BAB 7
POLIMORFISME

Tujuan

1. Memberikan pemahaman kepada mahasiswa tentang polimorfisme
2. Dapat membedakan perbedaan antara polymorfisme dan inheritance

Ringkasan Materi

A. Polymorfisme

Polimorfisme (memiliki banyak bentuk) menyatakan kemampuan untuk memperlakukan objek-objek dengan cara yang seragam meskipun objek-objek tersebut berbeda perilaku. Polimorfisme adalah suatu sifat yang merupakan efek langsung dari sifat inheritance pada OOP. Jika pada inheritance semua perilaku yang diturunkan kepada subclass memiliki bentuk yang sama, namun pada polimorfisme ini subclass dapat mendefinisikan perilaku yang sama dengan cara yang berbeda, dimana perilaku ini merupakan turunan dari super class.

Untuk bisa mengimplementasikan sifat polimorfisme, kita harus memakai method yang memungkinkan dibuat tanpa didefinisikan. Pada sub classnya nanti kita bisa mendefinisikan isi method tersebut sesuai tujuan masing-masing sub class. Dalam java kita dapat mendefinisikan method semacam ini dengan kata kunci **abstrak**, **dengan syarat kelas yang menampung method tersebut juga harus berupa Kelas abstrak**. **Kelas abstrak harus mengandung satu atau lebih method abstrak tapi boleh ada method selain abstrak**. Semua method abstrak yang ada di superclass harus dioverride di subclassnya

Cara pendeklarasian class abstract :

```
public abstract class <nama_kelas> {
   …
}
```

Cara pendeklarasian method abstract :

```
<modifier> <return_value> abstract <nama_method>;
```

Sebagai contoh, ada kelas MakhlukHidup sebagai kelas induk, yang mempunyai method tertentu misalnya *bernafas()*, *makan()*, *tidur()*, dan *berjalan()*. Kelas MakhlukHidup memiliki kelas turunan Manusia, Sapi, dan Kanguru. Dengan konsep inheritance, semua class turunan MakhlukHidup harus mengimpementasikan semua perilaku yang dimiliki kelas super tersebut. Meskipun subclass (Manusia, Sapi, dan Kanguru) bisa mengimplementasikan semua perilaku yang didefinisikan, namun sub class tersebut mengimplementasikannya dengan cara yang berbeda. Misalnya perilaku *berjalan()*, Manusia, Sapi dan Kanguru berjalan dengan cara yang berbeda. Jika Manusia berjalan dengan dua kaki, Sapi berjalan dengan empat kaki dan Kanguru berjalan dengan cara melompat dengan dua kaki. Implementasi contoh diatas adalah sebagai berikut.

```
public abstract class MakhlukHidup {
    public void bernafas(){
        System.out.println("adalah Makhluk hidup yang dapat
        bernafas");
    }
    public abstract void berjalan();
}
```

Untuk subclass yang mengimplementasikan kelas MakhlukHidup :

- Manusia

```
public class Manusia extends MakhlukHidup {
     public void berjalan(){
         System.out.println("Manusi berjalan dengan dua kaki");
```

```
    }
}
```

- Sapi

```
public class Sapi extends MakhlukHidup{
    public void berjalan(){
        System.out.println("Sapi berjalan dengan 4 kaki");
    }
}
```

- Kanguru

```
public class Kanguru {
    public void berjalan(){
        System.out.println("Kanguru berjalan dengan melompat pada 2
        kakinya");
    }
}
```

Pelaksanaan Percobaan

Polimorfisme

Ketikkan program di bawah ini. Buatlah class-class berikut dalam package yang sama.

