

Nama: Fahraj Ananta Aulia Arkan

NIM: 241511042

Prodi/Kelas: D3/2B

Link Github: <https://github.com/ananta-cloud/PBO-Te.git>

## Tugas – 2: Object & Class

Aplikasi ini adalah simulasi sistem peminjaman barang sederhana di lingkungan Teknisi JTK berbasis Java. Pengguna dapat berinteraksi melalui konsol untuk melakukan berbagai operasi seperti meminjam barang, mengembalikan barang, menambah data barang baru ke dalam sistem, serta melihat status ketersediaan semua barang dan daftar peminjam yang terdaftar.

Aplikasi ini mensimulasikan alur kerja dasar dari sebuah sistem inventaris, di mana data peminjam dan barang dikelola secara dinamis.

### Fitur Utama

#### 1. Menambah & Melihat Data

- Pengguna dapat menambahkan data barang baru yang akan tersedia untuk dipinjam.
- Sistem dapat menampilkan daftar lengkap semua barang beserta statusnya saat ini (apakah Tersedia atau sedang Dipinjam oleh siapa).
- Terdapat fitur untuk melihat daftar semua peminjam yang pernah berinteraksi dengan sistem, lengkap dengan Nama, NIM, dan Kelas mereka.

#### 2. Meminjam Barang

- Pengguna memulai proses dengan memasukkan NIM.
- Jika NIM belum terdaftar, sistem akan meminta pengguna untuk melengkapi data (Nama, Kelas) dan secara otomatis menyimpannya sebagai peminjam baru.
- Jika NIM sudah terdaftar, sistem akan mengenali pengguna.
- Pengguna kemudian akan diperlihatkan daftar barang yang tersedia dan dapat memilih barang yang ingin dipinjam.

#### 3. Mengembalikan Barang

- Untuk verifikasi, pengguna harus memasukkan NIM mereka.

- Sistem akan menampilkan daftar barang yang sedang dipinjam oleh pengguna tersebut.
  - Setelah pengguna memilih barang yang akan dikembalikan, status barang tersebut di sistem akan otomatis diperbarui menjadi Tersedia.
4. Sistem Pengguna Tunggal (Single User Role)
- Aplikasi ini saat ini berjalan dengan satu jenis peran (role) pengguna.
  - Artinya, setiap pengguna yang menjalankan aplikasi dapat mengakses semua fitur yang tersedia, seperti meminjam barang (fungsi 'user') serta menambah barang baru (fungsi 'admin').

## **Konsep yang Digunakan**

1. Encapsulation (Enkapsulasi) Semua atribut di kelas Peminjam dan Barang bersifat private. Akses terhadap data ini dikontrol sepenuhnya melalui method publik (getter), seperti getName() atau getStatus(). Hal ini memastikan integritas data dan menyembunyikan detail implementasi internal dari objek.
2. ArrayList Struktur data ArrayList digunakan sebagai "database" dinamis di dalam memori untuk menyimpan koleksi objek:
  - ArrayList<Peminjam> daftarPeminjam: Menampung semua objek peminjam yang telah terdaftar.
  - ArrayList<Barang> daftarBarang: Menampung semua objek barang yang ada di sistem.
3. Reusability (Reusabilitas) Kelas Peminjam dan Barang dirancang sebagai komponen yang dapat digunakan kembali. Contohnya:
  - Sebuah objek Peminjam yang dibuat akan disimpan dan dapat digunakan kembali untuk transaksi peminjaman berikutnya tanpa perlu input data ulang.
  - Sebuah objek Barang dapat dipinjam dan dikembalikan berkali-kali oleh objek Peminjam yang berbeda, menunjukkan bahwa objek tersebut memiliki siklus hidupnya sendiri dan dapat berinteraksi dengan banyak objek lain.

## Source Code

File	Code
Peminjam.class	<pre>class <u>Peminjam</u> {     private <i>String</i> nama;     private <i>String</i> nim;     private <i>String</i> kelas;      public Peminjam(<i>String</i> var1, <i>String</i> var2, <i>String</i> var3) {         this.nama = var1;         this.nim = var2;         this.kelas = var3;     }      public <i>String</i> getNama() {         return this.nama;     }      public <i>String</i> getNim() {         return this.nim;     }      public <i>String</i> getKelas() {         return this.kelas;     } }</pre>
Barang.class	<pre>class <u>Barang</u> {     private <i>String</i> namaBarang;     private <i>boolean</i> tersedia;     private <i>Peminjam</i> dipinjamOleh;      public Barang(<i>String</i> var1) {         this.namaBarang = var1;         this.tersedia = true;         this.dipinjamOleh = null;     }      public void pinjam(<i>Peminjam</i> var1) {         if (this.tersedia) {             this.tersedia = false;             this.dipinjamOleh = var1;         }     } }</pre>

