

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 2**



**NAMA:IMEL THERESIA BR  
SEMBIRING  
NIM. 2341760046  
SIB-1F / 12  
D-IV SISTEM INFORMASI BISNIS  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

## PRAKTIKUM 3

### 3.2 PERCOBAAN

#### 3.2.1 LANGKAH LANGKAH PERCOBAAN

1. Membuat class PersegiPanjang

```
Codeium: Refactor | Codeium: Explain
public class PersegiPanjang08 {
    public int panjang;
    public int lebar;
}
```

2. Menambahkan class PersegiPanjangDemo

```
PersegiPanjangDemo08.java > ...
Codeium: Refactor | Codeium: Explain
1 public class PersegiPanjangDemo08 {
   Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2     public static void main(String[] args) {
3         PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[3];
4     }
5 }
6
```

3. Melengkapi atribut

```
Codeium: Refactor | Codeium: Explain
public class PersegiPanjangDemo08 {
   Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
   public static void main(String[] args) {
       PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[3];

       arrayOfPersegiPanjang08[0] = new PersegiPanjang08();
       arrayOfPersegiPanjang08[0].panjang = 110;
       arrayOfPersegiPanjang08[0].lebar = 30;

       arrayOfPersegiPanjang08[1] = new PersegiPanjang08();
       arrayOfPersegiPanjang08[1].panjang = 80;
       arrayOfPersegiPanjang08[1].lebar = 30;

       arrayOfPersegiPanjang08[2] = new PersegiPanjang08();
       arrayOfPersegiPanjang08[2].panjang = 100;
       arrayOfPersegiPanjang08[2].lebar = 20;
   }
}
```

4. Menampilkan isi array

```
System.out.println("Persegi panjang ke-1, panjang: " + arrayOfPersegiPanjang08[0].
panjang + ", lebar: " + arrayOfPersegiPanjang08[0].lebar);

System.out.println("Persegi panjang ke-2, panjang: " + arrayOfPersegiPanjang08[1].
panjang + ", lebar: " + arrayOfPersegiPanjang08[1].lebar);

System.out.println("Persegi panjang ke-3, panjang: " + arrayOfPersegiPanjang08[2].
panjang + ", lebar: " + arrayOfPersegiPanjang08[2].lebar);
}
```

#### VERIFIKASI HASIL PERCOBAAN 3.2.2

```
ng\Code\User\workspaceStorage\1fa5903f2a48ca99c0e708af1d53db3f\
5e460\bin' 'PersegiPanjangDemo08'
Persegi panjang ke-1, panjang: 110, lebar: 30
Persegi panjang ke-2, panjang: 80, lebar: 30
Persegi panjang ke-3, panjang: 100, lebar: 20
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3>
```

### PERTANYAAN 3.2.3

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method? Jelaskan!
2. Apa yang dilakukan oleh kode program berikut?

```
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
```

3. Apakah class PersegiPanjang memiliki konstruktor? Jika tidak, kenapa bisa dilakukan pemanggilan konstruktor pada baris program berikut?

```
arrayOfPersegiPanjang[0] = new PersegiPanjang();
```

4. Apa yang dilakukan oleh kode program berikut?

```
arrayOfPersegiPanjang[0].panjang = 110;  
arrayOfPersegiPanjang[0].lebar = 30;
```

5. Mengapa class PersegiPanjang dan PersegiPanjangDemo dipisahkan pada uji coba 3.2?

Jawaban

1. Tidak, jika hanya untuk menampung data menampilkan data, maka tidak perlu menggunakan method, tetapi jika ada proses yang diperlukan seperti olah data, maka menggunakan method
2. Instansiasi array dengan length 3 pada class PersegiPanjang
3. Ya, memiliki konstruktor default yang secara otomatis dibuat oleh compiler, tidak terdefinisi secara khusus di class PersegiPanjang. Semua value nya akan menjadi null secara default
4. Mengakses index ke 0 dari arrayOfPersegiPanjang, kemudian mengatur nilai atribut Panjang dan lebar dengan nilai 110 dan 30
5. Karena memiliki fungsi tersendiri, class PersegiPanjang berfungsi untuk proses manipulasi data (jika ada), kemudian class PersegiPanjangDemo berfungsi sebagai interface dari class PersegiPanjang