Employee.java

```
1  public abstract class Employee {
2      private String  name;
3      private String noKTP;
4      public Employee(String name, String noKTP){
5          this.name = name;
6          this.noKTP = noKTP;
7      }
8      public String getName(){
9          return name;
10     }
11     public String getNoKTP(){
12         return noKTP;
13     }
14     public String toString(){
15         return     String.format("     "+getName()+"\nNo.     KTP
16 :"+getNoKTP());
17     }
18     public abstract double earnings();//pendapatan
19 }
```

SalariedEmployee.java

```
1  public class SalariedEmployee extends Employee {
2      private double weeklySalary; //gaji/minggu
3      public  SalariedEmployee(String  name,  String  noKTP,  double
   salary) {
4          super(name, noKTP);
5          setWeeklySalary(salary);
6      }
7      public void setWeeklySalary(double salary) {
8          weeklySalary = salary;
9      }
10     public double getWeeklySalary() {
11         return weeklySalary;
```

```
12      }
13      public double earnings() {
14          return getWeeklySalary();
15      }
16      public String toString() {
17          return    String.format("Salaried    employee:    "    +
18  super.toString() +
19                  "\nweekly salary:" + getWeeklySalary());
20      }
21  }
```

HourlyEmployee.java

```
1   public class HourlyEmployee extends Employee {
2       private double wage; //upah per jam
3       private double hours; //jumlah jam tiap minggu
4       public HourlyEmployee(String name, String noKTP,
                double hourlyWage, double hoursWorked) {
5           super(name, noKTP);
6           setWage(hourlyWage);
7           setHours(hoursWorked);
8       }
9       public void setWage(double hourlyWage){
10          wage = hourlyWage;
11      }
12      public double getWage(){
13          return wage;
14      }
15      public void setHours(double hoursWorked){
16          hours = hoursWorked;
17      }
18      public double getHours(){
19          return hours;
20      }
21      public double earnings(){
22          if(getHours()<=40)
23              return getWage() * getHours();
24          else
25              return 40 * getWage() + ( getHours()-40) *
    getWage() * 1.5;
26      }
27      public String toString(){
28          return          String.format("Hourly          employee:
    "+super.toString()
                    +"\nhourly    wage"+getWage()+"\nhours    worked:
    "+getHours());
29      }
30  }
```

Commission.java

```
1   public class CommissionEmployee extends Employee {
2       private double grossSales;//penjualan per minggu
3       private double commissionRate;//komisi
4       public CommissionEmployee(String name, String noKTP, double
```

```
5  sales, double rate){
6          super(name, noKTP);
7          setGrossSales(sales);
8          setCommissionRate(rate);
9      }
10     public void setGrossSales(double sales){
11         grossSales = sales;
12     }
13     public double getGrossSales(){
14         return grossSales;
15     }
16     public void setCommissionRate(double rate){
17         commissionRate = rate;
18     }
19     public double getCommissionRate(){
20         return commissionRate;
21     }
22     public double earnings(){
23         return getCommissionRate()*getGrossSales();
24     }
25     public String toString(){
26         return        String.format("Commision        employee:
27 "+super.toString()+"\ngross                                  sales:
28 "+getGrossSales()+"\ncommission rate"+getCommissionRate());
29     }
30 }
31
```

BasePlusCommissionEmployee.java

```
1  public      class      BasePlusCommissionEmployee      extends
2  CommissionEmployee {
3
4      private double baseSalary;//gaji pokok tiap minggu
5
6      public   BasePlusCommissionEmployee(String   name,   String
7  noKTP, double sales, double rate, double salary) {
8          super(name, noKTP, sales, rate);
9          setBaseSalary(salary);
10     }
11
12     public void setBaseSalary(double salary) {
13         baseSalary = salary;
14     }
15
16     public double getBaseSalary() {
17         return baseSalary;
18     }
19
20     public double earnings() {
21         return getBaseSalary() + super.earnings();
22     }
23
24     public String toString() {
```

```
25          return       String.format("Base-Salaried       "       +
26  super.toString() + "\nbase salary " + getBaseSalary());
27      }
28  }
```

