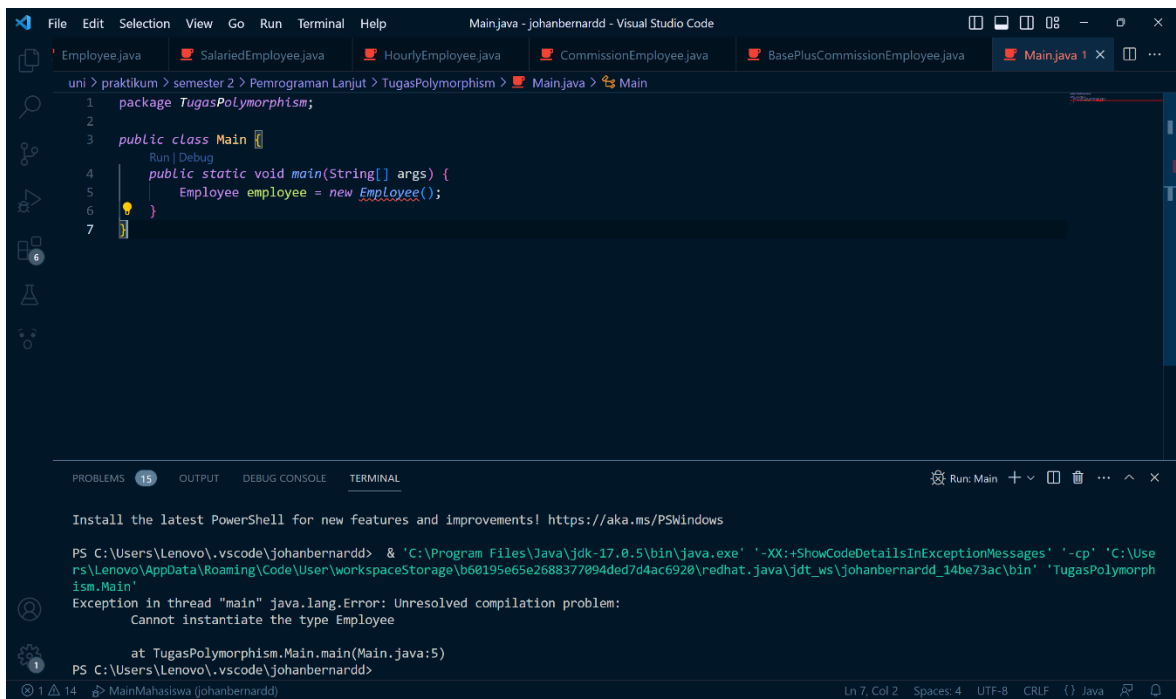


NAMA : Johanes Paulus Bernard Purek
NIM : 225150407111090
KELAS : B
BAB : 7 - Polymorphism
ASISTEN : Fahru Setiawan Iskandar dan Adin Rama Ariyanto Putra

1. Data dan Analisis hasil percobaan

Pertanyaan

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



```
1 package TugasPolymorphism;
2
3 public class Main {
4     public static void main(String[] args) {
5         Employee employee = new Employee();
6     }
7 }
```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL

