

NAMA : AKHMAD RIDLO RIFA'I

NIM : 1227050013

Praktikum Pemograman Berorientasi Objek – Pak Deden

Pasien.java

```
package praktikum.uts;
import java.util.Scanner;
public class pasien{
    private String nama;
    private String gejala;
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public String getGejala() {
        return gejala;
    }
    public void setGejala(String gejala) {
        this.gejala = gejala;
    }
    public void bacaInput(){
        Scanner input = new Scanner(System.in);
        System.out.println("Masukkan Nama Anda");
        nama = input.nextLine();
        System.out.println("Apa keluhan Anda");
        gejala = input.nextLine();
    }
}

// Interface untuk dokter
interface Dokter {
    void periksaPasien(pasien pasien);
    void berikanResep(pasien pasien);
}

// Class untuk dokter umum
class DokterUmum implements Dokter {

    @Override
    public void periksaPasien(pasien pasien) {
```

```

        System.out.println("Dokter umum memeriksa pasien " + pasien.getNama() +
"...");
        // Lakukan pemeriksaan umum
    }

    @Override
    public void berikanResep(pasien pasien) {
        System.out.println("Dokter umum memberikan resep untuk pasien " +
pasien.getNama() + "...");
        // Berikan resep umum
    }
}

```

Obat.java

```

package praktikum.uts;

import java.util.ArrayList;
import java.util.List;

public class obat {
    private String nama;
    private double harga;
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public double getHarga() {
        return harga;
    }
    public void setHarga(double harga) {
        this.harga = harga;
    }
    public obat(String nama, double harga) {
        this.nama = nama;
        this.harga = harga;
    }
    public List<obat> obatList(){
        List<obat> daftarObat = new ArrayList<>();
        daftarObat.add(new obat("Paramex", 2000));
        daftarObat.add(new obat("Panadol Anak", 3500));
        daftarObat.add(new obat("Betadine Salep", 10000));
    }
}

```

```

        daftarObat.add(new obat("Paracetamol", 15000));
        daftarObat.add(new obat("Fasidol", 15000));

        for (obat obat : daftarObat) {
            System.out.println("\nNama: " + obat.getNama());
            System.out.println("Harga: " + obat.getHarga());
            System.out.println("-----");
        }
        return daftarObat;
    }
}

class obatanak extends obat {
    public obatanak(String nama, double harga) {
        super(nama, harga);
        List<obat> obatAnakList = obatList();

        for (obat obatAnak : obatAnakList) {
            double diskon = 0.15 * obatAnak.getHarga();
            System.out.println("Nama Obat Anak: " + obatAnak.getNama());
            System.out.println("Harga Obat Anak: " + obatAnak.getHarga());
            System.out.println("Diskon Obat Anak: " + diskon);
            System.out.println("-----");
            System.out.println("Harga akhir: " + (obatAnak.getHarga() -
diskon)+"\n");
        }
        //TODO Auto-generated constructor stub
    }
}

```

Apotek.java (main class)

```

package praktikum.uts;
import java.util.Scanner;

public class apotek{
    public static void main(String[] args) {
        pasien pasien1 = new pasien();
        pasien1.bacaInput(); // Gunakan method tanpa parameter
        pasien pasien2 = new pasien();
        pasien2.bacaInput(); // Gunakan method dengan parameter custom
    }
}

```

```

        System.out.println("Pasien 1: " + pasien1.getNama() + " - Gejala: " +
pasien1.getGejala());
        System.out.println("Pasien 2: " + pasien2.getNama() + " - Gejala: " +
pasien2.getGejala());

        obat listObat = new obat(null, 0);
        listObat.obatList();
        Scanner input = new Scanner(System.in);
        System.out.println("Apakah mau membeli obat?");
        String pilih = input.nextLine().trim();
        if(pilih.equalsIgnoreCase("ya") || pilih.equalsIgnoreCase("y")){
            System.out.println("Apakah obat ini untuk anak kecil?");
            String anak = input.nextLine();
            if(anak.equalsIgnoreCase("ya") || anak.equalsIgnoreCase("y")){
                System.out.println("Mau pilih obat apa?");
                obatanak obatAnakList = new obatanak(null, 0);
            }else{
                Dokter dokterUmum = new DokterUmum();
                dokterUmum.periksaPasien(pasien1);
                dokterUmum.berikanResep(pasien1);
                dokterUmum.periksaPasien(pasien2);
                dokterUmum.berikanResep(pasien2);
            }
        }else{
            System.out.println("Error!!");
        }
    }
}

