

LAPORAN PRATIUM CODELAB PBO 2E

MODUL 4



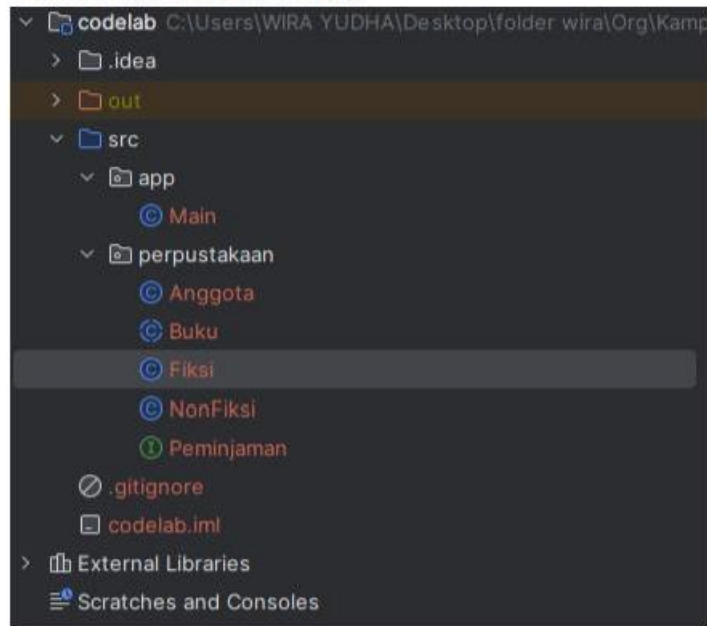
Nama: Muhammad Syakhish Al Hanif
NIM: 202410370110189
Kelas: PBO 2E
Github: github.com/Rofi69/PBO_2E_ROFI_214

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:
 - o **String** : nama
 - o **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: 8075)
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

```

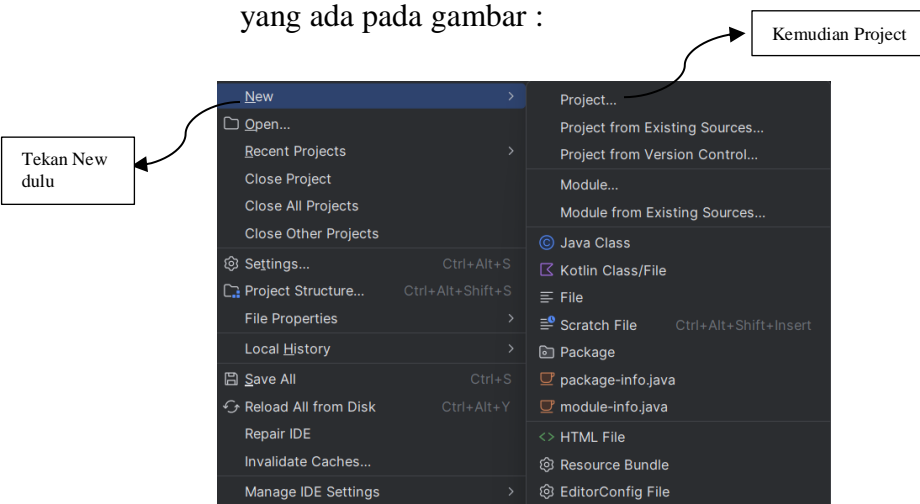
7. Catatan:

- o **idAnggota** merupakan **kelas kalian** dan **3-digit nim terakhir** kalian dan teman kalian
- o **Nama** menggunakan **nama kalian** dan **teman kalian**
- o Bagi yang tidak mengumpulkan codelab pada saat praktikum, kumpulkan di llab 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](#).