### LANGKAH LANGKAH PERCOBAAN 3.3.1

1. Import Scanner

```
import java.util.Scanner;
```

2. Modifikasi kode untuk menerima inputan dengan looping

```
Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[3];  
  
    for (int i = 0; i < 3; i++) {  
        arrayOfPersegiPanjang08[i] = new PersegiPanjang08();  
  
        System.out.println("Persegi panjang ke-" + (i + 1));  
        System.out.print(s:"Masukkan panjang: ");  
        arrayOfPersegiPanjang08[i].panjang = sc.nextInt();  
        System.out.print(s:"Masukkan lebar: ");  
        arrayOfPersegiPanjang08[i].lebar = sc.nextInt();  
    }  
}
```

### 3. Modifikasi menampilkan array

```
for (int i = 0; i < 3; i++) {  
    System.out.println("Persegi panjang ke-" + (i + 1) + ", panjang: "  
        + arrayOfPersegiPanjang08[i].panjang + ", lebar: " + arrayOfPersegiPanjang08[i].  
        lebar);  
}
```

#### VERIFIKASI HASIL PERCOBAAN 3.3.2

```
age\1fa5903f2a48ca99c0e708af1d53db3f\redhat.java\jdt_ws\pertemuan3_53a  
Demo08'  
Persegi panjang ke-1  
Masukkan panjang: 20  
Masukkan lebar: 10  
Persegi panjang ke-2  
Masukkan panjang: 40  
Masukkan lebar: 20  
Persegi panjang ke-3  
Masukkan panjang: 50  
Masukkan lebar: 60  
Persegi panjang ke-1, panjang: 20, lebar: 10  
Persegi panjang ke-2, panjang: 40, lebar: 20  
Persegi panjang ke-3, panjang: 50, lebar: 60  
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3> []
```

#### PERTANYAAN 3.3.3

1. Tambahkan method cetakInfo() pada class PersegiPanjang kemudian modifikasi kode program pada langkah no 3
2. Misalkan Anda punya array baru bertipe array of PersegiPanjang dengan nama myArrayOfPersegiPanjang. Mengapa kode berikut menyebabkan error?

Jawaban

1. Dengan membuat method cetakInfo yang mampu menerima array pada class persegiPanjangDemo

```
PersegiPanjangDemo08.java > ...  
import java.util.Scanner;  
  
Codeium: Refactor | Codeium: Explain  
public class PersegiPanjangDemo08 {  
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        PersegiPanjang08 persegiPanjang08 = new PersegiPanjang08();  
        PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[3];  
  
        for (int i = 0; i < 3; i++) {  
            arrayOfPersegiPanjang08[i] = new PersegiPanjang08();  
  
            System.out.println("Persegi panjang ke-" + (i + 1));  
            System.out.print(s:"Masukkan panjang: ");  
            arrayOfPersegiPanjang08[i].panjang = sc.nextInt();  
            System.out.print(s:"Masukkan lebar: ");  
            arrayOfPersegiPanjang08[i].lebar = sc.nextInt();  
        }  
  
        persegiPanjang08.cetakInfo(arrayOfPersegiPanjang08);  
    }  
}  
  
Codeium: Refactor | Explain | Generate Javadoc | X  
public void cetakInfo(PersegiPanjang08[] arrayOfPersegiPanjang08)  
{  
    for (int i = 0; i < 3; i++) {  
        System.out.println("Persegi panjang ke-" + (i + 1) + ", panjang: "  
            + arrayOfPersegiPanjang08[i].panjang + ", lebar: " + arrayOfPersegiPanjang08[i].  
            lebar);  
    }  
}
```

2. Karena terdapat duplikasi array dengan nama yang sama



### LANGKAH LANGKAH PERCOBAAN 3.4.1

1. Menambahkan konstruktor

```
public PersegiPanjang08(int p, int l) {  
    panjang = p;  
    lebar = l;  
}
```

2. Amati run hasil ubahan di atas  
Kode akan error karena telah didefinisikan konstruktor, maka pada class persegiPanjangDemo harus mengikuti konstruktor dari kelas persegiPanjang
3. Modifikasi input pada persegiPanjangDemo

```
for (int i = 0; i < 3; i++) {  
    System.out.println("Persegi panjang ke-" + (i + 1));  
    System.out.print(s:"Masukkan panjang: ");  
    panjang = sc.nextInt();  
    System.out.print(s:"Masukkan lebar: ");  
    lebar = sc.nextInt();  
  
    arrayOfPersegiPanjang08[i] = new PersegiPanjang08(panjang, lebar);  
}
```

