

# **LAPORAN PRAKTIKUM CODELAB MODUL 4**



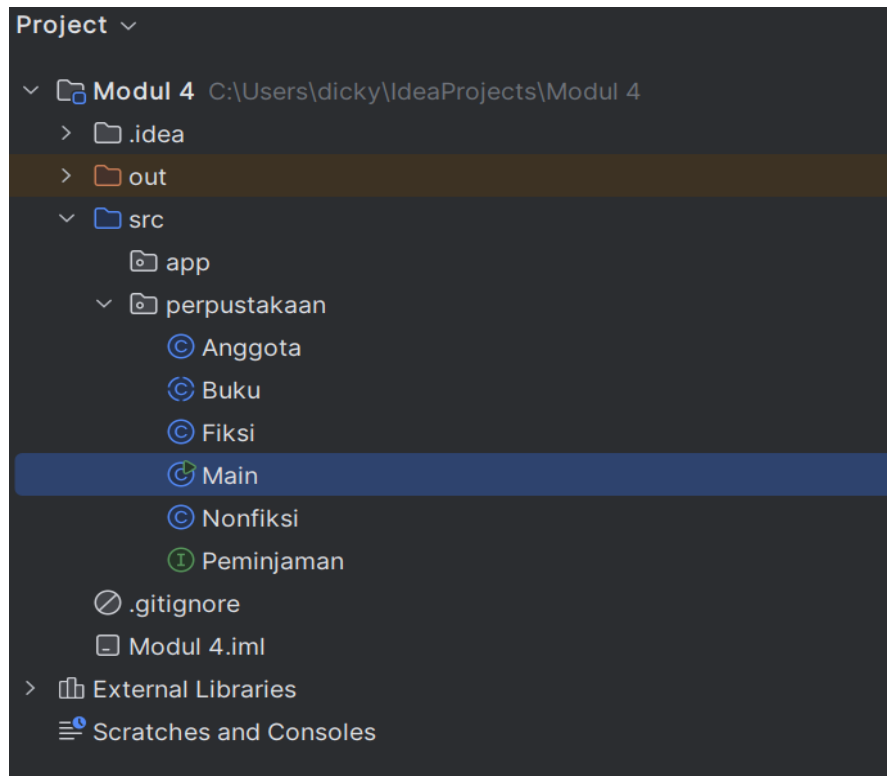
Nama: Dicky Ihsan Novanda

NIM: 202410370110426

Kelas: Informatika 2-I

### CODELAB:

1. Semua kelas harus disimpan dalam package perpustakaan (kecuali Main.java). Berikut adalah strukturnya:



2. Kelas Buku harus dibuat sebagai kelas abstrak dengan atribut judul dan penulis, serta memiliki method abstrak displayInfo()
3. Kelas Buku harus memiliki dua subclass: Fiksi dan NonFiksi, di mana masing-masing subclass mengimplementasikan method displayInfo() dengan cara yang berbeda.
4. Buatlah interface Peminjaman yang memiliki dua method: pinjamBuku() dan kembalikanBuku(). Kelas Anggota harus mengimplementasikan interface ini untuk mencetak keterangan peminjaman atau pengembalian.
5. Dalam kelas Anggota, buatlah method pinjamBuku() yang memiliki dua versi, satu menerima parameter berupa judul buku, dan satu lagi menerima parameter berupa judul dan durasi peminjaman. Kemudian buat 2 atribut yaitu:
  - String : nama
  - String : idAnggota
6. Contoh output yang diharapkan:

```
Buku Non-Fiksi: Madilog oleh Tan Malaka (Bidang: Sejarah & Ilmu Pengetahuan )
Buku Fiksi: Hainuwele: Sang Putri Kelapa oleh Lilis Hu (Genre: Dongeng)

Anggota: Wahyu Andika (ID: B075)
Anggota: Ega Faiz (ID: A047)

Wahyu Andika meminjam buku berjudul: Madilog
Ega Faiz meminjam buku "Hainuwele: Sang Putri Kelapa" selama 7 hari.

Wahyu Andika mengembalikan buku berjudul: Madilog
Ega Faiz mengembalikan buku berjudul: Hainuwele: Sang Putri Kelapa

Process finished with exit code 0
```

