

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : POLYMORFISME
 NAMA : Dani Adrian
 NIM : 225150201111009
 ASISTEN : Tengku Muhammad Rafi Rahardiansyah
 Muhammad Bin Djafar Almasyhur
 TGL PRAKTIKUM : 29 Maret 2023

Data dan Hasil Percobaan

Pertanyaan

1. Ketikkan kode ini.

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

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

Source Code

1	Tulis source code di sini pake courier new 12
---	---

Output

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Semester 2\OOP\Praktikum\Praktikum Modul 7\Teori> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\work
spaceStorage\3442933456500b1b16cbb41c382a549a\redhat.java\jdt_ws\Teori_e689e880\bin' 'Main'
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot instantiate the type Employee

    at Main.main(Main.java:3)
PS D:\Semester 2\OOP\Praktikum\Praktikum Modul 7\Teori>
  
```

The inset window shows a text editor titled 'Dani Adrian.txt' with the following content:

```

Dani Adrian
225150201111009
  
```

At the bottom of the inset window, it shows 'Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8'.

Penjelasan

Terjadi error karena Class Abstract tidak bisa di instansiasi.

2. Jalankan program dengan main sebagai berikut.

Main.java	
1.	public class Main {
2.	public static void main(String[] args) {
3.	SalariedEmployee salariedEmployee = new SalariedEmployee("Daniel", "135", 800.00);
4.	HourlyEmployee hourlyEmployee = new HourlyEmployee("Karina", "234", 16.75, 40);
5.	CommissionEmployee commissionEmployee = new CommissionEmployee("Keanu", "145", 10000, .06);
6.	BasePlusCommissionEmployee basePlusCommissionEmployee = new BasePlusCommissionEmployee("Bondan", "234", 5000, .04, 300);
7.	System.out.println("Employees diproses secara terpisah:\n");
8.	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());
9.	System.out.printf("%s\n%s: \$%,.2f\n\n", commissionEmployee, "pendapatan: ", commissionEmployee.earnings()); System.out.printf("%s\n%s: \$%,.2f\n\n", basePlusCommissionEmployee, "earned", basePlusCommissionEmployee.earnings());
10.	Employee[] employees = new Employee[4];
11.	employees[0] = salariedEmployee;
12.	employees[1] = hourlyEmployee;
13.	employees[2] = commissionEmployee;
14.	employees[3] = basePlusCommissionEmployee;
15.	System.out.println("Employees diproses secara polimorfisme:\n");
16.	for (Employee currentEmployee : employees) {
17.	System.out.println(currentEmployee);
18.	if (currentEmployee instanceof BasePlusCommissionEmployee) {
19.	BasePlusCommissionEmployee employee = (BasePlusCommissionEmployee) currentEmployee;
20.	employee.setBaseSalary(1.10 * employee.getBaseSalary());
21.	System.out.printf("Gaji pokok setelah dinaikkan 10% : 45 \$%,.2f\n", employee.getBaseSalary());
22.	}

23.	<code>System.out.printf("pendapatan: \$%,.2f\n\n",</code>
24.	<code>currentEmployee.earnings());</code>
25.	<code>for (int j = 0; j < employees.length; j++) {</code>
26.	<code>System.out.printf("Employee %d = %s\n", j,</code>
27.	<code>employees[j].getClass().getName());</code>
28.	<code>}</code>
29.	<code>}</code>

Source Code

1	Tulis source code di sini pake courier new 12
---	---

Ouput

The screenshot shows a Windows PowerShell terminal window with the following output:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspac
e\Storage\cfb8827f531be909cae84fd56f5a819\redhat.java\jdt_ws\praktikumketujuh_ea295537\bin' 'Main'
Employees      diproses      secara terpisah:

Salaried employee: Daniel
No.KTP:135
weekly salary: 800.0
pendapatan:      : $800.00

Hourly employee: Karina
No.KTP:234
hourly wage16.75
hours worked:40.0
pendapatan:      : $670.00

Commision employee: Keanu
No.KTP:145
gross sales: 10000.0
commission rate0.06
pendapatan:      : $600.00

Base-Salaried Commision employee: Bondan
No.KTP:234
gross sales: 5000.0
commission rate0.04
base salary 300.0
earned: $500.00

Employees      diproses      secara polimorfisme:

Salaried employee: Daniel
No.KTP:135
weekly salary: 800.0
pendapatan:      : $800.00

Hourly employee: Karina
No.KTP:234
hourly wage16.75
hours worked:40.0
pendapatan:      : $670.00

```

Overlaid on the terminal is a text editor window titled "Dani Adrian.txt" containing the text:

```

Dani Adrian
2251502011111009

```

The text editor window also shows status information at the bottom: "Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8".

