

LAPORAN PRAKTIKUM CODELAB MODUL 4



Disusun oleh:

Elvira Sangadji

202410370110186

Informatika 2E

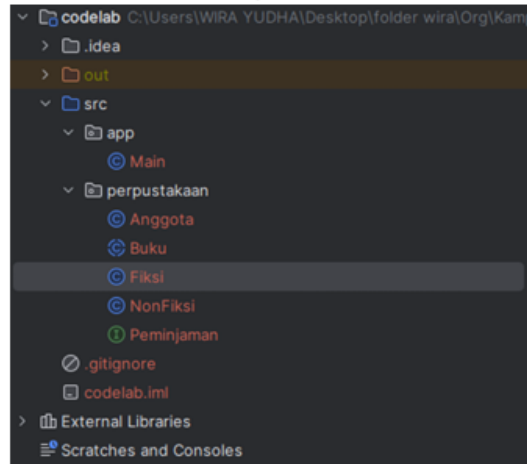
**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS
MUHAMADIYAH MALANG 2025**

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 struktur foldernya:



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
```

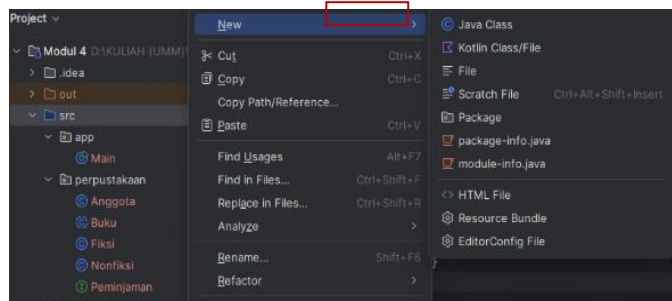
7. Catatan:

- **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:

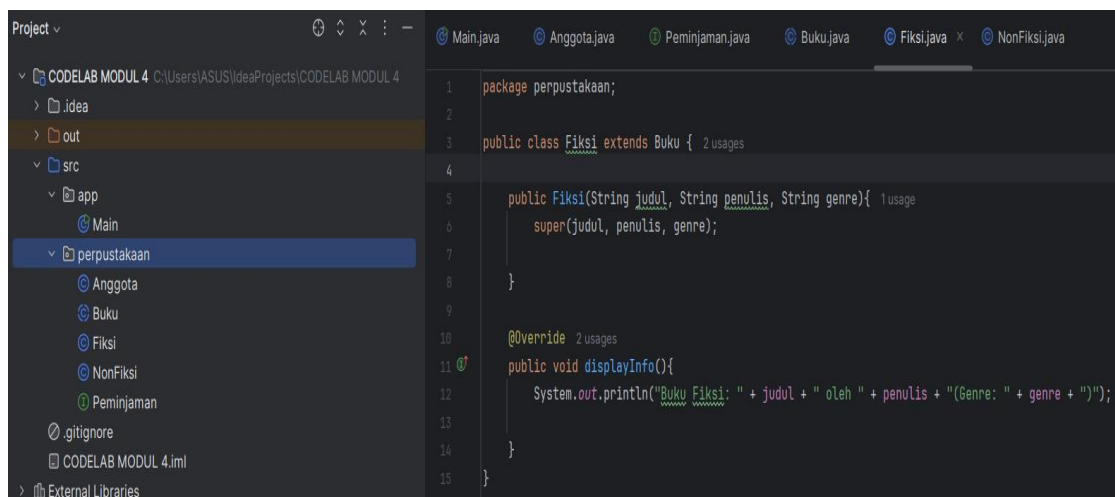
1. Membuat Package

- Langkah pertama kita harus membuat package terlebih dahulu.



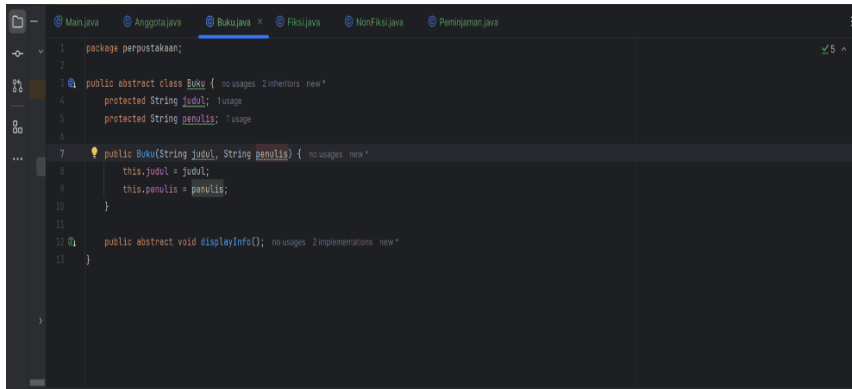
Caranya kita klik “java class” kemudian kita tulis sesuai dengan apa yang kita inginkan. Contohnya : app.Main

- Disini kita membuat 2 package, yang pertama package untuk perpustakaan dan yang kedua package app untuk class main



- Didalam package perpustakaan terdapat class buku yang memiliki subclass Fiksi dan Nonfiksi, class Anggota dan interface Peminjaman.

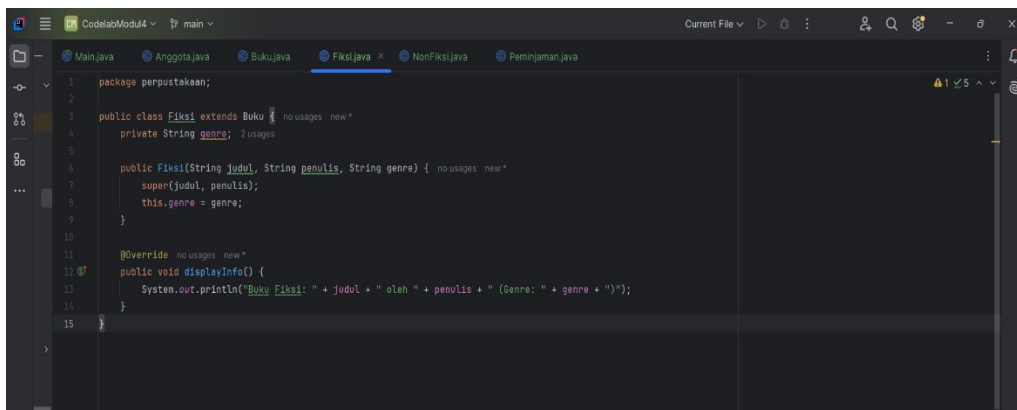
2. Membuat class Abstract Buku



