

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 5

Object Oriented Programming Java 2



Disusun Oleh:
Gerald Eberhard
(105223002)

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

1. Pendahuluan

Studi kasus praktikum ini adalah pembuatan Sistem Manajemen Perpustakaan sederhana untuk mengelola koleksi buku dan aktivitas peminjaman di sebuah perpustakaan. Sistem ini dirancang menggunakan konsep Pemrograman Berorientasi Objek (PBO) dalam bahasa Java untuk memodelkan entitas seperti buku, perpustakaan, dan pengguna atau user. Setiap entitas memiliki atribut dan perilaku spesifik, seperti informasi buku (judul, penulis, tahun terbit, status peminjaman), pengelolaan koleksi buku oleh perpustakaan, dan aktivitas peminjaman oleh pengguna (nama dan ID). Program ini mampu menambahkan buku, menampilkan daftar buku, serta memproses peminjaman dan pengembalian buku secara terstruktur.

Codingan ini terdiri dari empat kelas utama, yaitu Main.java dan package librarysystem yang memuat kelas Buku.java, Perpustakaan.java, dan User.java. Kelas Main.java berfungsi sebagai tempat pembuatan objek dan pemanggilan perintah-perintah dari kelas lain. Kelas Buku.java berisi informasi dan metode terkait buku, seperti status peminjaman. Kelas Perpustakaan.java mengelola koleksi buku menggunakan array of object, sedangkan kelas User.java merepresentasikan pengguna yang melakukan peminjaman dan pengembalian buku.

2. Variabel

No	Nama Variabel	Tipe Data	Fungsi
1	nama	String	Menyimpan nama User yang melakukan peminjaman buku.
2	idUser	String	Menyimpan ID User yang melakukan peminjaman buku
3	koleksiBuku	Buku	Array yang berfungsi untuk menyimpan daftar buku yang terdapat di perpustakaan.
4	jumlahBuku	integer	Menyimpan berapa jumlah buku yang tersimpan di dalam perpustakaan.
5	KAPASITAS	integer	Ini adalah konstansta yang dibuat untuk menyimpan nilai kapasitas dari perpustakaan.
6	judul	String	Variabel ini bertujuan untuk menyimpan judul buku.

