**Laporan Pengerjaan Praktikum Pertemuan 3 Teknik Pemrograman Praktik**

****

**Disusun oleh :**

**Astria Rizka Latifahsary (231524037)**

**Kelas :**

**D4 – 1B Teknik Informatika**

**Tahun Ajaran 2023 – 2024**

**Kasus 1**

**Sebelum :**

|  |
| --- |
| public class Restaurant {  public String[] nama\_makanan;  public double[] harga\_makanan;  public int[] stok;  public static byte id=0;g  public Restaurant() {  nama\_makanan = new String[10];  harga\_makanan = new double[10];  stok = new int[10];  }  public void tambahMenuMakanan(String nama, double harga, int stok){  this.nama\_makanan[id] = nama;  this.harga\_makanan[id] = harga;  this.stok[id] = stok;  }  public void tampilMenuMakanan() {  for(int i=0; i<=id; i++) {  if(!isOutOfStock(i)) {  System.out.println(nama\_makanan[i] +"["+stok[i]+"]"+"\tRp. "+harga\_makanan[i]);  }  }  }  public boolean isOutOfStock(int id) {  if(stok[id] == 0) {  return true;  }else {  return false;  }  }  public static void nextId() {  id++;  }  } |

|  |
| --- |
| public class RestaurantMain {  public static void main(String[] args) {  Restaurant menu = new Restaurant();  menu.tambahMenuMakanan("Bala-Bala", 1\_000,20);  Restaurant.nextId();  menu.tambahMenuMakanan("Gehu", 1\_000,20);  Restaurant.nextId();  menu.tambahMenuMakanan("Tahu", 1\_000,0);  Restaurant.nextId();  menu.tambahMenuMakanan("Molen", 1\_000,20);  menu.tampilMenuMakanan();  }  } |

**Setelah :**

|  |
| --- |
| public class Restaurant {  private String[] nama\_makanan;  private double[] harga\_makanan;  private int[] stok;  private static byte id=0;  public Restaurant() {  nama\_makanan = new String[10];  harga\_makanan = new double[10];  stok = new int[10];  menuMakanan();  }  private void menuMakanan(){  tambahMenuMakanan ("Bala-Bala", 1\_000, 20);  tambahMenuMakanan("Gehu ", 1\_000, 20);  tambahMenuMakanan("Tahu ", 1\_000, 0);  tambahMenuMakanan("Molen ", 1\_000, 20);  }  public void tambahMenuMakanan (String nama, double harga, int stok) {  this.nama\_makanan[id] = nama;  this.harga\_makanan[id] = harga;  this.stok[id] = stok;  nextId();  }  public void tampilMenuMakanan() {  for (int i=0; i<id; i++){  if(!isOutOfStock(i)) {  System.out.println(i + ". " + nama\_makanan[i] + "["+stok[i]+"]"+"\tRp. "+ harga\_makanan[i]);  }  }  }  public boolean isOutOfStock(int id) {  return stok[id] == 0;  }  public void pesanMenu(int id, int jumlah) {  if (id >= 0 && id < this.id && stok[id] >= jumlah) {  stok[id] -= jumlah;  System.out.println("Pesanan Anda: " + nama\_makanan[id] + " sejumlah " + jumlah);  } else {  System.out.println("Menu tidak tersedia atau stok tidak mencukupi.");  }  }    public static void nextId() {  id++;  }  } |

|  |
| --- |
| import java.util.Scanner;  public class RestaurantMain {  public static void main(String[] a2gs){  Scanner scanner = new Scanner(System.in);  Restaurant menu = new Restaurant();  boolean lanjut = true;  while (lanjut) {  System.out.println("\nSelamat datang di Restoran!");  System.out.println("1. Tampilkan Menu Makanan");  System.out.println("2. Pesan Menu");  System.out.println("3. Keluar");  System.out.print("Pilih menu: ");  int pilihan = scanner.nextInt();  switch (pilihan) {  case 1:  menu.tampilMenuMakanan();  break;  case 2:  System.out.print("Masukkan ID menu yang ingin dipesan: ");  int idPesan = scanner.nextInt();  System.out.print("Masukkan jumlah yang ingin dipesan: ");  int jumlahPesan = scanner.nextInt();  menu.pesanMenu(idPesan, jumlahPesan);  menu.tampilMenuMakanan();  break;  case 3:  lanjut = false;  break;  default:  System.out.println("Pilihan tidak valid.");  }  }  scanner.close();  System.out.println("Terima kasih telah menggunakan layanan kami. Sampai jumpa!");  }  } |

1. Saya menerapkan encapsulation dengan mengubah variable access modifier anggota dari ‘public’ menjadi ‘private’. Mengembangkan desain code menyesuaikan dengan desain OOP yang seharusnya. Selanjutnya saya atidak menggunakan getter setter, tetapi saya menggunakan custom method. Lalu agar membuat desain yang efesien mengubah pemanggilan method tambahMenuMakanan ke dalam contructor class Restaurant.
2. Dengan desain yang saya kembangkan dapat menambah fitur pemesanan dan mengurangi stok dengan menggunakan objek Scanner untuk mendapatkan input dari pengguna. Menambahkan metode ‘pesanMenu’ untuk memungkinkan pemesanan dan mengurangi stok dengan menggunakan struktur ‘switch case’ untuk memberikan pilihan kepada pegguna dan akan memberikan tindakan sesuai pilihan pengguna.

