

NAMA : Haniifan Thoriqul Ibad

NIM : 225150407111089

KELAS : SI-B

BAB : 7

ASISTEN : Adin Rama Ariyanto Putra dan Fahru Setiawan Iskandar

1. Data dan Analisis hasil percobaan

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

Jawab:

Terdapat error yang dimana tipe Employee tidak dapat diinstansiasi.

2. Analisis dan jelaskan output program (berdasarkan konsep polimorfisme)!

Jawab:

Program di atas menggunakan konsep polimorfisme dalam pemrosesan objek-objek yang berasal dari kelas-kelas yang berbeda namun memiliki hubungan ke "Employee" sebagai kelas dasar.

Output program akan mencetak informasi mengenai setiap karyawan yang diproses secara terpisah dan juga diproses secara polimorfisme.

Berikut adalah analisis dan penjelasan output program:

Pemrosesan secara terpisah:

- a. Untuk setiap objek karyawan yang diproses secara terpisah, program akan mencetak informasi karyawan, seperti nama, nomor identifikasi, dan pendapatan.
- b. Pendapatan karyawan dihitung menggunakan metode "earnings()" yang sesuai dengan jenis karyawan.
- c. Output program akan mencetak informasi karyawan dan pendapatan mereka secara terpisah.

Contoh output untuk pemrosesan secara terpisah:

Employees diproses secara terpisah:

Daniel: 135

pendapatan: \$800.00

Karina: 234

pendapatan: \$670.00

Keanu: 145

pendapatan: \$600.00

Bondan: 234

earned: \$740.00

Pemrosesan secara polimorfisme:

- Objek karyawan disimpan dalam array "employees" yang memiliki tipe data "Employee". Hal ini memungkinkan penggunaan polimorfisme.
- Program akan melalui setiap objek karyawan dalam array dan mencetak informasi karyawan dan pendapatan mereka.
- Penggunaan operator "instanceof" digunakan untuk memeriksa apakah objek karyawan adalah instance dari "BasePlusCommissionEmployee". Jika iya, maka gaji pokok akan dinaikkan sebesar 10%.
- Setelah peningkatan gaji pokok, program mencetak informasi karyawan dan pendapatan mereka.

Contoh output untuk pemrosesan secara polimorfisme:

Employees diproses secara polimorfisme:

Daniel: 135

pendapatan: \$800.00

Karina: 234

pendapatan: \$670.00

Keanu: 145

pendapatan: \$600.00

Bondan: 234

Gaji pokok setelah dinaikkan 10%: \$330.00

pendapatan: \$830.00

Informasi tentang kelas objek:

- a. Setelah pemrosesan secara polimorfisme, program akan mencetak informasi tentang kelas objek yang disimpan dalam array "employees" menggunakan metode "getClass().getName()".
- b. Output program akan mencetak indeks karyawan dan nama kelas karyawan.

Contoh output untuk informasi kelas objek:

Employee 0 = SalariedEmployee

Employee 1 = HourlyEmployee

Employee 2 = CommissionEmployee

Employee 3 = BasePlusCommissionEmployee

Dengan demikian, program ini menggambarkan penggunaan polimorfisme dalam memproses objek-objek yang berasal dari kelas-kelas yang berbeda namun memiliki hubungan ke "Employee" sebagai kelas dasar.

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

Jawab:

Tidak dapat membuat objek langsung dari kelas abstrak Employee karena kelas tersebut tidak dapat diinstansiasi. Namun, Anda dapat membuat objek dari kelas turunan Employee seperti HourlyEmployee, CommissionEmployee, dan BasePlusCommissionEmployee.

Berikut adalah contoh membuat objek dari kelas HourlyEmployee:

```
Employee employee1 = new HourlyEmployee("Jean", "123", 15.0, 50);  
System.out.println(employee1.toString());  
System.out.println("Earnings: " + employee1.earnings());
```

Hasil output program tersebut akan mencetak informasi tentang karyawan per jam, termasuk nama, nomor KTP, upah per jam, dan jumlah jam kerja. Selain itu, juga akan mencetak pendapatan karyawan berdasarkan upah per jam dan jam kerja yang dihitung dengan aturan lembur (jika jam kerja melebihi 40 jam).

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.

