

LAPORAN PRAKTIKUM 2B MODUL 4

Widya Amelia

202410370110053

PBO/2B

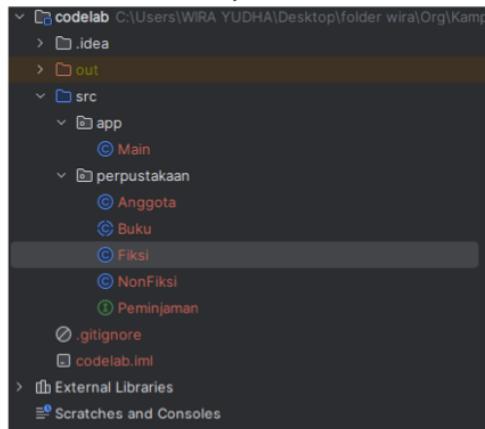
CODELAB

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:
 - o **String : nama**
 - o **String : idAnggota**

6. Contoh output yang diharapkan:

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

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

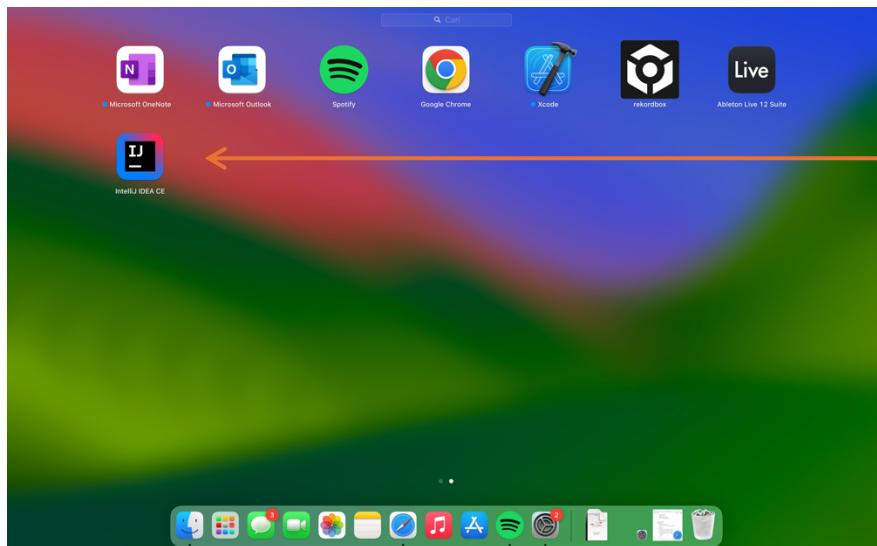
Wahyu Andika meminjam buku berjudul: Madilog
Ega Faiz meminjam buku "Hainuwele: Sang Putri Kelapa" selama 7 hari.