The screenshot shows a Java IDE with a dark theme. The main window displays the output of a Java program. The output includes details for three employees: Keanu (Commission employee), Bondan (Base-Salaried Commission employee), and Daniel (Salaried employee). It also shows the output of a casting operation: 'Employees diproses secara polimorfisme:'. A text editor window titled 'Dani Adrian.txt' is open in the foreground, showing the name 'Dani Adrian' and the ID '225150201111009'. The IDE status bar at the bottom indicates 'Ln 30, Col 9', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

```

Commission employee: Keanu
No.KTP:145
gross sales: 10000.0
commission rate0.06
pendapatan: : $600.00

Base-Salaried Commission employee: Bondan
No.KTP:234
gross sales: 5000.0
commission rate0.04
base salary 300.0
earned: $500.00

Employees diproses secara polimorfisme:

Salaried employee: Daniel
No.KTP:135
weekly salary: 800.0
pendapatan: $800.00

Hourly employee: Karina
No.KTP:234
hourly wage16.75
hours worked:40.0
pendapatan: $670.00

Commission employee: Keanu
No.KTP:145
gross sales: 10000.0
commission rate0.06
pendapatan: $600.00

Base-Salaried Commission employee: Bondan
No.KTP:234
gross sales: 5000.0
commission rate0.04
base salary 300.0
Gaji pokok setelah dinaikkan 10% : 45 $330.00
pendapatan: $530.00

Employee 0 = SalariedEmployee
Employee 1 = HourlyEmployee
Employee 2 = CommissionEmployee
Employee 3 = BasePlusCommissionEmployee
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh>
  
```

Penjelasan

Dalam kode tersebut, polimorfisme diterapkan pada karyawan Base Salaried Commission. Jika casting digunakan, hasil keluarannya akan bervariasi.

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

Source code

Main.java	
1.	public class Main {
2.	public static void main(String[] args) {
3.	FullTimeEmployee employee = new FullTimeEmployee("John", "1234567890", 5000.0);
4.	System.out.println(employee.toString());
5.	System.out.println("Earnings: \$" + employee.earnings());
6.	}
7.	}

FullTimeEmployee.java	
1.	public class FullTimeEmployee extends Employee{
2.	private double monthlySalary;

3.	
4.	<code>public FullTimeEmployee(String name, String noKTP,</code>
5.	<code>double monthlySalary) {</code>
6.	<code>super(name, noKTP);</code>
7.	<code>this.monthlySalary = monthlySalary;</code>
8.	<code>}</code>
9.	<code>@Override</code>
10.	<code>public double earnings() {</code>
11.	<code>return monthlySalary;</code>
12.	<code>}</code>
13.	<code>}</code>