```
1 package perpustakaan;
2
3 public abstract class Buku {
4     protected String judul;
5     protected String penulis;
6
7     public Buku(String judul, String penulis) {
8         this.judul = judul;
9         this.penulis = penulis;
10    }
11
12    public abstract void displayInfo();
13 }
```

- Di class ini dibuat abstract terdapat atribut judul dan penulis kemudian terdapat constructor juga untuk memberikan nilai awal
- Terdapat method abstrak “displayinfo()” untuk diimplementasikan di kelas kelas turunannya

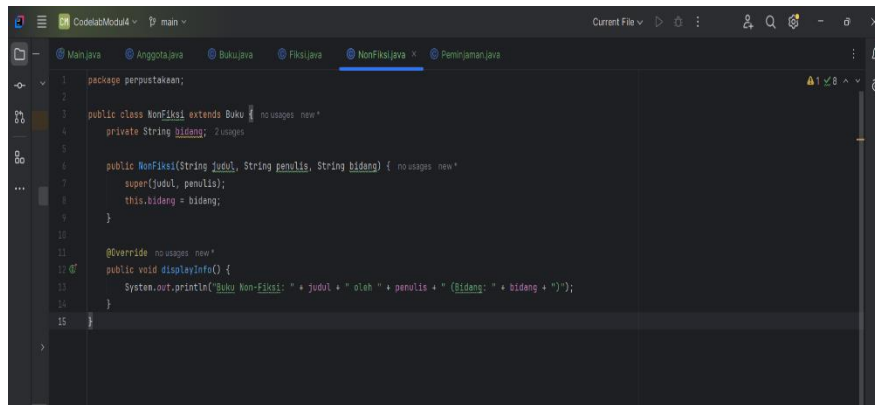
3. Membuat class turunan yaitu “Fiksi” dan “Nonfiksi”



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

- Dikelas fiksi kita mengextend Buku karna fiksi class turunannya kita memakai super untuk mengakses constructor yang ada di kelas Buku.
- Kemudian terdapat method displayinfo() yang di override karna mengimplementasikan method abstract displayinfo() dari class Buku yang di dalam bodynya terdapat info untuk buku fiksi.

- Kita lakukan sama di class nonfiksi. Perbedaan nya hanya terdapat di body method displayinfo().



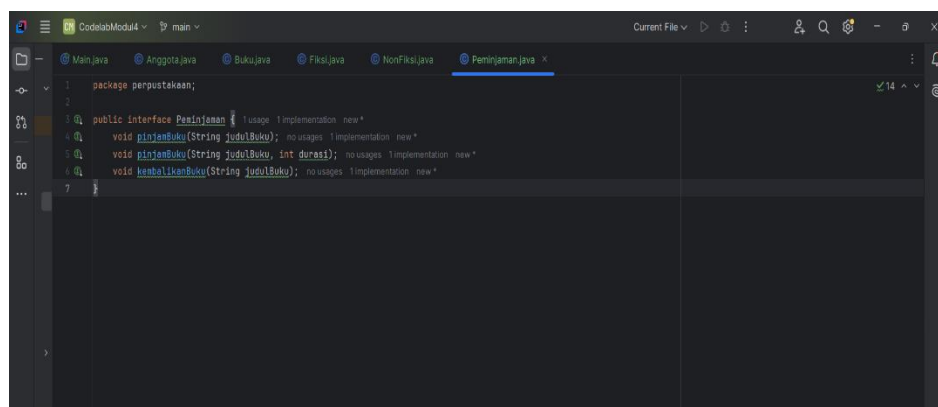
```

