

Nama : Roi damai zalukhu

Nim : 20220040186

Kelas : TI22E

• **Modifier Access pada Kelas dan Non Kelas:**

a. **Kelas (Class):**

- `public`: Memberikan akses yang tidak terbatas ke properti atau metode tersebut dari luar kelas.
- `protected`: Memberikan akses ke properti atau metode hanya kepada kelas itu sendiri dan kelas turunannya.
- `private`: Memberikan akses ke properti atau metode hanya kepada kelas itu sendiri.

b. **Non-Kelas (Property dan Method):**

- `public`: Membuat properti atau metode dapat diakses dari luar obyek.
- `protected`: Membuat properti atau metode hanya dapat diakses di dalam kelas itu sendiri dan di dalam kelas turunannya.
- `private`: Membuat properti atau metode hanya dapat diakses di dalam kelas itu sendiri.

• **Modifier non-Access pada Kelas dan Non-Kelas:**

a. **Kelas (Class):**

- `final`: Mencegah kelas untuk diwariskan.
- `abstract`: Menandakan bahwa kelas tersebut adalah kelas abstrak dan tidak dapat diinstansiasi.

b. **Non-Kelas (Property dan Method):**

- `final`: Mencegah properti atau metode untuk di-overwrite.
- `static`: Mendefinisikan properti atau metode sebagai milik kelas itu sendiri, bukan milik instance spesifik dari kelas tersebut.

3.Contoh Kode Implementasi:

java

```

// Contoh penggunaan modifier access dan non-access pada kelas dan
properti/metode non-kelas di Java

// Kelas dengan modifier access dan non-
access public class MyClass {      private
int privateVar;      protected int
protectedVar;      public int publicVar;
static int staticVar;

    // Constructor
public MyClass() {
this.privateVar = 0;
this.protectedVar = 0;
this.publicVar = 0;
}

    // Method dengan modifier access dan non-access
private void privateMethod() {
    System.out.println("Ini adalah metode private.");
}

    protected void protectedMethod() {
        System.out.println("Ini adalah metode protected.");
    }

    public void publicMethod() {
        System.out.println("Ini adalah metode public.");
    }

    final void finalMethod() {
        System.out.println("Ini adalah metode final.");
    }

    // Metode statis
static void staticMethod() {
    System.out.println("Ini adalah metode statis.");
}
}

// Kelas turunan dari MyClass
class SubClass extends MyClass {
void accessProtectedVar() {
    // Memperbolehkan akses ke protectedVar karena SubClass adalah
subclass dari MyClass
    System.out.println("Nilai protectedVar: " + protectedVar);
}
}

// Main class untuk pengujian
public class Main {
    public static void main(String[] args) {
        // Membuat objek MyClass
        MyClass obj = new MyClass();

        // Mengakses properti dan metode dengan berbagai modifier access
obj.publicMethod(); // Output: Ini adalah metode public.
        // obj.privateMethod(); // Compile error karena privateMethod()
tidak dapat diakses dari luar kelas MyClass
    }
}

```

```
        // obj.protectedMethod(); // Compile error karena protectedMethod()
tidak dapat diakses dari luar kelas MyClass atau kelas turunannya
obj.finalMethod(); // Output: Ini adalah metode final.
        MyClass.staticMethod(); // Output: Ini adalah metode statis.

        // Mengakses properti dengan berbagai modifier access
obj.publicVar = 10;
        // obj.protectedVar = 20; // Compile error karena protectedVar
tidak dapat diakses dari luar kelas MyClass atau kelas turunannya
        // obj.privateVar = 30; // Compile error karena privateVar tidak
dapat diakses dari luar kelas MyClass
        MyClass.staticVar = 40;

        // Membuat objek SubClass
SubClass subObj = new SubClass();
        subObj.accessProtectedVar(); // Output: Nilai protectedVar: 0

} }
```