

**LAPORAN PRAKTIKUM CODELAB PEMOGRAMAN  
BERORIENTASI OBJEK 2C MODUL 4**



**Nama: Zahra Zusseta Stiawan**

**Nim: 202410370110108**

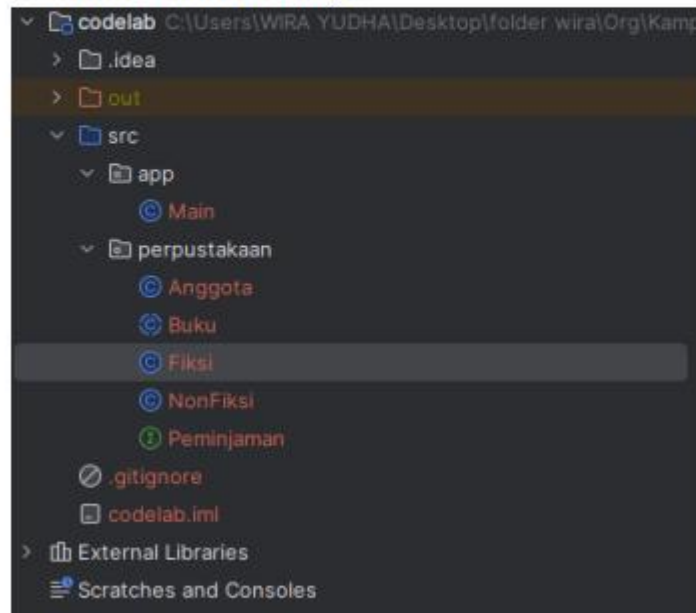
**Kelas: PBO – 2C**

## CODELAB :

### 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.

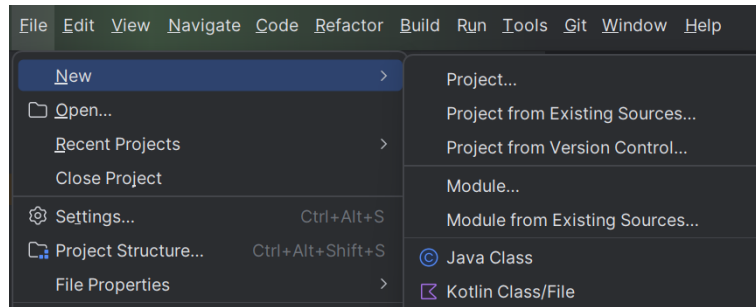
Wahyu Andika mengembalikan buku berjudul: Madilog
Ega Faiz mengembalikan buku berjudul: Hainuwele: Sang Putri Kelapa

Process finished with exit code 0
```

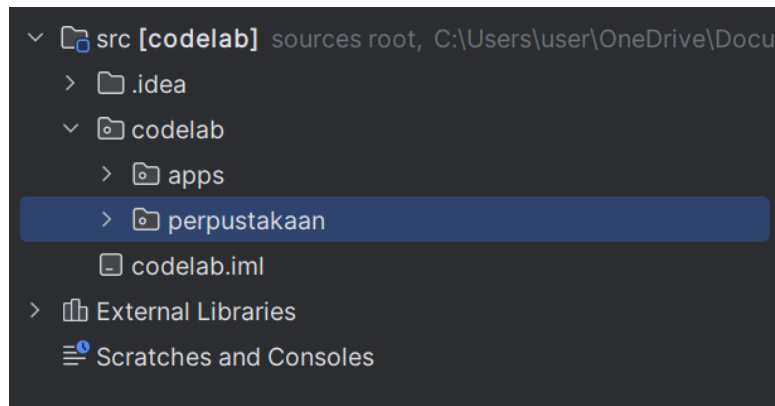
7. Catatan:
  - **idAnggota** merupakan **kelas kalian** dan **3-digit nim terakhir** kalian dan teman kalian
  - **Nama** menggunakan **nama kalian** dan **teman kalian**
  - Bagi yang tidak mengumpulkan codelab pada saat praktikum, kumpulkan di Ilab dalam bentuk laporan (pdf). Laporan harus menjelaskan step by step pengerjaan codelab. Cantumkan link repository kalian yang sudah kalian list di Spreadsheet modul 1 kemarin. Contoh laporan dapat dilihat [disini](#).

## Langkah-Langkah:

Langkah pertama kita membuat project dengan cara klik *file > New > Project*



Beri nama file baru, sebagai contoh sekarang saya beri nama *codelab.java*. lalu klik *create*. Kemudian buat struktur folder menggunakan package seperti dibawah ini



Lalu, pada *Package “perpustakaan”* buat class abstract, kemudian membuat 2 atribut, yaitu judul dan penulis, 2 atribut tersebut menggunakan tipe data *String* dan Modifier *Protected* agar atribut judul dan penulis bisa diakses langsung oleh kelas turunannya nanti tanpa menggunakan getter dan setter

Kemudian kedua atribut tersebut dimasukkan dalam *constructor*, karena kedua atribut tersebut adalah informasi penting dan wajib dimiliki, dengan meletakkannya di *constructor* maka setiap kali membuat objek Buku, program wajib memberikan nilai judul dan penulis, sehingga tidak ada buku yang tidak lengkap datanya.