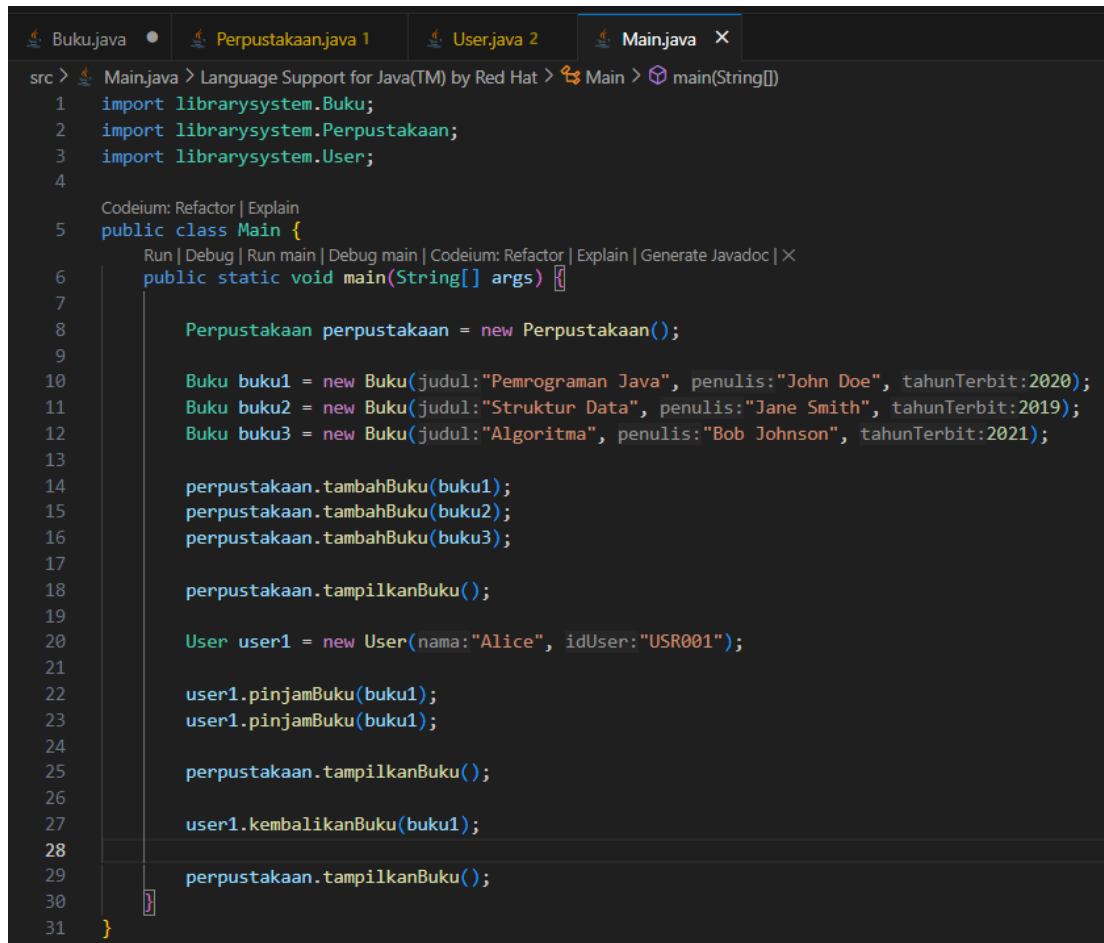
7	penulis	String	Variabel ini bertujuan untuk menyimpan nama penulis
8	tahunTerbit	integer	Variabel ini bertujuan untuk menyimpan tahun terbit dari buku yang disimpan
9	statusDipijam	boolean	Variabel ini diciptakan untuk menyimpan variabel kebenaran apakah buku ini benar sedang dipinjam ataukah tidak.
10	jumlahBukuTersedia	integer	Variabel ini diciptakan untuk menyimpan total sudah berapa banyak object buku yang dibuat.
11	buku1	Buku	Untuk menyimpan object-object dari data-data pada buku pertama.
12	buku2	Buku	Untuk menyimpan object-object dari data-data pada buku kedua.
13	buku3	Buku	Untuk menyimpan object dari data-data pada buku ketiga.
14	perpustakaan	Perpustakaan	Sebagai array of objects. Dalam hal ini array ini bertujuan untuk menyimpan object-object buku
15	user1	User	Untuk menyimpan data-data terkait user atau orang yang ingin meminjam buku.

3. Constructor dan Method

Nama Method	Jenis Method	Fungsi
User	Constructor	Membuat object User atau peminjam buku.
pinjamBuku	Prosedur atau void atau method	Memberikan output berupa teks yang diprint di terminal tentang user ini meminjam buku dan memanggil fungsi pinjam buku untuk kemudian memberikan keterangan statusDipinjam pada buku menjadi true.
kembalikanBuku	Prosedur atau void atau method	Memberikan output berupa teks yang diprint di terminal tentang user ini telah mengembalikan buku dan memanggil fungsi pinjam buku untuk kemudian