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

```

4. Membuat Interface Peminjaman

- Interface digunakan untuk mendefinisikan suatu sifat-sifat suatu class.
- Di interface ini kita memiliki 2 method yang akan diimplementasikan di class Anggota. Kemudian method kembalikanbuku() dengan satu parameter.



```

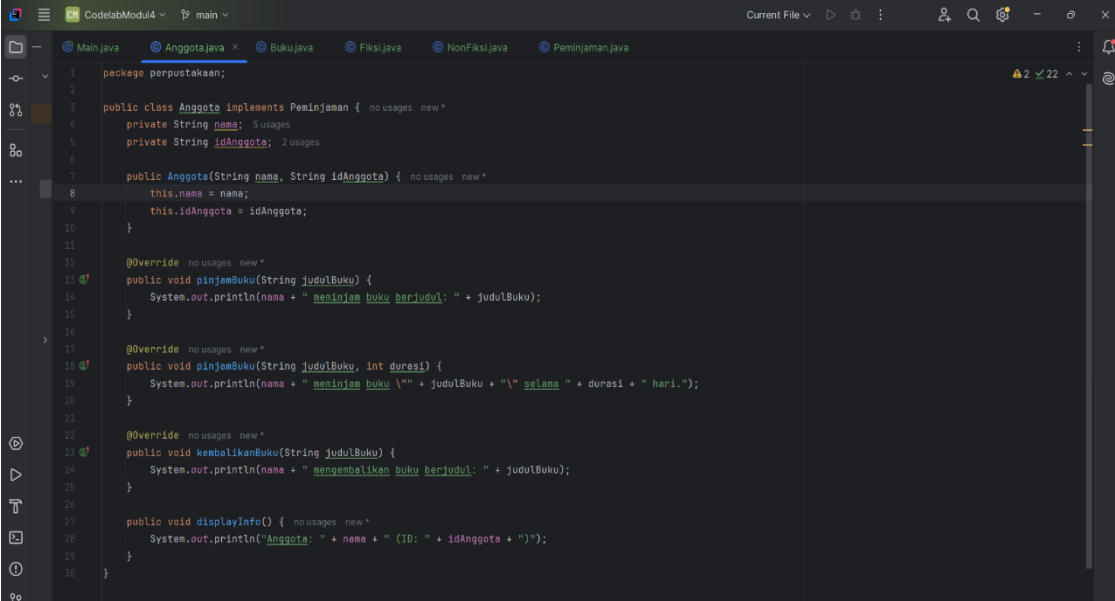
1 package perpustakaan;
2
3 public interface Peminjaman {
4     void pinjamBuku();
5     void pinjamBuku(String judulBuku);
6     void pinjamBuku(String judulBuku, int durasi);
7     void kembalikanBuku(String judulBuku);
8 }

```

- Kemudian semua method ini akan diimplementasi oleh class Anggota.

5. Class Anggota

- Di class ini kita mengimplementasi apa yang ada di dalam interface peminjaman