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

Process finished with exit code 0
```

7. Catatan:
 - o **idAnggota** merupakan **kelas kalian** dan **3-digit nim terakhir** kalian dan teman kalian
 - o **Nama** menggunakan **nama kalian** dan **teman kalian**

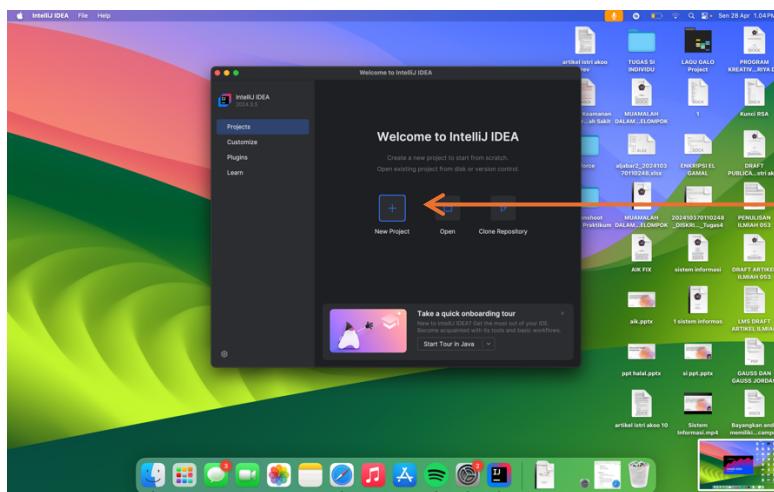
Langkah – Langkah :



Buka Aplikasi IntelliJ IDEA CE



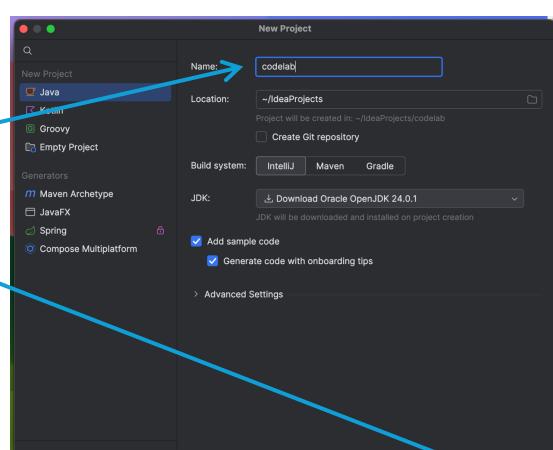
Akan muncul loading screen seperti gambar di samping



New Project

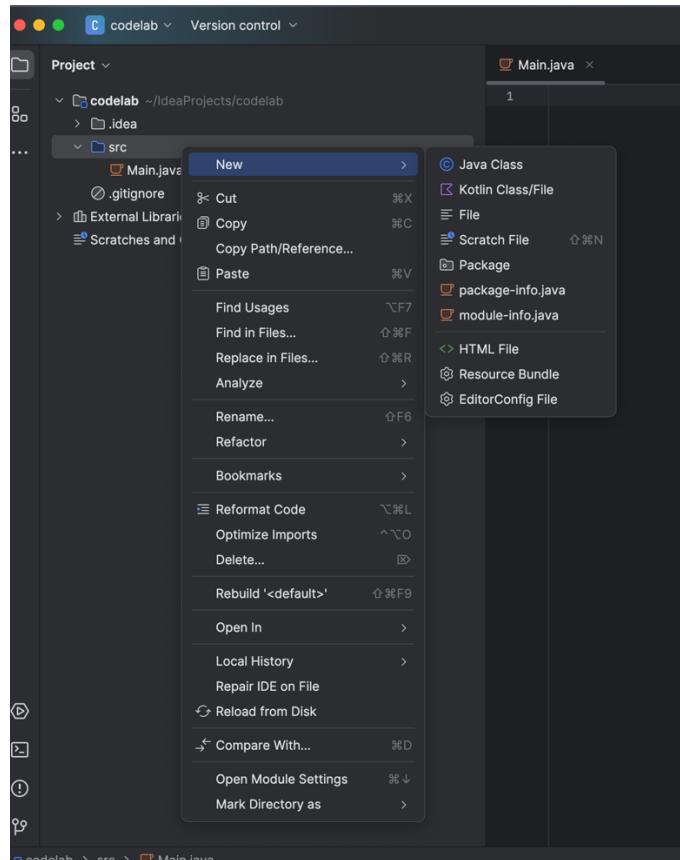
1. Langkah pertama di tampilan utama IntelliJ:

- Klik New Project
- Pilih Java
- Beri nama project misalnya codelab
- Klik Create

