

LAPORAN PRAKTIKUM PBO JAVA – MANAJEMEN STOK



Nama: Budi Anantha Trikurnia Irawan

NIM: 2024103701110337

Kelas: Pemrograman Berbasis Objek 2A

1. PENDAHULUAN

Pada Praktikum ini, dibuat sebuah simulasi Sistem Peminjaman Buku Perpustakaan menggunakan konsep Object Oriented Programming (OOP) di dalam Java. Tujuan diadakan praktikum ini adalah untuk memahami penerapan:

- Menggunakan ArrayList untuk menyimpan objek
- Menangani exception dengan try-catch
- Membuat custom exception

2. STRUKTUR PROGRAM

Berikut merupakan kelas-kelas yang digunakan dalam mengerjakan Codelab:

NAMA KELAS	TIPE	DESKRIPSI
Barang	Class	Representasi objek barang dengan atribut nama dan stok.
StokBarangTidakCukupException	Exception	Custom exception untuk menangani kasus stok tidak cukup.
ManajemenStok	Main	Berisi menu program dan logika.

3. PENJELASAN CODE

1. Kelas Barang

```
package PBOM5.CL;

public class Barang {
    private String nama;
    private int stok;

    public Barang(String nama, int stok) {
        this.nama = nama;
        this.stok = stok;
    }

    public String getNama() {
        return nama;
    }

    public int getStok() {
        return stok;
    }

    public void setStok(int stok) {
        this.stok = stok;
    }
}
```

Kelas barang disini berisi atribut yang bakal digunakan untuk kelas selanjutnya dengan menerapkan encapsulation dan atributnya bersifat private sehingga tidak dapat diakses langsung dari luar class. Untuk mengakses dan memodifikasi nilai atribut tersebut, class menyediakan method getter & setter yang membuat akses terkontrol terhadap data. Constructor disini berguna untuk menerima dua parameter yaitu nama dan stok yang digunakan untuk menginisialisasi objek Barang.

2. Kelas StokBarangTidakCukupException

```
1 package PBOM5.CL;
2
3 public class StokBarangTidakCukupException extends Exception {
4     public StokBarangTidakCukupException(String message) {
5         super(message);
6     }
7 }
8
9
```

Class StokBarangTidakCukupException merupakan implementasi custom exception yang meng-extend class Exception bawaan dari java. Class ini dibuat khusus untuk menangani kasus spesifik yaitu ketika stok barang tidak mencukupi untuk melakukan operasi pengurangan. Contructor class ini menerima parameter berupa string message yang diteruskan ke constructor superclass melalui keyword super. Dengan adanya custom exception, program ini dapat memberikan pesan error yang lebih spesifik, serta memudahkan pengembangan dalam menangani kasus error dengan stok barang.

3. Kelas ManajemenStok

```

package PBO09.Cl;
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;

public class ManajemenStok {
    private static ArrayList<Barang> daftarBarang = new ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        daftarBarang.add(new Barang("Pensil", 50));
        daftarBarang.add(new Barang("Buku", 10));
        daftarBarang.add(new Barang("Penghapus", 20));

        boolean running = true;
        while (running) {
            try {
                System.out.println("\n===== Menu Manajemen Stok =====");
                System.out.println("1. Tambah Barang Baru");
                System.out.println("2. Tampilkan Semua Barang");
                System.out.println("3. Kurangi Stok Barang");
                System.out.println("4. Keluar");
                System.out.print("Pilih opsi: ");

                int pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
                    case 1:
                        tambahBarang();
                        break;
                    case 2:
                        tampilkanDaftarBarang();
                        break;
                    case 3:
                        kurangiStok();
                        break;
                    case 4:
                        running = false;
                        System.out.println("Terima kasih! Program berakhir.");
                        break;
                    default:
                        System.out.println("Pilihan tidak valid!");
                }
            } catch (InputMismatchException e) {
                System.out.println("Input tidak valid! Masukkan angka.");
                scanner.nextLine();
            }
        }

        scanner.close();
    }
}

```

```

public static void tampilkanDaftarBarang() {
    System.out.println("===== Daftar Barang =====");
    if (daftarBarang.isEmpty()) {
        System.out.println("Tidak ada barang yang tersedia.");
        return;
    } else {
        Iterator<Barang> iterator = daftarBarang.iterator();
        int index = 1;
        while (iterator.hasNext()) {
            Barang barang = iterator.next();
            System.out.println(index + ". " + barang.getNama() + " - Stok: " +
            barang.getStok());
            index++;
        }
    }
}

public static void tambahBarang() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("===== Penambahan Stok Barang =====");
    System.out.print("Masukkan nama barang: ");
    String namaBarang = scanner.nextLine();
    System.out.print("Masukkan Jumlah Stok Barang: ");
    try {
        int jumlahStok = scanner.nextInt();
        scanner.nextLine();

        if (jumlahStok <= 0) {
            System.out.println("Jumlah stok harus lebih dari 0.");
            return;
        }

        daftarBarang.add(new Barang(namaBarang, jumlahStok));
        System.out.println("Barang '" + namaBarang + "' berhasil ditambahkan dengan
        stok " + jumlahStok);
    } catch (InputMismatchException e) {
        System.out.println("Input tidak valid. Silakan masukkan angka untuk jumlah
        stok.");
        scanner.nextLine();
    }
}