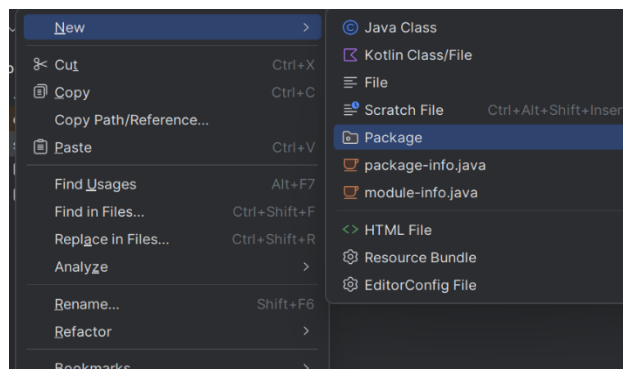
7. Catatan:

- idAnggota merupakan kelas kalian dan 3-digit nim terakhir kalian dan teman kalian
- Nama menggunakan nama kalian dan teman kalian

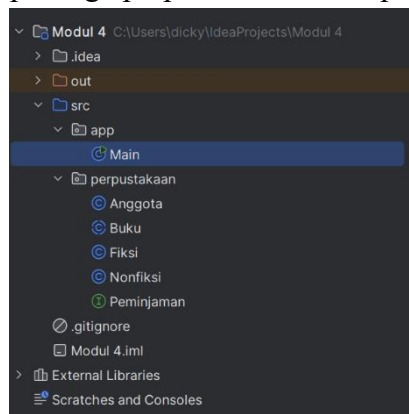
## LANGKAH-LANGKAH:

### 1. Buat Package

- Langkah pertama yang kita lakukan adalah membuat package, package adalah sebuah metode untuk mengelompokkan class dengan tujuan menghindari konflik nama class (jika ada yang bernama sama) dan memudahkan pengelolaan kode program, terutama pada aplikasi yang besar.:



- Ada dua package yang perlu di buat yaitu package app untuk class main dan package perpustakaan untuk parent class dan child class



## 2. Membuat Class

- Menurut panduan dari codelab, program yang saya buat memiliki 5 class dan 1 innterface.
  - Main class
  - Buku class
  - Anggota class
  - Fiksi class
  - NonFiksi class
  - Peminjamam (Interface)

## 3. Membuat Program di dalam class

- Buku class (Parent class)

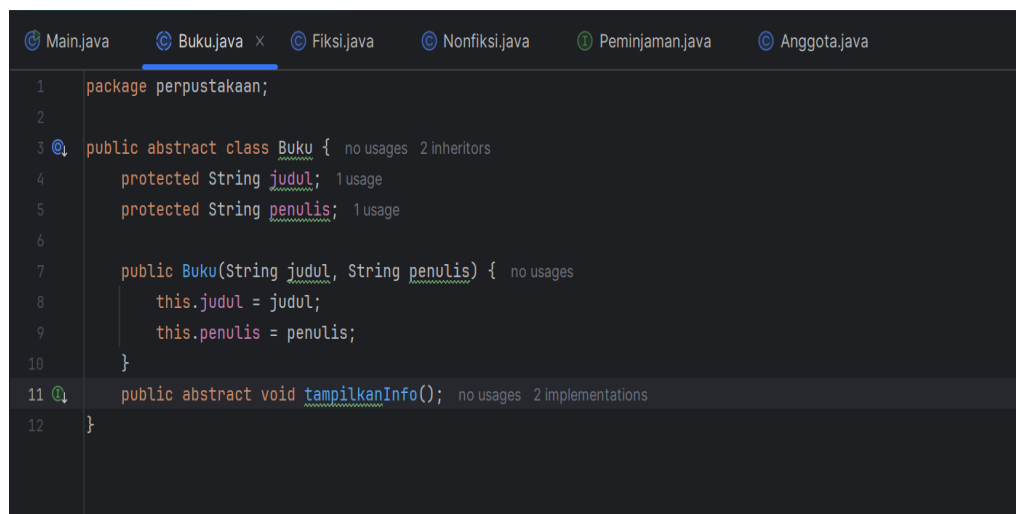
Buku class merupakan parent class yang memiliki attribute dan method untuk nantinya diturunkan ke child class. Kelas buku adalah kelas abstrak artinya:

- Tidak bisa dibuat objek langsung dari kelas ini.
- Digunakan sebagai template bagi subclass-nya.
- Harus memiliki setidaknya satu metode abstrak, yaitu metode yang tidak memiliki implementasi.

Atribut yang digunakan antara lain:

- judul => untuk menyimpan judul buku.
- penulis => Untuk menyimpan nama penulis.

Didalam atribut menggunakan kata kunci protected fungus=innya untuk dapat diakses oleh class turunannya nanti. Dalam Kelas buku juga ada constructor untuk menginisialisasi variable judul dan penulis.



```

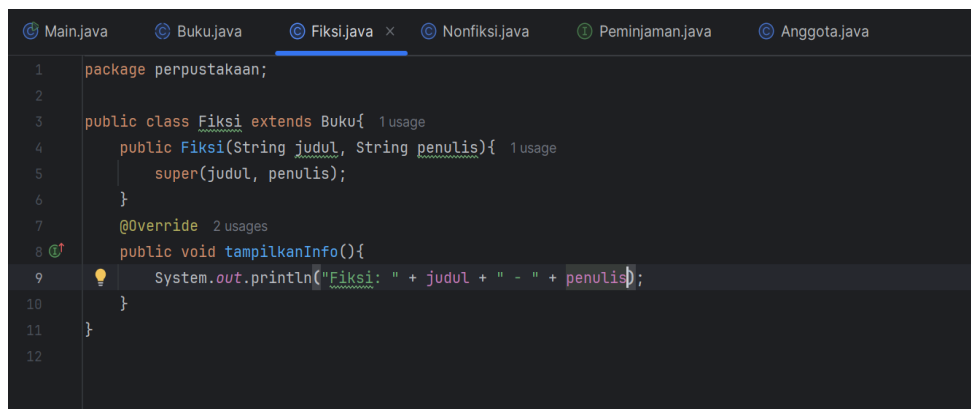
1 package perpustakaan;
2
3 public abstract class Buku { no usages 2 inheritors
4     protected String judul; 1 usage
5     protected String penulis; 1 usage
6
7     public Buku(String judul, String penulis) { no usages
8         this.judul = judul;
9         this.penulis = penulis;
10    }
11    public abstract void tampilkanInfo(); no usages 2 implementations
12 }
  
```

### ➤ Fiksi Class

Kelas fiksi berada di package perpustakaan seperti class induknya(Buku). Fiksi adalah class turunan dari class Buku, artinya ia akan diwariskan attribute dan method dari kelas induknya dan harus mengimplementasikan abstract `tampilkanInfo()` yang sudah didefinisikan di kelas induknya.

Fiksi kelas tentunya memiliki constructor yang menerima dua parameter yaitu judul dan penulis. **`super(...)`** digunakan untuk memanggil constructor dari kelas induk (Buku), agar atribut judul dan penulis terisi.