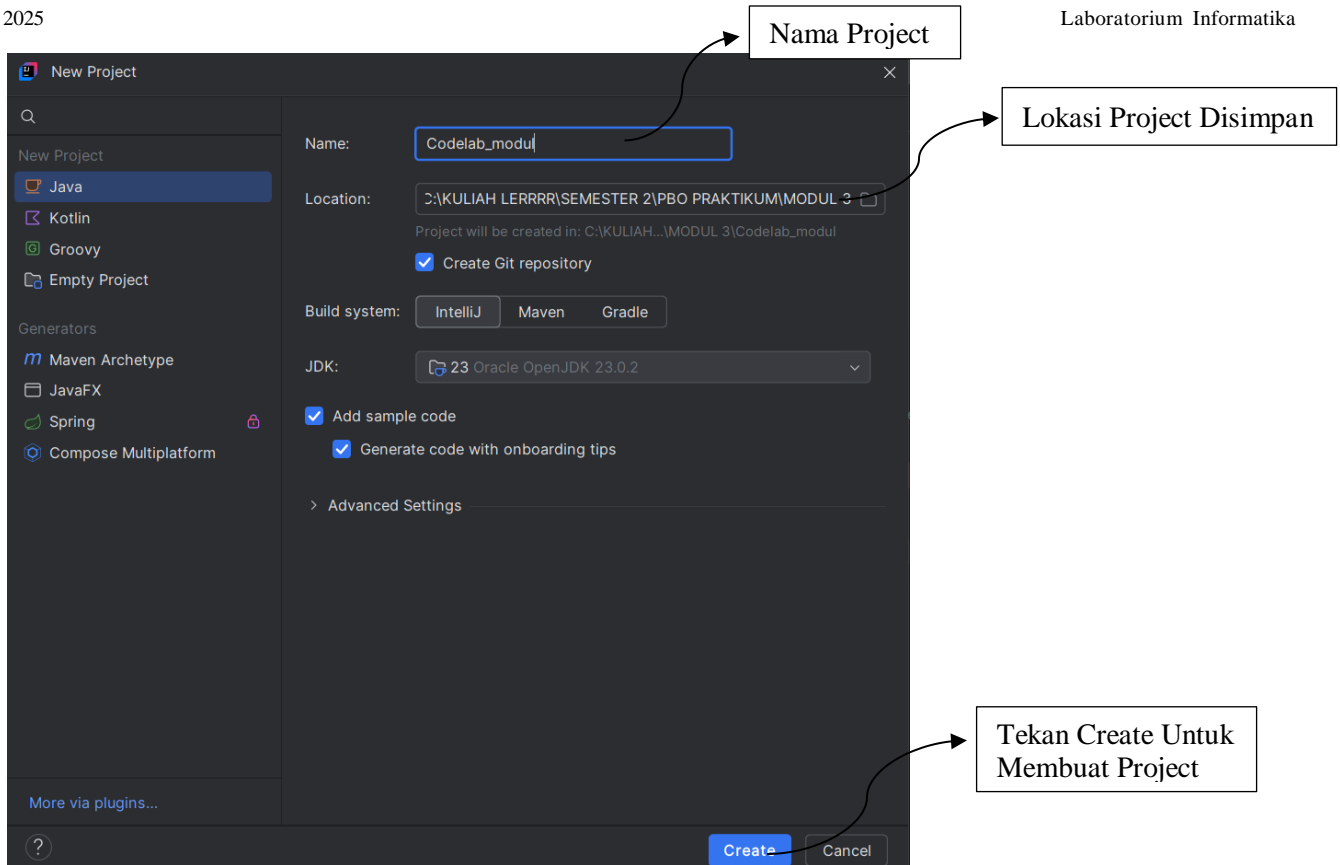
LANGKAH-LANGKAH:

1. Membuat New Project

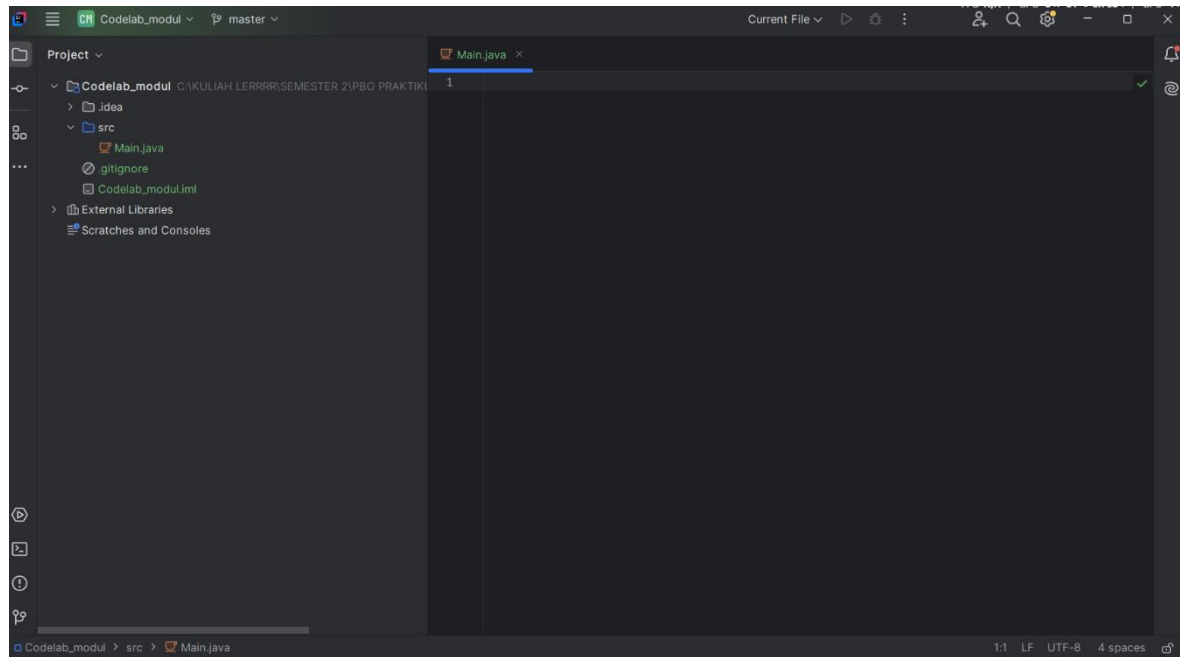
- Langkah pertama adalah membuat project Baru. Dengan cara tekan New kemudian Project seperti yang ada pada gambar :



- Setelah itu, Kalian namai project yang akan dibuat sesuai dengan instruksi, dan juga jangan lupa untuk mengecek apakah location sudah benar, jika sudah benar tekan tombol create di kanan bawah :

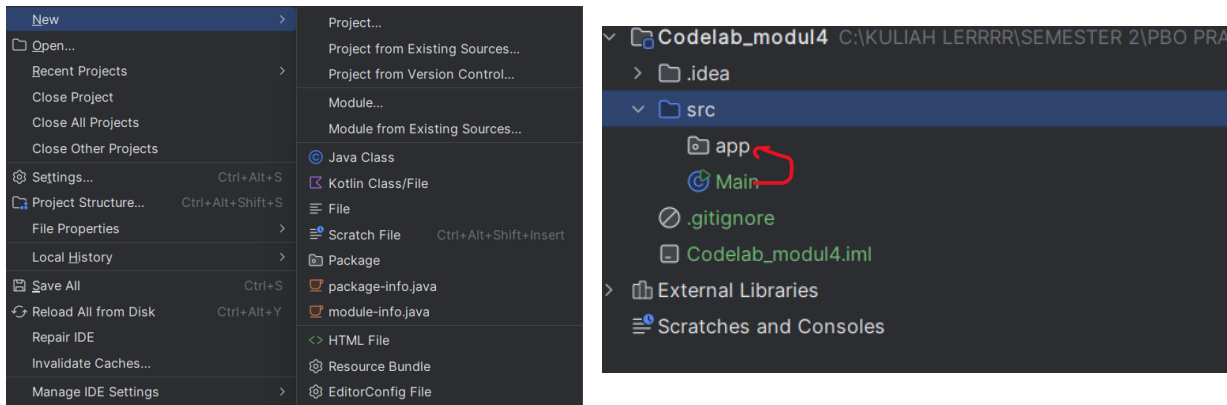


- Tampilan setelah membuat *project* baru:

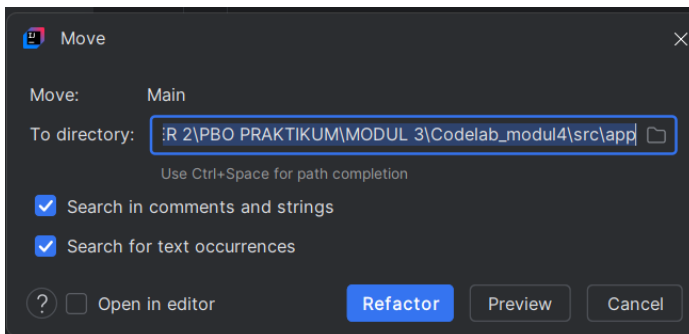


2. Package dan Class

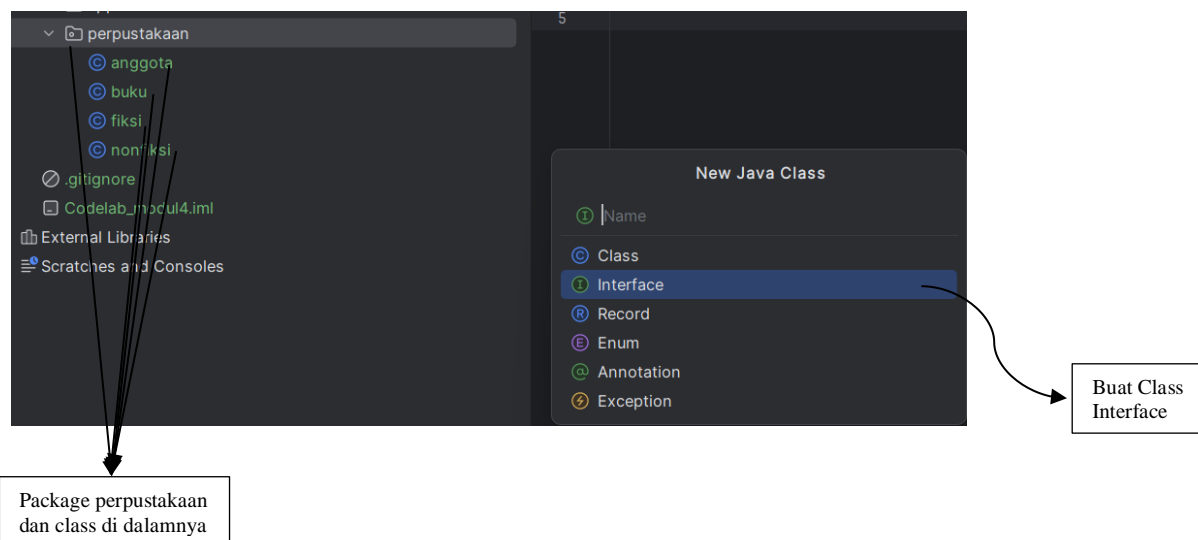
- Selanjutnya buat Package dengan cara tekan New Kemudian Package, dan jangan lupa untuk menamainya sesuai ketentuan :