### VERIFIKASI HASIL PERCOBAAN 3.4.2

```
Demo08'  
Persegi panjang ke-1  
Masukkan panjang: 2  
Masukkan lebar: 3  
Persegi panjang ke-2  
Masukkan panjang: 5  
Masukkan lebar: 6  
Persegi panjang ke-3  
Masukkan panjang: 8  
Masukkan lebar: 9  
Persegi panjang ke-1, panjang: 2, lebar: 3  
Persegi panjang ke-2, panjang: 5, lebar: 6  
Persegi panjang ke-3, panjang: 8, lebar: 9  
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertem
```

### PERTANYAAN 3.4.3

1. Mengapa terjadi compile error pada langkah no 2?
2. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya
3. Tambahkan method hitungLuas() dan hitungKeliling()
4. Lakukan modifikasi kode program untuk menampilkan luas dan keliling dengan contoh output

```
Persegi panjang ke-1  
Masukkan panjang: 2  
Masukkan lebar: 4  
Persegi panjang ke-2  
Masukkan panjang: 10  
Masukkan lebar: 20  
Persegi panjang ke-3  
Masukkan panjang: 5  
Masukkan lebar: 7  
Persegi panjang ke-1, panjang: 2, lebar: 4, luas: 8, keliling: 12  
Persegi panjang ke-2, panjang: 10, lebar: 20, luas: 200, keliling: 60  
Persegi panjang ke-3, panjang: 5, lebar: 7, luas: 35, keliling: 24
```

5. Modifikasi kode program pada praktikum 3.3 agar length array ditentukan dari user melalui input dengan Scanner

Jawaban

1. Karena belum menerapkan konstruktor yang telah di definisikan pada class PersegiPanjang
2. Bisa, seperti berikut. Konstruktor pertama dengan param p dan l, yang kedua tanpa param

```
public PersegiPanjang08(int p, int l) {  
    panjang = p;  
    lebar = l;  
}  
  
public PersegiPanjang08()  
{  
}
```

3. Membuat method hitungLuas dan hitungKeliling dengan param p dan l

```
Codeium: Refactor | Explain | Generate Javadoc | X  
public int hitungLuas(int p, int l)  
{  
    int luas = 0;  
    luas = p * l;  
    return luas;  
}  
  
Codeium: Refactor | Explain | Generate Javadoc | X  
public int hitungkeliling(int p, int l)  
{  
    int keliling = 0;  
    keliling = 2*(p+l);  
    return keliling;  
}
```

4. Melakukan ubahan pada saat menampilkan dengan memanggil method hitung luas dan hitungkeliling untuk mencari luas dan keliling

```
age\1fa5903f2a48ca99c0e708af1d53db3f\redhat.java\jdt_ws\pertemuan3_53a5e4  
Demo08'  
Persegi panjang ke-1  
Masukkan panjang: 10  
Masukkan lebar: 5  
Persegi panjang ke-2  
Masukkan panjang: 2  
Masukkan lebar: 5  
Persegi panjang ke-3  
Masukkan panjang: 9  
Masukkan lebar: 9  
Persegi panjang ke-1, panjang: 10, lebar: 5, luas: 50, keliling: 30  
Persegi panjang ke-2, panjang: 2, lebar: 5, luas: 10, keliling: 14  
Persegi panjang ke-3, panjang: 9, lebar: 9, luas: 81, keliling: 36  
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3>
```

5. Menambahkan input untuk length

```
Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    int length;  
    System.out.print(s:"Masukkan berapa banyak anda ingin menghitung : ");  
    length = sc.nextInt();  
  
    PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[length];  
    int panjang, lebar;
```

```
Demo09  
Masukkan berapa banyak anda ingin menghitung : 2  
Persegi panjang ke-1  
Masukkan panjang: 5  
Masukkan lebar: 10  
Persegi panjang ke-2  
Masukkan panjang: 5  
Masukkan lebar: 10  
Persegi panjang ke-1, panjang: 5, lebar: 10, luas: 50, keliling: 30  
Persegi panjang ke-2, panjang: 5, lebar: 10, luas: 50, keliling: 30  
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3>
```

### 3.5 TUGAS

1. Sebuah kampus membutuhkan program untuk menampilkan informasi mahasiswa berupa nama, NIM, jenis kelamin dan juga IPK mahasiswa. Program dapat menerima input semua informasi tersebut, kemudian menampilkannya kembali ke user. Tambahkan informasi rata-rata IPK pada bagian akhir. Gunakan looping dengan FOR untuk pembuatan object. Gunakan looping dengan FOREACH untuk pencetakan data ke layar.