```

1. Keberadaan Enkapsulasi

Class pasien pada pasien.java

```

private String nama;
private String gejala;
public String getNama() {
    return nama;
}
public void setNama(String nama) {
    this.nama = nama;
}
public String getGejala() {
    return gejala;
}
public void setGejala(String gejala) {

```

```

        this.gejala = gejala;
    }
    public void bacaInput(){
        Scanner input = new Scanner(System.in);
        System.out.println("Masukkan Nama Anda");
        nama = input.nextLine();
        System.out.println("Apa keluhan Anda");
        gejala = input.nextLine();
    }

```

Disini dijelaskan bahwa kita mendeskripsikan sebuah class bernama pasien yang bisa menampung nama serta gejala apa yang diderita, lalu di set dimana variabel nama akan menampung sebuah inputan dari String nama(lebih jelasnya adalah variabel nama di bacaInput()). Kemudian di get untuk bisa ditampilkan di main class(apotek.java) Deklarasinya di apotek.java

```

    pasien pasien1 = new pasien();
    pasien1.bacaInput(); // Gunakan method tanpa parameter
    pasien pasien2 = new pasien();
    pasien2.bacaInput(); // Gunakan method dengan parameter custom
    System.out.println("Pasien 1: " + pasien1.getNama() + " -
Gejala: " + pasien1.getGejala());
    System.out.println("Pasien 2: " + pasien2.getNama() + " -
Gejala: " + pasien2.getGejala());

```

Screenshot

```

Masukkan Nama Anda
ido
Apa keluhan Anda
demam
Masukkan Nama Anda
rena
Apa keluhan Anda
sakit kepala
Pasien 1: ido - Gejala: demam
Pasien 2: rena - Gejala: sakit kepala

```

2. Keberadaan Inheritance

Class obat pada obat.java

```
public class obat {
    private String nama;
    private double harga;
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public double getHarga() {
        return harga;
    }
    public void setHarga(double harga) {
        this.harga = harga;
    }
    public obat(String nama, double harga) {
        this.nama = nama;
        this.harga = harga;
    }
    public List<obat> obatList(){
        List<obat> daftarObat = new ArrayList<>();
        daftarObat.add(new obat("Paramex", 2000));
        daftarObat.add(new obat("Panadol Anak", 3500));
        daftarObat.add(new obat("Betadine Salep", 10000));
        daftarObat.add(new obat("Paracetamol", 15000));
        daftarObat.add(new obat("Fasidol", 15000));

        for (obat obat : daftarObat) {
            System.out.println("\nNama: " + obat.getNama());
            System.out.println("Harga: " + obat.getHarga());
            System.out.println("-----");
        }
        return daftarObat;
    }
}
```

Child class (obatanak)

```
class obatanak extends obat {
    public obatanak(String nama, double harga) {
        super(nama, harga);
        List<obat> obatAnakList = obatList();

        for (obat obatAnak : obatAnakList) {
```

```

        double diskon = 0.15 * obatAnak.getHarga();
        System.out.println("Nama Obat Anak: " +
obatAnak.getNama());
        System.out.println("Harga Obat Anak: " +
obatAnak.getHarga());
        System.out.println("Diskon Obat Anak: " + diskon);
        System.out.println("-----");
        System.out.println("Harga akhir: " + (obatAnak.getHarga()
- diskon)+"\n");
    }
    //TODO Auto-generated constructor stub
}
}

```

Di inheritance ini terlihat bahwa ada class obat yang mendeskripsikan nama serta harga obatnya, dan ditampung di sebuah list untuk menampung beberapa obat sekaligus. Nah tugas dari child class sendiri adalah mengambil data dari list yang terdapat di class obat dengan tetap menggunakan variabel yang sama dari class induknya. Namun dengan menambahkan diskon lalu mengubah harga aslinya menjadi harga yang sudah didiskon.

3.