Data dan Analisis hasil percobaan

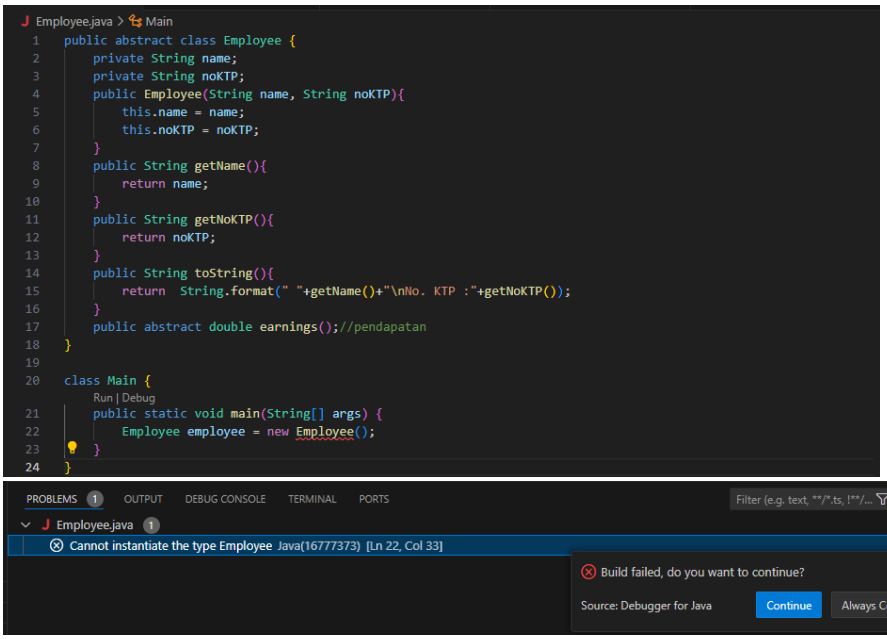
Pertanyaan

1. Ketikkan kode ini.

Main.java

```
1  public class Main {
2     public static void main(String[] args) {
3          Employee employee = new Employee();
4      }
5  }
```

Jalankan Main.java untuk polymorfisme Employee, analisis dan jelaskan keluaran program tersebut!



PENJELASAN : Keluaran apabila ditambahkan class main tersebut yaitu Cannot be Instantiated, yaitu pesan kesalahan yang menunjukkan bahwa suatu objek tidak dapat dibuat atau diinstansiasi. Cara yang tepat dalam mengatasi hal tersebut adalah dengan memperbaiki kode program agar dapat membuat objek yang valid dengan mengimplementasikan konstruktor yang tepat atau membuat kelas turunan yang mengimplementasikan metode-metode yang diperlukan.