Jawab:

Berikut adalah modifikasi dari program yang diberikan dengan menambahkan atribut "tanggalLahir" di kelas Employee, serta menambahkan metode accessor dan mutator untuk atribut tersebut. Selain itu, gaji yang diterima diasumsikan per bulan, dan ditambahkan logika untuk menambahkan gaji sebesar 100.000 jika bulan ini adalah bulan ulang tahun karyawan.

```
import java.time.LocalDate;
```

```
public abstract class Employee {  
    private String name;  
    private String noKTP;  
    private LocalDate tanggalLahir;
```

```
    public Employee(String name, String noKTP, LocalDate tanggalLahir) {  
        this.name = name;  
        this.noKTP = noKTP;
```

```
this.tanggalLahir = tanggalLahir;  
}
```

Laporan Praktikum Pemrograman Lanjut SI 4



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
public String getName() {  
    return name;  
}  
  
public String getNoKTP() {  
    return noKTP;  
}  
  
public LocalDate getTanggalLahir() {  
    return tanggalLahir;  
}  
  
public void setTanggalLahir(LocalDate tanggalLahir) {  
    this.tanggalLahir = tanggalLahir;  
}  
  
public String toString() {  
    return String.format("Name: %s\nNo. KTP: %s\nTanggal Lahir: %s", getName(),  
        getNoKTP(), getTanggalLahir());  
}  
  
public abstract double earnings(); // Pendapatan  
}
```

Selanjutnya, modifikasi kelas-kelas turunan Employee:

```
import java.time.LocalDate;
```

```

public class HourlyEmployee extends Employee {
    public HourlyEmployee(String name, String noKTP, LocalDate tanggalLahir, double
    hourlyWage, double hoursWorked) {
        super(name, noKTP, tanggalLahir);

```

Laporan Praktikum Pemrograman Lanjut SI 5



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

}

```

```

// ...

```

```

public double earnings() {
    double earnings = getWage() * getHours();
    if (isBirthday()) {
        earnings += 100000;
    }
    return earnings;
}

```

```

private boolean isBirthday() {
    LocalDate today = LocalDate.now();
    LocalDate tanggalLahir = getTanggalLahir();
    return today.getMonthValue() == tanggalLahir.getMonthValue() &&
    today.getDayOfMonth() == tanggalLahir.getDayOfMonth();
}

```

```

// ...

```

```

}

```

```

public class CommissionEmployee extends Employee {

```

```

    public CommissionEmployee(String name, String noKTP, LocalDate tanggalLahir,
    double sales, double rate) {

```

```

        super(name, noKTP, tanggalLahir);
    }

```

```

    public double earnings() {
        double earnings = getCommissionRate() * getGrossSales();
        if (isBirthday()) {

```

Laporan Praktikum Pemrograman Lanjut SI 6



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

            earnings += 100000;
        }
        return earnings;
    }

```

```

    private boolean isBirthday() {
        LocalDate today = LocalDate.now();
        LocalDate tanggalLahir = getTanggalLahir();
        return today.getMonthValue() == tanggalLahir.getMonthValue() &&
            today.getDayOfMonth() == tanggalLahir.getDayOfMonth();
    }
}

```

```

    public class BasePlusCommissionEmployee extends CommissionEmployee {
        public BasePlusCommissionEmployee(String name, String noKTP, LocalDate tanggalLahir,
            double sales, double rate, double salary) {
            super(name, noKTP, tanggalLahir, sales, rate);
        }

```

```

        public double earnings() {
            double earnings = getBaseSalary() + super.earnings();
            if (isBirthday()) {
                earnings += 100000;
            }
            return earnings;
        }
    }

```

```
}
```

```
private boolean isBirthday() {  
    LocalDate today = LocalDate.now();  
    LocalDate tanggalLahir = getTanggalLahir();  
    return today
```

Laporan Praktikum Pemrograman Lanjut SI 7



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

Kelas turunan Employee yang belum dimodifikasi adalah SalariedEmployee. Berikut adalah modifikasi kelas SalariedEmployee:

```
import java.time.LocalDate;
```

```
public class SalariedEmployee extends Employee {  
    private double monthlySalary; // gaji/bulan
```

```
    public SalariedEmployee(String name, String noKTP, LocalDate tanggalLahir, double  
salary) {  
        super(name, noKTP, tanggalLahir);  
        setMonthlySalary(salary);  
    }
```

```
    public void setMonthlySalary(double salary) {  
        monthlySalary = salary;  
    }
```

```
    public double getMonthlySalary() {  
        return monthlySalary;  
    }
```