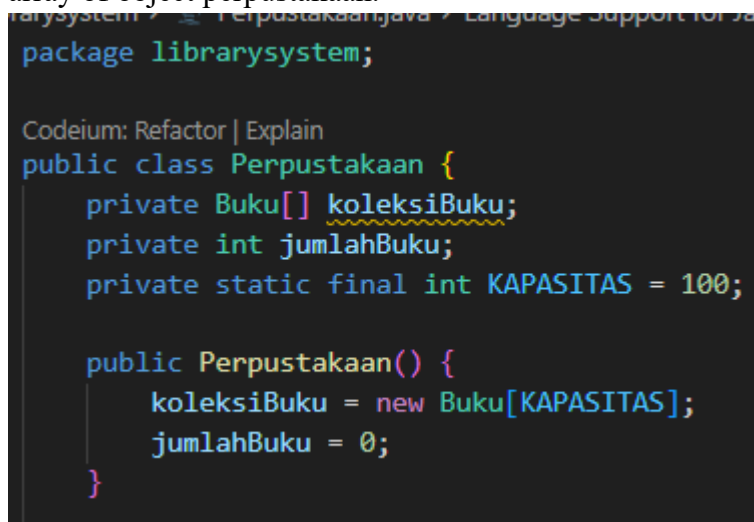
		memberikan keterangan statusDipinjam pada buku menjadi false.
perpustakaan	Constructor	Membuat array of object buku yang disebut dengan object perpustakaan.
tambahkanBuku	Prosedur atau void atau method	Menambahkan buku kedalam koleksi buku yang dimiliki di perpustakaan.
tampilkanBuku	Prosedur atau void atau method	Menampilkan daftar buku di perpustakaan.
Buku	Constructor	Membuat object buku.
getJudul	Fungsi	Mengembalikan nilai judul
getPenulis	Fungsi	Mengembalikan nilai penulis
getTahunTerbit	Fungsi	Mengembalikan nilai tahun terbit
isStatusDipinjam	Fungsi	Mengembalikan nilai status buku apakah dipinjam atau tidak
getJumlahBukuTersedia	Fungsi	Mengembalikan jumlah buku yang tersedia.
setJudul	Fungsi	Menginput atau mengganti nilai judul
setPenulis	Fungsi	Menginput atau mengganti nilai penulis
setTahunTerbit	Fungsi	Menginput atau mengganti tahun terbit
pinjamBuku	Prosedur atau void atau method	Mengeluarkan output berupa keterangan apakah buku dapat dipinjam ataukah tidak. Kemudian mengubah nilai statusDipinjam menjadi true apabila buku dapat dipinjam.
kembalikanBuku	Prosedur atau void atau method	Mengeluarkan output berupa keterangan apakah buku dapat dikembalikan ataukah tidak. Kemudian apabila bisa maka statusDipinjam akan diganti menjadi false.

4. Dokumentasi dan Pembahasan Code



```
src > Main.java > Language Support for Java(TM) by Red Hat > Main > main(String[])
1  import librarysystem.Buku;
2  import librarysystem.Perpustakaan;
3  import librarysystem.User;
4
5  public class Main {
6      public static void main(String[] args) {
7
8          Perpustakaan perpustakaan = new Perpustakaan();
9
10         Buku buku1 = new Buku(judul:"Pemrograman Java", penulis:"John Doe", tahunTerbit:2020);
11         Buku buku2 = new Buku(judul:"Struktur Data", penulis:"Jane Smith", tahunTerbit:2019);
12         Buku buku3 = new Buku(judul:"Algoritma", penulis:"Bob Johnson", tahunTerbit:2021);
13
14         perpustakaan.tambahBuku(buku1);
15         perpustakaan.tambahBuku(buku2);
16         perpustakaan.tambahBuku(buku3);
17
18         perpustakaan.tampilkanBuku();
19
20         User user1 = new User(nama:"Alice", idUser:"USR001");
21
22         user1.pinjamBuku(buku1);
23         user1.pinjamBuku(buku1);
24
25         perpustakaan.tampilkanBuku();
26
27         user1.kembalikanBuku(buku1);
28
29         perpustakaan.tampilkanBuku();
30     }
31 }
```

Program pada THT kali ini cukup simple dimana pertama-tama kita akan membuat array of object perpustakaan.



```
librarysystem > Perpustakaan.java > Language Support for Java(TM) by Red Hat > Perpustakaan
package librarysystem;

Codeium: Refactor | Explain
public class Perpustakaan {
    private Buku[] koleksiBuku;
    private int jumlahBuku;
    private static final int KAPASITAS = 100;

    public Perpustakaan() {
        koleksiBuku = new Buku[KAPASITAS];
        jumlahBuku = 0;
    }
}
```

Kemudian kita akan membuat 3 object buku menggunakan constructor buku.

```
Codeium: Refactor | Explain
public class Buku {
    private String judul;
    private String penulis;
    private int tahunTerbit;
    private boolean statusDipinjam;
    private static int jumlahBukuTersedia = 0;

    public Buku(String judul, String penulis, int tahunTerbit) {
        this.judul = judul;
        this.penulis = penulis;
        this.tahunTerbit = tahunTerbit;
        this.statusDipinjam = false;
        jumlahBukuTersedia++;
    }
}
```