Screenshot

```

y
Mau pilih obat apa?

Nama: Paramex
Harga: 2000.0
-----

Nama: Panadol Anak
Harga: 3500.0
-----

Nama: Betadine Salep
Harga: 10000.0
-----

Nama: Paracetamol
Harga: 15000.0
-----

Nama: Fasidol
Harga: 15000.0
-----

Nama Obat Anak: Paramex
Harga Obat Anak: 2000.0
Diskon Obat Anak: 300.0
-----

Harga akhir: 1700.0
Nama Obat Anak: Panadol Anak
Harga Obat Anak: 3500.0
Diskon Obat Anak: 525.0
-----

Harga akhir: 2975.0
Nama Obat Anak: Betadine Salep
Harga Obat Anak: 10000.0
Diskon Obat Anak: 1500.0
-----

Harga akhir: 8500.0
Nama Obat Anak: Paracetamol
Harga Obat Anak: 15000.0
-----

Nama Obat Anak: Fasidol
Harga Obat Anak: 15000.0

```

3. Keberadaan Polimorphism

Class pasien yang sama

Dengan deklarasi class pasien yang sama seperti sebelumnya, tinggal menambahkan object menjadi pasien1 dan pasien2 untuk menciptakan Overload dan lalu ditampilkan di apotek.java
Deklarasinya di apotek.java

```

pasien pasien1 = new pasien();
pasien1.bacaInput(); // Gunakan method tanpa parameter
pasien pasien2 = new pasien();

```



```

        pasien2.bacaInput(); // Gunakan method dengan parameter custom
        System.out.println("Pasien 1: " + pasien1.getNama() + " -
Gejala: " + pasien1.getGejala());
        System.out.println("Pasien 2: " + pasien2.getNama() + " -
Gejala: " + pasien2.getGejala());

```

Screenshot

```

Masukkan Nama Anda
ido
Apa keluhan Anda
demam
Masukkan Nama Anda
rena
Apa keluhan Anda
sakit kepala
Pasien 1: ido - Gejala: demam
Pasien 2: rena - Gejala: sakit kepala

```

4. Keberadaan Interface

Interface dokter di class pasien.java yang sama

```

// Interface untuk dokter
interface Dokter {
    void periksaPasien(pasien pasien);
    void berikanResep(pasien pasien);
}

// Class untuk dokter umum
class DokterUmum implements Dokter {

    @Override
    public void periksaPasien(pasien pasien) {
        System.out.println("Dokter umum memeriksa pasien " +
pasien.getNama() + "...");
        // Lakukan pemeriksaan umum
    }

    @Override
    public void berikanResep(pasien pasien) {
        System.out.println("Dokter umum memberikan resep untuk pasien
" + pasien.getNama() + "...");
        // Berikan resep umum
    }
}

```

Bayangkan interface seperti peraturan. Peraturan ini menentukan apa yang harus dokter lakukan: "memeriksa pasien" dan "memberikan resep".

Dokter umum seperti dokter yang mengikuti peraturan ini. Dia harus melakukan dua hal yang disebutkan dalam peraturan.

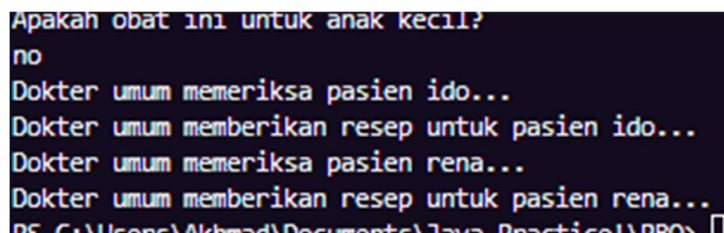
Di apotek, kita mempekerjakan dokter umum. Kita bisa menyuruhnya memeriksa pasien dan memberikan resep karena dia sudah mengikuti peraturan (mengimplementasikan interface).

Deklarasinya di apotek.java

```
Dokter dokterUmum = new DokterUmum();
dokterUmum.periksaPasien(pasien1);
dokterUmum.berikanResep(pasien1);
dokterUmum.periksaPasien(pasien2);
dokterUmum.berikanResep(pasien2);
```

3.

Screenshot



```
Apakah obat ini untuk anak kecil?
no
Dokter umum memeriksa pasien ido...
Dokter umum memberikan resep untuk pasien ido...
Dokter umum memeriksa pasien rena...
Dokter umum memberikan resep untuk pasien rena...
PS C:\Users\Alhamdulillah\Documents\Java-Practical\080>
```