```
    public double earnings() {
```



```

double earnings = getMonthlySalary();
if (isBirthday()) {
    earnings += 100000;
}
return earnings;
}

private boolean isBirthday() {

```

Laporan Praktikum Pemrograman Lanjut SI 8



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

    LocalDate today = LocalDate.now();
    LocalDate tanggalLahir = getTanggalLahir();
    return today.getMonthValue() == tanggalLahir.getMonthValue() &&
today.getDayOfMonth() == tanggalLahir.getDayOfMonth();
}

public String toString() {
    return String.format("Salaried employee: %s\nMonthly salary: $%.2f",
super.toString(), getMonthlySalary());
}
}

```

Selanjutnya, buatlah kelas uji untuk menguji program yang telah dimodifikasi: `import java.time.LocalDate;`

```

public class Main {
    public static void main(String[] args) {
        LocalDate tanggalLahirSalaried = LocalDate.of(1990, 5, 20);
        LocalDate tanggalLahirHourly = LocalDate.of(1992, 8, 15);
        LocalDate tanggalLahirCommission = LocalDate.of(1988, 10, 5);
        LocalDate tanggalLahirBasePlusCommission = LocalDate.of(1995, 12, 25);

```

```
SalariedEmployee salariedEmployee = new SalariedEmployee("Daniel", "135",  
tanggalLahirSalaried, 800.00);
```

```
HourlyEmployee hourlyEmployee = new HourlyEmployee("Karina", "234",  
tanggalLahirHourly, 16.75, 40);
```

```
CommissionEmployee commissionEmployee = new  
CommissionEmployee("Keanu", "145", tanggalLahirCommission, 10000, .06);
```

```
BasePlusCommissionEmployee basePlusCommissionEmployee = new  
BasePlusCommissionEmployee("Bondan", "234", tanggalLahirBasePlusCommission,  
5000, .04, 300);
```

Laporan Praktikum Pemrograman Lanjut SI 9



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
Employee[] employees = new Employee[4];  
employees[0] = salariedEmployee;  
employees[1] = hourlyEmployee;  
employees[2] = commissionEmployee;  
employees[3] = basePlusCommissionEmployee;  
  
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()); }  
}  
}
```

Dalam kelas Main, objek-objek dibuat dengan tanggal lahir yang sesuai. Kemudian, pengujian dilakukan dengan mencetak informasi karyawan dan pendapatan mereka. Jika bulan ini adalah bulan ulang tahun karyawan, gaji akan ditambahkan sebesar 100.000. Berikut adalah contoh implementasinya:

```
import java.time.LocalDate;

public class Main {
    public static void main(String[] args) {
        LocalDate tanggalLahirSalaried = LocalDate.of(1990, 5, 20);
        LocalDate tanggalLahirHourly = LocalDate.of(1992, 8, 15);
```

Laporan Praktikum Pemrograman Lanjut SI 10



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
        LocalDate tanggalLahirCommission = LocalDate.of(1988, 10, 5);
        LocalDate tanggalLahirBasePlusCommission = LocalDate.of(1995, 12, 25);

        SalariedEmployee salariedEmployee = new SalariedEmployee("Daniel", "135",
tanggalLahirSalaried, 800.00);

        HourlyEmployee hourlyEmployee = new HourlyEmployee("Karina", "234",
tanggalLahirHourly, 16.75, 40);

        CommissionEmployee commissionEmployee = new
CommissionEmployee("Keanu", "145", tanggalLahirCommission, 10000, .06);

        BasePlusCommissionEmployee basePlusCommissionEmployee = new
BasePlusCommissionEmployee("Bondan", "234", tanggalLahirBasePlusCommission,
5000, .04, 300);

        // Set tanggal lahir baru
        salariedEmployee.setTanggalLahir(LocalDate.of(1990, 5, 20));
        hourlyEmployee.setTanggalLahir(LocalDate.of(1992, 8, 15));
        commissionEmployee.setTanggalLahir(LocalDate.of(1988, 10, 5));
        basePlusCommissionEmployee.setTanggalLahir(LocalDate.of(1995, 12, 25));
```

```

Employee[] employees = new Employee[4];
employees[0] = salariedEmployee;
employees[1] = hourlyEmployee;
employees[2] = commissionEmployee;
employees[3] = basePlusCommissionEmployee;

for (Employee employee : employees) {
    System.out.println(employee);
    System.out.printf("Pendapatan: $%.2f\n", employee.earnings());
    System.out.println();
}
}
}