```
package codelab.perpustakaan;

public abstract class Buku { 2 usages 2 inheritors
    protected String judul; 4 usages
    protected String penulis; 3 usages

    public Buku(String judul, String penulis) { 2 usages
        this.judul = judul;
        this.penulis = penulis;
    }

    public abstract void DisplayInfo(); 2 usages 2 implementations

    public String getJudul() { 4 usages
        return judul;
    }
}
```

Kemudian kita membuat class baru untuk mewarisi dari kelas Buku, disini saya akan membuat 2 subclass Bernama Fiksi dan NonFiksi.

Kelas Fiksi dan NonFiksi dibuat dengan menggunakan konsep pewarisan dari kelas abstrak Buku. Keyword `extends` digunakan untuk menunjukkan bahwa kelas tersebut merupakan turunan dari kelas Buku. Dengan pewarisan ini, Fiksi dan NonFiksi secara otomatis mewarisi atribut judul dan penulis, serta diwajibkan mengimplementasikan method abstrak `displayInfo()`.

```
package codeLab.perpustakaan;

public class Fiksi extends Buku { 2 usages 1 stealouve
    public Fiksi(String judul, String penulis) { super(judul, penulis); }

    @Override 2 usages 1 stealouve
    public void DisplayInfo(){
        System.out.println("Buku Fiksi: "+judul+" Oleh "+penulis);
    }
}
```

Constructor Fiksi menerima parameter judul dan penulis, lalu meneruskannya ke constructor Buku menggunakan keyword `super`. Method `displayInfo()` diimplementasikan ulang (override) untuk menampilkan informasi buku fiksi dengan format yang lebih spesifik, yaitu menuliskan label "Buku Fiksi".

```
package codeLab.perpustakaan;

public class NonFiksi extends Buku { 2 usages 1 stealouve
    public NonFiksi(String judul, String penulis) { super(judul, penulis); }

    @Override 2 usages 1 stealouve
    public void DisplayInfo(){
        System.out.println("Buku Non Fiksi: "+judul+" Oleh "+penulis);
    }
}
```

Sama seperti Fiksi, constructor NonFiksi meneruskan nilai atribut ke constructor superclass Buku. `displayInfo()` diimplementasikan ulang untuk menampilkan label "Buku Non-Fiksi" agar pengguna bisa membedakan jenis bukunya.

Kemudian kita membuat sebuah interface bernama Peminjaman yang mendefinisikan dua perilaku umum: `pinjamBuku()` dan `kembalikanBuku()`. Interface ini akan diimplementasikan oleh kelas Anggota.

```
package codeLab.perpustakaan;

public interface Peminjaman { 1 usage 1 implementation 1 stealouve
    void PinjamBuku(String judul); 2 usages 1 implementation 1 stealouve
    void KembalikanBuku(String judul, int hari); 2 usages 1 implementation 1 stealouve
}
```

Pada gambar diatas, ada 2 method void yang akan saya perinci lebih lanjut. Method *void pinjamBuku(String judul)* ini dideklarasikan untuk menangani proses peminjaman buku berdasarkan judul buku yang dipinjam, dan parameter berupa String judul digunakan agar metode ini bisa dipanggil dengan informasi buku yang spesifik. Kemudian bagian method *void kembalikanBuku(String judul, int hari)* digunakan untuk menangani pengembalian buku, dan sama seperti peminjaman, parameter judul digunakan untuk menyebut buku mana yang dikembalikan dan ditambahi parameter hari untuk menjelaskan berapa hari peminjaman buku tersebut.

Selanjutnya, kita membuat kelas Anggota yang berfungsi sebagai representasi pengguna/peminjam buku di sistem perpustakaan. Kelas ini mengimplementasikan interface Peminjaman.

```
package codelab.perpustakaan;

public class Anggota implements Peminjaman { 4 usages 1 stealouve
    private String nama; 4 usages
    private String idAnggota; 2 usages
    public String Info; no usages

    public Anggota(String nama, String idAnggota) { 2 usages 1 stealouve
        this.nama = nama;
        this.idAnggota = idAnggota;
    }

    public String getInfo() { return "Anggota: " + nama + " (ID: " + idAnggota + ")"; }

    @Override 2 usages 1 stealouve
    public void PinjamBuku(String judul){
        System.out.println(nama+" meminjam buku berjudul: "+judul);
    }

    @Override 2 usages 1 stealouve
    public void KembalikanBuku(String judul,int hari){
        System.out.println(nama+" mengembalikan buku berjudul: "+judul+" selama "+hari+" hari.");
    }
}
```

Pada kelas Anggota berada dalam package codelab.perpustakaan dan **mengimplementasikan interface Peminjaman**, yang berarti kelas ini wajib mengisi method-method yang dideklarasikan di interface tersebut. Kemudian kita menambahkan atribut berupa nama, idAnggota, dan info, semuanya bertipe String namun bedanya pada Modifiernya, nama dan idAnggota menggunakan modifier private agar atribut ini tidak bisa diakses langsung dari luar kelas, sebagai bentuk enkapsulasi. Sedangkan pada atribut info menggunakan public agar bisa diakses dari luar kelas.

Kemudian ada constructor yang digunakan untuk mengisi nilai awal atribut nama dan idAnggota saat objek Anggota dibuat. Method *getInfo()* memberikan informasi keanggotaan dalam format yang rapi, bisa digunakan saat mencetak identitas anggota. Ini juga bentuk enkapsulasi, di mana data pribadi anggota tidak dibuka langsung, tapi disediakan melalui method khusus.