Didalam kelas fiksi ada overriding implementasi konkret dari metode abstrak `tampilkanInfo()` di kelas Buku. Tugasnya adalah menampilkan informasi buku dengan format khusus untuk buku fiksi.



```

1  package perpustakaan;
2
3  public class Fiksi extends Buku{ 1 usage
4      public Fiksi(String judul, String penulis){ 1 usage
5          super(judul, penulis);
6      }
7
8      @Override 2 usages
9      public void tampilkanInfo(){
10         System.out.println("Fiksi: " + judul + " - " + penulis);
11     }
12 }

```

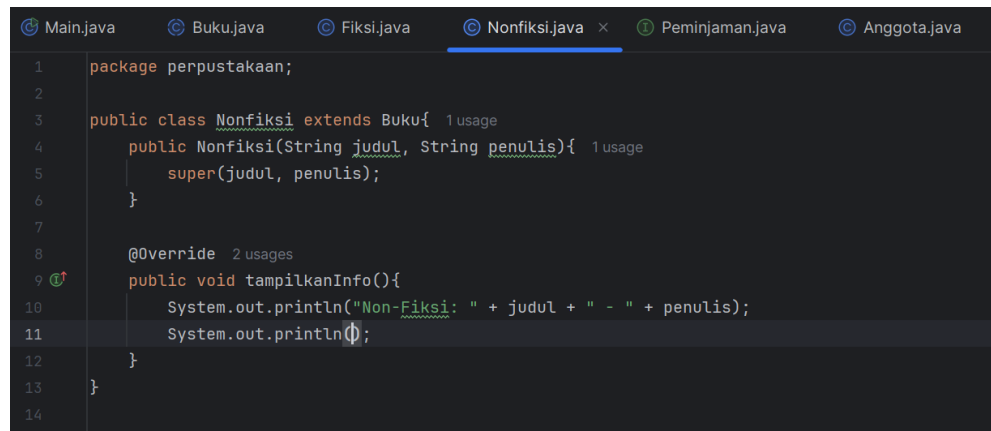
### ➤ Nonfiksi class

Seperti kelas fiksi, nonfiksi berada di dalam package perpustakaan. Kelas Nonfiksi merupakan kelas turunan dari kelas Buku. Nonfiksi mewarisi attribute dan method dari kelas Buku, seperti judul, penulis dan **`tampilkanInfo()`**. Nonfiksi tidak bersifat abstract, jadi bisa dibuat objeknya.

Adapun constructor yang menerima dua parameter yaitu judul dan penulis. **`super(...)`** memanggil constructor dari kelas Buku, agar atribut judul dan penulis diisi.

Override method pada kelas Nonfiksi mengimplementasikan dari metode abstrak `tampilkanInfo()` di kelas Buku. Tugasnya menampilkan informasi buku nonfiksi dalam format yang sesuai.

Tujuan dari kelas Nonfiksi berfungsi untuk menpresentasikan buku nonfiksi disistem perpustakaan dan juga memastikan buku nonfiksi punya cara tersendiri untuk menampilkan informasi berbeda dengan kelas Fiksi.



```

1 package perpustakaan;
2
3 public class Nonfiksi extends Buku{ 1 usage
4     public Nonfiksi(String judul, String penulis){ 1 usage
5         super(judul, penulis);
6     }
7
8     @Override 2 usages
9     public void tampilkanInfo(){
10         System.out.println("Non-Fiksi: " + judul + " - " + penulis);
11         System.out.println();
12     }
13 }
14

```

### ➤ Peminjaman (Interface)

Peminjaman merupakan sebuah interface yang digunakan untuk mendefinisikan kontrak perilaku yang harus diikuti oleh kelas yang mengimplementasikannya. Interface tidak memiliki implementasi (isi method), ia hanya berupa deklarasi.

Method yang dideklarasikan:

1. void pinjamBuku();

Method ini dideklarasikan untuk menangani proses peminjaman buku.

2. void kembalikanBuku();

Method ini dideklarasikan untuk menangani proses pengembalian buku.