```

Laporan Praktikum Pemrograman Lanjut SI 11



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

Dalam contoh ini, objek-objek Employee dibuat dengan tanggal lahir yang sesuai. Kemudian, tanggal lahir karyawan diperbarui menggunakan metode `setTanggalLahir()`. Kemudian, dalam loop `for`, informasi karyawan dan pendapatan mereka dicetak. Jika bulan ini adalah bulan ulang tahun karyawan, gaji akan ditambahkan sebesar 100.000, sesuai dengan logika yang sudah diimplementasikan dalam metode `earnings()` di setiap kelas turunan Employee.

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 diproduksi per minggu dikalikan upah per barangnya.
 - a. Analisis dan jelaskan proses modifikasi program diatas (dimulai dari pemilihan jenis class, perancangan class, dan penempatan class)
 - b. Implementasi hasil analisis tersebut ke dalam program dan buat kelas uji dengan minimal 4 objek yang dibentuk.

Jawab:

a. Analisis dan Jelaskan Proses Modifikasi Program di Atas:

Untuk memenuhi kriteria baru penggajian berdasarkan banyaknya barang yang diproduksi, kita dapat membuat kelas turunan baru dari kelas Employee yang sesuai dengan kriteria tersebut. Karena kriteria tersebut berkaitan dengan produksi barang, maka kita dapat membuat kelas turunan baru bernama ProductionEmployee.

Proses modifikasi program akan melibatkan beberapa langkah:

Perancangan Kelas ProductionEmployee:

Kelas ProductionEmployee akan menjadi turunan dari kelas Employee. Atribut yang diperlukan adalah upah per barang dan jumlah barang yang diproduksi per minggu. Diperlukan metode untuk mengakses dan memperbarui atribut tersebut.

Laporan Praktikum Pemrograman Lanjut SI 12



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

Metode earnings() akan diimplementasikan untuk menghitung pendapatan karyawan berdasarkan produksi barang.

Penempatan Kelas ProductionEmployee:

Kelas ProductionEmployee akan ditempatkan dalam package yang sama dengan kelas-kelas lainnya (misalnya TP7).

Modifikasi dilakukan pada file TP7.java dengan menambahkan kelas ProductionEmployee.

b. Implementasi Hasil Analisis ke dalam Program dan Pembuatan Kelas Uji dengan Minimal 4 Objek:

Berikut adalah implementasi hasil analisis ke dalam program dengan pembuatan kelas

uji: package TP7;

```
public class ProductionEmployee extends Employee {  
    private double wagePerItem; // upah per barang
```

```

private int quantityProduced; // jumlah barang yang diproduksi per minggu

public ProductionEmployee(String name, String noKTP, double wagePerItem, int
quantityProduced) {
    super(name, noKTP);
    setWagePerItem(wagePerItem);
    setQuantityProduced(quantityProduced);
}

public void setWagePerItem(double wagePerItem) {
    this.wagePerItem = wagePerItem;
}

public double getWagePerItem() {
    return wagePerItem;
}

```

Laporan Praktikum Pemrograman Lanjut SI 13



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

}

public void setQuantityProduced(int quantityProduced) {
    this.quantityProduced = quantityProduced;
}

public int getQuantityProduced() {
    return quantityProduced;
}

public double earnings() {
    return getWagePerItem() * getQuantityProduced();
}

public String toString() {
    return String.format("Production employee: " + super.toString() + "\nwage per item:

```

```

" + getWagePerItem() +
        "\nquantity produced: " + getQuantityProduced());
    }
}

package TP7;