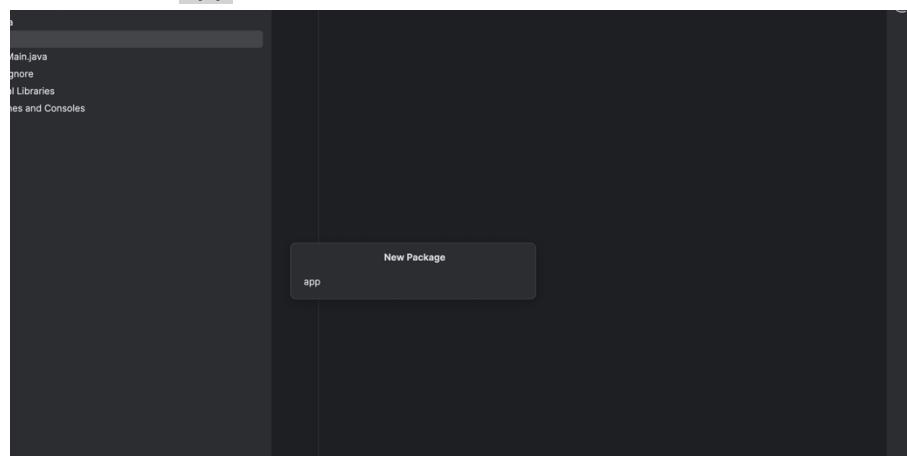


2. Membuat struktur folder

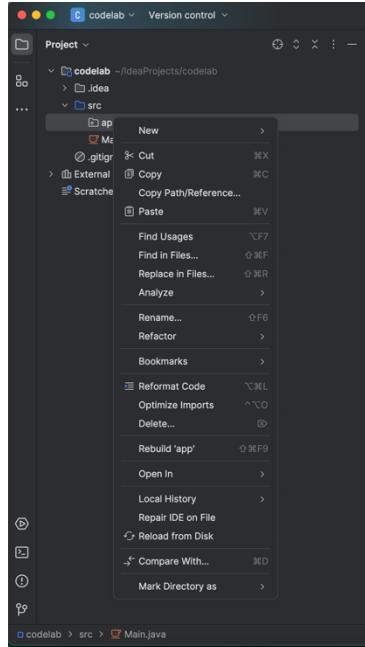
- Klik kanan di `src` → New → Package



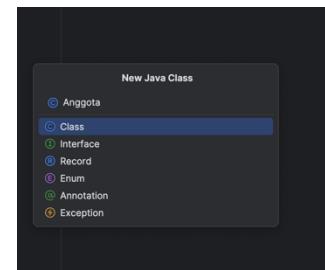
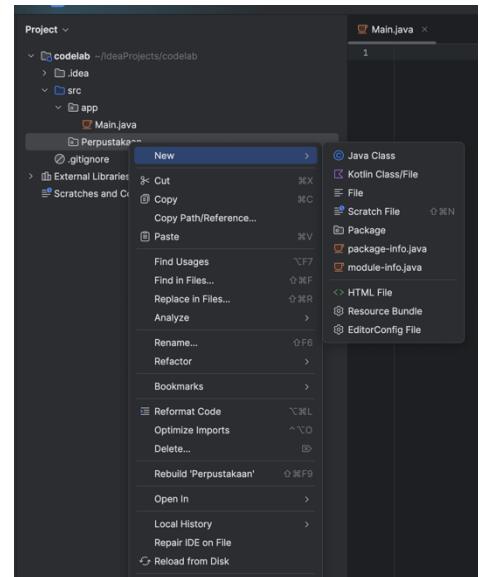
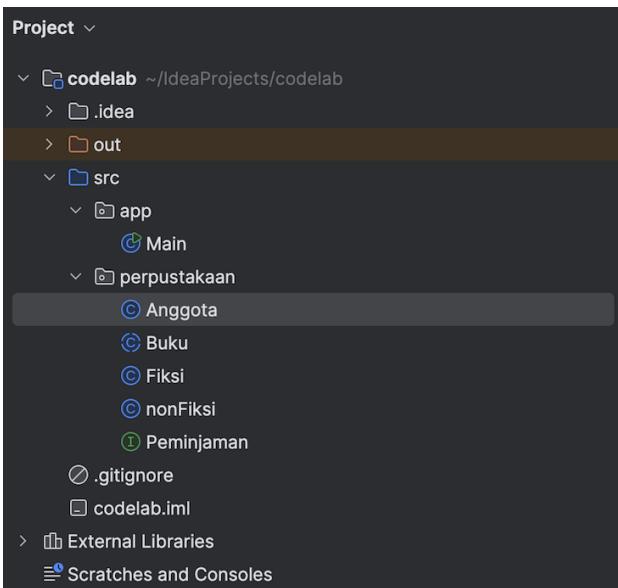
- beri nama `app` kemudian enter



- Selanjutnya, klik kanan di app → New → Package → beri nama perpustakaan kemudian enter



- Berikutnya pada perpustakaan, klik New → Java Class → buat kelas kelas ini:
 - Anggota**
 - Buku**
 - Fiksi**
 - nonFiksi**
 - Peminjaman**



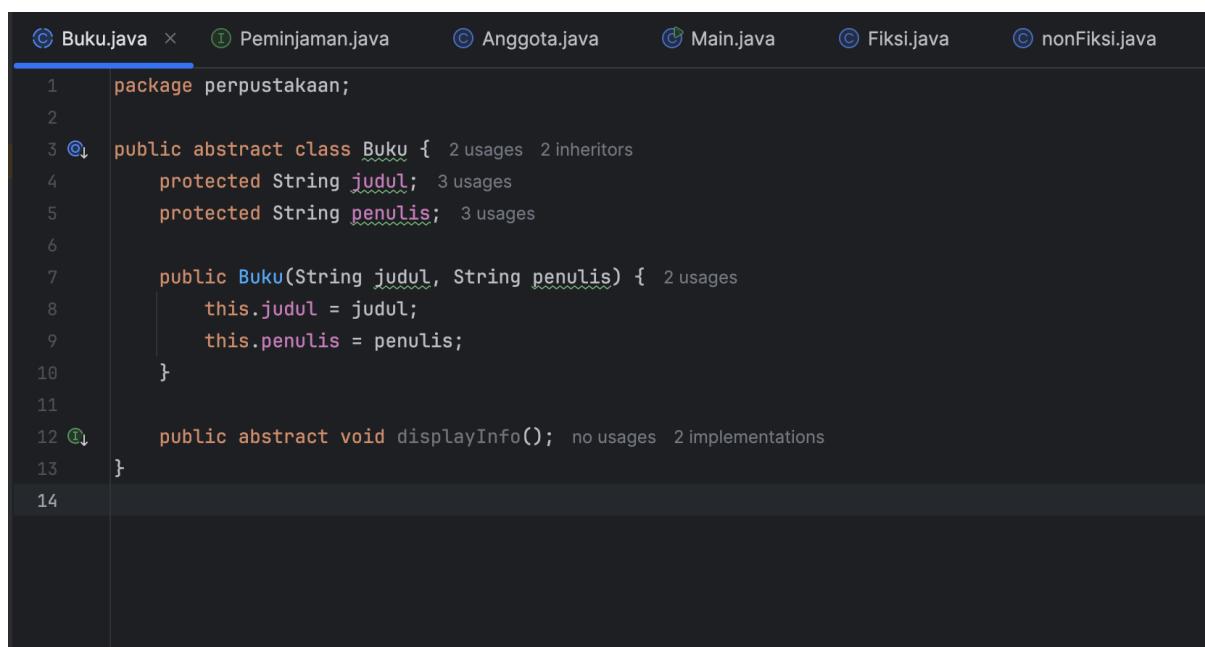
Isi untuk Buku.java di dalam package perpustakaan