```

```

public static void kurangiStok() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("==== Pengurangan Stok Barang ====");
    if (daftarBarang.isEmpty()) {
        System.out.println("Tidak ada barang yang tersedia.");
        return;
    }

    tampilkanDaftarBarang();

    System.out.print("Masukkan nomor indeks barang yang akan dikurangi stoknya: ");
    try {
        int indeksBarang = scanner.nextInt();
        scanner.nextLine();

        if (indeksBarang < 1 || indeksBarang > daftarBarang.size()) {
            System.out.println("Indeks barang tidak valid.");
            return;
        }

        Barang barang = daftarBarang.get(indeksBarang - 1);
        System.out.print("Masukkan jumlah stok yang akan dikurangi: ");
        int jumlahKurang = scanner.nextInt();
        scanner.nextLine();

        if (jumlahKurang <= 0) {
            System.out.println("Jumlah stok yang akan dikurangi harus lebih dari 0.");
            return;
        }

        int stokSaatIni = barang.getStok();
        if (jumlahKurang > stokSaatIni) {
            throw new StokBarangTidakCukupException("Stok " + barang.getNama() + " tidak cukup. Stok saat ini: " + stokSaatIni);
        } else {
            barang.setStok(stokSaatIni - jumlahKurang);
            System.out.println("Stok barang '" + barang.getNama() + "' berhasil dikurangi. Sisa stok: " + barang.getStok());
        }
    } catch (InputMismatchException e) {
        System.out.println("Input tidak valid. Silakan masukkan angka.");
        scanner.nextLine();
    } catch (StokBarangTidakCukupException e) {
        System.out.println(e.getMessage());
    } catch (IndexOutOfBoundsException e) {
        System.out.println("Indeks barang tidak valid: " + e.getMessage());
    }
}
}

```

Class ManajemenStok merupakan class utama yang berfungsi sebagai entry point program dengan method main(). Pada bagian awal, class ini mendeklarasikan ArrayList dengan nama daftarBarang sebagai struktur data yang digunakan untuk menyimpan objek-objek Barang.

Method tampilkanDaftarBarang() yang berfungsi untuk menampilkan semua objek barang dalam ArrayList. Method ini juga menggunakan Iterator yang berguna untuk menelusuri atau mengakses elemen elemen dari suatu struktur data secara berurutan tanpa perlu mengetahui detail struktur internalnya. Iterator menyediakan metode hasNext() untuk memeriksa apakah masih ada elemen berikutnya, dan next() untuk mengambil elemen tersebut.

Fungsi try-catch didalam kode tersebut untuk menangani error atau exception yang mungkin akan terjadi saat program dijalankan, sehingga program tidak langsung berhenti secara tiba-tiba. Try digunakan untuk menuliskan kode yang berpotensi menimbulkan exception, sedangkan catch digunakan untuk menangani exception apabila error itu terjadi. Dengan try-catch programmer bisa menampilkan pesan error yang informatif.

4. OUTPUT PROGRAM

```
==== Menu Manajemen Stok ====
1. Tambah Barang Baru
2. Tampilkan Semua Barang
3. Kurangi Stok Barang
4. Keluar
Pilih opsi: 1
===== Penambahan Stok Barang =====
Masukkan nama barang: Excalibur
Masukkan Jumlah Stok Barang: 1
Barang 'Excalibur' berhasil ditambahkan dengan stok 1

==== Menu Manajemen Stok ====
1. Tambah Barang Baru
2. Tampilkan Semua Barang
3. Kurangi Stok Barang
4. Keluar
Pilih opsi: 3
===== Pengurangan Stok Barang =====
===== Daftar Barang =====
1. Pensil - Stok: 50
2. Buku - Stok: 30
3. Penghapus - Stok: 20
4. Excalibur - Stok: 1
Masukkan nomor indeks barang yang akan dikurangi stoknya: 2
Masukkan jumlah stok yang akan dikurangi: 10
Stok barang 'Buku' berhasil dikurangi. Sisa stok: 20

==== Menu Manajemen Stok ====
1. Tambah Barang Baru
2. Tampilkan Semua Barang
3. Kurangi Stok Barang
4. Keluar
Pilih opsi: 2
===== Daftar Barang =====
1. Pensil - Stok: 50
2. Buku - Stok: 20
3. Penghapus - Stok: 20
4. Excalibur - Stok: 1

==== Menu Manajemen Stok ====
1. Tambah Barang Baru
2. Tampilkan Semua Barang
3. Kurangi Stok Barang
4. Keluar
Pilih opsi: 4
Terima kasih! Program berakhir.
```

LINK REPOSITORY:

https://github.com/Jaeeyy/OOP_A_Anantha_337.git