#### Class Tugas108

```
J Tugas108.java > ...
Codeium: Explain
1 public class Tugas108 {
2     public double ipk, countIpk;
3     public String nama, nim, jenisKelamin;
4
5     public Tugas108(String nama, String nim, String jenisKelamin, double ipk) {
6         this.nama = nama;
7         this.nim = nim;
8         this.jenisKelamin = jenisKelamin;
9         this.ipk = ipk;
10    }
11
12    public Tugas108()
13    {
14    }
15
16    Codeium: Refactor | Explain | Generate Javadoc | X
17    public void cetakInfo(Tugas108[] mahasiswa)
18    {
19        int i = 1;
20        for (Tugas108 mhs : mahasiswa) {
21            countIpk += mhs.ipk;
22
23            System.out.println("Data Mahasiswa ke-" + i);
24            System.out.println("Nama: " + mhs.nama);
25            System.out.println("Nim: " + mhs.nim);
26            System.out.println("Jenis kelamin: " + mhs.jenisKelamin);
27            System.out.println("IPK: " + mhs.ipk);
28            System.out.println();
29            i++;
30        }
31        System.out.println("Rata-rata IPK : " + getAvgIpk(countIpk, mahasiswa.length));
32    }
33
34    Codeium: Refactor | Explain | Generate Javadoc | X
35    public double getAvgIpk(double ipk, int mhs)
36    {
37        double avg;
38        avg = ipk / Double.valueOf(mhs);
39        return avg;
40    }
41 }
```

## Class TugasDemo108

```
import java.util.Scanner;

Codeium: Refactor | Codeium: Explain
public class PersegiPanjangDemo08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int length;
        System.out.print(s:"Masukkan berapa banyak anda ingin menghitung : ");
        length = sc.nextInt();

        PersegiPanjang08[] arrayOfPersegiPanjang08 = new PersegiPanjang08[length];
        int panjang, lebar;

        for (int i = 0; i < arrayOfPersegiPanjang08.length; i++) {
            System.out.println("Persegi panjang ke-" + (i + 1));
            System.out.print(s:"Masukkan panjang: ");
            panjang = sc.nextInt();
            System.out.print(s:"Masukkan lebar: ");
            lebar = sc.nextInt();

            arrayOfPersegiPanjang08[i] = new PersegiPanjang08(panjang, lebar);
        }
        PersegiPanjang08 persegiPanjang08 = new PersegiPanjang08();
        persegiPanjang08.cetakInfo(arrayOfPersegiPanjang08);
    }
}
```

## Hasil Compile

```
Masukkan berapa mahasiswa yang ingin di data : 2
Masukkan data mahasiswa ke-1
Masukkan nama: Dzulfikar
Masukkan NIM: 2341760071
Masukkan jenis kelamin: L
Masukkan IPK: 3.6

Masukkan data mahasiswa ke-2
Masukkan nama: Joni
Masukkan NIM: 2341760072
Masukkan jenis kelamin: L
Masukkan IPK: 4

Data Mahasiswa ke-1
Nama: Dzulfikar
Nim: 2341760071
Jenis kelamin: L
IPK: 3.6

Data Mahasiswa ke-2
Nama: Joni
Nim: 2341760072
Jenis kelamin: L
IPK: 4.0

Rata-rata IPK : 3.8
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3> █
```



- Implementasikan tugas ASD Teori no 1 ke dalam kode program. Buatlah array of objects berdasarkan input dari user dan tampilkan data seperti pada Tugas no 1

