

**LAPORAN PRAKTIKUM CODELAB**  
**PEMROGRAMAN BERORIENTASI OBJEK 2E**  
**MODUL 4**



**Nama : Gita Fadila Mayangsari**

**Nim : 202410370110215**

**Kelas : Pemrograman Berorientasi Objek 2E**

## CODELAB

Buatlah sistem manajemen perpustakaan sederhana yang menerapkan konsep-konsep dasar pemrograman berorientasi objek dalam Java, yaitu **Package**, **Polymorphism**, **Overloading**, **Interface**, dan **Abstraction**.

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.

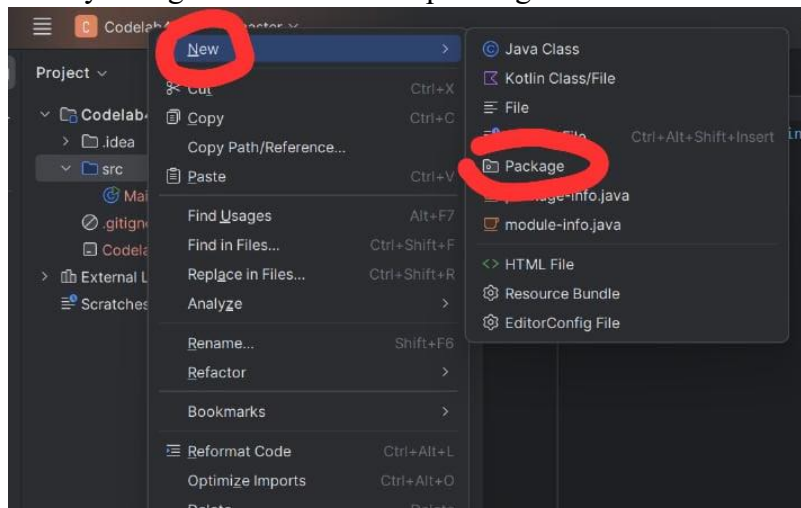
Wahyu Andika mengembalikan buku berjudul: Madilog
Ega Faiz mengembalikan buku berjudul: Hainuwele: Sang Putri Kelapa

Process finished with exit code 0
```

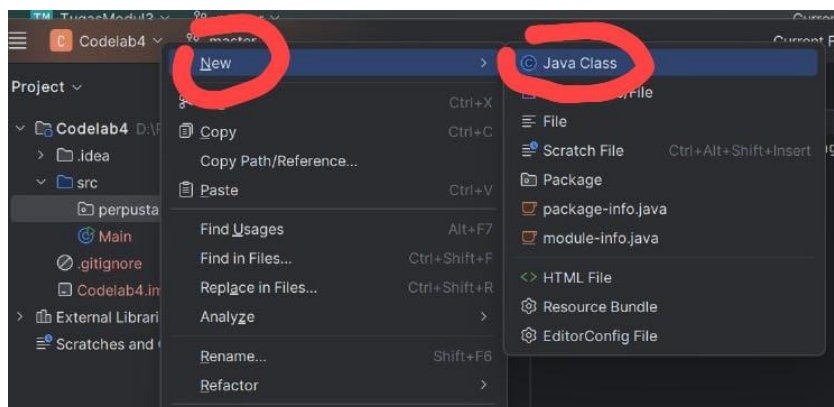
**Langkah-Langkah :**

**1. Membuat package app dan perpustakaan**

Caranya dengan klik “New” → “package”



- Pada package perpustakaan diisi dengan class abstract buku, class turunan fiksi dan non fiksi, interface peminjaman, dan class anggota
- Pada package app diisi dengan class main
- Caranya klik “New” → “Java Class”



## 2. Membuat class Abstract Buku

```
1 package perpustakaan;
2
3 public abstract class Buku { no usages 2 inheritors new *
4     protected String judul; 1 usage
5     protected String penulis; 1 usage
6
7     public Buku(String judul, String penulis) { no usages new *
8         this.judul = judul;
9         this.penulis = penulis;
10    }
11
12    public abstract void displayInfo(); no usages 2 implementations new *
13 }
14
```

- Pada class abstract ini terdapat atribut judul, penulis, dan genre.
- Terdapat juga constructor untuk memberikan nilai awal
- Terdapat method abstrak “displayInfo()” untuk diimplementasikan di class turunannya

## 3. Membuat class turunan “Fiksi” dan “Nonfiksi”

```
1 package perpustakaan;
2
3 public class Fiksi extends Buku { no usages new *
4     private String genre; 2 usages
5
6     public Fiksi(String judul, String penulis, String genre) { no usages new *
7         super(judul, penulis);
8         this.genre = genre;
9     }
10
11    @Override no usages new *
12    public void displayInfo() {
13        System.out.println("Buku Fiksi: " + judul + " oleh " + penulis + " (Genre: " + genre + ")");
14    }
15 }
16
```

