# LAPORAN TUGAS 2

Diajukan untuk memenuhi salah satu tugas praktikum mata kuliah Pemograman Beroientasi Objek (Teori).



**Disusun Oleh:**

**Muhammd Ihsan Ramadhan 241511083**

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**

**Politeknik Negeri Bandung**

**2025**

# DAFTAR ISI

[KOVER 1](#_Toc207400955)

[DAFTAR ISI 2](#_Toc207400956)

[1. Deskripsi Aplikasi 3](#_Toc207400957)

[2. Source Code 3](#_Toc207400958)

[2.1. Barang.java 3](#_Toc207400959)

[2.2. Peminjaman.java 4](#_Toc207400960)

[2.3. Main.java 5](#_Toc207400961)

[3. Screenshot Hasil Implementasi 7](#_Toc207400962)

[3.1. Menu Utama 7](#_Toc207400963)

[3.2. Menu 1. Pinjam Barang 8](#_Toc207400964)

[3.3. Menginputkan Kode Barang dan Nama Peminjam 8](#_Toc207400965)

[3.4. Tampilan Menu 1 Setelah Barang di Pinjam (Proyektor tidak tersedia) 8](#_Toc207400966)

[3.5. Menu 2. Kembalikan Barang (Proyektor) 8](#_Toc207400967)

[3.6. Menu 3.Lihat Daftar Peminjaman 8](#_Toc207400968)

[4. Lesson Learned 8](#_Toc207400969)

## Deskripsi Aplikasi

Aplikasi yang dibuat adalah Sistem Peminjaman Barang di Teknisi JTK.  
Program ini menggunakan konsep Class dan Object dengan penerapan Encapsulation untuk mengatur atribut dari setiap barang yang dipinjam.

Dalam aplikasi ini:

* Terdapat class Barang yang merepresentasikan data barang (nama barang, kode, jumlah, dan status).
* Terdapat class Peminjaman yang mengatur data peminjam (nama peminjam, tanggal pinjam, tanggal kembali, serta barang yang dipinjam).
* Aplikasi dijalankan melalui class Main untuk melakukan simulasi peminjaman barang.

Aplikasi ini bertujuan untuk:

* Mencatat barang yang dipinjam oleh pengguna.
* Menampilkan detail peminjaman dengan data terenkapsulasi.
* Memberikan gambaran sederhana implementasi OOP dengan Java.

## Source Code

### Barang.java

public class Barang {

    private String kodeBarang;

    private String namaBarang;

    private boolean tersedia;

    public Barang(String kodeBarang, String namaBarang) {

        this.kodeBarang = kodeBarang;

        this.namaBarang = namaBarang;

        this.tersedia = true;

    }

    public String getKodeBarang() {

        return kodeBarang;

    }

    public String getNamaBarang() {

        return namaBarang;

    }

    public boolean isTersedia() {

        return tersedia;

    }

    public void setTersedia(boolean tersedia) {

        this.tersedia = tersedia;

    }

    public String toString() {

        return kodeBarang + " - " + namaBarang + (tersedia ? " (Tersedia)" : " (Dipinjam)");

    }

}

### Peminjaman.java

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class Peminjaman {

    private String namaPeminjam;

    private Barang barang;

    private LocalDateTime tanggalPinjam;

    private LocalDateTime tanggalKembali;

    public Peminjaman(String namaPeminjam, Barang barang) {

        this.namaPeminjam = namaPeminjam;

        this.barang = barang;

        this.tanggalPinjam = LocalDateTime.now();

        this.tanggalKembali = null;

    }

    public String getNamaPeminjam() {

        return namaPeminjam;

    }

    public Barang getBarang() {

        return barang;

    }

    public LocalDateTime getTanggalPinjam() {

        return tanggalPinjam;

    }

    public LocalDateTime getTanggalKembali() {

        return tanggalKembali;

    }

    public void setTanggalKembali(LocalDateTime tanggalKembali) {

        this.tanggalKembali = tanggalKembali;

    }

    public String toString() {

        DateTimeFormatter fmt = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");

        return "Peminjam: " + namaPeminjam +

                ", Barang: " + barang.getNamaBarang() +

                ", Tanggal Pinjam: " + tanggalPinjam.format(fmt) +

                (tanggalKembali != null ? ", Tanggal Kembali: " + tanggalKembali.format(fmt) : " (Belum dikembalikan)");

    }

}

### Main.java

import java.time.LocalDateTime;

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ArrayList<Barang> daftarBarang = new ArrayList<>();

        daftarBarang.add(new Barang("A01", "Proyektor"));

        daftarBarang.add(new Barang("A02", "Terminal Listrik"));

        daftarBarang.add(new Barang("A03", "Kabel HDMI"));

        ArrayList<Peminjaman> daftarPeminjaman = new ArrayList<>();

        int pilih;

        do {

            System.out.println("\n=== Sistem Peminjaman Barang JTK ===");