public class Main {
    public static void main(String[] args) {
        LocalDate tanggalLahirSalaried = LocalDate.of(1990, 5, 20);
        LocalDate tanggalLahirHourly = LocalDate.of(1992, 8, 15);
        LocalDate tanggalLahirCommission = LocalDate.of(1988, 10, 5);
        LocalDate tanggalLahirBasePlusCommission = LocalDate.of(1995, 12, 25);
        LocalDate tanggalLahirProduction = LocalDate.of(1993, 4, 10);
    }
}

```

Laporan Praktikum Pemrograman Lanjut SI 14



LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

        SalariedEmployee salariedEmployee = new SalariedEmployee("Daniel", "135",
tanggalLahirSalaried, 800.00);

        HourlyEmployee hourlyEmployee = new HourlyEmployee("Karina", "234",
tanggalLahirHourly, 16.75, 40);

        CommissionEmployee commissionEmployee = new
CommissionEmployee("Keanu", "145", tanggalLahirCommission, 10000, .06);

        BasePlusCommissionEmployee basePlusCommissionEmployee = new
BasePlusCommissionEmployee("Bondan", "234", tanggalLahirBasePlusCommission,
5000, .04, 300);

        ProductionEmployee productionEmployee = new ProductionEmployee("Maya",
"456", tanggalLahirProduction, 10.5, 1000);

        System.out.println("Employees diproses secara terpisah:\n");

        System.out.printf("%s\n%s: $%,.2f\n\n", salariedEmployee, "pendapatan",
salariedEmployee.earnings());
    }
}

```

```

        System.out.printf("%s\n%s: $%,.2f\n\n", hourlyEmployee, "pendapatan",
hourlyEmployee.earnings());

        System.out.printf("%s\n%s: $%,.2f\n\n", commissionEmployee, "pendapatan",
commissionEmployee.earnings());

        System.out.printf("%s\n%s: $%,.2f\n\n", basePlusCommissionEmployee,
"pendapatan", basePlusCommissionEmployee.earnings());

        System.out.printf("%s\n%s: $%,.2f\n\n", productionEmployee, "pendapatan",
productionEmployee.earnings());

```

```

Employee[] employees = new Employee[5];
employees[0] = salariedEmployee;
employees[1] = hourlyEmployee;
employees[2] = commissionEmployee;
employees[3] = basePlusCommissionEmployee;
employees[4] = productionEmployee;

```

```

System.out.println("Employees diproses secara polimorfisme:\n");

```

Laporan Praktikum Pemrograman Lanjut SI 15

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

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());
    }

    if (currentEmployee instanceof ProductionEmployee) {
        ProductionEmployee employee = (ProductionEmployee) currentEmployee;
        if (employee.isBirthday()) {
            employee.addBonus(100000);

```



```

        System.out.println("Penambahan bonus $100.000 karena hari ulang
tahun\n"); }
    }

    System.out.printf("pendapatan: $%,.2f\n\n",
currentEmployee.earnings()); }
    }
}

```

Dalam contoh di atas, kelas `ProductionEmployee` ditambahkan dan objek `Production`

3. Tugas Praktikum

2.1 Source code

```

3
    package Prak7;
4
5
    public abstract class Kue
6
    {
7
        private String nama;
        private double harga;

```

Laporan Praktikum Pemrograman Lanjut SI 16

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

8	18
9	19
10	20
11	21
12	22
13	23
14	24
15	25
16	26
17	27 } 28

```

public Kue(String nama, {
double harga) { this.nama=return this.harga;
= nama; }
this.harga = harga;
}
public abstract double
hitungHarga();

public String getNama()
{
return this.nama;
}
public String toString() {
return "" + nama + "(
Harga " + (int) harga; }
public double getHarga()

```

```

29 package Prak7;
30
31 public class KueJadi extends Kue {
32
33     private double jumlah;
34
35     public KueJadi(String nama, double harga, double jumlah)
36     {
37         super(nama, harga);
38         this.jumlah = jumlah;
39     }
40     public double getJumlah() {
41         return this.jumlah;
42     }
43 }
44

```

Laporan Praktikum Pemrograman Lanjut SI 17

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

40

41

42

43

44

```

45         return super.getHarga() *
46             this.jumlah * 2;
47     }
48
49     @Override
50     public String toString() {
51 } 52         return super.toString() + " per " +
           (int) jumlah + " buah ); }
53     }

```

```

public double hitungHarga() {

```

```

53 package Prak7;
54
55 public class KuePesanan extends Kue {
56     private double berat;
57
58     public KuePesanan(String nama, double harga, double berat)
59     {
60         super(nama, harga);
61         this.berat = berat;
62     }
63
64     public double getBerat() {
65         return this.berat;
66     }
67
68     @Override
69     public double hitungHarga() {

```