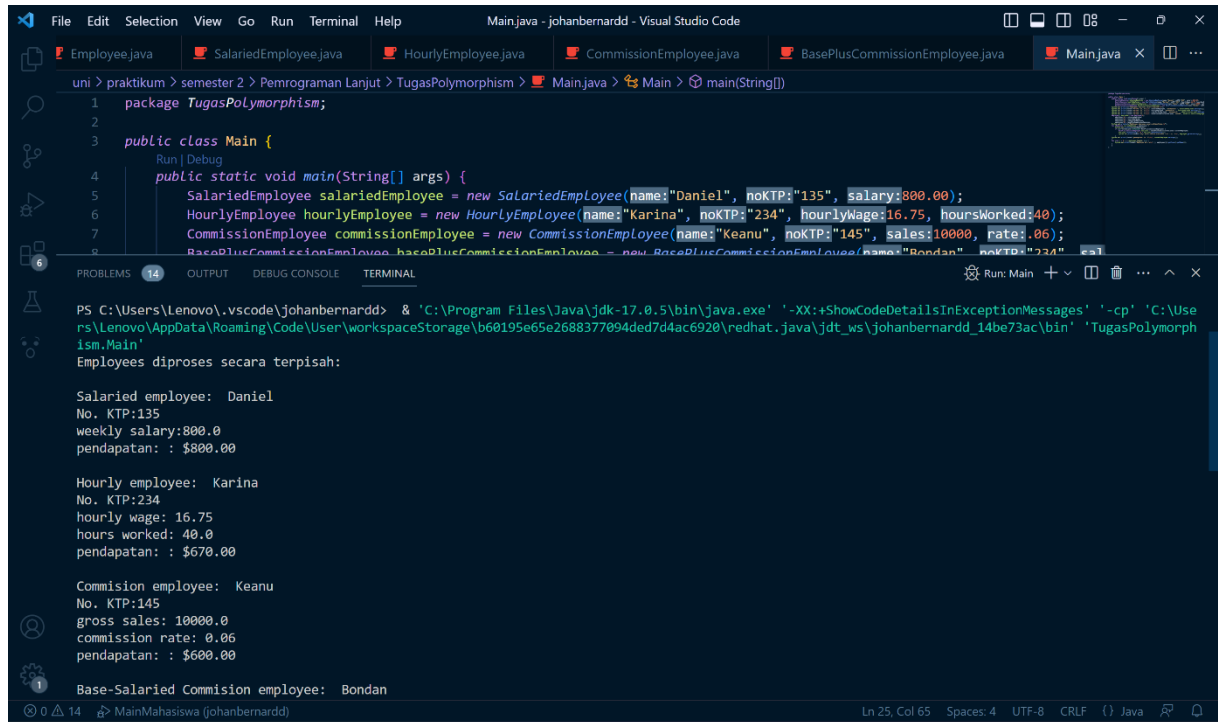
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\Lenovo\.vscode\johanbernardd> & 'C:\Program Files\Java\jdk-17.0.5\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Lenovo\AppData\Roaming\Code\User\workspaceStorage\b68195e65e2688377094ded7d4ac6920\redhat.java\jdt_ws\johanbernardd_14be73ac\bin' 'TugasPolymorphism.Main'

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Cannot instantiate the type Employee

at TugasPolymorphism.Main.main(Main.java:5)
PS C:\Users\Lenovo\.vscode\johanbernardd>

- Ketika kode program instansiasi employee di Main.java dijalankan, akan menghasilkan keluaran berupa error dimana hal ini terjadi karena class Employee bersifat abstract class dan mengakibatkan class tersebut tidak dapat diinstansiasikan sebagai objek ketika akan dijalankan di class Main. Sebuah objek akan dijalankan di Main apabila menginstansiasikan subclass dari abstract class Employee.
2. Jalankan program dengan main sebagai berikut. Analisis dan jelaskan output program (berdasarkan konsep polimorfisme)!



```

package TugasPolymorphism;

public class Main {
    public static void main(String[] args) {
        SalariedEmployee salariedEmployee = new SalariedEmployee(name:"Daniel", noKTP:"135", salary:800.00);
        HourlyEmployee hourlyEmployee = new HourlyEmployee(name:"Karina", noKTP:"234", hourlyWage:16.75, hoursWorked:40);
        CommissionEmployee commissionEmployee = new CommissionEmployee(name:"Keanu", noKTP:"145", sales:10000, rate:0.06);
        BasePlusCommissionEmployee basePlusCommissionEmployee = new BasePlusCommissionEmployee(name:"Bondan", noKTP:"234", salary:300.00, commissionRate:0.04, grossSales:5000.00);
    }
}

