

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL V
CLASS dan OBJECT



Disusun Oleh:
Mohamad Dandung Sadat 105222033

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. Pendahuluan

Sistem Perpustakaan yang dirancang. Sistem ini dibuat untuk mendukung proses peminjaman dan pengembalian buku oleh pengguna, serta merekam koleksi buku yang ada. Dalam pelaksanaannya, program ini memanfaatkan empat kelas utama, yaitu User, Buku, Perpustakaan, dan Main.

Kelas User menggambarkan entitas pengguna perpustakaan yang memiliki kemampuan untuk meminjam serta mengembalikan buku. Kelas Buku berfungsi untuk menyimpan data rinci tentang buku seperti nama, pengarang, tahun terbit, dan status apakah buku tersebut sedang dipinjam atau tidak. Kelas Perpustakaan mengelola koleksi buku dengan menggunakan struktur data ArrayList, dan juga menawarkan fitur untuk menambah serta menampilkan daftar buku. Di sisi lain, kelas Main berfungsi sebagai titik awal program untuk menginisialisasi objek dan melaksanakan simulasi proses peminjaman serta pengembalian buku oleh pengguna.

Studi kasus ini tidak hanya melatih pemahaman sintaksis Java, tetapi juga kemampuan untuk merancang program yang efisien dan terstruktur secara modular. Mahasiswa diharuskan untuk mengetahui urutan pelaksanaan program, hubungan antar objek, serta pemanfaatan fitur seperti konstruktor, metode, dan variabel statis. Contohnya, pemanfaatan `static int jumlahBukuTersedia` dalam kelas Buku menunjukkan cara nilai dapat dibagi di antara semua objek buku yang dibuat.

Melalui penerapan pembuatan sistem ini, mahasiswa mendapatkan pengalaman praktis dalam merancang program berbasis objek yang dapat mengelola berbagai situasi nyata seperti buku yang sedang dipinjam, usaha meminjam buku yang tidak ada, hingga proses pengembalian buku yang belum dipinjam. Langkah ini krusial untuk memperkuat pemahaman tentang logika pemrograman serta keterampilan analisis dalam penerapan kode Java.

II. Variabel

No	Nama Variabel	Tipe data	Fungsi
1	<i>nama</i>	String	Menyimpan nama pengguna
2	Iduser	String	Menyimpan ID pengguna
3	Judul	String	Menyimpan judul buku
4	Penulis	String	Menyimpan nama penulis buku
5	Tahunterbit	Int	Menyimpan tahun terbit buku
6	Statusdipinjam	Boolean	Menandakan apakah buku sedang dipinjam
7	Jumlahbukutersedia	Static int	Menyimpan jumlah total buku yang tersedia
8	Koleksibuku	Arraylist	Menyimpan koleksi buku dalam perpustakaan
9	namaperpustakaan	String	Menyimpan nama perpustakaan

III. Constructor dan Method

No	Nama Metode	Jenis Metode	Fungsi
1	User(...)	Constructor	Menginisialisasi objek pengguna
2	Buku(...)	Constructor	Menginisialisasi objek buku
3	Perpustakaan(.. .)	Constructor	Membuat perpustakaan baru
4	pinjamBuku	Procedural	Meminjam buku oleh pengguna
5	kembalikanBuku	Procedural	Mengembalikan buku oleh pengguna
6	tampilkanBuku	Procedural	Menampilkan semua buku di perpustakaan
7	tambahBuku	Procedural	Menambahkan buku ke koleksi perpustakaan
8	pinjamBuku	Functional	Mengatur status peminjaman buku
9	kembalikanBuku	Functional	Mengatur status pengembalian buku

IV. Dokumentasi dan Pembahasan Code

4.1. Kelas User

```
public class User {  
    private String nama;  
    private String idUser;  
  
    public User(String nama, String idUser) {  
        this.nama = nama;  
        this.idUser = idUser;  
        System.out.println("Pengguna '" + this.nama + "' dengan ID '" + this.idUser + "'");  
    }  
}
```

Mendeklarasikan **kelas User**, yang merepresentasikan pengguna dalam sistem perpustakaan. Dua atribut privat digunakan untuk menyimpan data pribadi pengguna yaitu nama dan idUser, menjaga prinsip enkapsulasi. Constructor digunakan untuk menginisialisasi objek User. Parameter nama dan idUser diisi saat pembuatan objek, lalu ditampilkan pesan konfirmasi ke konsol sebagai bentuk interaksi pengguna.

4.2. Kelas Buku

```
1 public class Buku {  
2     private String judul;  
3     private String penulis;  
4     private int tahunTerbit;  
5     private boolean statusDipinjam;  
6     public static int jumlahBukuTersedia;  
7  
8     public Buku(String judul, String penulis, int tahunTerbit) {  
9         this.judul = judul;  
10        this.penulis = penulis;  
11        this.tahunTerbit = tahunTerbit;  
12        this.statusDipinjam = false;  
13        jumlahBukuTersedia++;  
14    }  
}
```

Kelas Buku menyimpan data sebuah buku. Atribut statusDipinjam menunjukkan apakah buku sedang dipinjam, dan jumlahBukuTersedia adalah variabel static yang menghitung jumlah buku yang tersedia secara global.