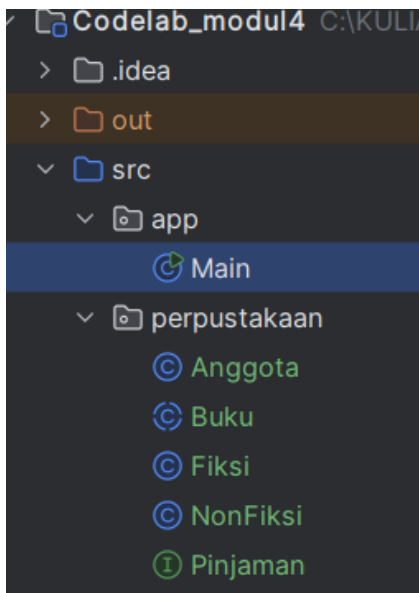


- Setelah itu drag and drop Class Main kedalam Package app untuk di Refactor



- Kemudian Buat Package perpustakaan dan juga Class Anggota, Buku, Fiksi, dan Nonfiksi di dalam package perpustakaan
- Jangan lupa untuk membuat Interface yang bernama pinjaman dengan cara yang sama





Jadilah seperti gambar disamping

3. Code

a. Class buku

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

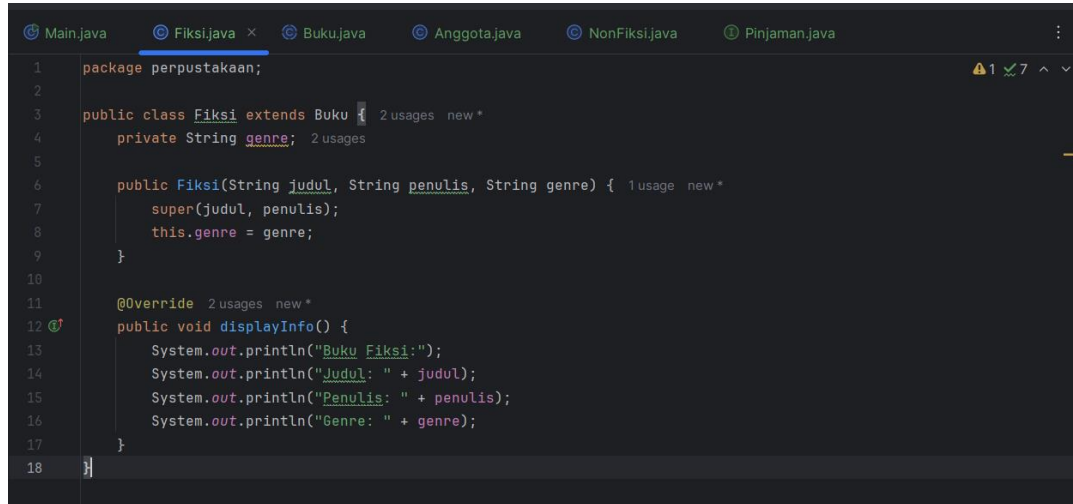
Penjelasan:

- **Atribut:**
 - judul (protected): Menyimpan judul buku
 - penulis (protected): Menyimpan nama penulis
- **Method:**
 - Constructor: Menginisialisasi judul dan penulis

- `displayInfo()`: Method abstrak yang harus diimplementasikan subclass untuk menampilkan informasi buku

b. Subclass fiksi dan nonfiksi