Output

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\cfb8827f5531be909cae84fd56f5a819\redhat.java\jdt_ws\praktikumketujuh_ea295537\bin' 'Main'
John
No.KTP:1234567890
Earnings: $5000.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh>

```

Dani Adrian
225150201111009

Penjelasan

Kelas FullTimeEmployee adalah turunan dari kelas abstrak Employee. Hal ini disebabkan karena kelas Employee tidak dapat diinstansiasi secara langsung, melainkan harus melalui instansiasi subclassnya.

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.

Source Code

Main.java	
1.	<code>public class Main{</code>
2.	<code>public static void main(String[] args) {</code>
3.	<code>SalariedEmployee salariedEmployee = new</code>
	<code>SalariedEmployee("Daniel", "135", 2000, 4, 6, 800.00);</code>

4.	HourlyEmployee hourlyEmployee = new HourlyEmployee("Karina", "234",2000, 5, 6, 16.75, 40);
5.	CommissionEmployee commissionEmployee = new CommissionEmployee("Keanu", "145",2000, 6, 6, 10000, .06);
6.	BasePlusCommissionEmployee basePlusCommissionEmployee = new BasePlusCommissionEmployee("Bondan", "234",2000, 7, 6, 5000, .04, 300);
7.	
8.	Employee[] employees = new Employee[4];
9.	employees[0] = salariedEmployee;
10.	employees[1] = hourlyEmployee;
11.	employees[2] = commissionEmployee;
12.	employees[3] = basePlusCommissionEmployee;
13.	
14.	System.out.println("Tanggal hari ini : 2023/04/11\n");
15.	
16.	System.out.println("Employees diproses secara polimorfisme:\n");
17.	
18.	for (Employee currentEmployee : employees) {
19.	System.out.println(currentEmployee);
20.	if (currentEmployee instanceof BasePlusCommissionEmployee){
21.	BasePlusCommissionEmployee employee = (BasePlusCommissionEmployee) currentEmployee;
22.	employee.setBaseSalary(1.10 * employee.getBaseSalary());
23.	System.out.printf("Gaji pokok setelah dinaikkan 10% : 45 \$%,.2f\n", employee.getBaseSalary());}
24.	System.out.printf("pendapatan: \$%,.2f\n\n", currentEmployee.earnings());}
25.	
26.	for (int j = 0; j < employees.length; j++) {
27.	System.out.printf("Employee %d = %s\n", j, employees[j].getClass().getName());
28.	}
29.	}
30.	}

Employee.java

1.	import java.time.LocalDate;
----	-----------------------------

2.	
3.	<code>public abstract class Employee</code>
4.	<code>{</code>
5.	<code> private String name;</code>
6.	<code> private String noKTP;</code>
7.	<code> private LocalDate tanggalLahir;</code>
8.	
9.	<code> public Employee(String name, String noKTP, int year,</code> <code>int month, int day){</code>
10.	<code> this.name = name;</code>
11.	<code> this.noKTP = noKTP;</code>
12.	<code> this.tanggalLahir = LocalDate.of(year, month,</code> <code>day);</code>
13.	<code> }</code>
14.	<code> public String getName(){</code>
15.	<code> return name;</code>
16.	<code> }</code>
17.	<code> public String getNoKTP(){</code>
18.	<code> return noKTP;</code>
19.	<code> }</code>
20.	<code> public LocalDate getTanggalLahir()</code>
21.	<code> {</code>
22.	<code> return this.tanggalLahir;</code>
23.	<code> }</code>
24.	<code> public void setTanggalLahir(LocalDate p)</code>
25.	<code> {</code>
26.	<code> this.tanggalLahir = p;</code>
27.	<code> }</code>
28.	<code> public int getBonus(int i)</code>
29.	<code> {</code>
30.	<code> return tanggalLahir.getMonthValue() == i ? 100000</code> <code>: 0;</code>
31.	<code> }</code>
32.	<code> public String toString(){</code>
33.	<code> return String.format(" "+getName()+"\nNo.</code> <code>: "+getNoKTP() + "\nBd. : " + getTanggalLahir());</code>
34.	<code> }</code>
35.	<code> public abstract double earnings();//pendapatan</code>
36.	<code>}</code>

SalariedEmployee.java

1.	<code>public class SalariedEmployee extends Employee{</code>
----	--

2.	private double weeklySalary; //gaji/minggu
3.	public SalariedEmployee(String name, String noKTP, int year, int month, int day, double salary) {
4.	super(name, noKTP, year, month, day); setWeeklySalary(salary);
5.	}
6.	public void setWeeklySalary(double salary) {
7.	weeklySalary = salary;
8.	}
9.	public double getWeeklySalary() {
10.	return weeklySalary;
11.	}
12.	public double earnings() {
13.	return (getWeeklySalary()*4) + getBonus(4);
14.	}
15.	public String toString() {
16.	return String.format("Salaried employee: " + super.toString() + "\nweekly salary:" + getWeeklySalary());
17.	}
18.	}

HourlyEmployee.java	
1.	public class HourlyEmployee extends Employee{
2.	private double wage; //upah per jam
3.	private double hours; //jumlah jam tiap minggu
4.	
5.	public HourlyEmployee(String name, String noKTP, int year, int month, int day, double hourlyWage, double hoursWorked) {
6.	super(name, noKTP, year, month, day);
7.	setWage(hourlyWage);
8.	setHours(hoursWorked);
9.	}
10.	
11.	public void setWage(double hourlyWage){
12.	wage = hourlyWage;
13.	}
14.	
15.	public double getWage(){
16.	return wage;
17.	}

18.	
19.	public void setHours(double hoursWorked) {
20.	hours = hoursWorked;
21.	}
22.	
23.	public double getHours() {
24.	return hours;
25.	}
26.	
27.	public double earnings() {
28.	if(getHours() <= 40)
29.	return getWage() * getHours();
30.	else
31.	return 40 * getWage() + (getHours()-40) * getWage() * 1.5 + getBonus(4);
32.	}
33.	
34.	public String toString() {
35.	return String.format("Hourly employee: "+super.toString() + "\nhourly wage"+getWage()+"\nhours worked: "+getHours());
36.	}
37.	}

CommissionEmployee.java	
1.	public class CommissionEmployee extends Employee {
2.	private double grossSales;//penjualan per minggu
3.	private double commissionRate;//komisi
4.	
5.	public CommissionEmployee(String name, String noKTP, int year, int month, int day, double sales, double rate) {
6.	super(name, noKTP, year, month, day);
7.	setGrossSales(sales);
8.	setCommissionRate(rate);
9.	}
10.	
11.	public void setGrossSales(double sales) {
12.	grossSales = sales;
13.	}
14.	
15.	public double getGrossSales() {
16.	return grossSales;

17.	}
18.	
19.	public void setCommissionRate(double rate){
20.	commissionRate = rate;
21.	}
22.	
23.	public double getCommissionRate(){
24.	return commissionRate;
25.	}
26.	
27.	public double earnings(){
28.	return getCommissionRate()*getGrossSales() + getBonus(4);
29.	}
30.	
31.	public String toString(){
32.	return String.format("Commision employee: "+super.toString()+"\ngross sales: "+getGrossSales()+"\ncommission rate"+getCommissionRate());
33.	}
34.	}

BasePlusCommissionEmployee.java	
1.	public class BasePlusCommissionEmployee extends CommissionEmployee {
2.	
3.	private double baseSalary;//gaji pokok tiap minggu
4.	
5.	public BasePlusCommissionEmployee(String name, String noKTP, int year, int month, int day, double sales, double rate, double salary) {
6.	super(name, noKTP, year, month, day, sales, rate);
7.	setBaseSalary(salary);
8.	}
9.	
10.	public void setBaseSalary(double salary) {
11.	baseSalary = salary;
12.	}
13.	
14.	public double getBaseSalary() {

15.	<code>return baseSalary;</code>
16.	<code>}</code>
17.	
18.	<code>public double earnings() {</code>
19.	<code>return getBaseSalary() + super.earnings() +</code> <code>getBonus(4);</code>
20.	<code>}</code>
21.	
22.	<code>public String toString() {</code>
23.	<code>return String.format("Base-Salaried" +</code> <code>super.toString() + "\nbase salary " + getBaseSalary());</code>
24.	<code>}</code>
25.	<code>}</code>

Output

