

LAPORAN PROYEK AKHIR
PEMROGRAMAN BERORIENTASI OBJEK
Sistem Perpustakaan Sederhana



DISUSUN OLEH:
2409116048 Rini Wulandari

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	3
1.1 Latar Belakang	3
1.2 Tujuan	3
BAB II PENJELASAN CODINGAN	4
2.1 Source Code & Penjelasan.....	4
2.2 Output	8
BAB III KESIMPULAN.....	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era modern seperti sekarang, teknologi semakin banyak digunakan untuk mempermudah pekerjaan manusia, termasuk dalam mengelola data. Salah satu contoh yang sering kita temui adalah perpustakaan. Di dalam perpustakaan, ada banyak data buku yang harus dicatat, ditampilkan, dan diperbarui agar mudah diakses oleh pengelola maupun pengguna. Jika dilakukan secara manual, tentu akan memakan waktu dan rawan kesalahan. Oleh karena itu, diperlukan sistem sederhana yang dapat membantu proses pengelolaan data buku agar lebih praktis, cepat, dan rapi.

Dalam pembelajaran pemrograman berorientasi objek (PBO), pembuatan sistem perpustakaan sederhana dijadikan sebagai latihan untuk memahami konsep dasar pemrograman. Melalui tugas ini, mahasiswa belajar bagaimana membuat class, objek, menggunakan enkapsulasi, serta menyimpan data dengan ArrayList. Dengan begitu, mahasiswa tidak hanya memahami teori saja, tetapi juga mampu menerapkannya ke dalam program nyata. Selain itu, latihan ini juga bermanfaat untuk melatih logika, keterampilan coding, dan membiasakan mahasiswa dalam merancang aplikasi sederhana yang dapat digunakan dalam kehidupan sehari-hari.

1.2 Tujuan

Tujuan dari tugas ini adalah agar mahasiswa dapat memahami dan menerapkan dasar-dasar pemrograman berorientasi objek melalui pembuatan sistem perpustakaan sederhana. Mahasiswa diharapkan mampu membuat class, objek, serta menggunakan fitur seperti enkapsulasi dan ArrayList untuk mengelola data buku. Selain itu, tugas ini bertujuan untuk melatih keterampilan dalam menyusun logika program, menampilkan data dengan rapi, serta melakukan pembaruan data sesuai kebutuhan.

Dengan adanya tugas ini, mahasiswa tidak hanya belajar menulis kode, tetapi juga berlatih berpikir sistematis dalam merancang sebuah program yang memiliki fungsi nyata. Harapannya, melalui latihan ini mahasiswa dapat lebih siap untuk mengembangkan aplikasi lain yang lebih kompleks di masa depan, serta terbiasa menghubungkan teori dengan praktik pemrograman dalam kehidupan sehari-hari.

BAB II

PENJELASAN CODINGAN

2.1 Source Code & Penjelasan

A. Import

```
package com.mycompany.sistemperpustakaan_tugaspbo;

import java.util.ArrayList;
import java.util.Scanner;
```

- Package, digunakan untuk mengelompokkan class-class yang berhubungan agar lebih terorganisir.
- Import, digunakan untuk mengimpor beberapa kelas terkait pemograman yang memiliki fungsi masing-masing. Dalam program ini saya menggunakan ArrayList dan Scanner yang berfungsi sebagai penghasil list dan alat untuk menginput.

B. Class Buku

```
// ===== CLASS BUKU =====
class Buku {
    private String judul;
    private String pengarang;
    private int tahunTerbit;
    private String isbn;
    private int jumlahStok;

    // Constructor
    public Buku(String judul, String pengarang, int tahunTerbit, String isbn, int jumlahStok) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.tahunTerbit = tahunTerbit;
        this.isbn = isbn;
        this.jumlahStok = jumlahStok;
    }

    // Getter & Setter
    public String getJudul() { return judul; }
    public void setJudul(String judul) { this.judul = judul; }

    public String getPengarang() { return pengarang; }
    public void setPengarang(String pengarang) { this.pengarang = pengarang; }

    public int getTahunTerbit() { return tahunTerbit; }
    public void setTahunTerbit(int tahunTerbit) { this.tahunTerbit = tahunTerbit; }

    public String getIsbn() { return isbn; }
    public void setIsbn(String isbn) { this.isbn = isbn; }

    public int getJumlahStok() { return jumlahStok; }
```

```

    public void setJumlahStok(int jumlahStok) { this.jumlahStok =
jumlahStok; }
}

```

- Class Buku ini digunakan untuk merepresentasikan data buku dalam sistem dengan struktur yang lebih rapi meliputi identitas, stok, dan bisa diakses lewat method.
- Pada kode, diberi akses modifier private yang artinya atribut-atribut ini hanya bisa diakses dari dalam class buku sendiri dan tidak bisa diakses langsung dari luar class buku.
- Constructor dalam kode ini berguna untuk menganalisis objek saat pertama kali dibuat.
- Getter & Setter pada kode ini merupakan method yang berguna untuk mengambil nilai yang bersifat private dan mengubah nilai atribut yang bersifat private.