2. Jalankan program dengan main sebagai berikut.

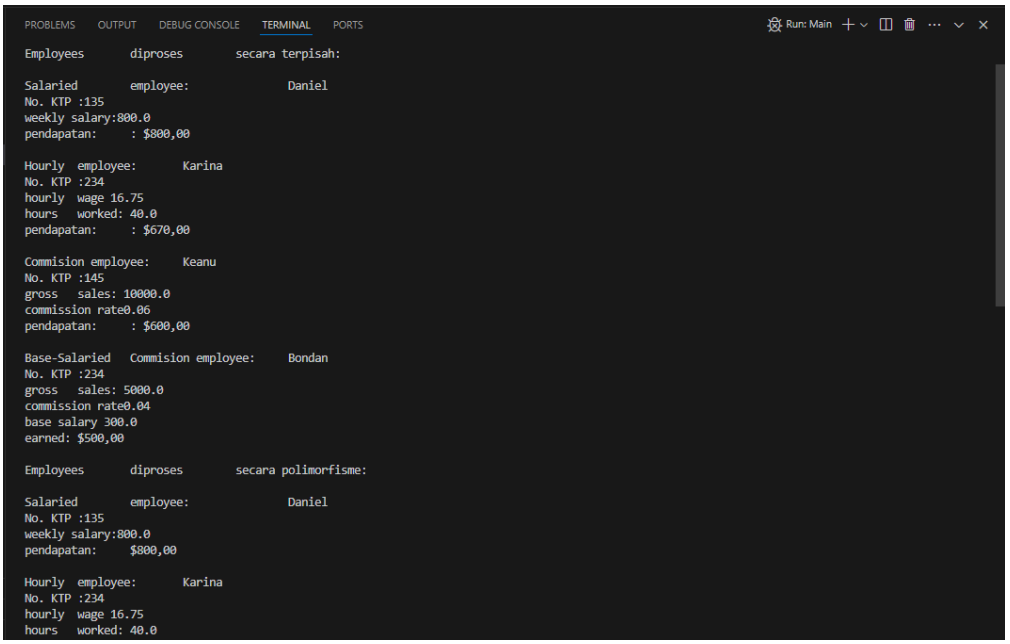
Main.java

```
1  public class Main {
2     public static void main(String[] args) {
3          SalariedEmployee       salariedEmployee       =       new
4  SalariedEmployee("Daniel", "135", 800.00);
5          HourlyEmployee        hourlyEmployee        =        new
6  HourlyEmployee("Karina", "234", 16.75, 40);
7          CommissionEmployee      commissionEmployee      =      new
8  CommissionEmployee("Keanu", "145", 10000, .06);
9          BasePlusCommissionEmployee basePlusCommissionEmployee =
10  new  BasePlusCommissionEmployee("Bondan",  "234",  5000,  .04,
11 300);
12         System.out.println("Employees      diproses      secara
```

```
13  terpisah:\n");
14          System.out.printf("%s\n%s: $%,.2f\n\n",
15                  salariedEmployee,         "pendapatan:         ",
16  salariedEmployee.earnings());
17          System.out.printf("%s\n%s: $%,.2f\n\n",
18                  hourlyEmployee,          "pendapatan:          ",
19  hourlyEmployee.earnings());
20          System.out.printf("%s\n%s: $%,.2f\n\n",
21                  commissionEmployee,        "pendapatan:        ",
22  commissionEmployee.earnings());
23          System.out.printf("%s\n%s: $%,.2f\n\n",
24                  basePlusCommissionEmployee,
25                  "earned",
26  basePlusCommissionEmployee.earnings());
27
28          Employee[] employees = new Employee[4];
29          employees[0] = salariedEmployee;
30          employees[1] = hourlyEmployee;
31          employees[2] = commissionEmployee;
```

```
        employees[3] = basePlusCommissionEmployee;
        System.out.println("Employees diproses secara
polimorfisme:\n");
        for (Employee currentEmployee : employees) {
            System.out.println(currentEmployee);
            if (currentEmployee instanceof
BasePlusCommissionEmployee) {
                BasePlusCommissionEmployee employee =
(BasePlusCommissionEmployee) currentEmployee;
                employee.setBaseSalary(1.10 *
employee.getBaseSalary());
                System.out.printf(
                        "Gaji pokok setelah dinaikkan 10%% :
$%,.2f\n",
                        employee.getBaseSalary());
            }
            System.out.printf("pendapatan: $%,.2f\n\n",
currentEmployee.earnings());
        }
        for (int j = 0; j < employees.length; j++) {
            System.out.printf("Employee %d = %s\n", j,
employees[j].getClass().getName());
        }
    }
}
```

Analisis dan jelaskan output program (berdasarkan konsep polimorfisme)!



PENJELASAN : Program tersebut merupakan program yang menunjukkan penggunaan konsep inheritance dan polymorphism dalam pemrograman berorientasi objek. Program ini memiliki beberapa kelas seperti Employee, SalariedEmployee, HourlyEmployee, CommissionEmployee, dan BasePlusCommissionEmployee. Output dari program tersebut yaitu : 1. Data dari setiap objek SalariedEmployee, HourlyEmployee, CommissionEmployee, dan BasePlusEmployee yang dibuat tercetak pada layer. Data yang ditampilkan diantaranya yaitu nama, nomor KTP, gaji pokok, jumlah jam kerja, dan pendapatan karyawan tersebut. 2. Data dari setiap objek diproses secara polimorfisme. Data yang dicetak sama seperti pada pemrosesan secara terpisah, tetapi juga termasuk data gaji pokok setelah dinaikkan 10% bagi objek BasePlusCommissionEmployee. 3. Terdapat juga output yang meunjukkan jenis dari setiap objek yang disimpan dalam array employees, jenis dari setiap objek tersebut dicetak dengan menggunakan getClass().getName().