```

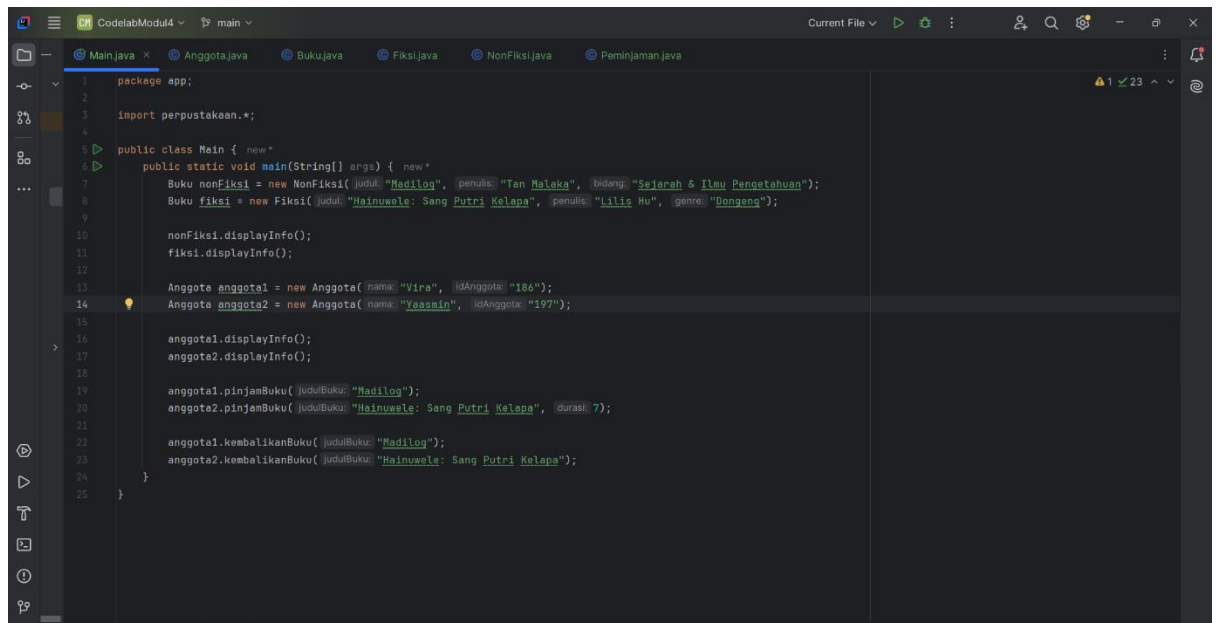
1 package perpustakaan;
2
3 public class Anggota implements Peminjaman {
4     private String nama;
5     private String idAnggota;
6
7     public Anggota(String nama, String idAnggota) {
8         this.nama = nama;
9         this.idAnggota = idAnggota;
10    }
11
12    @Override
13    public void pinjamBuku(String judulBuku) {
14        System.out.println(nama + " meminjam buku berjudul: " + judulBuku);
15    }
16
17    @Override
18    public void pinjamBuku(String judulBuku, int durasi) {
19        System.out.println(nama + " meminjam buku \"" + judulBuku + "\" selama " + durasi + " hari.");
20    }
21
22    @Override
23    public void kembalikanBuku(String judulBuku) {
24        System.out.println(nama + " mengembalikan buku berjudul: " + judulBuku);
25    }
26
27    public void displayInfo() {