```
package perpustakaan;

public abstract class Buku {
    protected String judul;
    protected String penulis;

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

    public abstract void displayInfo();
}
```



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

Penjelasan Singkat :

- judul dan penulis adalah atributnya
- displayInfo() harus di-override oleh anak-anaknya (Fiksi, nonFiksi)

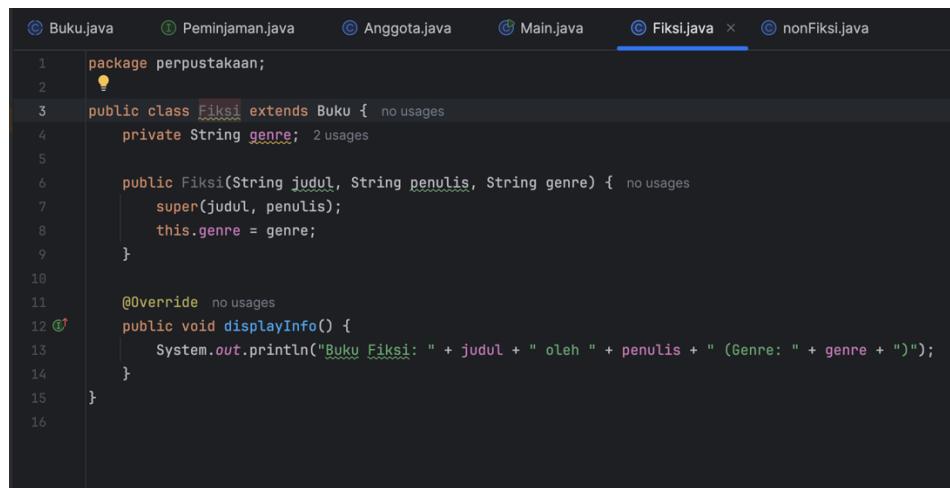
Isi untuk Fiksi.java di dalam package perpustakaan

