



## PEMROGRAMAN BERBASIS OBJEK

Nama : Ujang Herlan

Nim : 20220040028

Kelas : TI22F

Tugas : Sesi 5

Dosen : Nugraha, M. Kom

### 1. Modifier Akses

Modifier akses digunakan untuk mengatur tingkat aksesibilitas kelas, properti, dan metode dalam sebuah program. Berikut adalah modifier akses yang umum digunakan dalam bahasa pemrograman:

Kelas:

- **Public:** Kelas dapat diakses dari mana saja dalam program.
- **Internal:** Kelas hanya dapat diakses dari dalam assembly (namespace) yang sama.
- **Protected:** Kelas hanya dapat diakses dari dalam kelas turunannya.
- **Private:** Kelas hanya dapat diakses dari dalam kelas itu sendiri.

Properti dan Method

- **Public:** Properti atau metode dapat diakses dari mana saja dalam program.
- **Internal:** Properti atau metode hanya dapat diakses dari dalam assembly (namespace) yang sama.
- **Protected:** Properti atau metode hanya dapat diakses dari dalam kelas turunannya.
- **Private:** Properti atau metode hanya dapat diakses dari dalam kelas itu sendiri.

Fungsi:

- Meningkatkan keamanan dan modularitas program.
- Mengontrol akses ke data dan fungsionalitas program.
- Membantu mencegah modifikasi data yang tidak disengaja.

### 2. Modifier Non-Akses

Modifier non-akses tidak mengatur tingkat aksesibilitas, tetapi memberikan informasi tambahan tentang kelas, properti, dan metode. Berikut adalah beberapa modifier non-akses yang umum digunakan:

Kelas:

- **Abstract:** Kelas abstrak tidak dapat diinstansiasi secara langsung, tetapi hanya dapat digunakan sebagai kelas dasar untuk kelas turunan.
- **Sealed:** Kelas sealed tidak dapat diturunkan oleh kelas lain.

Properti dan Method

- **Static:** Properti atau metode statis terkait dengan kelas itu sendiri, bukan dengan instans individual dari kelas tersebut.
- **Abstract:** Properti atau metode abstrak tidak memiliki implementasi dan harus diimplementasikan dalam kelas turunan.
- **Virtual:** Properti atau metode virtual dapat ditimpa (override) dalam kelas turunan.
- **Override:** Properti atau metode override menerima implementasi dari kelas dasar.
- **Final:** Properti atau metode final tidak dapat ditimpa dalam kelas turunan.



## Fungsi

- Meningkatkan fleksibilitas dan reusability program.
- Mengontrol perilaku kelas, properti, dan metode.
- Meningkatkan performa program.

### 3. Contoh sample kode

#### A. Library

##### **Modifier Akses untuk Kelas:**

##### 1. **Kelas Library:**

- Modifier akses: **public**
- Fungsinya: Mengatur kelas **Library** agar dapat diakses dari mana saja dalam program.

##### 2. **Kelas Book dan Member:**

- Tidak memiliki modifier akses tambahan.
- Dalam Java, jika tidak disertakan modifier akses, maka kelas memiliki akses package-private, yang berarti kelas tersebut dapat diakses hanya dari package yang sama.

##### **Modifier Akses untuk Non-Kelas (Properti dan Metode):**

##### 1. **Properti books dalam kelas Library:**

- Modifier akses: **private**
- Fungsinya: Menyimpan daftar buku secara privat, sehingga hanya dapat diakses dari dalam kelas itu sendiri.

##### 2. **Properti title dalam kelas Book dan name dalam kelas Member:**

- Modifier akses: **private**
- Fungsinya: Menyimpan informasi judul buku dan nama member secara privat, sehingga hanya dapat diakses dari dalam kelas tersebut.

##### 3. **Metode addBook dalam kelas Library:**

- Modifier akses: **protected**
- Fungsinya: Memberikan akses ke metode **addBook** hanya kepada kelas turunannya, sehingga metode ini dapat diakses dari dalam kelas turunan.

##### 4. **Metode getNumberOfBooks dalam kelas Library dan getName dalam kelas Member:**

- Modifier akses: **public**
- Fungsinya: Memberikan akses yang luas sehingga metode ini dapat diakses dari mana saja dalam program.

##### 5. **Konstruktor Book dan Member:**

- Tidak disertakan, tetapi karena tidak ada modifier akses tambahan, maka memiliki akses package-private secara default.
- Fungsinya: Membuat instance dari kelas Book dan Member hanya dapat dilakukan dari package yang sama.



## B. Bank

### 1. **Kelas Bank:**

- Modifier non-akses: **final**
- Fungsinya: Mencegah kelas Bank dari diturunkan oleh kelas lain dengan membuatnya final, sehingga tidak dapat diwarisi.

### 2. **Kelas Account:**

- Modifier non-akses: **abstract**
- Fungsinya: Mengatur kelas Account sebagai kelas abstrak, yang tidak dapat diinstansiasi secara langsung tetapi hanya dapat digunakan sebagai kelas dasar untuk kelas turunan.

### 3. **Kelas Transaction:**

- Modifier non-akses: **final**
- Fungsinya: Mencegah kelas Transaction dari dimodifikasi oleh kelas lain dengan membuatnya final, sehingga tidak dapat diturunkan atau diubah.

### **Modifier Non-Akses untuk Properti dan Metode:**

#### 1. **Properti balance dalam kelas Account:**

- Modifier non-akses: **private**
- Fungsinya: Menyimpan saldo akun secara privat, sehingga hanya dapat diakses dari dalam kelas itu sendiri.

#### 2. **Metode deposit dan withdraw dalam kelas Account:**

- Tidak memiliki modifier non-akses tambahan.
- Kedua metode ini dapat diakses dari kelas turunannya dan digunakan untuk melakukan deposit dan penarikan saldo.

#### 3. **Metode execute dalam kelas Transaction:**

- Modifier non-akses: **final**
- Fungsinya: Mencegah metode execute dari ditimpa oleh kelas turunannya, sehingga implementasinya tidak dapat diubah.