1. fiksi

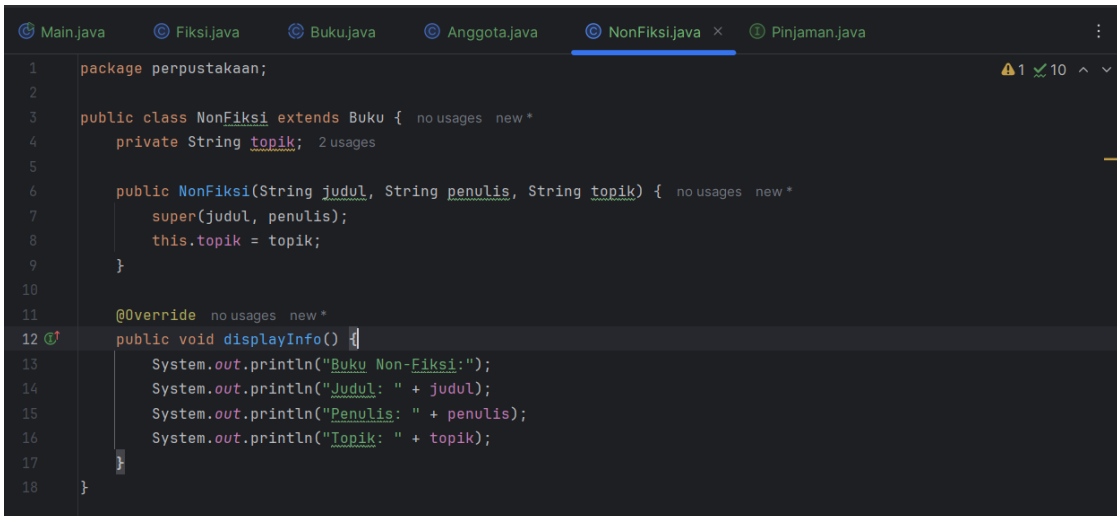


```
1 package perpustakaan;
2
3 public class Fiksi extends Buku { 2 usages new *
4     private String genre; 2 usages
5
6     public Fiksi(String judul, String penulis, String genre) { 1 usage new *
7         super(judul, penulis);
8         this.genre = genre;
9     }
10
11     @Override 2 usages new *
12     public void displayInfo() {
13         System.out.println("Buku Fiksi:");
14         System.out.println("Judul: " + judul);
15         System.out.println("Penulis: " + penulis);
16         System.out.println("Genre: " + genre);
17     }
18 }
```

Penjelasan:

- **Atribut Tambahan:**
 - genre: Menyimpan genre buku fiksi (misal: Drama, Fantasi)
- **Method:**
 - Constructor: Memanggil constructor parent class dan menambahkan inisialisasi genre