28        System.out.println("Anggota: " + nama + " (ID: " + idAnggota + ")");
29    }
30 }

```

- “public class Anggota Implements Peminjaman” itu artinya kita mengimplementasi apa yang ada di dalam interface tersebut.
- Di class ini ada atribut tambahan yaitu nama dan anggota untuk mengetahui siapa yang meminjam buku dan mengembalikan buku.
- Karna interface itu masih berkaitan dengan abstrak maka kita harus meng override method dari interface Peminjaman.
- Di sini ada 4 method yang sudah di override, untuk method pinjambuku() terdapat 2, dengan 1 parameter yaitu String judul untuk mengetahui judul buku apa yang dipinjam dan dengan 2 parameter yaitu String judul dan String durasi untuk mengetahui berapa lama ia meminjam buku.

6. Membuat class Main

- Class main digunakan untuk menjalankan program yang sudah kita buat.

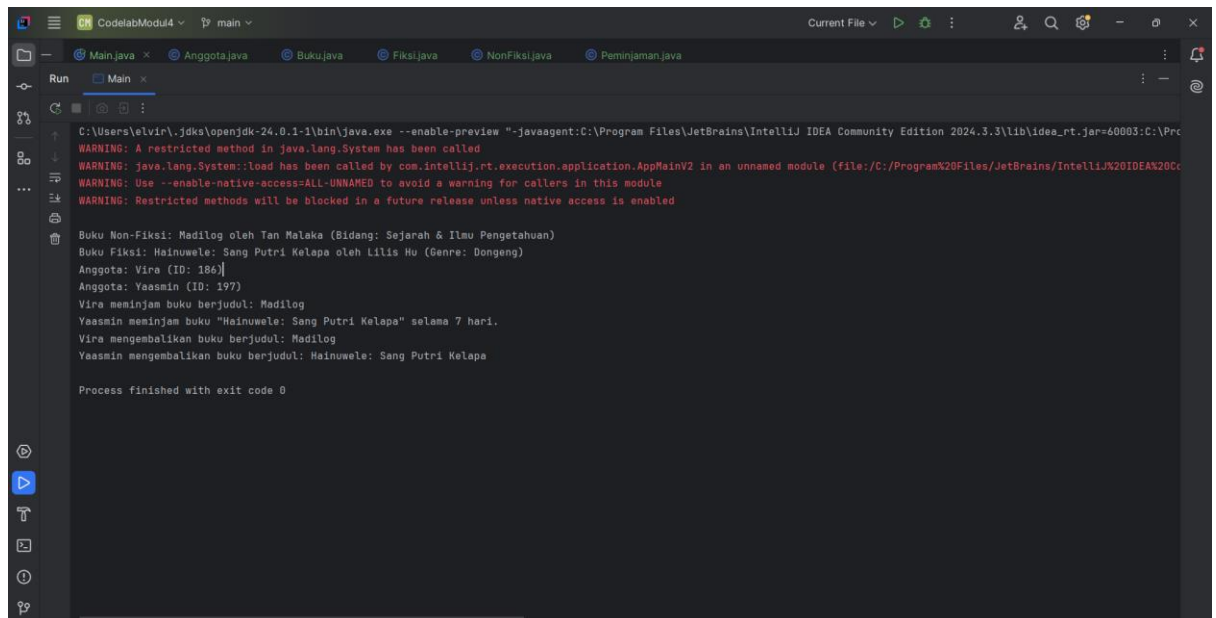