	<pre> }  public void kembalikan() {     this.tersedia = true;     this.dipinjamOleh = null; }  public String getNamaBarang() {     return this.namaBarang; }  public boolean isTersedia() {     return this.tersedia; }  public Peminjam getDipinjamOleh() {     return this.dipinjamOleh; }  public String getStatus() {     if (this.tersedia) {         return " Tersedia";     } else {         String var10000 = this.dipinjamOleh.getNama();         return "Dipinjam oleh: " + var10000 + " (NIM: " + this.dipinjamOleh.getNim() + ")";     } } } </pre>
AplikasiPeminjaman.class	<pre> import java.io.PrintStream; import java.util.ArrayList; import java.util.Iterator; import java.util.Scanner;  public class <u>AplikasiPeminjaman</u> {     static ArrayList&lt;Peminjam&gt; daftarPeminjam = new ArrayList();     static ArrayList&lt;Barang&gt; daftarBarang = new ArrayList();      public AplikasiPeminjaman() {     }      public static void main(String[] var0) {         inialisasiData();         Scanner var1 = new Scanner(System.in); </pre>

```

int var2 = -1;

while(var2 != 0) {
    tampilkanMenu();
    System.out.print("Masukkan pilihan Anda: ");

    try {
        var2 = Integer.parseInt(var1.nextLine());
    } catch (NumberFormatException var4) {
        System.out.println("Input tidak valid, silakan masukkan angka.");
        var2 = -1;
        continue;
    }

    switch (var2) {
        case 0:
            System.out.println("Terima kasih, program selesai.");
            break;
        case 1:
            prosesPinjam(var1);
            break;
        case 2:
            prosesKembalikan(var1);
            break;
        case 3:
            lihatStatusSemuaBarang();
            break;
        case 4:
            lihatDaftarPeminjam();
            break;
        case 5:
            tambahBarang(var1);
            break;
        default:
            System.out.println("Pilihan tidak valid. Silakan coba lagi.");
    }

    if (var2 != 0) {
        System.out.println("\nTekan Enter untuk kembali ke menu...");
        var1.nextLine();
    }
}

var1.close();
}

```

```

public static void tampilkanMenu() {
    System.out.println("\n===== MENU PEMINJAMAN BARANG
=====");
    System.out.println("1. Pinjam Barang");
    System.out.println("2. Kembalikan Barang");
    System.out.println("3. Lihat Status Semua Barang");
    System.out.println("4. Lihat Daftar Peminjam");
    System.out.println("5. Tambah Barang Baru");
    System.out.println("0. Keluar");
    System.out.println("=====");
}

public static void tambahBarang(Scanner var0) {
    System.out.println("\n--- Tambah Barang Baru ---");
    System.out.print("Masukkan nama barang baru: ");
    String var1 = var0.nextLine();
    if (var1 != null && !var1.trim().isEmpty()) {
        Barang var2 = new Barang(var1);
        daftarBarang.add(var2);
        System.out.println("\nBerhasil! Barang '" + var1 + "' telah ditambahkan
ke sistem.");
    } else {
        System.out.println("\nGagal! Nama barang tidak boleh kosong.");
    }
}

public static void inisialisasiData() {
    daftarPeminjam.add(new Peminjam("Budi Santoso", "2301001",
"Informatika-A"));
    daftarPeminjam.add(new Peminjam("Citra Lestari", "2302005", "Sistem
Informasi-B"));
    daftarBarang.add(new Barang("Proyektor Epson EB-S41"));
    daftarBarang.add(new Barang("Kabel HDMI 5m"));
}

public static void prosesPinjam(Scanner var0) {
    System.out.println("\n--- Proses Peminjaman Barang ---");
    System.out.print("Masukkan NIM Anda: ");
    String var1 = var0.nextLine();
    Peminjam var2 = null;
    Iterator var3 = daftarPeminjam.iterator();

    while(var3.hasNext()) {
        Peminjam var4 = (Peminjam)var3.next();
        if (var4.getNim().equals(var1)) {
            var2 = var4;

