

Laporan Pertemuan 3 Teknik Pemrograman
Praktik



Disusun oleh :

Nesta Rizkia Saputra (231524060)

Kelas :

D4 – 1B Teknik Informatika

Tahun Ajaran 2023 – 2024

Kasus 1

```
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];
    }

    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++;
    }
}
```

Diatas merupakan class Restaurant dari program awal sebelum diubah

```
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, 0 );
        Restaurant.nextId();
        menu.tambahMenuMakanan("Molen",1_000, 20 );

        menu.tampilMenuMakanan();
    }
}
```

Diatas merupakan class RestaurantMain dari program awal sebelum diubah

Perubahan Program

```
class MenuItem {
    private String nama;
    private double harga;
    private int stok;

    public MenuItem(String nama, double harga, int stok) {
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    public String getNama() {
        return nama;
    }

    public double getHarga() {
        return harga;
    }

    public int getStok() {
        return stok;
    }

    public boolean isOutOfStock() {
        return stok == 0;
    }
}
```

1. Yang pertama adalah saya menambah class yaitu class MenuItem dimana dalam kelas ini berguna untuk menampung semua item dalam Restaurant. Dengan class ini mengurangi deklarasi basic type pada class Restaurant juga untuk memecah responsibilities dari class Restaurant.

```
public class Restaurant
{
    private MenuItem[] daftarMenu;
    private static byte id = 0;

    public Restaurant()
    {
        daftarMenu = new MenuItem[10];
    }

    public void tambahMenuMakanan(String nama, double harga, int stok)
    {
        MenuItem item = new MenuItem(nama, harga, stok);
    }
}
```

```

        daftarMenu[id] = item;
        id++;
    }

    public void tampilMenuMakanan()
    {
        for (int i = 0; i < id; i++)
        {
            if (!daftarMenu[i].isOutOfStock())
            {
                System.out.println(daftarMenu[i].getNama() + "[" +
daftarMenu[i].getStok() + "]" + "\tRp. " + daftarMenu[i].getHarga());
            }
        }
    }
}

```

2. variabel object 'nextid' dihapus karena kita sudah memiliki class sendiri untuk 'MenuItem' dan menggunakan attributes 'id'

```

public class RestaurantMain
{
    public static void main(String[] args)
    {
        Restaurant menu = new Restaurant();
        menu.tambahMenuMakanan("bala-bala", 1_000, 20);
        menu.tambahMenuMakanan("Gehu", 1_000, 0);
        menu.tambahMenuMakanan("Molen", 1_000, 20);

        menu.tampilMenuMakanan();
    }
}

```

3. Terakhir dalam class main hanya merubah method yang sebelumnya terdapat method menggunakan attributes nextid sekarang sudah tidak ada.

Upgrade Program

```
class MenuItem {
    private String nama;
    private double harga;
    private int stok;

    public MenuItem(String nama, double harga, int stok) {
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    public String getName() {
        return nama;
    }

    public double getHarga() {
        return harga;
    }

    public int getStok() {
        return stok;
    }

    public void kurangiStok(int jumlah) {
        stok -= jumlah;
    }

    public boolean isOutOfStock() {
        return stok == 0;
    }
}
```

1. Dalam class MenuItem saya menambahkan Method untuk mengurangi stok setiap kali ada pembelian

```
public class Restaurant {
    private MenuItem[] daftarMenu;
    private int id;

    public Restaurant() {
        daftarMenu = new MenuItem[10];
        id = 0;
    }

    public void tambahMenuMakanan(String nama, double harga, int stok) {
        if (id < daftarMenu.length) {
            MenuItem item = new MenuItem(nama, harga, stok);
        }
    }
}
```

```

        daftarMenu[id] = item;
        id++;
    } else {
        System.out.println("Tidak bisa menambahkan menu, sudah mencapai
batas maksimum.");
    }
}

public void PesanMenu(int idMenu, int jumlah)
{
    if (idMenu >= 0 && idMenu < id && daftarMenu[idMenu].getStok() >=
jumlah)
    {
        daftarMenu[idMenu].kurangiStok(jumlah);
        System.out.println("Pesanan Anda berhasil.");
    } else {
        System.out.println("Menu tidak tersedia atau stok tidak
mencukupi.");
    }
}

public void tampilMenuMakanan()
{
    for (int i = 0; i < id; i++)
    {
        if (!daftarMenu[i].isOutOfStock())
        {
            System.out.println(daftarMenu[i].getNama() + "[" +
daftarMenu[i].getStok() + "]" + "\tRp. " + daftarMenu[i].getHarga());
        }
    }
}
}