```
• package perpustakaan;

public class Fiksi extends Buku {
    private String genre;

    public Fiksi(String judul, String penulis, String genre) {
        super(judul, penulis);
        this.genre = genre;
    }

    @Override
    public void displayInfo() {
        System.out.println("Buku Fiksi: " + judul + " oleh " +
        penulis + " (Genre: " + genre + ")");
    }
}
```



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

Penjelasan Singkat :

- Fiksi turunan dari Buku
- Tambahan atribut baru genre

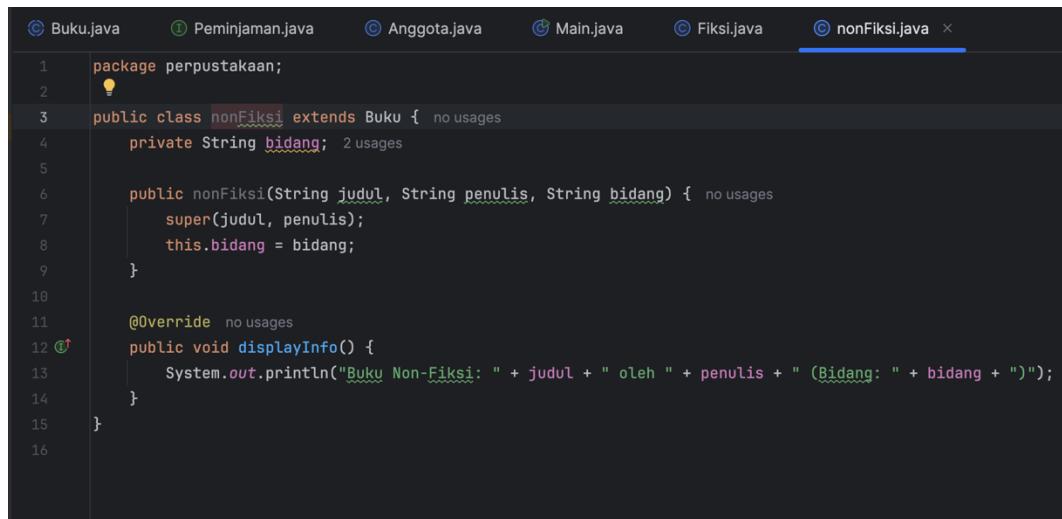
Isi untuk nonFiksi.java di dalam package perpustakaan

```
package perpustakaan;

public class nonFiksi extends Buku {
    private String bidang;

    public nonFiksi(String judul, String penulis, String bidang) {
        super(judul, penulis);
        this.bidang = bidang;
    }

    @Override
    public void displayInfo() {
        System.out.println("Buku Non-Fiksi: " + judul + " oleh " + penulis
+ " (Bidang: " + bidang + ")");
    }
}
```



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

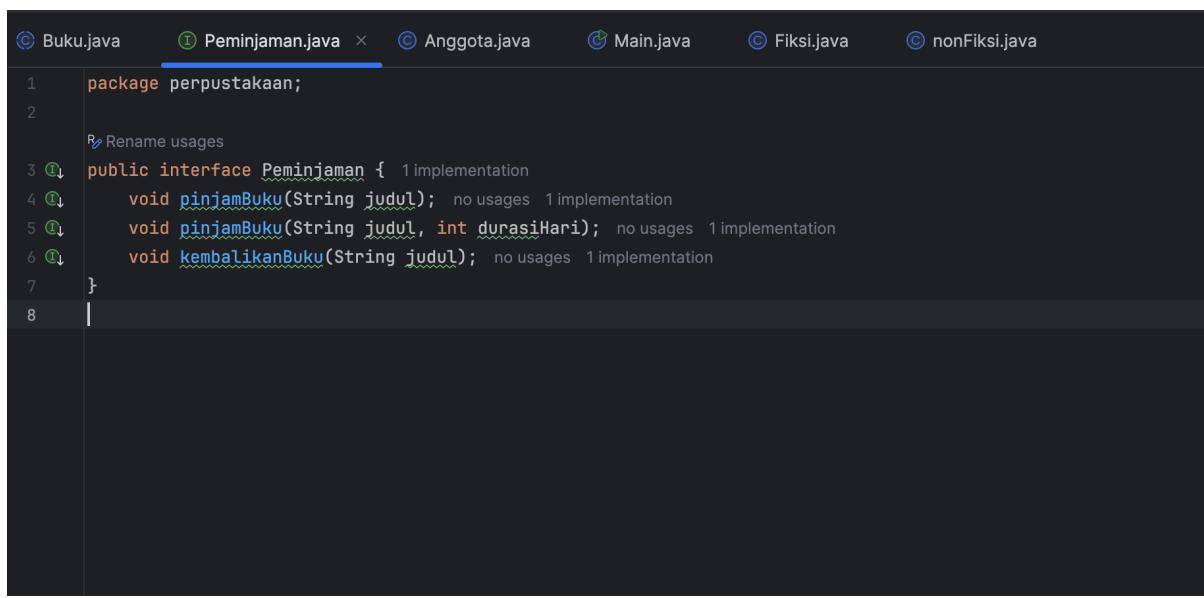
Penjelasan singkat :

- nonFiksi juga turunan dari Buku
- Tambahan atribut baru bidang

Isi untuk Peminjaman.java di dalam package perpustakaan

