tapi jika public/private tidak ditulis dalam getter dan setter bagaimana misalnya:

String getMerk() {

return merk;

}

Nah, ini juga pertanyaan yang **sangat bagus** — kamu sudah mulai peka dengan **aturan default access modifier** di Java 👏

Jadi, kalau kamu menulis method seperti ini:

String getMerk() {

return merk;

}

…artinya kamu **tidak menulis public, private, atau protected** — dan di Java itu **bukan berarti "public" secara otomatis**.  
Yang terjadi adalah method tersebut memiliki **default access modifier**, atau disebut juga **“package-private”**.

**🧩 1. Apa itu *default* (package-private) access modifier?**

Kalau kamu **tidak menulis** kata kunci public, private, atau protected,  
Java secara otomatis memberi akses **default**, yang berarti:

Method hanya bisa diakses oleh kelas-kelas **dalam package yang sama**.

Artinya, fungsi getMerk() hanya bisa dipanggil dari **file atau class lain yang ada dalam package yang sama**.  
Kalau kamu coba panggil dari **package lain**, maka akan error.

Contohnya:

// File: Mobil.java

package kendaraan;

public class Mobil {

private String merk = "Toyota";

String getMerk() { // default access

return merk;

}

}

// File: Main.java

package mainapp;

import kendaraan.Mobil;

public class Main {

public static void main(String[] args) {

Mobil m = new Mobil();

System.out.println(m.getMerk()); // ❌ ERROR!

}

}

🟥 **Error:**  
getMerk() tidak bisa diakses dari package mainapp  
karena ia tidak bersifat public.

**✅ 2. Jadi akses “default” (tanpa modifier) berarti:**

| **Modifier** | **Akses di dalam kelas** | **Akses di package sama** | **Akses di subclass** | **Akses di luar package** |
| --- | --- | --- | --- | --- |
| public | ✅ | ✅ | ✅ | ✅ |
| protected | ✅ | ✅ | ✅ | ❌ (kecuali subclass) |
| *(default)* *(tidak ditulis)* | ✅ | ✅ | ❌ | ❌ |
| private | ✅ | ❌ | ❌ | ❌ |

**⚙️ 3. Jadi dalam konteks getter dan setter:**

Kalau kamu menulis:

String getMerk() {

return merk;

}

…maka:

* Hanya **kelas lain dalam package yang sama** yang bisa mengaksesnya.
* Kelas dari package berbeda **tidak bisa** memanggilnya.

Biasanya, getter dan setter memang ditulis **public**, karena kita ingin:

* Atribut tetap private (hanya bisa diakses lewat metode),
* Tapi **metode-nya bisa diakses dari mana pun** (termasuk class lain, atau package lain seperti Main).

**🧠 4. Analogi sederhana**

Bayangkan kamu punya rumah:

* private → hanya kamu sendiri yang boleh masuk kamar.
* *(default)* → tetangga satu kompleks boleh masuk.
* protected → anak kamu (kelas turunan) boleh masuk juga.
* public → siapa pun boleh masuk lewat pintu depan 😄

Nah, getter/setter biasanya dibuat **public**, karena mereka adalah **pintu depan** untuk akses data dari luar.

**💬 5. Kesimpulan**

| **Penulisan** | **Arti** | **Dapat diakses oleh** |
| --- | --- | --- |
| public String getMerk() | Bisa diakses dari mana saja | Semua kelas, semua package |
| String getMerk() *(tanpa modifier)* | Default / package-private | Kelas dalam package yang sama saja |
| private String getMerk() | Hanya di dalam kelas Mobil | Tidak bisa diakses dari luar |
| protected String getMerk() | Kelas dalam package sama + subclass | Subclass, meskipun di package berbeda |

🟢 **Jadi kesimpulannya:**  
Kalau kamu tidak menulis public atau private, maka **getter itu bukan public**, tapi **hanya bisa diakses dalam package yang sama**.  
Makanya di kebanyakan program Java (terutama kalau ada class Main di package berbeda), getter dan setter hampir selalu ditulis public.

Kalau kamu mau, aku bisa kasih contoh konkret di mana getter tanpa public **gagal dipanggil** dari class lain supaya kamu bisa lihat bedanya secara langsung. Mau saya tunjukkan?