 - `displayInfo()`: Implementasi spesifik untuk menampilkan info buku fiksi dengan format khusus

2. nonfiksi



```

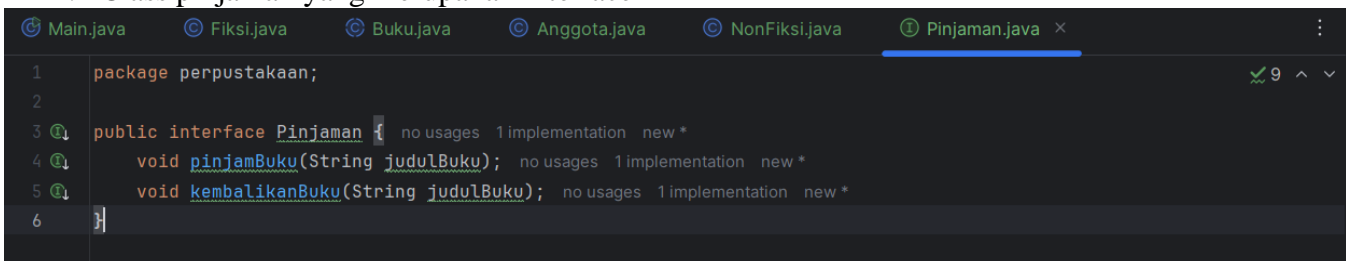
1 package perpustakaan;
2
3 public class NonFiksi extends Buku { no usages new *
4     private String topik; 2 usages
5
6     public NonFiksi(String judul, String penulis, String topik) { no usages new *
7         super(judul, penulis);
8         this.topik = topik;
9     }
10
11     @Override no usages new *
12     public void displayInfo() {
13         System.out.println("Buku Non-Fiksi:");
14         System.out.println("Judul: " + judul);
15         System.out.println("Penulis: " + penulis);
16         System.out.println("Topik: " + topik);
17     }
18 }

```

Penjelasan:

- **Atribut Tambahan:**
 - topik: Menyimpan topik buku non-fiksi (misal: Sains, Sejarah)
- **Method:**
 - displayInfo(): Implementasi spesifik untuk menampilkan info buku non-fiksi dengan format berbeda dari fiksi

4. Class pinjaman yang merupakan interface



```

1 package perpustakaan;
2
3 public interface Pinjaman { no usages 1 implementation new *
4     void pinjamBuku(String judulBuku); no usages 1 implementation new *
5     void kembalikanBuku(String judulBuku); no usages 1 implementation new *
6 }

```

Penjelasan:

- **Method Wajib:**
 - pinjamBuku(String): Untuk menandai peminjaman buku
 - kembalikanBuku(String): Untuk menandai pengembalian buku

Tujuan: Memastikan semua kelas yang mengimplementasikan interface ini memiliki kemampuan dasar peminjaman.