Class transaksi digunakan untuk operasi berkaitan dengan transaksi

```
Codeium: Explain
public class Transaksi {
    public double totalBayar, jumlahUang, diskon;

    public Transaksi (double totalBayar, double jumlahUang, double diskon){
        this.totalBayar = totalBayar;
        this.jumlahUang = jumlahUang;
        this.diskon = diskon;
    }

    public Transaksi (){}

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void bayar(Transaksi[] data){
        for(int i =0; i<data.length; i++){
            if(data[i].jumlahUang>data[i].totalBayar){
                System.out.println("Maaf, uang anda kurang");
            }else{
                double total = (data[i].totalBayar + data[i].diskon) / 100;

                double diskon = data[i].totalBayar*data[i].diskon/100;
                System.out.println("Total Bayar \t : " + data[i].totalBayar);
                System.out.println("Jumlah Uang \t : " + data[i].jumlahUang);
                System.out.println("Diskon \t\t : " + diskon);
                System.out.println("Kembalian anda \t:" + kembalian(total, data[i].jumlahUang));
                System.out.println();
            }
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public double kembalian(double totalBayar, double jumlahUang){
        double back = 0;
        back = jumlahUang - totalBayar;
        return back;
    }
}
```

Class omset digunakan untuk operasi berkaitan dengan omset

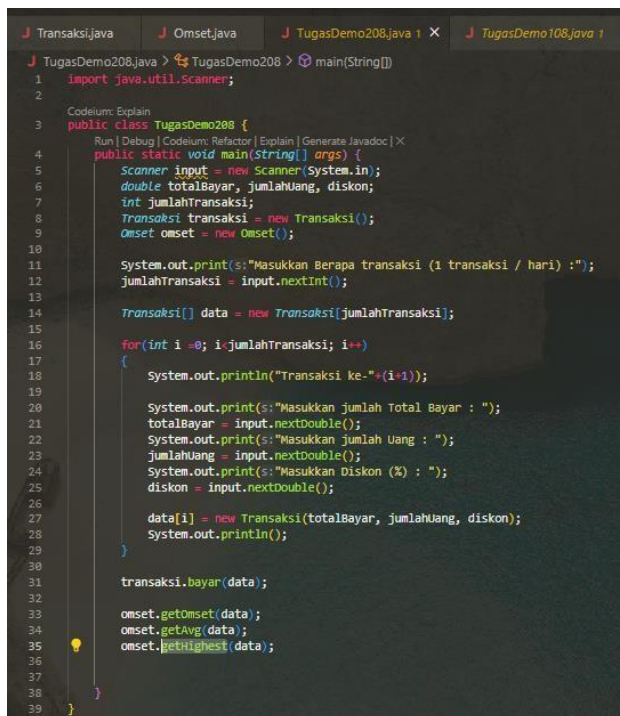
```
Omset.java > ...
public class Omset {
    public Omset (){}

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void getOmset(Transaksi[] data)
    {
        for(int i =0; i<data.length; i++){
            double total = data[i].totalBayar - ((data[i].totalBayar*data[i].diskon)/100);
            System.out.println("Omset hari ke-"+(i+1)+" : " + total);
        }
        System.out.println();
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void getAvg(Transaksi[] data)
    {
        double avg =0;
        for(int i =0; i<data.length; i++){
            avg += data[i].totalBayar - ((data[i].totalBayar*data[i].diskon)/100);
        }
        System.out.println("Rata rata omset : " + (avg/data.length));
        System.out.println();
    }

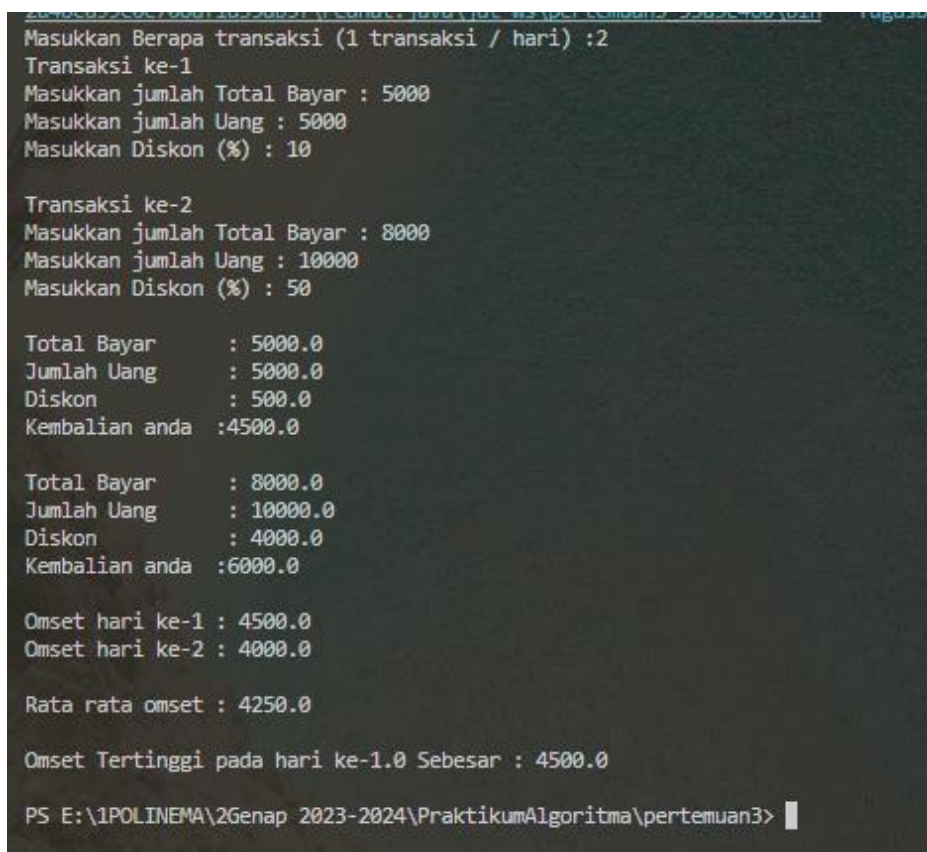
    Codeium: Refactor | Explain | Generate Javadoc | X
    public void getHighest(Transaksi[] data)
    {
        double highest = 0;
        double index = 0;
        for(int i =0; i<data.length; i++){
            double total = data[i].totalBayar - ((data[i].totalBayar*data[i].diskon)/100);
            if(total > highest){
                index = i;
                highest = total;
            }
        }
        System.out.println("Omset Tertinggi pada hari ke-"+(index+1)+" Sebesar : " + highest);
        System.out.println();
    }
}
```