```

-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\cfb8827f5531be909cae84fd56f5a819\redhat.java\jdt_ws\praktikumketujuh_ea295537\bin' 'Main'
Tanggal hari ini : 2023/04/11

Employees diproses secara polimorfisme:

Salaried      employee:      Daniel
No. :135
Bd. :2000-04-06
weekly salary:800.0
pendapatan: $103,200.00

Hourly employee:      Karina
No. :234
Bd. :2000-05-06
hourly wage16.75
hours worked: 40.0
pendapatan: $670.00

Commision employee:      Keanu
No. :145
Bd. :2000-06-06
gross sales: 10000.0
commission rate0.06
pendapatan: $600.00

Base-SalariedCommision employee:      Bondan
No. :234
Bd. :2000-07-06
gross sales: 5000.0
commission rate0.04
base salary 300.0
gaji pokok setelah dinaikkan 10% : 45 $330.00
pendapatan: $530.00

Employee 0 = SalariedEmployee
Employee 1 = HourlyEmployee
Employee 2 = CommisionEmployee
Employee 3 = BasePlusCommisionEmployee
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh>

```

Penjelasan

Hal yang telah dipenuhi diantaranya sebagai berikut :

1. Menambahkan atribut tanggal lahir di kelas Employee beserta accessor dan mutator-nya.
2. Memodifikasi program agar sesuai dengan penambahan atribut dan method tersebut.
3. Mengasumsikan bahwa gaji yang diterima adalah per bulan.
4. Membuat kelas uji untuk menguji program yang sudah dimodifikasi.

5. Membuat objek dari semua kelas (`salariedEmployee`, `hourlyEmployee`, `commissionEmployee`, `basePlusCommissionEmployee`) dan menghitung gajinya secara polimorfisme.
6. Menambahkan gaji sebesar 100.000 jika bulan ini adalah bulan ulang tahun karyawan.

Dalam hal ini, penambahan atribut tanggal lahir dan method pendukungnya berguna untuk menghitung apakah saat ini adalah bulan ulang tahun karyawan. Jika ya, maka gaji karyawan akan ditambah sebesar 100.000.

Setelah itu, dibuatlah kelas uji untuk menguji program yang telah dimodifikasi. Kemudian, dibuat objek dari semua kelas dan menghitung gajinya secara polimorfisme dengan menggunakan metode yang telah diimplementasikan sebelumnya.

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.

Source Code

Main.java	
1.	<code>public class Main {</code>
2.	<code> public static void main(String[] args) {</code>
3.	<code> PerBarangEmployee pb1 = new</code> <code> PerBarangEmployee("Adit", "1231", 2000, 1, 1, 20, 40);</code>
4.	<code> PerBarangEmployee pb2 = new</code> <code> PerBarangEmployee("Arit", "1232", 2000, 2, 1, 30, 40);</code>
5.	<code> PerBarangEmployee pb3 = new</code> <code> PerBarangEmployee("Atit", "1233", 2000, 3, 1, 40, 40);</code>
6.	<code> PerBarangEmployee pb4 = new</code> <code> PerBarangEmployee("Apit", "1234", 2000, 4, 1, 50, 40);</code>
7.	<code> System.out.println(pb1 + "\nEarning: " +</code> <code> pb1.earnings());</code>
8.	<code> System.out.println(pb2 + "\nEarning: " +</code> <code> pb2.earnings());</code>
9.	<code> System.out.println(pb3 + "\nEarning: " +</code> <code> pb3.earnings());</code>
10.	<code> System.out.println(pb4 + "\nEarning: " +</code> <code> pb4.earnings());</code>
11.	<code> }</code>
12.	<code>}</code>

PerBarangEmployee.java	
1.	public class PerBarangEmployee extends Employee{
2.	private int jumlahBarang;
3.	private int upahBarang;
4.	
5.	public PerBarangEmployee(String name, String noKTP, int year, int month, int day, int jumlahBarang, int hargaBarang){
6.	super(name, noKTP, year, month, day);
7.	setJumlahBarang(jumlahBarang);
8.	setUpahBarang(hargaBarang);
9.	}
10.	
11.	@Override
12.	public double earnings(){
13.	return getUpahBarang()*getJumlahBarang();
14.	}
15.	
16.	@Override
17.	public String toString(){
18.	return String.format("Commision employee: "+super.toString()+"\nJumlah Barang: "+getJumlahBarang()+"\nHarga Barang: "+getUpahBarang());
19.	}
20.	
21.	public int getJumlahBarang() {
22.	return jumlahBarang;
23.	}
24.	
25.	public void setJumlahBarang(int jumlahBarang) {
26.	this.jumlahBarang = jumlahBarang;
27.	}
28.	
29.	public int getUpahBarang() {
30.	return upahBarang;
31.	}
32.	
33.	public void setUpahBarang(int hargaBarang) {
34.	this.upahBarang = hargaBarang;
35.	}
36.	}

Output

The screenshot shows a Java IDE with a command prompt window and a text editor window. The command prompt displays the output of a Java program, showing four instances of a class with attributes like No., Bd., Jumlah Barang, Harga Barang, and Earning. The text editor window shows the content of a file named 'Dani Adrian.txt', which contains the name 'Dani Adrian' and a long alphanumeric string.

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh> c::; cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\cfb8827f5531be909cae84fd56f5a819\redhat.java\jdt_ws\praktikumketujuh_ea295537\bin' 'Main'
Commision employee:    Adit
No. :1231
Bd. :2000-01-01
Jumlah Barang: 20
Harga Barang: 40
Earning: 800.0
Commision employee:    Arit
No. :1232
Bd. :2000-02-01
Jumlah Barang: 30
Harga Barang: 40
Earning: 1200.0
Commision employee:    Atit
No. :1233
Bd. :2000-03-01
Jumlah Barang: 40
Harga Barang: 40
Earning: 1600.0
Commision employee:    Apit
No. :1234
Bd. :2000-04-01
Jumlah Barang: 50
Harga Barang: 40
Earning: 2000.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuh>
  