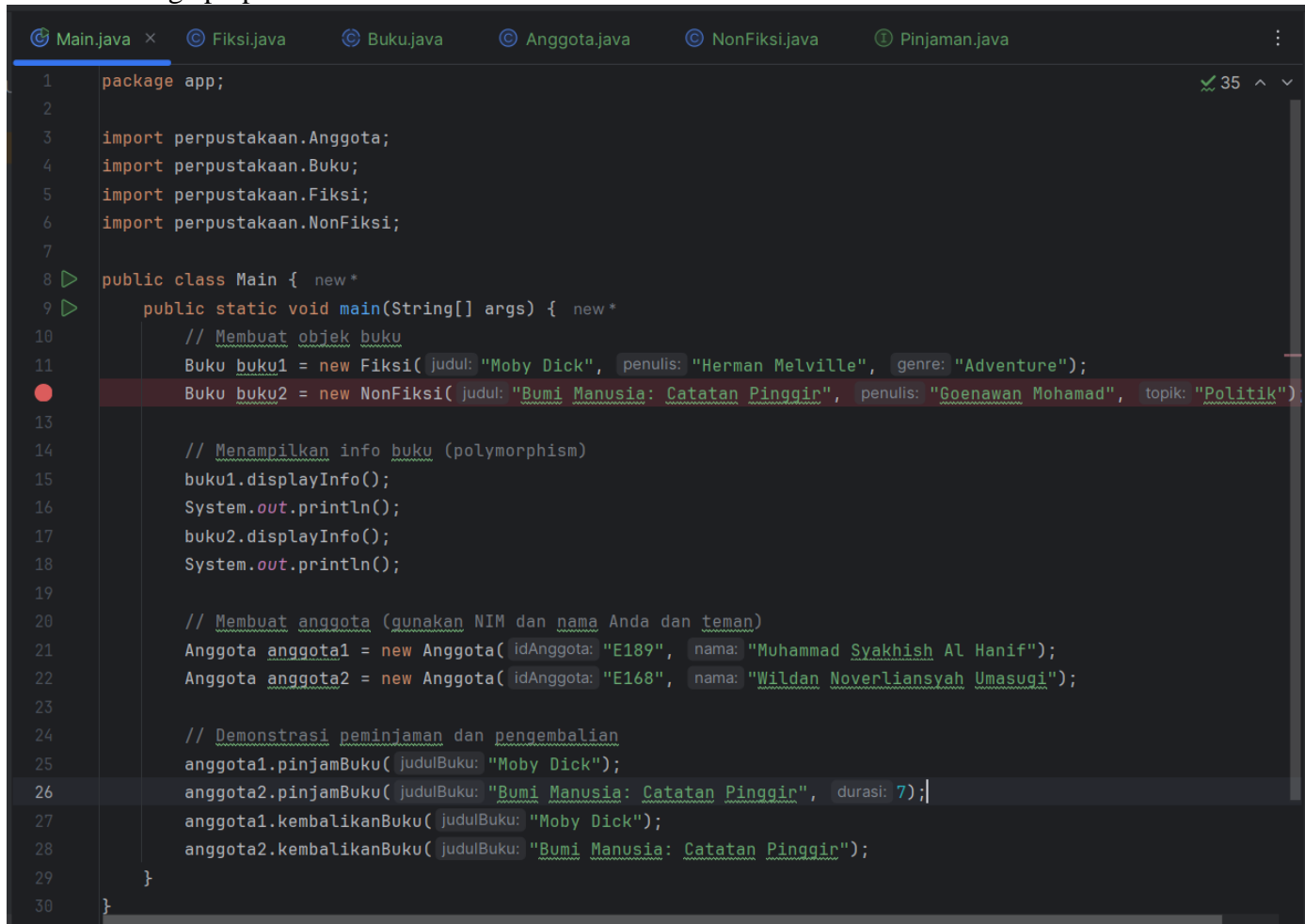
5. Class anggota, mengimplementasi interface peminjaman

```
1 package perpustakaan;
2
3 public class Anggota implements Pinjaman { no usages new *
4     private String idAnggota; 4 usages
5     private String nama; 4 usages
6
7     public Anggota(String idAnggota, String nama) { no usages new *
8         this.idAnggota = idAnggota;
9         this.nama = nama;
10    }
11
12    // Implementasi interface Pinjaman
13    @Override no usages new *
14    public void pinjamBuku(String judulBuku) {
15        System.out.println(nama + " (ID: " + idAnggota + ") meminjam buku: " + judulBuku);
16    }
17
18    @Override no usages new *
19    public void kembalikanBuku(String judulBuku) {
20        System.out.println(nama + " (ID: " + idAnggota + ") mengembalikan buku: " + judulBuku);
21    }
22
23    // Overloading method pinjamBuku
24    public void pinjamBuku(String judulBuku, int durasi) { no usages new *
25        System.out.println(nama + " (ID: " + idAnggota + ") meminjam buku: " + judulBuku +
26            " untuk " + durasi + " hari");
27    }
28 }
```

Penjelasan:

- **Atribut:**
 - idAnggota: Nomor identifikasi unik anggota
 - nama: Nama lengkap anggota
- **Method:**
 - Implementasi interface Pinjaman dengan dua method wajib
 - Overloading pinjamBuku() dengan versi yang menerima durasi peminjaman
 - Semua method mencetak informasi transaksi

6. Class Main yang berada di package app, digunakan untuk menjalankan semua komponen yang ada di Package perpustakaan