```

2. Pada class Resturant saya menambahkan beberapa method untuk menambahkan sebuah menu.
3. Saya juga menambahkan method untuk mengatur sebuah pemesanan baik jika pemesanan berhasil maupun pemesanan gagal.

```

public class RestaurantMain
{
    public static void main(String[] args)
    {
        Restaurant menu = new Restaurant();
        menu.tambahMenuMakanan("bala-bala", 1_000, 20);
        menu.tambahMenuMakanan("Gehu", 1_000, 100);
        menu.tambahMenuMakanan("Molen", 1_000, 20);

        menu.tampilMenuMakanan();
    }
}

```

```

// Contoh pemesanan
menu.PesanMenu (3,10);
menu.tampilMenuMakanan(); // Memperlihatkan stok yang berkurang setelah
pemesanan
}
}

```

4. Dalam class main saya tidak melakukan perubahan, namun hanya menambahkan sebuah method untuk pemesanan

Implementasi

```

// Contoh pemesanan
menu.PesanMenu (3,10);

```

Output seperti diatas karena id menu yang dipesan adalah 3 dimana id menu yang tersedia hanya 0,1,2 karena menu hanya ada tiga. Oleh karena itu output yang ditampilkan seperti diatas.

Kasus 2

Class Produk

```
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) {
        stok -= jumlah;
    }
}
```

Class Penjualan

```
class Penjualan {
    private String namaProduk;
    private int quantity;
    private double hargaTotal;

    public Penjualan(String namaProduk, int quantity, double hargaTotal) {
        this.namaProduk = namaProduk;
        this.quantity = quantity;
        this.hargaTotal = hargaTotal;
    }

    public String getNamaProduk() {
        return namaProduk;
    }

    public int getQuantity() {
```

```

        return quantity;
    }

    public double getHargaTotal() {
        return hargaTotal;
    }
}

```

Class Pemesanan (Main)

```

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

public class Pemesanan {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Produk> daftarMenu = initializeMenu();

        // Tampilkan menu
        System.out.println("Menu Makanan:");
        for (Produk produk : daftarMenu) {
            System.out.println(produk.getNamaProduk() + " - Rp. " +
produk.getHarga());
        }

        // Pesan menu
        ArrayList<Penjualan> pesanan = new ArrayList<>();
        String pesanLagi;

        do {
            System.out.print("\nMasukkan nama produk yang ingin dipesan: ");
            String namaProduk = scanner.nextLine();
            System.out.print("Masukkan jumlah yang ingin dipesan: ");
            int quantity = scanner.nextInt();
            scanner.nextLine(); // Membuang karakter newline

            Produk produkDipesan = cariProduk(daftarMenu, namaProduk);
            if (produkDipesan != null && produkDipesan.getStok() >= quantity)
            {
                double hargaTotal = produkDipesan.getHarga() * quantity;
                pesanan.add(new Penjualan(namaProduk, quantity, hargaTotal));
                produkDipesan.kurangiStok(quantity);
                System.out.println("Produk berhasil dipesan.");
            } else {
                System.out.println("Maaf, produk tidak tersedia atau stok
tidak mencukupi.");
            }
        }
    }
}