```

Dani Adrian.txt

```

File Edit View
Dani Adrian
225150201111009
Ln 1, Col 1 100% Windows (CRLF) UTF-8
  
```

Penjelasan

- Dalam pemilihan jenis kelas, dipilihlah kelas publik yang tidak bersifat abstrak karena nantinya akan diinstansiasi di dalam kelas Main. Kelas tersebut akan dibuat sebagai turunan dari kelas Employee karena membutuhkan method dan atribut yang telah ada di kelas tersebut.

Dalam perancangan kelas baru ini, akan ditambahkan atribut dan method setter getter sesuai dengan instruksi soal, yaitu jumlahBarang dan hargaBarang. Method earning dari superclass harus diimplementasikan karena merupakan method abstrak. Method earning akan mengembalikan nilai dari jumlahBarang dikalikan dengan hargaBarang, yang bertipe data double. Kelas baru ini akan dijadikan subclass dari kelas Employee.

- Untuk menerapkan kelas tersebut, kita membuat 4 instance kelas baru yang dinamakan PerBarangEmployee. Instance pertama memiliki nama Adit dengan jumlah barang per minggu sebesar 20 dan upah per barang sebesar 40. Earning diperoleh dengan mengalikan upah per barang dengan jumlah barang per minggunya sehingga didapatkan nilai sebesar 800. Hal yang sama juga berlaku untuk semua instance yang telah dibuat.

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

Source Code

Kue.java	
1.	public abstract class Kue {
2.	private String nama;
3.	private double harga;
4.	
5.	Kue(String nama, double harga)
6.	{
7.	setNama(nama);
8.	setHarga(harga);

9.	}
10.	
11.	public String getName() {
12.	return nama;
13.	}
14.	public void setName(String nama) {
15.	this.nama = nama;
16.	}
17.	public double getHarga() {
18.	return harga;
19.	}
20.	public void setHarga(double harga) {
21.	this.harga = harga;
22.	}
23.	
24.	public abstract double hitungHarga();
25.	public String toString()
26.	{
27.	String output =
28.	"Nama : " + getName() + "\n" +
29.	"Harga : " + getHarga() + "\n";
30.	
31.	return output;
32.	}
33.	}

KueJadi.java	
1.	public class KueJadi extends Kue{
2.	private double jumlah;
3.	
4.	KueJadi(String nama, double harga, double jumlah) {
5.	super(nama, harga);
6.	setJumlah(jumlah);
7.	}
8.	
9.	@Override
10.	public double hitungHarga() {
11.	return getJumlah()*getHarga()*2;
12.	}
13.	
14.	@Override
15.	public String toString()
16.	{
17.	String output =
18.	super.toString() +
19.	"Jumlah : " + getJumlah() + "\n" +

20.	"Jenis Kue : Kue Jadi\n" +
21.	"Total Harga : " + hitungHarga();
22.	
23.	return output;
24.	}
25.	
26.	public double getJumlah() {
27.	return jumlah;
28.	}
29.	
30.	public void setJumlah(double jumlah) {
31.	this.jumlah = jumlah;
32.	}
33.	
34.	}