Tujuan penggunaan Interface Peminjaman yaitu menyediakan kontrak umum bagi semua kelas yang ingin bisa melakukan peminjaman buku dan pengembalian buku. Mendorong konsistensi kelas seperti kelas Anggota yang mengimplementasikan interface ini yang harus menyediakan implementasi untuk **pinjamBuku()** dan **kembalikanBuku()**.

### ➤ Anggota class

Anggota adalah kelas konkret (biasa) yang bisa dibuat objeknya. Kelas ini mengimplementasikan interface Peminjaman, artinya wajib menyediakan implementasi metode **pinjamBuku()** dan **kembalikanBuku()**.

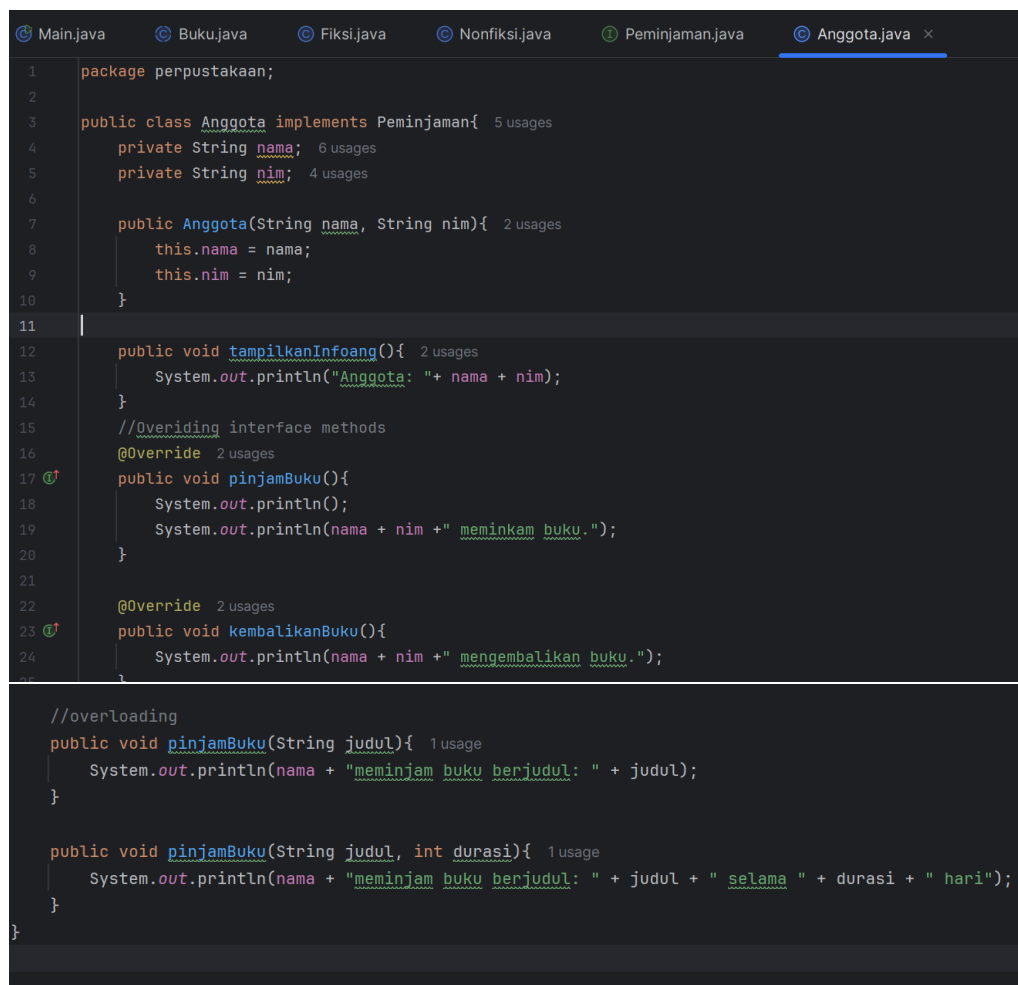
Ada pula attribute yang dipakai di kelas Anggota antara lain: nama yang menyediakan nama anggota perpustakaan, nim untuk menyimpan nim mahasiswa atau anggota, keduanya bersifat private yang hanya bisa diakses langsung dari dalam kelas Anggota.

