
| Tidak ada package sama sekali | ✅ Ya, karena semua class otomatis di *default package* |  |
| Ada package dan class lain di package berbeda | ❌ Tidak, karena default access hanya berlaku dalam 1 package |  |
| Tambahkan public di method | ✅ Ya, bisa diakses dari mana saja |  |

**🎯 Kesimpulan Akhir**

Jika package **tidak ditulis**, maka semua class berada di **default package** yang sama,  
sehingga **method tanpa modifier (default access)** masih bisa diakses di antara class-class tersebut.

Tapi begitu kamu mulai menulis package secara eksplisit, **aturan access modifier** akan berlaku sepenuhnya:

* public → bisa diakses dari mana pun.
* *(default)* → hanya dari package yang sama.
* private → hanya dari dalam class itu sendiri.

Kalau kamu mau, aku bisa tunjukkan **contoh folder proyek Java sederhana** dengan dan tanpa package, supaya kamu bisa lihat bagaimana perbedaan aksesnya terjadi secara nyata di struktur file.  
Apakah kamu ingin saya buatkan contoh visualnya juga?