```

```

        System.out.println("Selamat datang kembali, " + var4.getNama() +
"!");
        break;
    }
}

String var10;
if (var2 == null) {
    System.out.println("NIM tidak terdaftar. Silakan masukkan data baru.");
    System.out.print("Masukkan Nama Anda: ");
    String var8 = var0.nextLine();
    System.out.print("Masukkan Kelas Anda: ");
    var10 = var0.nextLine();
    var2 = new Peminjam(var8, var1, var10);
    daftarPeminjam.add(var2);
    System.out.println("\nData peminjam baru '" + var8 + "' berhasil
disimpan.");
}

System.out.println("\nDaftar barang yang tersedia:");
ArrayList var9 = new ArrayList();
Iterator var11 = daftarBarang.iterator();

while(var11.hasNext()) {
    Barang var5 = (Barang)var11.next();
    if (var5.isTersedia()) {
        var9.add(var5);
        System.out.println("- " + var5.getNamaBarang());
    }
}

if (var9.isEmpty()) {
    System.out.println("Maaf, tidak ada barang yang tersedia untuk
dipinjam.");
} else {
    System.out.print("\nKetik nama barang yang ingin dipinjam: ");
    var10 = var0.nextLine();
    boolean var12 = false;
    Iterator var6 = var9.iterator();

    while(var6.hasNext()) {
        Barang var7 = (Barang)var6.next();
        if (var7.getNamaBarang().equalsIgnoreCase(var10)) {
            var7.pinjam(var2);
            PrintStream var10000 = System.out;
            String var10001 = var2.getNama();

```

```

        var10000.println("\nBerhasil! " + var10001 + " telah meminjam "
+ var7.getNamaBarang() + ".");
        var12 = true;
        break;
    }
}

    if (!var12) {
        System.out.println("\nGagal! Barang tidak ditemukan atau sudah
dipinjam.");
    }

}
}

public static void prosesKembalikan(Scanner var0) {
    System.out.println("\n--- Proses Pengembalian Barang ---");
    System.out.print("Masukkan NIM Anda untuk verifikasi: ");
    String var1 = var0.nextLine();
    Peminjam var2 = null;
    Iterator var3 = daftarPeminjam.iterator();

    while(var3.hasNext()) {
        Peminjam var4 = (Peminjam)var3.next();
        if (var4.getNim().equals(var1)) {
            var2 = var4;
            break;
        }
    }

    if (var2 == null) {
        System.out.println("\nGagal! Peminjam dengan NIM " + var1 + " tidak
ditemukan.");
    } else {
        ArrayList var6 = new ArrayList();
        Iterator var7 = daftarBarang.iterator();

        Barang var5;
        while(var7.hasNext()) {
            var5 = (Barang)var7.next();
            if (!var5.isTersedia() && var5.getDipinjamOleh() == var2) {
                var6.add(var5);
            }
        }

        if (var6.isEmpty()) {

```



```

        System.out.println("\n" + var2.getNama() + " tidak memiliki barang
yang sedang dipinjam.");
    } else {
        System.out.println("\nBarang yang Anda pinjam (" + var2.getNama()
+ "):");

        int var8;
        for(var8 = 0; var8 < var6.size(); ++var8) {
            System.out.println(var8 + 1 + ". " +
((Barang)var6.get(var8)).getNamaBarang());
        }

        System.out.print("Pilih nomor barang yang ingin dikembalikan: ");
        var8 = Integer.parseInt(var0.nextLine());
        if (var8 > 0 && var8 <= var6.size()) {
            var5 = (Barang)var6.get(var8 - 1);
            var5.kembalikan();
            System.out.println("\n Berhasil! " + var5.getNamaBarang() + " telah
dikembalikan.");
        } else {
            System.out.println("\n Pilihan tidak valid.");
        }

    }
}

public static void lihatStatusSemuaBarang() {
    System.out.println("\n--- Status Semua Barang ---");
    Iterator var0 = daftarBarang.iterator();

    while(var0.hasNext()) {
        Barang var1 = (Barang)var0.next();
        PrintStream var10000 = System.out;
        String var10001 = var1.getNamaBarang();
        var10000.println("- " + var10001 + " | Status: " + var1.getStatus());
    }
}

public static void lihatDaftarPeminjam() {
    System.out.println("\n--- Daftar Peminjam Terdaftar ---");
    if (daftarPeminjam.isEmpty()) {
        System.out.println("Belum ada peminjam yang terdaftar.");
    }
}

```

	<pre> Iterator var0 = daftarPeminjam.iterator();  while(var0.hasNext()) {     Peminjam var1 = (Peminjam)var0.next();     PrintStream var10000 = System.out;     String var10001 = var1.getNama();     var10000.println("- Nama: " + var10001 + ", NIM: " + var1.getNim() + ", Kelas: " + var1.getKelas()); } } } </pre>
--	---