```
1 package perpustakaan;
2
3 public class Nonfiksi extends Buku { 1 usage new *
4     private String bidang; 2 usages
5
6     public Nonfiksi(String judul, String penulis, String bidang) { 1 usage new *
7         super(judul, penulis);
8         this.bidang = bidang;
9     }
10
11    @Override 2 usages new *
12    public void displayInfo() {
13        System.out.println("Buku Non-Fiksi: " + judul + " oleh " + penulis + " (Bidang: " + bidang + ")");
14    }
15 }
16
```

- Pada class Fiksi dan Nonfiksi tersebut mengextends class Buku

- Pada konstruktornya menggunakan super untuk mengakses constructor dari class Buku
- Terdapat method displayinfo() yang di override karna mengimple mentasikan method abstract displayinfo() dari class Buku yang di dalam bodynya terdapat info untuk buku fiksi dan nonfiksi.

#### 4. Membuat Interface “Peminjaman”

```

1 package perpustakaan;
2
3 public interface Peminjaman { 1 usage 1 implementation new *
4     void pinjamBuku(String judulBuku); 1 usage 1 implementation new *
5     void pinjamBuku(String judulBuku, int durasi); 1 usage 1 implementation new *
6     void kembalikanBuku(String judulBuku); 2 usages 1 implementation new *
7 }
8

```

- Interface digunakan untuk mendefinisikan suatu sifat-sifat suatu class
- Pada interface tersebut terdapat 2 method yang akan diimplementa- sikan di class Anggota yaitu, method “pinjamBuku” dan “kembalikanBuku” dengan 1 parameter yang merupakan contoh **overloading**

#### 5. Membuat class “Anggota”

```

public class Anggota implements Peminjaman { 4 usages new *
    private String nama; 5 usages
    private String idAnggota; 2 usages

    public Anggota(String nama, String idAnggota) { 2 usages new *
        this.nama = nama;
        this.idAnggota = idAnggota;
    }

    @Override 1 usage new *
    public void pinjamBuku(String judulBuku) {
        System.out.println(nama + " meminjam buku berjudul: " + judulBuku);
    }

    @Override 1 usage new *
    public void pinjamBuku(String judulBuku, int durasi) {
        System.out.println(nama + " meminjam buku \" + judulBuku + \" selama " + durasi + " hari.");
    }

    @Override 2 usages new *
    public void kembalikanBuku(String judulBuku) {
        System.out.println(nama + " mengembalikan buku berjudul: " + judulBuku);
    }
}

```

- Pada class ini yaitu mengimplimentasikan isi dari interface peminjaman yang ditandai pada “public class Anggota implements Peminjaman
- Terdapat tambahan atribut, yaitu “nama” dan “idAnggota” untuk data peminjaman dan pengembalian buku
- Terdapat method yang di **override**, karena interface masih berkaitan dengan abstract

## 6. Membuat class “Main”

```
1 package app;
2
3 import perpustakaan.*;
4
5 public class Main {
6     public static void main(String[] args) {
7         Buku nonFiksi = new NonFiksi( judul: "Madilog", penulis: "Tan Malaka", bidang: "Sejarah & Ilmu Pengetahuan");
8         Buku fiksi = new Fiksi( judul: "Hainuwele: Sang Putri Kelapa", penulis: "Lilis Hu", genre: "Dongeng");
9
10        nonFiksi.displayInfo();
11        fiksi.displayInfo();
12
13        Anggota anggota1 = new Anggota( nama: "Shonia", idAnggota: "209");
14        Anggota anggota2 = new Anggota( nama: "Fadila", idAnggota: "215");
15
16        anggota1.displayInfo();
17        anggota2.displayInfo();
18
19        anggota1.pinjamBuku( judulBuku: "Madilog");
20        anggota2.pinjamBuku( judulBuku: "Hainuwele: Sang Putri Kelapa", durasi: 7);
21
22        anggota1.kembalikanBuku( judulBuku: "Madilog");
23        anggota2.kembalikanBuku( judulBuku: "Hainuwele: Sang Putri Kelapa");
24    }
25 }
```

- Class main digunakan menjalankan program yang sudah dibuat
- Untuk menjalankan package dari perpustakaan, maka harus mengimport package tersebut, dengan cara menulis “import Perpustakaan.\*”. Tanda \* untuk mengimport semua class yang ada di package tersebut.
- Terdapat 2 objek untuk class Fiksi dan Nonfiksi, dan terdapat 2 objek untuk class Anggota
- Untuk memanggil method harus menulis objek dan methodnya, contoh “buku.displayInfo()”
- Karena Anggota implements dari Peminjaman, maka harus mendeklarasikan objek dengan cara menulis nama interface kemudian objek kemudian = new Anggota dan didalam parameter diberi nilai
- Untuk memanggil method “pinjamBuku()” dan “kembalikanBuku()”, maka harus memberikan nilai didalam parameter, karena method tersebut menggunakan parameter

## 7. Hasil Output

```
Buku Non-Fiksi: Madilog oleh Tan Malaka (Bidang: Sejarah & Ilmu Pengetahuan)
Buku Fiksi: Hainuwele: Sang Putri Kelapa oleh Lilis Hu (Genre: Dongeng)
Anggota: Shonia (ID: 209)
Anggota: Fadila (ID: 215)
Shonia meminjam buku berjudul: Madilog
Fadila meminjam buku "Hainuwele: Sang Putri Kelapa" selama 7 hari.
Shonia mengembalikan buku berjudul: Madilog
Fadila mengembalikan buku berjudul: Hainuwele: Sang Putri Kelapa

Process finished with exit code 0
```