3. **Buat objek dari method Employee? Jelaskan hasil dari output program tersebut!**

```
Employee employee = new Employee() {

    @Override
    public double earnings() {
        return 0;
    }
}
```

PENJELASAN : Akan terjadi error pada program tersebut, alasannya karena tidak dapat membuat objek method yang merupakan bagian dari sebuah kelas yang harus memiliki objek terlebih dahulu sebelum pemanggilan dilakukan.

4. **Tambahkan atribut tanggal lahir di Kelas Employee, serta tambahkan method pendukungnya (accesor dan mutator). Modifikasi program agar sesuai. Asumsikan gaji yang diterima adalah per bulan, buat kelas uji untuk menguji program yang sudah anda modifikasi, kemudian buat objek dari semua class (salariedEmployee, hourlyEmployee, commissionEmployee, basePlusCommissionEmployee dan hitung gajinya secara polimorfisme, serta tambahkan gajinya sebesar 100.000 jika bulan ini adalah bulan ulang tahunnya.**

Employee.java > Employee

```
public abstract class Employee {
    private String name;
    private String noKTP;
    private String tanggalLahir;
    public Employee(String name, String noKTP){
        this.name = name;
        this.noKTP = noKTP;
        this.tanggalLahir = tanggalLahir;
    }
    public String getName(){
        return name;
    }
    public String getNoKTP(){
        return noKTP;
    }
```

Menambahkan atribut tanggal lahir

SalariedEmployee.java > SalariedEmployee > setWeeklySalary(double)

```
public class SalariedEmployee extends Employee {
    private double weeklySalary; //gaji/minggu
    public SalariedEmployee(String name, String noKTP, double salary) {
        super(name, noKTP);
        setWeeklySalary(salary);
    }
    public void setWeeklySalary(double salary) {
        weeklySalary = salary < 0.0 ? 0.0 : salary;
    }
    public double getWeeklySalary() {
        return weeklySalary;
    }
    public double earnings() {
        return getWeeklySalary();
    }
    public String toString() {
```

HourlyEmployee.java > HourlyEmployee > getTanggalLahir()

```
public class HourlyEmployee extends Employee {
    private double wage; //upah per jam
    private double hours; //jumlah jam tiap minggu
    private String tanggalLahir; // tanggal lahir dalam format dd-mm-yyyy
    public HourlyEmployee(String name, String noKTP, double hourlyWage, double hoursWorked, String tanggalLahir)
        super(name, noKTP, tanggalLahir);
        setWage(hourlyWage);
        setHours(hoursWorked);
        setTanggalLahir(tanggalLahir);
    }
    public void setWage(double hourlyWage){
        wage = hourlyWage;
    }
    public double getWage(){
        return wage;
    }
    public void setHours(double hoursWorked){
        hours = hoursWorked;
    }
    public double getHours(){
        return hours;
    }
    public void setTanggalLahir(String tanggalLahir){
        tanggalLahir = tanggalLahir;
    }
    public String getTanggalLahir(){
        return tanggalLahir;
    }
```

CommissionEmployee.java > ...

```java
public class CommissionEmployee extends Employee {
    private double grossSales;//penjualan per minggu
    private double commissionRate;//komisi
    public CommissionEmployee(String name, String noKTP, double sales, double rate, String tanggalLahir){
        super(name, noKTP, tanggalLahir);
        setGrossSales(sales);
        setCommissionRate(rate);
    }
    public void setGrossSales(double sales){
        grossSales = sales;
    }
    public double getGrossSales(){
        return grossSales;
    }
    public void setCommissionRate(double rate){
        commissionRate = rate;
    }
    public double getCommissionRate(){
        return commissionRate;
    }
    public double earnings(){
        return getCommissionRate()*getGrossSales();
    }
    public String toString(){
        return String.format("Commision employee: "+super.toString()+"\ngross   sales: "+getGrossSales()+"\ncomm
    }
}
```