```

```

        System.out.print("Apakah Anda ingin memesan lagi? (ya/tidak): ");
        pesanLagi = scanner.nextLine();
    } while (pesanLagi.equalsIgnoreCase("ya"));

    // Tampilkan pesanan dan total bayar
    System.out.println("\nPesanan Anda:");
    double totalBayar = 0;
    for (Penjualan pesananItem : pesanan) {
        System.out.println(pesananItem.getNamaProduk() + " - " +
            pesananItem.getQuantity() + " pcs - Rp. " + pesananItem.getHargaTotal());
        totalBayar += pesananItem.getHargaTotal();
    }
    System.out.println("Total Bayar: Rp. " + totalBayar);

    scanner.close();
}

private static ArrayList<Produk> initializeMenu() {
    ArrayList<Produk> daftarMenu = new ArrayList<>();
    daftarMenu.add(new Produk("Nasi Goreng", 15000, 20));
    daftarMenu.add(new Produk("Mie Goreng", 6000, 15));
    daftarMenu.add(new Produk("Ayam Goreng", 8000, 10));
    daftarMenu.add(new Produk("Sate Ayam", 15000, 8));
    daftarMenu.add(new Produk("Nasi Kuning", 8000, 12));
    daftarMenu.add(new Produk("Bakso", 10000, 20));
    daftarMenu.add(new Produk("Soto Ayam", 12000, 15));
    daftarMenu.add(new Produk("Ikan Bakar", 25000, 10));
    daftarMenu.add(new Produk("Capcay", 12000, 18));
    daftarMenu.add(new Produk("Gado-gado", 10000, 10));
    daftarMenu.add(new Produk("Tahu Goreng", 2000, 25));
    daftarMenu.add(new Produk("Sambal Goreng", 2000, 20));
    daftarMenu.add(new Produk("Pecel Lele", 14000, 15));
    daftarMenu.add(new Produk("Rendang", 15000, 10));
    daftarMenu.add(new Produk("Sop Ayam", 10000, 12));

    return daftarMenu;
}

private static Produk cariProduk(ArrayList<Produk> daftarMenu, String
namaProduk) {
    for (Produk produk : daftarMenu) {
        if (produk.getNamaProduk().equalsIgnoreCase(namaProduk)) {
            return produk;
        }
    }
    return null;
}
}

```

Implementasi

The screenshot shows the VS Code interface with the Explorer on the left, the Terminal in the center, and the Run and Debug sidebar on the right. The Explorer shows a project structure for 'PERTEMUAN 03' with files like 'Pemesanan.java', 'PemesananMakanan.java', 'Penjualan.java', and 'Produk.java'. The Terminal displays the output of a Java program. The Run and Debug sidebar shows the 'Run: Pemesanan' command.

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Bones\Kuliah\Semester 2\Tugas Kuliah\Teknik Pemograman\PR\Pertemuan 03> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\user\AppData\Roaming\Code\User\workspaceStorage\b0283484939ec5a6bd4e87e3bfcac138\redhat.java\jdt_ws\Pertemuan_03_fe52e57f\bin' 'Pemesanan'
Menu Makanan:
Nasi Goreng - Rp. 15000.0
Mie Goreng - Rp. 6000.0
Ayam Goreng - Rp. 8000.0
Sate Ayam - Rp. 15000.0
Nasi Kuning - Rp. 8000.0
Bakso - Rp. 10000.0
Soto Ayam - Rp. 12000.0
Ikan Bakar - Rp. 25000.0
Capcay - Rp. 12000.0
Gado-gado - Rp. 10000.0
Tahu Goreng - Rp. 2000.0
Sambal Goreng - Rp. 2000.0
Pecel Lele - Rp. 14000.0
Rendang - Rp. 15000.0
Sop Ayam - Rp. 10000.0

Masukkan nama produk yang ingin dipesan: nasi goreng
Masukkan jumlah yang ingin dipesan: 2
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: mie goreng
Masukkan jumlah yang ingin dipesan: 1
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: sate ayam
Masukkan jumlah yang ingin dipesan: 3
```

The screenshot shows the VS Code interface with the Explorer on the left, the Terminal in the center, and the Run and Debug sidebar on the right. The Explorer shows the same project structure as the previous screenshot. The Terminal displays the continuation of the Java program's output. The Run and Debug sidebar shows the 'Run: Pemesanan' command.

```
Masukkan jumlah yang ingin dipesan: 3
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: tahu goreng
Masukkan jumlah yang ingin dipesan: 4
Maaf, produk tidak tersedia atau stok tidak mencukupi.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: rendang
Masukkan jumlah yang ingin dipesan: 1
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: tahu goreng 2
Masukkan jumlah yang ingin dipesan: 2
Maaf, produk tidak tersedia atau stok tidak mencukupi.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: tahu goreng
Masukkan jumlah yang ingin dipesan: 2
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): ya

Masukkan nama produk yang ingin dipesan: sop ayam
Masukkan jumlah yang ingin dipesan: 3
Produk berhasil dipesan.
Apakah Anda ingin memesan lagi? (ya/tidak): tidak

Pesanan Anda:
nasi goreng - 2 pcs - Rp. 30000.0
mie goreng - 1 pcs - Rp. 6000.0
sate ayam - 3 pcs - Rp. 45000.0
rendang - 1 pcs - Rp. 15000.0
tahu goreng - 2 pcs - Rp. 4000.0
sop ayam - 3 pcs - Rp. 30000.0
Total Bayar: Rp. 130000.0
```