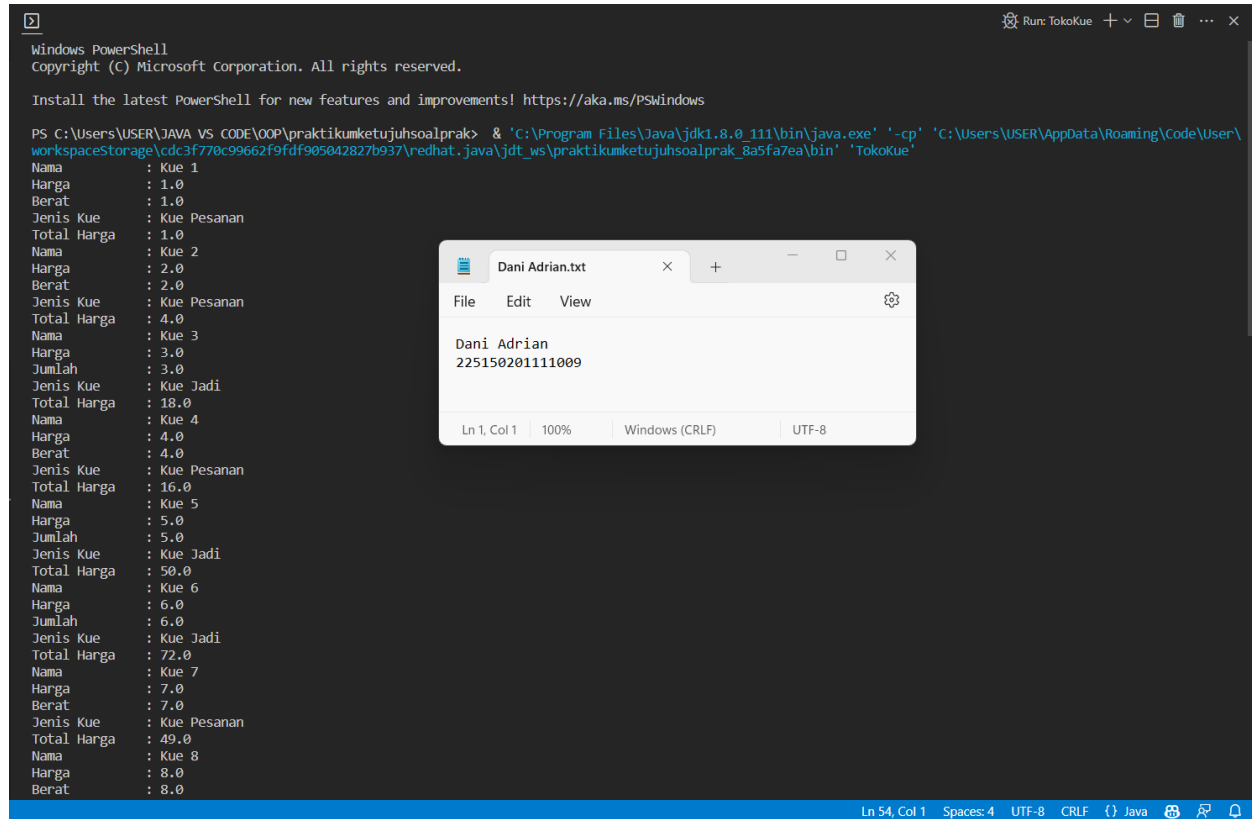
KuePesanan.java	
1.	public class KuePesanan extends Kue{
2.	private double berat;
3.	
4.	KuePesanan(String nama, double harga, double berat)
5.	{
6.	super(nama, harga);
7.	setBerat(berat);
8.	}
9.	
10.	public double getBerat() {
11.	return berat;
12.	}
13.	
14.	public void setBerat(double berat) {
15.	this.berat = berat;
16.	}
17.	
18.	@Override
19.	public double hitungHarga()
20.	{
21.	return getHarga()*getBerat();
22.	}
23.	
24.	@Override
25.	public String toString()
26.	{
27.	String output =
28.	super.toString() +
29.	"Berat : " + getBerat() + "\n" +

30.	"Jenis Kue : Kue Pesanan\n" +
31.	"Total Harga : " + hitungHarga() ;
32.	
33.	return output;
34.	}
35.	}

TokoKue.java	
1.	public class TokoKue {
2.	public static void main(String[] args) {
3.	Kue[] toko = new Kue[20];
4.	
5.	for (int i = 0; i < toko.length; i++) {
6.	if ((i+1)%5 == 0)
7.	{
8.	toko[i] = new KueJadi("Kue " + (i+1), i +
9.	1 , i + 1);
10.	} else if ((i+1)%3 == 0)
11.	{
12.	toko[i] = new KueJadi("Kue " + (i+1), i +
13.	1, i + 1);
14.	} else if ((i+1)%2 == 0)
15.	{
16.	toko[i] = new KuePesanan("Kue " + (i+1),
17.	i + 1, i + 1);
18.	} else
19.	{
20.	toko[i] = new KuePesanan("Kue " + (i+1),
21.	i + 1, i + 1);
22.	}
23.	System.out.println(toko[i]);
24.	}
25.	
26.	double totalHargaSemuaJenis = 0;
27.	double totalHargaJadi = 0;
28.	double totalHargaPesanan = 0;
29.	double totalJumlah = 0;
30.	double totalBerat = 0;
31.	double hargaTerbesar = 0;
32.	
33.	for (int i = 0; i < toko.length; i++) {
34.	totalHargaSemuaJenis +=
35.	toko[i].hitungHarga();
36.	hargaTerbesar = (toko[i].hitungHarga() >
37.	hargaTerbesar) ? toko[i].hitungHarga() : hargaTerbesar;