```
1 package app;
2
3 import perpustakaan.*;
4
5 public class Main {
6     public static void main(String[] args) {
7         Buku nonFiksi = new NonFiksi(judul: "Madilog", penulis: "Tan Malaka", bidang: "Sejarah & Ilmu Pengetahuan");
8         Buku fiksi = new Fiksi(judul: "Hainuwele: Sang Putri Kelapa", penulis: "Lilis Hu", genre: "Bongeng");
9
10        nonFiksi.displayInfo();
11        fiksi.displayInfo();
12
13        Anggota anggota1 = new Anggota(nama: "Vira", idAnggota: "186");
14        Anggota anggota2 = new Anggota(nama: "Yesshin", idAnggota: "197");
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 }
```

- Karna kita ingin mengeksekusi atau menjalankan program dari package perpustakaan, maka kita harus mengimport package tersebut yaitu dengan cara menulis “import perpustakaan.*”
- Tanda * karna kita ingin mengimport semua class yang ada di package perpustakaan, itu agar lebih ringkas atau agar kita tidak menuliskannya satu satu.
- Disini kita membuat 4 objek, 2 objek untuk kelas Fiksi dan Nonfiksi, dan 2 lainnya untuk kelas anggota.
- Karna anggota itu implements dari peminjaman, maka kita mendeklarasikan objek dengan cara menulis nama interface kemudian objek kemudian = new Anggota dan didalam parameter kita memberi nilai.
- Setelah itu untuk memanggil method kita tulis objek dan methodnya, contoh “buku1.displayinfo()”

- Untuk memanggil method `pinjambuku()`, kita harus memberikan nilai didalam parameter, karena method tersebut menggunakan parameter.

7. Hasil Output



```

C:\Users\elvir\.jdk\openjdk-24.0.1-1\bin\java.exe --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\lib\idea_rt.jar=60003:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\config -Didea.copyright.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\copyright -Didea.home.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\bin -Didea.platform.prefix=JDK -Djava.class.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\bin\idea_rt.jar -Djava.runtime.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.3\bin\jre -Djava.vendor=JetBrains s.r.o. -Djava.version=24.0.1-1 --enable-native-access=ALL-UNNAMED
WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::load has been called by com.intellij.rt.execution.application.AppMainV2 in an unnamed module (file:/C:/Program%20Files/JetBrains/IntelliJ%20IDEA%20Community%20Edition%2024.3.3/bin/)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this module
WARNING: Restricted methods will be blocked in a future release unless native access is enabled

Buku Non-Fiksi: Madilog oleh Tan Malaka (Bidang: Sejarah & Ilmu Pengetahuan)
Buku Fiksi: Hainuwele: Sang Putri Kelapa oleh Lilis Hu (Genre: Dongeng)
Anggota: Vira (ID: 186)
Anggota: Yaasmin (ID: 197)
Vira meminjam buku berjudul: Madilog
Yaasmin meminjam buku "Hainuwele: Sang Putri Kelapa" selama 7 hari.
Vira mengembalikan buku berjudul: Madilog
Yaasmin mengembalikan buku berjudul: Hainuwele: Sang Putri Kelapa

Process finished with exit code 0
  
```

- Output program menampilkan informasi detail dari sistem manajemen perpustakaan yang telah kita buat, dan ini menunjukkan bahwa program berhasil dan berfungsi dengan baik

https://github.com/Vira1924/Modul4_E_Vira_186