**Kasus 2**

**Source Code :**

|  |
| --- |
| public class Produk {  private String namaProduk;  private double harga;  private int stok;  public Produk(String namaProduk, double harga, int stok) {  this.namaProduk = namaProduk;  this.harga = harga;  this.stok = stok;  }  public String getNamaProduk(){  return namaProduk;  }  public double getHarga(){  return harga;  }  public int getStok(){  return stok;  }  public void kurangiStok(int jumlah) {  if (jumlah > stok) {  System.out.println("Stok tidak mencukupi!");  } else {  stok -= jumlah;  }  }  // Validasi input jumlah produk  public boolean validasiJumlah(int jumlah) {  return jumlah > 0 && jumlah <= stok;  }  } |

|  |
| --- |
| class Penjualan {  private Produk barang;  private int quantity;  public Penjualan(Produk barang, int quantity) {  this.barang = barang;  this.quantity = quantity;  }  public double hitungTotal(){  return barang.getHarga() \* quantity;  }  // public String getNamaProduk() {  // return barang.getNamaProduk();  // }  public int getQuantity(){  return quantity;  }  public String toString() {  return barang.getNamaProduk() + "\t" + quantity + "\tRp. " + hitungTotal();  }  } |

|  |
| --- |
| import java.util.ArrayList;  import java.util.Scanner;  public class PemesananMakananMain {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Inisialisasi menu makanan  ArrayList<Produk> menu = new ArrayList<>();  menu.add(new Produk("Seblak", 15000, 10));  menu.add(new Produk("Basreng", 12000, 5));  menu.add(new Produk("Cireng", 20000, 3));  menu.add(new Produk("Cilok", 18000, 8));  menu.add(new Produk("Telur Gulung", 25000, 2));  menu.add(new Produk("Nugget", 10000, 6));  menu.add(new Produk("Sotong", 15000, 4));  menu.add(new Produk("Tahu Bulat", 17000, 7));  menu.add(new Produk("Batagor", 20000, 3));  menu.add(new Produk("Spatula", 18000, 5));  ArrayList<Penjualan> pesanan = new ArrayList<>();  boolean lanjutPesan = true;  while (lanjutPesan) {  System.out.println("Menu Makanan:");  for (int i = 0; i < menu.size(); i++) {  System.out.println((i + 1) + ". " + menu.get(i).getNamaProduk() + " - Rp " + menu.get(i).getHarga() + " - Stok :" + menu.get(i).getStok());  }  System.out.print("Pilih nomor menu yang ingin dipesan: ");  int pilihanMenu = -1;  while (!scanner.hasNextInt()) {  System.out.println("Input tidak valid! Masukkan nomor menu yang ingin dipesan: ");  scanner.next(); // Mengabaikan input yang tidak valid  }  pilihanMenu = scanner.nextInt();  if (pilihanMenu < 1 || pilihanMenu > menu.size()) {  System.out.println("Nomor menu tidak valid!");  continue;  }  System.out.print("Masukkan jumlah yang ingin dipesan: ");  int jumlahPesanan = -1;  while (!scanner.hasNextInt()) {  System.out.println("Input tidak valid! Masukkan jumlah yang ingin dipesan: ");  scanner.next(); // Mengabaikan input yang tidak valid  }  jumlahPesanan = scanner.nextInt();  Produk produkDipesan = menu.get(pilihanMenu - 1);  if (!produkDipesan.validasiJumlah(jumlahPesanan)) {  System.out.println("Jumlah pesanan tidak valid!");  continue;  }  produkDipesan.kurangiStok(jumlahPesanan);  Penjualan penjualan = new Penjualan(produkDipesan, jumlahPesanan);  pesanan.add(penjualan);  System.out.print("Ingin memesan lagi? (ya/tidak): ");  String lanjut = scanner.next();  if (!lanjut.equalsIgnoreCase("ya")) {  lanjutPesan = false;  }  }  System.out.println("\nPesanan Anda:");  double totalBayarSemua = 0;  for (Penjualan item : pesanan) {  double total = item.hitungTotal();  System.out.println(item.toString() + " - " + item.getQuantity() + " pcs - Rp " + total);  totalBayarSemua += total;  }  System.out.println("Total Bayar: Rp " + totalBayarSemua);  scanner.close();  }  } |

**Output :**

|  |
| --- |
|  |

**Repository**

|  |
| --- |
| <https://github.com/AstriaRLatifahsary/TeknikPemogramanAstria> |