```

69         return this.berat * super.getHarga();
70     }
71

```

Laporan Praktikum Pemrograman Lanjut SI 18

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

72 @Override
73 public String toString() {
74     return super.toString() + " per kg ";
75 }
76 }
77
78 package Prak7;
79
80 public class Main {
81     public static void main(String[] args) {
82         Kue[] kueArray = new Kue[20];
83         kueArray[0] = new KuePesanan("Kue Bolu ", 10000, 0.5);
84         kueArray[1] = new KuePesanan("Kue Brownies ", 12000, 0.4);
85         kueArray[2] = new KuePesanan("Kue Lapis ", 15000, 0.8);
86         kueArray[3] = new KuePesanan("Kue Tart ", 20000, 1.2);
87         kueArray[4] = new KuePesanan("Kue Sus ", 8000, 0.3);
88         kueArray[5] = new KueJadi("Kue Donat ", 5000, 10);
89         kueArray[6] = new KueJadi("Kue Pie ", 15000, 5);
90         kueArray[7] = new KueJadi("Kue Croissant ", 12000, 8);
91         kueArray[8] = new KueJadi("Kue Roll ", 10000, 12);
92         kueArray[9] = new KueJadi("Kue Cheese Cake ", 25000, 3);
93         kueArray[10] = new KueJadi("Kue Coklat", 10000, 3);
94         kueArray[11] = new KueJadi("Kue Strawberry", 5000, 1);
95         kueArray[12] = new KueJadi("Kue Lemon", 7000, 2);
96         kueArray[13] = new KueJadi("Kue Cream", 12000, 4);
97         kueArray[14] = new KueJadi("Kue Susu", 10000, 4);
98         kueArray[15] = new KuePesanan("Kue Nastar", 10000, 1);
99         kueArray[16] = new KuePesanan("Kue Geprek", 2000, 0.5);
100

```

```
kueArray[17] = new KuePesanan("Kue Manis", 10000, 0.7); 101
kueArray[18] = new KuePesanan("Kue Cantik", 8000, 1); 102
kueArray[19] = new KuePesanan("Kue Freya", 2000, 0.8); 103
```

Laporan Praktikum Pemrograman Lanjut SI 19

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
104// Tampilkan semua kue dan jenisnya
105for (Kue kue : kueArray) {
106 System.out.println(kue.toString());
107 if (kue instanceof KuePesanan) {
108 System.out.println("Jenis Kue: Kue Pesanan");
109 } else {
110 System.out.println("Jenis Kue: Kue Jadi");
111 }
112}
113
114// Hitung total harga dari semua jenis kue
115double totalHarga = 0;
116for (Kue kue : kueArray) {
117 totalHarga += kue.hitungHarga();
118}
119System.out.println("\n|=====
    = ===");
120System.out.println("Total Harga Semua Jenis Kue : " + (int)totalHarga);
121
122// Hitung total harga dan total berat dari KuePesanan
123double totalHargaKuePesanan = 0;
124double totalBerat = 0;
125for (Kue kue : kueArray) {
126 if (kue instanceof KuePesanan) {
127 totalHargaKuePesanan += kue.hitungHarga();
128 totalBerat += ((KuePesanan) kue).getBerat();
129 }
130}
```

```

131 System.out.println("Total Harga      Kue Pesanan 133
    Kue Pesanan      : " +
(int)totalHargaKuePesanan);
132 System.out.println("Total Berat      : " + totalBerat +"kg");

```

Laporan Praktikum Pemrograman Lanjut SI 20

LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```

134 // Hitung total harga dan total jumlah dari KueJadi
135 double totalHargaKueJadi = 0;
136 double totalJumlah = 0;
137 for (Kue kue : kueArray) {
138     if (kue instanceof KueJadi) {
139         totalHargaKueJadi += kue.hitungHarga();
140         totalJumlah += ((KueJadi) kue).getJumlah();
141     }
142 }
143 System.out.println("Total Harga Kue Jadi : " + (int)totalHargaKueJadi);
144
    System.out.println("Total Jumlah Kue Jadi : " + (int)totalJumlah +" Kue");
145
146 // Tampilkan informasi kue dengan harga (harga akhir) terbesar
147 Kue kueTerbesar = kueArray[0];
148 for (Kue kue : kueArray) {
149     if (kue.hitungHarga() > kueTerbesar.hitungHarga()) {
150         kueTerbesar = kue;
151     }
152 }
153
    System.out.println("Kue dengan Harga Terbesar : " +
        kueTerbesar.toString());
154
    System.out.println("=====

```

```
==|");
```

```
155 }
```

156.1 Screenshot hasil

```
156}
```

Laporan Praktikum Pemrograman Lanjut SI 21

**LABORATORIUM PEMBELAJARAN
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**