33.	if (toko[i] instanceof KueJadi)
34.	{
35.	totalHargaJadi += toko[i].hitungHarga();
36.	totalJumlah += ((KueJadi)toko[i]).getJumlah();
37.	}
38.	if (toko[i] instanceof KuePesanan)
39.	{
40.	totalHargaPesanan += toko[i].hitungHarga();
41.	totalBerat += ((KuePesanan)toko[i]).getBerat();
42.	}
43.	}
44.	
45.	System.out.println();
46.	System.out.println("Hasil : ");
47.	System.out.println("Total Harga Semua Jenis Kue : " + totalHargaSemuaJenis);
48.	System.out.println("Total Harga Kue Jadi : " + totalHargaJadi);
49.	System.out.println("Total Jumlah Kue Jadi : " + totalJumlah);
50.	System.out.println("Total Harga Kue Pesanan : " + totalHargaPesanan);
51.	System.out.println("Total Berat Kue Pesanan : " + totalBerat);
52.	}
53.	}

Output



The screenshot shows a Windows PowerShell terminal window with a dark background. The terminal output displays the results of a Java program, listing details for eight different kue (cakes) and their total prices. A text editor window titled 'Dani Adrian.txt' is overlaid on the terminal, showing the name 'Dani Adrian' and the ID '225150201111009'. The terminal window has a title bar that says 'Run: TokoKue' and a status bar at the bottom showing 'Ln 54, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and '{ } Java'.

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuhsoalprak> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\cdc3f770c99662f9fdf905042827b937\redhat.java\jdt_ws\praktikumketujuhsoalprak_8a5fa7ea\bin' 'TokoKue'
Nama      : Kue 1
Harga     : 1.0
Berat     : 1.0
Jenis Kue : Kue Pesanan
Total Harga : 1.0
Nama      : Kue 2
Harga     : 2.0
Berat     : 2.0
Jenis Kue : Kue Pesanan
Total Harga : 4.0
Nama      : Kue 3
Harga     : 3.0
Jumlah    : 3.0
Jenis Kue : Kue Jadi
Total Harga : 18.0
Nama      : Kue 4
Harga     : 4.0
Berat     : 4.0
Jenis Kue : Kue Pesanan
Total Harga : 16.0
Nama      : Kue 5
Harga     : 5.0
Jumlah    : 5.0
Jenis Kue : Kue Jadi
Total Harga : 50.0
Nama      : Kue 6
Harga     : 6.0
Jumlah    : 6.0
Jenis Kue : Kue Jadi
Total Harga : 72.0
Nama      : Kue 7
Harga     : 7.0
Berat     : 7.0
Jenis Kue : Kue Pesanan
Total Harga : 49.0
Nama      : Kue 8
Harga     : 8.0
Berat     : 8.0
```

Dani Adrian
225150201111009

Ln 54, Col 1 Spaces: 4 UTF-8 CRLF { } Java

```

Berat      : 7.0
Jenis Kue  : Kue Pesanan
Total Harga : 49.0
Nama       : Kue 8
Harga      : 8.0
Berat      : 8.0
Jenis Kue  : Kue Pesanan
Total Harga : 64.0
Nama       : Kue 9
Harga      : 9.0
Jumlah     : 9.0
Jenis Kue  : Kue Jadi
Total Harga : 162.0
Nama       : Kue 10
Harga      : 10.0
Jumlah     : 10.0
Jenis Kue  : Kue Jadi
Total Harga : 200.0
Nama       : Kue 11
Harga      : 11.0
Berat      : 11.0
Jenis Kue  : Kue Pesanan
Total Harga : 121.0
Nama       : Kue 12
Harga      : 12.0
Jumlah     : 12.0
Jenis Kue  : Kue Jadi
Total Harga : 288.0
Nama       : Kue 13
Harga      : 13.0
Berat      : 13.0
Jenis Kue  : Kue Pesanan
Total Harga : 196.0
Nama       : Kue 15
Harga      : 15.0
Jumlah     : 15.0
Jenis Kue  : Kue Jadi
Total Harga : 450.0
Nama       : Kue 16
Harga      : 16.0
Berat      : 16.0
Jenis Kue  : Kue Pesanan
Total Harga : 256.0
Nama       : Kue 17
Harga      : 17.0

```

Dani Adrian
225150201111009

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Ln 54, Col 1 | Spaces: 4 | UTF-8 | CRLF | {} Java

```

Total Harga : 196.0
Nama       : Kue 15
Harga      : 15.0
Jumlah     : 15.0
Jenis Kue  : Kue Jadi
Total Harga : 450.0
Nama       : Kue 16
Harga      : 16.0
Berat      : 16.0
Jenis Kue  : Kue Pesanan
Total Harga : 256.0
Nama       : Kue 17
Harga      : 17.0
Berat      : 17.0
Jenis Kue  : Kue Pesanan
Total Harga : 289.0
Nama       : Kue 18
Harga      : 18.0
Jumlah     : 18.0
Jenis Kue  : Kue Jadi
Total Harga : 648.0
Nama       : Kue 19
Harga      : 19.0
Berat      : 19.0
Jenis Kue  : Kue Pesanan
Total Harga : 361.0
Nama       : Kue 20
Harga      : 20.0
Jumlah     : 20.0
Jenis Kue  : Kue Jadi
Total Harga : 800.0