## Hasil Program

Gambar		Fitur
	<pre> ===== MENU PEMINJAMAN BARANG ===== 1. Pinjam Barang 2. Kembalikan Barang 3. Lihat Status Semua Barang 4. Lihat Daftar Peminjam 5. Tambah Barang Baru 0. Keluar ===== Masukkan pilihan Anda: █ </pre>	Menu utama (user/admin)
	<pre> ===== Masukkan pilihan Anda: 1  --- Proses Peminjaman Barang --- Masukkan NIM Anda: 241511042 NIM tidak terdaftar. Silakan masukkan data baru. Masukkan Nama Anda: Fahraj Ananta Aulia Arkan Masukkan Kelas Anda: 2B D3  Data peminjam baru 'Fahraj Ananta Aulia Arkan' berhasil disimpan.  Daftar barang yang tersedia: - Proyektor Epson EB-S41 - Kabel HDMI 5m  Ketik nama barang yang ingin dipinjam: Proyektor Epson EB-S41  Berhasil! 'Fahraj Ananta Aulia Arkan' telah meminjam 'Proyektor Epson EB-S41'.  Tekan Enter untuk kembali ke menu... </pre>	Peminjaman barang (user)

<pre>===== Masukkan pilihan Anda: 2  --- Proses Pengembalian Barang --- Masukkan NIM Anda untuk verifikasi: 241511042  Barang yang Anda pinjam (Fahraj Ananta Aulia Arkan): 1. Proyektor Epson EB-S41 Pilih nomor barang yang ingin dikembalikan: 1      Berhasil! 'Proyektor Epson EB-S41' telah dikembalikan.  Tekan Enter untuk kembali ke menu...</pre>	<p>Pengembalian barang (user)</p>
<pre>===== Masukkan pilihan Anda: 3  --- Status Semua Barang --- - Proyektor Epson EB-S41   Status: Tersedia - Kabel HDMI 5m   Status: Tersedia  Tekan Enter untuk kembali ke menu...</pre>	<p>Menampilkan data informasi status ketersediaan barang (user/admin)</p>
<pre>===== Masukkan pilihan Anda: 4  --- Daftar Peminjam Terdaftar --- - Nama: Budi Santoso, NIM: 2301001, Kelas: 1B D3 - Nama: Citra Lestari, NIM: 2302005, Kelas: 2B D4  Tekan Enter untuk kembali ke menu...</pre>	<p>Menampilkan data peminjam yang pernah terdaftar (admin/user)</p>

<pre> ===== Masukkan pilihan Anda: 5  --- Tambah Barang Baru --- Masukkan nama barang baru: Kabel HDMI 10m  Berhasil! Barang 'Kabel HDMI 10m' telah ditambahkan ke sistem.  Tekan Enter untuk kembali ke menu... </pre>	Tambah data barang (admin)
---	-------------------------------

## Lesson Learned

### 1. Pemahaman Enkapsulasi

Praktik nyata membantu mengerti alur bagaimana object dibuat, bagaimana atribut dan method saling berhubungan, serta bagaimana konsep OOP diterapkan dalam kode.

### 2. Pemahaman Getter dan Setter

Saya jadi paham bahwa getter digunakan untuk mengambil nilai atribut secara aman, sementara setter dipakai untuk mengubah nilai atribut dengan kendali tertentu. Konsep ini membantu menjaga prinsip enkapsulasi, supaya atribut tidak bisa diakses/diubah sembarangan dari luar class.

### 3. Object dalam Array dan ArrayList

Object tidak hanya dibuat satu per satu, tetapi juga dapat disimpan dalam bentuk Array maupun ArrayList. Array cocok untuk jumlah data yang tetap, sedangkan ArrayList lebih fleksibel karena ukurannya bisa berubah.

## Referensi

[https://www.bing.com/search?q=UT&pq=java+encap&sk=CSYN1&sc=10-10&pqlt=161&q=java+encapsulation&cvid=0f8ecf58bcad43958a3931bf80078456&gs\\_lcrp=EgRlZGdlKgcIABAAGPkHMgcIABAAGPkHMgYIARBFgDkyBggCEAAYQDIGCAMQABhAMgYIBBAAGEAyBggFEAAYQDIGCAYQABhAMgYIBxAAGEAyBggIEAAYQNIBCDQxMDVqMGoxqAIAA&FORM=ANSPA1&PC=ESBXSP](https://www.bing.com/search?q=UT&pq=java+encap&sk=CSYN1&sc=10-10&pqlt=161&q=java+encapsulation&cvid=0f8ecf58bcad43958a3931bf80078456&gs_lcrp=EgRlZGdlKgcIABAAGPkHMgcIABAAGPkHMgYIARBFgDkyBggCEAAYQDIGCAMQABhAMgYIBBAAGEAyBggFEAAYQDIGCAYQABhAMgYIBxAAGEAyBggIEAAYQNIBCDQxMDVqMGoxqAIAA&FORM=ANSPA1&PC=ESBXSP)

Java Encapsulation and Getters and Setters

[https://www.w3schools.com/java/java\\_arraylist.asp](https://www.w3schools.com/java/java_arraylist.asp)

<https://stackoverflow.com/questions/3982550/creating-an-arraylist-of-objects>