```
package perpustakaan;

public interface Peminjaman {
    void pinjamBuku(String judul);
    void pinjamBuku(String judul, int durasiHari);
    void kembalikanBuku(String judul);
}
```



```
1 package perpustakaan;
2
3 public interface Peminjaman {
4     void pinjamBuku(String judul); no usages 1 implementation
5     void pinjamBuku(String judul, int durasiHari); no usages 1 implementation
6     void kembalikanBuku(String judul); no usages 1 implementation
7 }
8 |
```

Penjelasan singkat :

- Ini interface
- Ada dua method: `pinjamBuku()` dan `kembalikanBuku()`
- Tanpa isi, karena nanti isi method ini akan dibuat di kelas `Anggota`

Isi untuk Anggota.java di dalam package perpustakaan\

```
package perpustakaan;

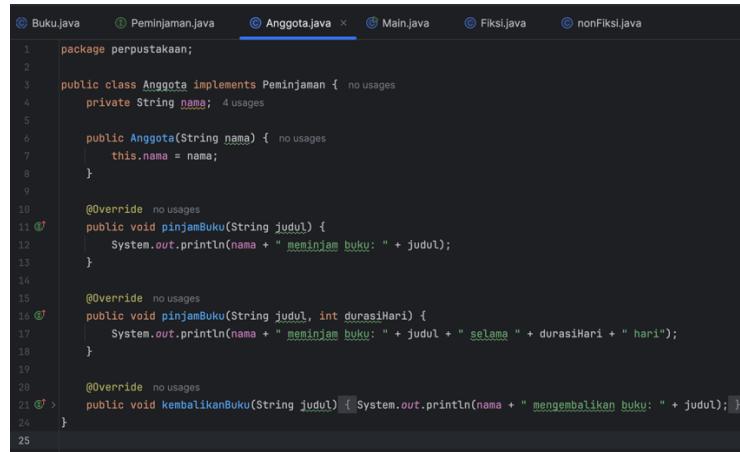
public class Anggota implements Peminjaman {
    private String nama;

    public Anggota(String nama) {
        this.nama = nama;
    }

    @Override
    public void pinjamBuku(String judul) {
        System.out.println(nama + " meminjam buku: " + judul);
    }

    @Override
    public void pinjamBuku(String judul, int durasiHari) {
        System.out.println(nama + " meminjam buku: " + judul + " selama " +
durasiHari + " hari");
    }

    @Override
    public void kembalikanBuku(String judul) {
        System.out.println(nama + " mengembalikan buku: " + judul);
    }
}
```



```
1 package perpustakaan;
2
3 public class Anggota implements Peminjaman { no usages
4     private String nama; 4 usages
5
6     public Anggota(String nama) { no usages
7         this.nama = nama;
8     }
9
10    @Override no usages
11    public void pinjamBuku(String judul) {
12        System.out.println(nama + " meminjam buku: " + judul);
13    }
14
15    @Override no usages
16    public void pinjamBuku(String judul, int durasiHari) {
17        System.out.println(nama + " meminjam buku: " + judul + " selama " + durasiHari + " hari");
18    }
19
20    @Override no usages
21    public void kembalikanBuku(String judul) System.out.println(nama + " mengembalikan buku: " + judul); 1
22
23 }
```

Penjelasan singkat :

- Anggota mengimplementasikan Peminjaman
- Wajib isi pinjamBuku(String) dan kembalikanBuku(String)
- Overloading terjadi di kembalikanBuku(String, int) (nama sama, parameter berbeda)

Berikut struktur file dan codeline supaya project bisa berjalan :

1. Struktur Folder :