Hasil :
Total Harga Semua Jenis Kue : 4214.0
Total Harga Kue Jadi       : 2688.0
Total Jumlah Kue Jadi      : 98.0
Total Harga Kue Pesanan    : 1526.0
Total Berat Kue Pesanan    : 112.0

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumketujuhsoalprak>

Ln 54, Col 1 | Spaces: 4 | UTF-8 | CRLF | {} Java

Penjelasan

1. Kelas Kue

Kelas abstrak Kue memiliki dua variabel data pribadi (private), yaitu nama dan harga dengan tipe data String dan double masing-masing. Saat objek Kue dibuat, konstruktor Kue akan menginisialisasi nilai variabel nama dan harga. Metode `getNama()` dan `setNama(String nama)` digunakan untuk mengakses dan memodifikasi variabel nama, sedangkan metode `getHarga()` dan `setHarga(double harga)` digunakan untuk mengakses dan memodifikasi variabel harga.

Selain itu, kelas Kue memiliki dua metode abstrak yaitu `hitungHarga()` dan `toString()`. Metode `hitungHarga()` akan diimplementasikan oleh kelas turunan dari Kue sesuai dengan jenis kue yang diwakili oleh kelas turunan tersebut. Metode `toString()` akan mengembalikan informasi tentang objek Kue seperti nama dan harga dalam bentuk string.

Kelas Kue tidak dapat diinstansiasi secara langsung karena merupakan kelas abstrak. Sebaliknya, kelas turunan dari Kue harus dibuat dengan mengimplementasikan metode `hitungHarga()` sesuai dengan jenis kue yang diwakili oleh kelas turunan tersebut.

2. KelasKueJadi

Kelas KueJadi merupakan turunan dari kelas abstrak Kue. Kelas ini memiliki tambahan satu variabel data pribadi yaitu jumlah dengan tipe data double, yang merepresentasikan jumlah kue jadi yang dibeli. Konstruktor KueJadi memiliki tiga parameter yaitu nama, harga, dan jumlah. Nilai nama dan harga akan diinisialisasi oleh konstruktor superclass (Kue) menggunakan `super(nama, harga)`, sedangkan nilai jumlah akan diinisialisasi menggunakan `setJumlah(jumlah)`.

Kelas KueJadi mengimplementasikan metode abstrak `hitungHarga()` yang mengembalikan hasil perkalian harga dan jumlah, kemudian dikalikan dengan 2. Selain itu, kelas KueJadi juga mengoverride metode `toString()` dari superclass (Kue) untuk menambahkan informasi mengenai jenis kue (Kue Jadi) dan total harga (`hitungHarga()`).

Kelas KueJadi memiliki metode akses (`getJumlah()`) dan modifikasi (`setJumlah(double jumlah)`) untuk mengakses dan memodifikasi variabel jumlah.

3. KelasKuePesanan

Kelas KuePesanan adalah turunan dari kelas abstrak Kue. Kelas ini memiliki variabel data pribadi tambahan yaitu berat dengan tipe data double, yang digunakan untuk merepresentasikan berat kue pesanan.

Untuk menginisialisasi nilai variabel nama dan harga pada saat objek KuePesanan dibuat, konstruktor KuePesanan mengambil tiga parameter yaitu nama, harga, dan berat. Konstruktor ini menggunakan `super(nama, harga)` untuk menginisialisasi nilai nama dan harga, dan nilai berat diinisialisasi dengan menggunakan metode `setBerat(berat)`.

Kelas KuePesanan mengimplementasikan metode abstrak `hitungHarga()` yang mengembalikan hasil perkalian harga dan berat. Selain itu, kelas KuePesanan juga mengoverride metode `toString()` dari superclass (Kue) untuk menambahkan informasi mengenai jenis kue (Kue Pesanan) dan total harga (`hitungHarga()`).

Untuk mengakses dan memodifikasi variabel berat, kelas KuePesanan memiliki metode akses (`getBerat()`) dan modifikasi (`setBerat(double berat)`).

4. Kelas TokoKue

Dalam method main, terdapat pengisian array toko dengan 20 objek Kue, dimana jenis kue yang diisi bergantung pada aturan yang ditentukan dalam kondisi if-else. Setelah diisi, array toko diiterasi untuk menampilkan setiap objek Kue yang diisi dengan memanggil method toString(). Kemudian, dilakukan penghitungan total harga semua jenis kue, total harga kue jadi, total jumlah kue jadi, total harga kue pesanan, dan total berat kue pesanan dengan mengiterasi array toko dan memanggil method hitungHarga(), sambil melakukan pengecekan instance of pada setiap objek Kue untuk menentukan jenis kue yang dihitung. Hasil perhitungan tersebut kemudian ditampilkan di akhir program.