Kemudian program akan berlanjut ke baris 14, 15, dan 16. Pada bagian ini buku-buku yang sudah dibuat tersebut akan dimasukkan ke dalam perpustakaan melalui

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void tambahBuku(Buku buku) {
    if (jumlahBuku < koleksiBuku.length) {
        koleksiBuku[jumlahBuku] = buku;
        jumlahBuku++;
        System.out.println("Buku '" + buku.getJudul() + "' ditambahkan ke perpustakaan");
    } else {
        System.out.println(x:"Kapasitas perpustakaan penuh!");
    }
}
}
```

Kemudian dengan memanfaatkan method tampilkanBuku maka program akan menampilkan buku-buku yang tersedia.

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void tampilkanBuku() {
    System.out.println(x:"\nDaftar Buku di Perpustakaan:");
    for (int i = 0; i < jumlahBuku; i++) {
        Buku b = koleksiBuku[i];
        System.out.println((i + 1) + ". " + b.getJudul() +
            " - " + b.getPenulis() +
            " (" + b.getTahunTerbit() + ")" +
            " [Dipinjam: " + (b.isStatusDipinjam() ? "Ya" : "Tidak") + "]);
    }
    System.out.println("Total buku tersedia: " + Buku.getJumlahBukuTersedia());
}
}
```

Kemudian pada main kelas akan deconstruct sebuah object user bernama user1 menggunakan constructor yang telah dibuat pada kelas user.

```
package librarysystem;

Codeium: Refactor | Explain
public class User {
    private String nama;
    private String idUser;

    public User(String nama, String idUser) {
        this.nama = nama;
        this.idUser = idUser;
    }
}
```

Kemudian berikutnya program akan berlanjut dengan memberikan uji coba user1. Untuk dua kali melakukan peminjaman buku yang sama. Hal ini bertujuan agar kita dapat menguji apakah method pinjamBuku pada kelas user

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void pinjamBuku(Buku buku) {
    System.out.println("\nUser " + nama + " (ID: " + idUser + ") meminjam buku:");
    buku.pinjamBuku();
}
```

dan method pinjamBuku pada kelas Buku.

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void pinjamBuku() {
    if (!statusDipinjam) {
        statusDipinjam = true;
        System.out.println("Buku " + judul + " berhasil dipinjam");
    } else {
        System.out.println("Buku " + judul + " sedang dipinjam");
    }
}
```

Kemudian buku dengan menggunakan method tampilkanBuku pada kelas perpustakaan akan menampilkan daftar buku pada perpustakaan. Kemudian setelah itu user akan mengembalikan buku menggunakan method kembalikan buku pada kelas user

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void kembalikanBuku(Buku buku) {
    System.out.println("\nUser " + nama + " (ID: " + idUser + ") mengembalikan buku:");
    buku.kembalikanBuku();
}
```

dan method kembalikanBuku pada kelas buku

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void kembalikanBuku() {
    if (statusDipinjam) {
        statusDipinjam = false;
        System.out.println("Buku '" + judul + "' telah dikembalikan");
    } else {
        System.out.println("Buku '" + judul + "' tidak sedang dipinjam");
    }
}
```

Kemudian terakhir program akan menampilkan daftar buku yang terdapat di dalam perpustakaan.

5. Kesimpulan

Praktikum ini berhasil mengimplementasikan Sistem Manajemen Perpustakaan sederhana menggunakan konsep Pemrograman Berorientasi Objek seperti enkapsulasi (melalui atribut private dan getter/setter) serta hubungan antar kelas dalam package librarysystem. Program ini memenuhi kebutuhan studi kasus dengan mengelola data buku, perpustakaan, dan pengguna secara terstruktur menggunakan array of object. Pengalaman ini sangat membantu saya memahami penerapan materi PBO dalam menyelesaikan masalah nyata, seperti pengelolaan perpustakaan. Sistem ini dapat menambahkan buku, menampilkan daftar buku, serta memproses peminjaman dan pengembalian dengan baik. Sekian dan terima kasih.

6. Daftar Pustaka

Modul 5: Obejct Oriented Programming Java 2