BasePlusCommissionEmployee.java > BasePlusCommissionEmployee > BasePlusCommissionEmployee(String, String, double, double, double, String)

```java
public  class   BasePlusCommissionEmployee  extends CommissionEmployee {
    private double baseSalary;//gaji pokok tiap minggu
    private String tanggalLahir;

    public  BasePlusCommissionEmployee(String   name,   String noKTP, double sales, double rate, double salary,
        super(name, noKTP, sales, rate, tanggalLahir);
        setBaseSalary(salary);
        setTanggalLahir(tanggalLahir);
    }

    private void setTanggalLahir(String tanggalLahir2) {
        // TODO Auto-generated method stub
        throw new UnsupportedOperationException(message:"Unimplemented method 'setTanggalLahir'");
    }

    public void setBaseSalary(double salary) {
        baseSalary = salary;
    }

    public double getBaseSalary() {
        return baseSalary;
    }

    public double earnings() {
        return getBaseSalary() + super.earnings();
    }
```

Modifikasi subclass

5. **Perusahaan yang mengaplikasikan program polimorfisme diatas ingin menambahkan kriteria baru untuk penggajian karyawannya, yaitu penggajian berdasarkan banyaknya barang yang diproduksi. Dengan ketentuan gaji karyawan tersebut adalah hasil dari banyaknya barang yang dipoduksi per minggu dikalikan upah per barangnya.**
   a. **Analisis dan jelaskan proses modifikasi program diatas (dimulai dari pemilihan jenisclass, perancangan class, dan penempatan class)**
   b. **Implementasi hasil analisis tersebut ke dalam program dan buat kelas uji denganminimal 4 objek yang dibentuk.**

JAWABAN :

a. Dalam memodifikasi program diatas, terdapat penambahan kriteria baru untuk penggajian karyawan berdasarkan banyaknya barang yang diproduksi per minggu nya. Proses modifikasi program tersebut melibatkan beberapa tahap, yaitu:

   1. Pemilihan jenis class

      Dalam menambahkan kriteria baru untuk penggajian karyawan, diperlukan sebuah class baru yang mampu menghitung gaji berdasarkan banyaknya barang yang diproduksi. Salah satu jenis class yang dapat digunakan adalah class ProductionWorker, yang merepresentasikan karyawan produksi yang memiliki upah per barang.

   2. Perancangan class

      Setelah jenis class dipilih, dilakukan perancangan class ProductionWorker dengan menentukan atribut dan method yang dibutuhkan. Atribut yang dibutuhkan adalah upah per barang dan jumlah barang yang diproduksi per minggu. Method yang dibutuhkan adalah accessor dan mutator untuk mengakses dan mengubah nilai atribut, serta method untuk menghitung gaji.

   3. Penempatan class

Setelah class ProductionWorker dirancang, selanjutnya dilakukan penempatan class tersebut ke dalam program utama. Class ProductionWorker dapat ditempatkan bersama dengan class Employee dan subclassnya, karena keduanya saling terkait dengan karyawan dan penggajian. Selain itu, diperlukan juga penyesuaian pada class main untuk memperhitungkan penggajian berdasarkan produksi.

Dengan adanya modifikasi program tersebut, perusahaan dapat melakukan penggajian karyawan berdasarkan produksi dengan mudah dan efisien. Hal ini memungkinkan perusahaan untuk memberikan gaji yang lebih adil dan sesuai dengan kinerja karyawan.

b. Implementasi Program

```
public class ProductionWorker extends Employee {

private int shift;

private double payRate;

private int totalProduced;



public ProductionWorker(String name, String
socialSecurityNumber, Date tanggalLahir, int shift,
double payRate) {

super(firstName, lastName, socialSecurityNumber,birthDate);

setShift(shift);

setPayRate(payRate);

totalProduced = 0;

}



public int getShift() {

return shift;

}



public void setShift(int shift) { if

(shift == 1 || shift == 2) {

this.shift = shift;

} else {
```

```
            System.out.println("Invalid shift, pleaseenter 1 or
            2");
      }
}


public double getPayRate() {return
      payRate;
}


public void setPayRate(double payRate) {if (payRate
      >= 0) {
            this.payRate = payRate;
      } else {
            System.out.println("Pay rate cannot be
negative");
      }
}


public int getTotalProduced() {return
      totalProduced;
}


public void addProduced(int amount) {if (amount
      >= 0) {
            totalProduced += amount;
      } else {
```

```
                System.out.println("Invalid amount, cannotadd
negative value");

           }

     }



     @Override

     public double calculatePay() { return

           payRate * totalProduced;

     }
  }
```