C. Class Perpustakaan

```

// ===== CLASS PERPUSTAKAAN =====
class Perpustakaan {
    private final ArrayList<Buku> daftarBuku;

    public Perpustakaan() {
        daftarBuku = new ArrayList<>();

        // Tambahkan beberapa buku default
        daftarBuku.add(new Buku("Dasar Pemrograman", "Strolen Tasik",
2020, "ISBN001", 5));
        daftarBuku.add(new Buku("Pemrograman Berorientasi Objek",
"Nurhadi", 2023, "ISBN002", 3));
        daftarBuku.add(new Buku("Algoritma & Struktur Data", "Damayanti",
2019, "ISBN003", 4));
    }

    // Tambah buku
    public void tambahBuku(Buku buku) {
        daftarBuku.add(buku);
    }

    // Tampilkan semua buku dalam format tabel
    public void tampilkanDaftarBuku() {
        if (daftarBuku.isEmpty()) {
            System.out.println("Belum ada buku di perpustakaan.");
        } else {
            System.out.println("\n===== DAFTAR
BUKU PERPUSTAKAAN =====");
            System.out.printf("%-5s %-30s %-20s %-10s %-10s %-5s\n",
"No", "Judul", "Pengarang", "Tahun", "ISBN", "Stok");
            System.out.println("-----");
            System.out.println("-----");

            int no = 1;
            for (Buku buku : daftarBuku) {
                System.out.printf("%-5d %-30s %-20s %-10d %-10s %-5d\n",
no++, buku.getJudul(), buku.getPengarang(),
buku.getTahunTerbit(), buku.getIsbn(),
buku.getJumlahStok());
            }
        }
    }
}

```

```

        System.out.println("=====\n");
    }

    // Update stok buku berdasarkan judul
    public void updateStokBuku(String judul, int stokBaru) {
        for (Buku buku : daftarBuku) {
            if (buku.getJudul().equalsIgnoreCase(judul)) {
                buku.setJumlahStok(stokBaru);
                System.out.println("Stok buku \"" + judul + "\" berhasil
diperbarui.");
                return;
            }
        }
        System.out.println("Buku dengan judul \"" + judul + "\" tidak
ditemukan.");
    }
}

```

- Class Perpustakaan dibuat dengan atribut daftarBuku yang bertipe *ArrayList* yang berfungsi sebagai pengelola daftar buku yang mencakup menambahkan, menampilkan dan memperbarui stok baru.
- Public Perpustakaan, berguna saat objek perpustakaan dibuat daftarBuku diinisialisasi sebagai *ArrayList* kosong. Sebelumnya, daftar buku default telah ditambahkan sebanyak tiga buku agar perpustakaan tidak kosong.
- Method tambahBuku(), saya masukkan untuk menambahkan objek buku baru dala daftarBuku.
- Method tampilkanDaftarBuku(), berguna untuk mengecek apakah daftar buku kosong atau ada isi. Jika ada isinya, akan menampilkan daftar semua buku dengan format table yang telah disesuaikan.
- System.out.printf berguna agar tampilan table lebih rapi.
- Method updateStokBuku(), digunakan untuk memperbarui stok suatu buku berdasarkan judul.
- Penggunaan equalsIgnoreCase() digunakan agar pencarian tidak terpaku pada huruf besar/kecil.

D. Class Main

```

// ===== CLASS MAIN =====
public class SistemPerpustakaan_TugasPBO {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Perpustakaan perpustakaan = new Perpustakaan();
        int pilihan;

        do {
            System.out.println("\n== MENU PERPUSTAKAAN ==");
            System.out.println("1. Tambah Buku");
            System.out.println("2. Tampilkan Daftar Buku");
            System.out.println("3. Update Stok Buku");
            System.out.println("4. Keluar");
            System.out.print("Pilih menu : ");
            pilihan = input.nextInt();
        }
    }
}