```

@Override 2 usages stealouve
public void PinjamBuku(String judul){
    System.out.println(nama+" meminjam buku berjudul: "+judul);
}

@Override 2 usages stealouve
public void KembalikanBuku(String judul,int hari){
    System.out.println(nama+" mengembalikan buku berjudul: "+judul+" selama "+hari+" hari.");
}

```

Kedua method ini mengimplementasikan method *pinjamBuku()* dan *kembalikanBuku()* dari interface.

Langkah yang terakhir yaitu membuat file Main.java yang berfungsi untuk menguji seluruh kelas yang telah dibuat dan memastikan semua metode dan interaksi antar kelas berjalan dengan baik. Di sini kita akan membuat objek dari kelas Anggota dan memanggil method-method seperti *PinjamBuku()* dan *KembalikanBuku()*.

```

package codelab.apps;

import codelab.perpustakaan.*;

public class main {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota( nama: "Zahra Zusseta", idAnggota: "C108");
        Anggota anggota2 = new Anggota( nama: "Dhea Suin", idAnggota: "C076");

        Fiksi bukuFiksi = new Fiksi( judul: "Luka Cita", penulis: "Valerie Patkar");
        NonFiksi bukuNonFiksi = new NonFiksi( judul: "Ensiklopedia Dunia", penulis: "John Doe");

        bukuFiksi.DisplayInfo();
        bukuNonFiksi.DisplayInfo();
        System.out.println(" ");

        System.out.println(anggota1.getInfo());
        System.out.println(anggota2.getInfo());
        System.out.println(" ");

        anggota1.PinjamBuku(bukuFiksi.getJudul());
        anggota2.PinjamBuku(bukuNonFiksi.getJudul());
        System.out.println(" ");

        anggota1.KembalikanBuku(bukuFiksi.getJudul(), hari: 7);
        anggota2.KembalikanBuku(bukuNonFiksi.getJudul(), hari: 9);
    }
}

```

Kita menggunakan package codelab.apps karena kelas main ini diletakkan pada package apps, dan import diatas agar semua kelas dalam codelab.perpustakaan (Anggota, Fiksi, NonFiksi, dll) dapat digunakan.

Saya akan membuat 2 objek anggota dibuat dengan nama dan ID berbeda untuk outputnya dan constructor Anggota digunakan untuk menginisialisasi atribut nama dan idAnggota.



Lalu kita membuat objek Buku, objek Fiksi dan NonFiksi dibuat dari kelas turunan Buku, dan parameter judul dan penulis dikirim ke constructor masing-masing.

```
bukuFiksi.DisplayInfo();
bukuNonFiksi.DisplayInfo();
System.out.println(" ");

System.out.println(anggota1.getInfo());
System.out.println(anggota2.getInfo());
System.out.println(" ");

anggota1.PinjamBuku(bukuFiksi.getJudul());
anggota2.PinjamBuku(bukuNonFiksi.getJudul());
System.out.println(" ");

anggota1.KembalikanBuku(bukuFiksi.getJudul(), hari: 7);
anggota2.KembalikanBuku(bukuNonFiksi.getJudul(), hari: 9);
```

Pada bagian ini, semua nya akan ditampilkan, mulai dari informasi buku, nama dan id anggota, kemudian PinjamBuku() dipanggil untuk masing-masing anggota, dengan judul diambil dari method getJudul() milik buku, dan yang terakhir mengembalikan buku sambil mencatat lama peminjaman dalam satuan hari.

Dari semua kode kelas yang telah diperinci, akan menghasilkan output seperti dibawah ini.

```
Buku Fiksi: Luka Cita Oleh Valerie Patkar
Buku Non Fiksi: Ensiklopedia Dunia Oleh John Doe

Anggota: Zahra Zusseta (ID: C108)
Anggota: Dhea Suin (ID: C076)

Zahra Zusseta meminjam buku berjudul: Luka Cita
Dhea Suin meminjam buku berjudul: Ensiklopedia Dunia

Zahra Zusseta mengembalikan buku berjudul: Luka Cita selama 7 hari.
Dhea Suin mengembalikan buku berjudul: Ensiklopedia Dunia selama 9 hari.

Process finished with exit code 0
```

\*Link repository: [https://github.com/stealouve/PBO\\_C\\_Zeta\\_108.git](https://github.com/stealouve/PBO_C_Zeta_108.git)