Tugas Praktikum

I. Buatlah sebuah klas **abstract** Kue yang memiliki attribut dan method sebagai berikut

- nama : String
- harga : double
+ *hitungHarga()***** : double
+ toString : String *(menampilkan nama kue dan harga)*

**** abstract**

II. Buatlah 2 subklas dari klas Kue yaitu

a. KuePesanan
   - berat : double
   + hitungHarga() : double
     Hitung harga berdasarkan harga x berat

b. KueJadi
   - jumlah : double
   + hitungHarga() : double
     Hitung harga berdasarkan harga x jumlah x 2

III. Berdasarkan 2 kelas tersebut, buatlah :

1. Array yang terdiri dari 20 kue
2. Isikan 20 objek kue dengan berbagai jenis kue (KuePesanan atau KueJadi)
3. Dari array tersebut :
   a. Tampilkan semua kue dan harus ditampilkan jenis kuenya
   b. Hitung total harga yang didapat dari semua jenis kue
   c. Hitung total harga dan total berat dari KuePesanan
   d. Hitung total harga dan total jumlah dari KueJadi
   e. Tampilkan informasi kue dengan harga (harga akhir) terbesar

```
package Prak;


abstract class Kue {
   private String nama;
   private int jumlah;
   private int harga;

   public Kue(String nama, int jumlah, int harga) {
       this.nama = nama;
       this.jumlah = jumlah;
       this.harga = harga;
   }

   public Kue(String nama, double harga) {
   }

   public String getNama() {
       return nama;
```

```
}
```

```
public void setNama(String nama) {
    this.nama = nama;
}

public int getJumlah() {
    return jumlah;
}

public void setJumlah(int jumlah) {
    this.jumlah = jumlah;
}

public int getHarga() {
    return harga;
}

public void setHarga(int harga) {
    this.harga = harga;
}

public int hitungTotalHarga() {
    return jumlah * harga;
}
```

```
    public String toString() {

        return "Nama kue: " + nama + "\nJumlah kue: " +
jumlah + "\nHarga per kue: " + harga + "\nTotal harga: "
+ hitungTotalHarga();

    }

}
```

CLASS KUE

```
package Prak;


public class KuePesanan {

    private Kue jenisKue;

    private int jumlah;

    private String namaPemesan;


    public KuePesanan(Kue jenisKue, int jumlah, String
namaPemesan) {

        this.jenisKue = jenisKue;

        this.jumlah = jumlah;

        this.namaPemesan = namaPemesan;

    }


    public Kue getJenisKue() {

        return jenisKue;

    }
```

```
        public void setJenisKue(Kue
    jenisKue) {this.jenisKue = jenisKue;
}

public int getJumlah() {
    return jumlah;
}

public void setJumlah(int jumlah) {
    this.jumlah = jumlah;
}

public String getNamaPemesan() {
    return namaPemesan;
}

public void setNamaPemesan(String namaPemesan) {
    this.namaPemesan = namaPemesan;
}

public double getTotalHarga() {
     return jenisKue.getHarga() * jumlah;
}
```

```
    @Override
    public String toString() {
        return "Pesanan untuk " + namaPemesan + ", " +
jumlah + " " + jenisKue.getNama() + " dengan total harga
" + getTotalHarga();
    }
}
```

CLASS KUE PESANAN

```
package Prak;

public class KueJadi extends Kue {
    private double jumlah;

    public KueJadi(String nama, double harga, double
jumlah) {
        super(nama, harga);
        this.jumlah = jumlah;
    }

    public int getJumlah() {
        return (int) this.jumlah;
    }

    public double hitungHarga() {
        return this.jumlah * super.getHarga() * 2;
    }
```

```
    public String toString() {

        return super.toString() + "\nJumlah Kue\t\t: " +
this.jumlah + "\nTotal Harga\t\t: " + hitungHarga()

                + "\nJenis Kue\t\t: Kue Jadi";

    }

}
```