```

```

        input.nextLine(); //

        switch (pilihan) {
            case 1 -> {
                System.out.println("\nMasukkan data buku baru:");
                System.out.print("Judul : ");
                String judul = input.nextLine();
                System.out.print("Pengarang : ");
                String pengarang = input.nextLine();
                System.out.print("Tahun Terbit: ");
                int tahun = input.nextInt();
                input.nextLine();
                System.out.print("ISBN : ");
                String isbn = input.nextLine();
                System.out.print("Jumlah Stok : ");
                int stok = input.nextInt();
                input.nextLine();

                Buku buku = new Buku(judul, pengarang, tahun, isbn,
stok);
                perpustakaan.tambahBuku(buku);
                System.out.println("Buku berhasil ditambahkan!");
            }

            case 2 -> perpustakaan.tampilkanDaftarBuku();

            case 3 -> {
                System.out.print("\nMasukkan judul buku yang ingin
diupdate stoknya: ");
                String judulUpdate = input.nextLine();
                System.out.print("Masukkan stok baru: ");
                int stokBaru = input.nextInt();
                perpustakaan.updateStokBuku(judulUpdate, stokBaru);
            }

            case 4 -> System.out.println("Terima kasih. Program
selesai ^-^.");
        }
    }
}

```

- Class Main dibuat sebagai class utama sebagai tempat program dijalankan.
- Scanner input, berguna untuk membaca input dari data yang akan dimasukkan. Untuk objek Perpustakaan agar pengelolaan data buku bisa dilakukan.
- Menu Perpustakaan, yang menerapkan *Loop Do-While* yang akan menampilkan menu secara berulang sampai user memilih keluar dari program.
- Penerapan *Switch-Case* untuk menu, agar user bisa memilih pilihan yang telah diberikan.
- Input.close(), untuk menutup objek *Scanner* untuk menghindari kebocoran *resource*.

2.2 Output

- Menu Perpustakaan.

```
==== MENU PERPUSTAKAAN ====
1. Tambah Buku
2. Tampilkan Daftar Buku
3. Update Stok Buku
4. Keluar
Pilih menu : 2
```

- Daftar Buku Perpustakaan.

```
===== DAFTAR BUKU PERPUSTAKAAN =====
No   Judul           Pengarang       Tahun     ISBN      Stok
-----
1    Dasar Pemrograman  Strolen Tasik  2020     ISBN001    5
2    Pemrograman Berorientasi Objek Nurhadi 2023     ISBN002    3
3    Algoritma & Struktur Data  Damayanti   2019     ISBN003    4
=====
```

- Data Buku Berhasil ditambahkan.

```
Masukkan data buku baru:
Judul      : Pengantar Manajemen Bisnis
Pengarang   : Nurul Hidayat
Tahun Terbit: 2025
ISBN       : ISBN004
Jumlah Stok : 8
Buku berhasil ditambahkan!
```

```
===== DAFTAR BUKU PERPUSTAKAAN =====
No   Judul           Pengarang       Tahun     ISBN      Stok
-----
1    Dasar Pemrograman  Strolen Tasik  2020     ISBN001    5
2    Pemrograman Berorientasi Objek Nurhadi 2023     ISBN002    3
3    Algoritma & Struktur Data  Damayanti   2019     ISBN003    4
4    Pengantar Manajemen Bisnis  Nurul Hidayat 2025     ISBN004    8
=====
```

- Update Stok Berhasil diupdate.

```
Masukkan judul buku yang ingin diupdate stoknya: Dasar Pemrograman  
Masukkan stok baru: 3  
Stok buku "Dasar Pemrograman" berhasil diperbarui.
```

===== DAFTAR BUKU PERPUSTAKAAN =====					
No	Judul	Pengarang	Tahun	ISBN	Stok
1	Dasar Pemrograman	Strolen Tasik	2020	ISBN001	3
2	Pemrograman Berorientasi Objek	Nurhadi	2023	ISBN002	3
3	Algoritma & Struktur Data	Damayanti	2019	ISBN003	4
4	Pengantar Manajemen Bisnis	Nurul Hidayat	2025	ISBN004	8

- Program Selesai.

```
==== MENU PERPUSTAKAAN ====  
1. Tambah Buku  
2. Tampilkan Daftar Buku  
3. Update Stok Buku  
4. Keluar  
Pilih menu : 4  
Terima kasih. Program selesai ^-^.  
-----  
BUILD SUCCESS  
-----
```

BAB III

KESIMPULAN

Berdasarkan hasil pembuatan dan penjelasan program **Sistem Perpustakaan Sederhana** ini, dapat disimpulkan bahwa penerapan konsep **Pemrograman Berorientasi Objek (PBO)** seperti class, objek, enkapsulasi, constructor, serta penggunaan ArrayList berhasil membantu mengelola data buku dengan lebih terstruktur. Program ini memungkinkan pengguna untuk menambahkan buku baru, menampilkan daftar buku dalam format tabel yang rapi, serta memperbarui stok buku sesuai kebutuhan.

Dengan adanya sistem ini, pengelolaan data perpustakaan menjadi lebih mudah, cepat, dan efisien dibandingkan cara manual. Selain itu, pembuatan program ini juga melatih kemampuan mahasiswa dalam memahami teori OOP dan mengimplementasikannya dalam bentuk aplikasi nyata yang dapat dikembangkan lebih lanjut di masa depan.