            System.out.println("1. Pinjam Barang");

            System.out.println("2. Kembalikan Barang");

            System.out.println("3. Lihat Daftar Peminjaman");

            System.out.println("4. Keluar");

            System.out.print("Pilih menu: ");

            pilih = sc.nextInt();

            sc.nextLine();

            switch (pilih) {

                case 1:

                    System.out.println("\n=== Daftar Barang Tersedia ===");

                    for (Barang b : daftarBarang) {

                        if (b.isTersedia()) {

                            System.out.println(b);

                        }

                    }

                    System.out.print("Masukkan kode barang: ");

                    String kodePinjam = sc.nextLine();

                    System.out.print("Masukkan nama peminjam: ");

                    String nama = sc.nextLine();

                    boolean ditemukan = false;

                    for (Barang b : daftarBarang) {

                        if (b.getKodeBarang().equalsIgnoreCase(kodePinjam) && b.isTersedia()) {

                            b.setTersedia(false);

                            Peminjaman p = new Peminjaman(nama, b);

                            daftarPeminjaman.add(p);

                            System.out.println("Barang berhasil dipinjam: " + b.getNamaBarang());

                            ditemukan = true;

                            break;

                        }

                    }

                    if (!ditemukan) {

                        System.out.println("Barang tidak tersedia atau kode salah!");

                    }

                    break;

                case 2:

                    System.out.print("Masukkan kode barang yang dikembalikan: ");

                    String kodeKembali = sc.nextLine();

                    boolean ketemu = false;

                    for (Peminjaman p : daftarPeminjaman) {

                        if (p.getBarang().getKodeBarang().equalsIgnoreCase(kodeKembali) && p.getTanggalKembali() == null) {

                            p.setTanggalKembali(LocalDateTime.now());

                            p.getBarang().setTersedia(true);

                            System.out.println("Barang berhasil dikembalikan: " + p.getBarang().getNamaBarang());

                            ketemu = true;

                            break;

                        }

                    }

                    if (!ketemu) {

                        System.out.println("Data peminjaman tidak ditemukan!");

                    }

                    break;

                case 3:

                    System.out.println("\n=== Daftar Peminjaman ===");

                    if (daftarPeminjaman.isEmpty()) {

                        System.out.println("Belum ada data peminjaman.");

                    } else {

                        for (Peminjaman p : daftarPeminjaman) {

                            System.out.println(p);

                        }

                    }

                    break;

                case 4:

                    System.out.println("Terima kasih!");

                    break;

                default:

                    System.out.println("Menu tidak valid!");

            }

        } while (pilih != 4);

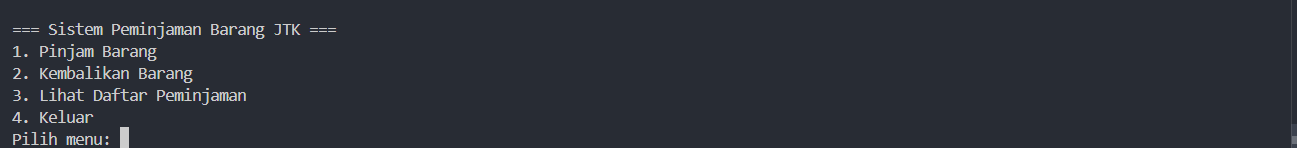
        sc.close();

    }

}

## Screenshot Hasil Implementasi

### Menu Utama



### Menu 1. Pinjam Barang

Sebuah gambar berisi teks, Font, cuplikan layar, tipografi

Konten yang dihasilkan AI mungkin salah.

### Menginputkan Kode Barang dan Nama Peminjam

Sebuah gambar berisi teks, Font, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

### Tampilan Menu 1 Setelah Barang di Pinjam (Proyektor tidak tersedia)

Sebuah gambar berisi teks, Font, cuplikan layar, deasin

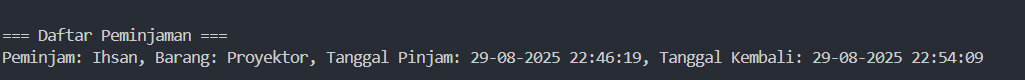
Konten yang dihasilkan AI mungkin salah.

### Menu 2. Kembalikan Barang (Proyektor)

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

### Menu 3.Lihat Daftar Peminjaman



## Lesson Learned

Dari tugas ini, saya mempelajari:

1. Bagaimana cara membuat class dan object di Java.
2. Pentingnya penggunaan enkapsulasi (private attribute + getter & setter) agar data lebih aman dan terkontrol.
3. Penerapan OOP membantu menyusun kode lebih rapi, terstruktur, dan mudah dikembangkan.
4. Bagaimana membuat relasi antar class, misalnya Peminjaman memiliki atribut Barang.
5. Praktik implementasi aplikasi sederhana yang bisa dikembangkan lebih lanjut menjadi sistem peminjaman yang nyata.