CLASS KUE JADI

```
package Prak;


public class Main {

    public static void main(String[]

   args) {Kue[] kues = new Kue[20];

   kues[0] = new KuePesanan("Lemper", 1000, 6);

   kues[1] = new KuePesanan("Bolu", 2000, 5);

   kues[2] = new KuePesanan("Pastel", 1000, 7);

   kues[3] = new KuePesanan("Kue Lapis", 1500, 2);

   kues[4] = new KuePesanan("Bolu kukus", 1500, 4);

   kues[5] = new KuePesanan("Red Velvet", 5000, 3);

   kues[6] = new KuePesanan("Bika Ambon", 2500, 5);

   kues[7] = new KuePesanan("Kue Cucur", 1500, 2);

   kues[8] = new KuePesanan("Kue Apem", 1000, 9);

   kues[9] = new KuePesanan("Naga Sari", 1000, 8);

   kues[10] = new KueJadi("Lemper", 1000, 23);

   kues[11] = new KueJadi("Bolu", 2000, 45);
```

```
        kues[12] = new KueJadi("Pastel", 1000, 21);

        kues[13] = new KueJadi("Kue Lapis", 1500, 10);

        kues[14] = new KueJadi("Bolu kukus", 1500, 5);

        kues[15] = new KueJadi("Red Velvet", 5000, 8);

        kues[16] = new KueJadi("Bika Ambon", 2500, 25);

        kues[17] = new KueJadi("Kue Cucur", 1500, 17);

        kues[18] = new KueJadi("Kue Apem", 1000, 11);

        kues[19] = new KueJadi("Naga Sari", 1000, 3);

        System.out.println(); System.out.println("Kue
    Pesanan :   "); System.out.println();

        for (int i = 0; i < 20; i++) {if

            (i == 10) {

                System.out.println("");
    System.out.println("Kue Jadi : ");
    System.out.println(""); System.out.println();

                    }
    System.out.println(kues[i].toString());
    System.out.println();

    }

        double totalHarga = 0;

        for (int i = 0; i < 20; i++) { totalHarga

            += kues[i].hitungHarga();

        }

        double hargaP = 0;

        double beratP = 0;

        String jumlahJ = null;

        for (int i = 0; i < 10; i++) {

        hargaP += kues[i].hitungHarga();
```

```
if (kues[i] instanceof KuePesanan) {
        KuePesanan pesanan = (KuePesanan)
        KueJadi jadi = null;
        jumlahJ += jadi.getJumlah();
   }
           }
           double harga[] = new
        double[20];
           for (int i = 0; i < 20; i++) {
               harga[i] =
        kues[i].hitungHarga();
           }
   String termahal = "";
   double tertinggi =
        getTertinggi(harga); for (int
   j = 0; j < 20; j++) {
               if (kues[j].hitungHarga()
        == tertinggi) {
                   termahal =
        kues[j].getNama();
               }
   }
   System.out.println("==================
        ================== ========");
   System.out.println("Total Harga
        Kue\t\t\t: Rp. " + totalHarga);
   System.out.println("Total Harga Kue
        Pesanan\t: Rp. " + hargaP);
```

```
    System.out.println("Total Berat Kue

        Pesanan\t: "+ beratP);
```

```
        String hargaJ = null;

        System.out.println("Total Harga Kue Jadi\t: Rp. "
+ hargaJ);

        System.out.println("Total Jumlah Kue Jadi\t: " +
jumlahJ);

        System.out.println("Kue Termahal\t\t\t: " +
termahal);


System.out.println("==================================
========");

    }

    public static double getTertinggi(double[]

                                            inputArray)
{

        double nilaiTertinggi = inputArray[0];

        for (int i = 1; i < inputArray.length; i++) {

            if (inputArray[i] > nilaiTertinggi) {

                nilaiTertinggi = inputArray[i];

            } }

        return nilaiTertinggi;

    }

}
```

CLASS MAIN