Constructor yang digunakan untuk membuat objek Anggota dan mengisi nilai attribute nama dan nim.

Method di dalam kelas Anggota untuk menampilkan identitas mahasiswa atau anggota perpustakaan yang akan diinisialisasikan ke dalam kelas Main.

Sebagai bagian dari kontrak interface, kelas ini harus memiliki dua method yaitu **pinjamBuku()**(Overloading). Overloading method yaitu method dengan nama yang sama tapi parameternya berbeda dan juga memberika fleksibilitas saat meminjam buku, baik tanpa detail, dengan judul atau dengan lama pinjam.

Tujuan dan fungsi dari kelas Anggota yaitu merepresentasikan seseorang yang terdaftar di perpustakaan. Selain itu, menjalankan fungsi pemrosesan peminjaman dan pengembalian buku sesuai kontrak interface Peminjaman.



```

1 package perpustakaan;
2
3 public class Anggota implements Peminjaman{ 5 usages
4     private String nama; 6 usages
5     private String nim; 4 usages
6
7     public Anggota(String nama, String nim){ 2 usages
8         this.nama = nama;
9         this.nim = nim;
10    }
11
12    public void tampilkanInfoang(){ 2 usages
13        System.out.println("Anggota: " + nama + nim);
14    }
15    //Overriding interface methods
16    @Override 2 usages
17    public void pinjamBuku(){
18        System.out.println();
19        System.out.println(nama + nim + " meminjam buku.");
20    }
21
22    @Override 2 usages
23    public void kembalikanBuku(){
24        System.out.println(nama + nim + " mengembalikan buku.");
25    }
26
27    //overloading
28    public void pinjamBuku(String judul){ 1 usage
29        System.out.println(nama + " meminjam buku berjudul: " + judul);
30    }
31
32    public void pinjamBuku(String judul, int durasi){ 1 usage
33        System.out.println(nama + " meminjam buku berjudul: " + judul + " selama " + durasi + " hari");
34    }
35 }
  
```

#### ➤ Main class

Kelas Main berada di package app, terpisah dari package perpustakaan tempat kelas-kelas seperti Kelas Buku, Fiksi, Nonfiksi, dan Anggota disimpan. Dalam kelas main mengimport 4 class dari package perpustakaan, yang semuanya digunakan di dalam method **main()**.

Objek **buku1** dan **buku2** bertipe Buku, tapi masing- masing adalah instance dari subclass Fiksi dan Nonfiksi. Ini adalah contoh polymorphism yaitu objek induk (Buku) dapat mereferensikan subclass-nya.

Memanggil method **tampilkanInfo()** (Hasil override dari methode abstract). Output yang dihasilkan sesuai dengan jenis bukunya apakah Fiksi atau Nonfiksi.

Membuat object identitas dari mahasiswa atau anggota yaitu nama dan nim. Kemudian akan ditampilkan dengan menggunakan:

- anggota1.tampilkanInfoang();
- anggota2.tampilkanInfoang();

Kemudian peminjaman dan pengembalian buku akan ditampilkan dengan menggunakan:

Anggota 1:

- anggota1.pinjamBuku();
- anggota1.pinjamBuku("Si Kancil Mencuri Timun");
- anggota1.kembalikanBuku();

Anggota 2:

- anggota2.pinjamBuku();
- anggota2.pinjamBuku("Nation Geographic...", 7);
- anggota2.kembalikanBuku();

Disini saya menggunakan method overloading pada pinjamBuku() tanpa parameter, dengan parameter String(judul), dan dengan String dan int (judul & durasi peminjaman).