Class tugasDemo208 berfungsi sebagai main yang pertama kali di jalankan



```
1 import java.util.Scanner;
2
3 public class TugasDemo208 {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         double totalBayar, jumlahUang, diskon;
7         int jumlahTransaksi;
8         Transaksi transaksi = new Transaksi();
9         Omset omset = new Omset();
10
11         System.out.print("Masukkan Berapa transaksi (1 transaksi / hari) :");
12         jumlahTransaksi = input.nextInt();
13
14         Transaksi[] data = new Transaksi[jumlahTransaksi];
15
16         for(int i =0; i<jumlahTransaksi; i++)
17         {
18             System.out.println("Transaksi ke-"+(i+1));
19
20             System.out.print("Masukkan jumlah Total Bayar : ");
21             totalBayar = input.nextDouble();
22             System.out.print("Masukkan jumlah Uang : ");
23             jumlahUang = input.nextDouble();
24             System.out.print("Masukkan Diskon (%) : ");
25             diskon = input.nextDouble();
26
27             data[i] = new Transaksi(totalBayar, jumlahUang, diskon);
28             System.out.println();
29         }
30
31         transaksi.bayar(data);
32
33         omset.getOmset(data);
34         omset.getAvg(data);
35         omset.getHighest(data);
36     }
37 }
38
39 }
```

Hasil compile program



```
Masukkan Berapa transaksi (1 transaksi / hari) :2
Transaksi ke-1
Masukkan jumlah Total Bayar : 5000
Masukkan jumlah Uang : 5000
Masukkan Diskon (%) : 10

Transaksi ke-2
Masukkan jumlah Total Bayar : 8000
Masukkan jumlah Uang : 10000
Masukkan Diskon (%) : 50

Total Bayar      : 5000.0
Jumlah Uang      : 5000.0
Diskon           : 500.0
Kembalian anda   :4500.0

Total Bayar      : 8000.0
Jumlah Uang      : 10000.0
Diskon           : 4000.0
Kembalian anda   :6000.0

Omset hari ke-1 : 4500.0
Omset hari ke-2 : 4000.0

Rata rata omset : 4250.0

Omset Tertinggi pada hari ke-1.0 Sebesar : 4500.0

PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan3>
```