Constructor menerima data judul, penulis, dan tahun terbit, lalu menyimpan informasi ini ke dalam atribut yang sesuai. Nilai statusDipinjam diset false sebagai default, dan jumlahBukuTersedia ditingkatkan sebagai indikator global.

4.3.Kelas Perpustakaan

```
1 public class Perpustakaan {
2     private ArrayList<Buku> koleksiBuku;
3     private String namaPerpustakaan;
4
5     public Perpustakaan(String namaPerpustakaan) {
6         this.namaPerpustakaan = namaPerpustakaan;
7         this.koleksiBuku = new ArrayList<>();
8         System.out.println("Perpustakaan '" + this.namaPerpustakaan
9     }
```

Kelas ini bertanggung jawab menyimpan koleksi buku. Menggunakan ArrayList<Buku> untuk fleksibilitas dalam menambahkan dan menampilkan daftar buku.

Constructor menerima nama perpustakaan dan menginisialisasi list kosong. Pesan konfirmasi ditampilkan untuk validasi pembuatan objek.

4.4.Kelas Main

```
1 public class Main {
2     public static void main(String[] args) {
3         Perpustakaan perpustakaanKota = new Perpustakaan("Perpustakaan Kota");
4         Buku buku1 = new Buku("Pemrograman Java", "Budi Santoso", 2023);
5         Buku buku2 = new Buku("Algoritma Data", "Citra Dewi", 2022);
6         User user1 = new User("Alice", "U001");
7         user1.pinjamBuku(buku1);
8         user1.kembalikanBuku(buku1);
9     }
}
```

Kelas main ini fungsi nya untuk membuat objek” pada kelas lain
Membuat objek Perpustakaan dengan nama "Perpustakaan Kota".

Membuat dua objek Buku dengan data berbeda. Ini menunjukkan fleksibilitas instansiasi,Membuat objek User dengan ID unik.Menunjukkan interaksi nyata, di mana pengguna dapat meminjam dan mengembalikan buku. Fungsi-fungsi ini secara tidak langsung memanggil metode dari kelas Buku, mencerminkan hubungan antar objek (komposisi).

V. **Kesimpulan**

Sistem Perpustakaan yang dirancang ini secara menyeluruh menunjukkan aplikasi praktis dari prinsip-prinsip Pemrograman Berorientasi Objek.

Dalam strukturnya yang terorganisir, program ini memisahkan tanggung jawab pengelolaan menjadi komponen-komponen yang berbeda: satu untuk merepresentasikan setiap buku, satu untuk mengelola koleksi perpustakaan secara keseluruhan, dan satu lagi untuk mewakili para pengguna. Setiap komponen ini, atau 'objek', memiliki serangkaian karakteristik uniknya sendiri yang disimpan dengan aman, memastikan bahwa detail seperti judul buku, nama penulis, atau identitas pengguna dijaga kerahasiaannya dan hanya dapat diakses melalui interaksi yang terkontrol. Lebih dari sekadar penyimpanan data, program ini juga mampu melakukan berbagai tindakan yang relevan. Misalnya, dapat melacak jumlah keseluruhan buku yang telah dibuat dalam sistem. Selain itu, ia secara cermat mengelola proses peminjaman dan pengembalian buku, di mana status setiap buku secara otomatis diperbarui untuk mencerminkan ketersediaannya. Interaksi ini difasilitasi oleh metode yang memungkinkan pengguna untuk meminjam atau mengembalikan buku, dengan program yang secara cerdas memeriksa apakah buku tersebut tersedia atau sudah dipinjam sebelum melanjutkan transaksi. Seluruh rancangan ini mengedepankan filosofi di mana informasi internal dijaga kerahasiaannya, namun disediakan cara yang jelas dan terstruktur bagi bagian lain dari program untuk berinteraksi dengannya. Ini menghasilkan sistem yang tidak hanya terorganisir dengan baik tetapi juga mudah dipahami dan diperluas di masa mendatang, menunjukkan kekuatan pendekatan berorientasi objek dalam membangun aplikasi yang kokoh dan dapat diskalakan.

VI. Daftar Pustaka

- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- Liang, Y. D. (2018). *Introduction to Java Programming and Data Structures*. Pearson.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Oracle Java Documentation. <https://docs.oracle.com/javase/8/docs/api/>
- Wibowo, A., & Widodo, A. (2020). Penerapan Object Oriented Programming dalam Pengembangan Sistem Informasi. *Jurnal Teknologi dan Sistem Komputer*, 8(2), 87-93. <https://doi.org/10.14710/jtsiskom.8.2.2020.87-93>