```
src/
└── app/
    └── Main.java
└── perpustakaan/
    ├── Anggota.java
    ├── Buku.java
    ├── Fiksi.java
    ├── nonFiksi.java
    └── Peminjam.java
```

2. Isi File Main.java

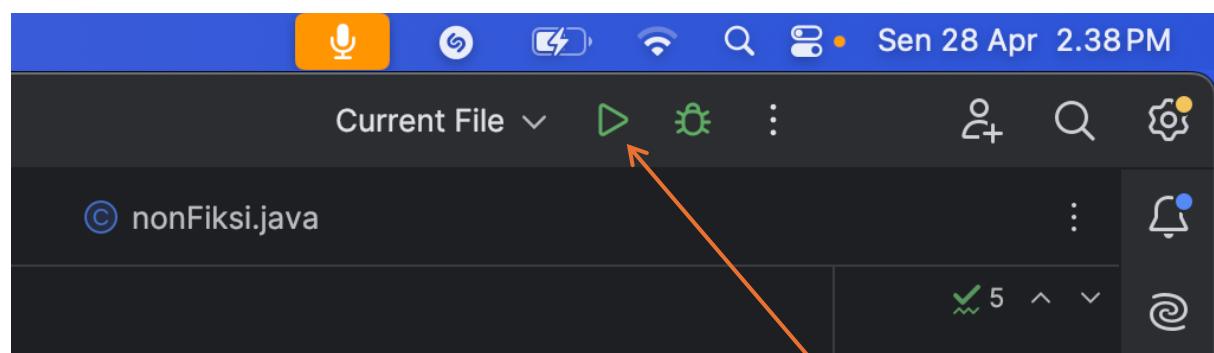
```
3. package app;

import perpustakaan.Anggota;

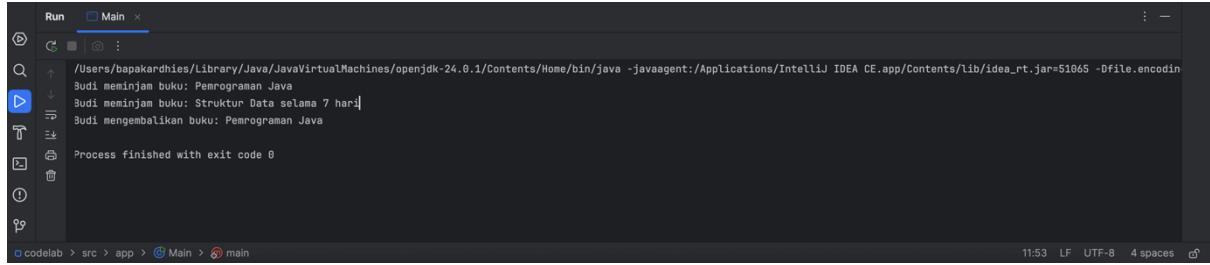
public class Main {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota("Budi");

        anggota1.pinjamBuku("Pemrograman Java");
        anggota1.pinjamBuku("Struktur Data", 7);
        anggota1.kembalikanBuku("Pemrograman Java");
    }
}
```

Setelah itu :



Klik Play Button di pojok kanan atas, dan output di console akan menampilkan seperti ini



The screenshot shows the IntelliJ IDEA interface with the 'Run' tab selected. In the center, there is a terminal window displaying the following text:

```
/Users/bapakardhies/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=51065 -Dfile.encoding=UTF-8
Budi meminjam buku: Pemrograman Java
Budi meminjam buku: Struktur Data selama 7 hari
Budi mengembalikan buku: Pemrograman Java
Process finished with exit code 0
```

At the bottom of the terminal window, the file path 'codelab > src > app > Main > main' is shown. On the right side of the terminal, the status bar displays '11:53 LF UTF-8 4 spaces'.