```
1 package app;
2
3 import perpustakaan.Anggota;
4 import perpustakaan.Buku;
5 import perpustakaan.Fiksi;
6 import perpustakaan.NonFiksi;
7
8 public class Main {
9     public static void main(String[] args) {
10         // Membuat objek buku
11         Buku buku1 = new Fiksi( judul: "Moby Dick", penulis: "Herman Melville", genre: "Adventure");
12         Buku buku2 = new NonFiksi( judul: "Bumi Manusia: Catatan Pinggir", penulis: "Goenawan Mohamad", topik: "Politik");
13
14         // Menampilkan info buku (polymorphism)
15         buku1.displayInfo();
16         System.out.println();
17         buku2.displayInfo();
18         System.out.println();
19
20         // Membuat anggota (gunakan NIM dan nama Anda dan teman)
21         Anggota anggota1 = new Anggota( idAnggota: "E189", nama: "Muhammad Syakhish Al Hanif");
22         Anggota anggota2 = new Anggota( idAnggota: "E168", nama: "Wildan Noverliansyah Umasugi");
23
24         // Demonstrasi peminjaman dan pengembalian
25         anggota1.pinjamBuku( judulBuku: "Moby Dick");
26         anggota2.pinjamBuku( judulBuku: "Bumi Manusia: Catatan Pinggir", durasi: 7);
27         anggota1.kembalikanBuku( judulBuku: "Moby Dick");
28         anggota2.kembalikanBuku( judulBuku: "Bumi Manusia: Catatan Pinggir");
29     }
30 }
```

Penjelasan:

- Demonstrasi semua fitur:
 1. Pembuatan objek buku dengan tipe berbeda
 2. Polymorphism melalui displayInfo()
 3. Penggunaan interface dan overloading method