```

```

PS C:\Users\Lenovo\.vscode\johanbernardd & 'C:\Program Files\Java\jdk-17.0.5\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Lenovo\AppData\Roaming\Code\User\workspaceStorage\b60195e65e2688377094ded7d4ac6920\redhat.java\jdt_ws\johanbernardd_14be73ac\bin' 'TugasPolymorphism.Main'
Employees diproses secara terpisah:

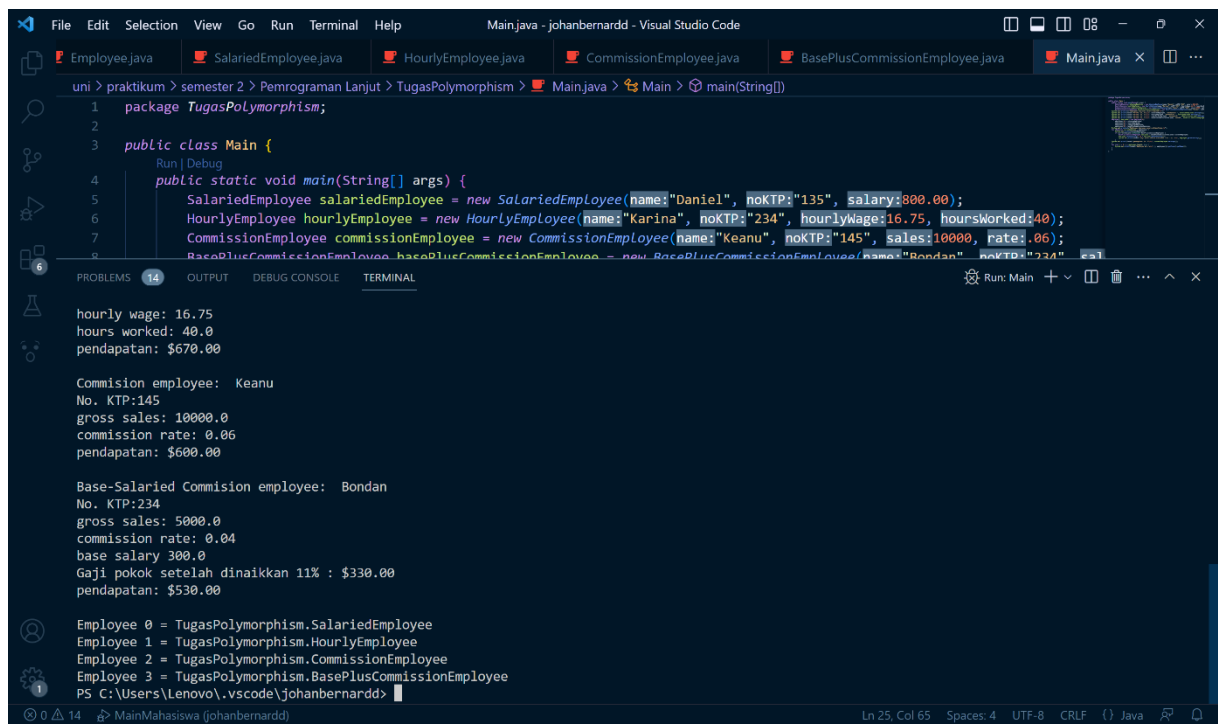
Salaried employee: Daniel
No. KTP:135
weekly salary:800.0
pendapatan : $800.00

Hourly employee: Karina
No. KTP:234
hourly wage: 16.75
hours worked: 40.0
pendapatan : $670.00

Commision employee: Keanu
No. KTP:145
gross sales: 10000.0
commission rate: 0.06
pendapatan : $600.00

Base-Salaried Commision employee: Bondan

```



```

hourly wage: 16.75
hours worked: 40.0
pendapatan : $670.00

Commision employee: Keanu
No. KTP:145
gross sales: 10000.0
commission rate: 0.06
pendapatan : $600.00

Base-Salaried Commision employee: Bondan
No. KTP:234
gross sales: 5000.0
commission rate: 0.04
base salary 300.0
Gaji pokok setelah dinaikkan 11% : $330.00
pendapatan : $530.00

Employee 0 = TugasPolymorphism.SalariedEmployee
Employee 1 = TugasPolymorphism.HourlyEmployee
Employee 2 = TugasPolymorphism.CommissionEmployee
Employee 3 = TugasPolymorphism.BasePlusCommissionEmployee
PS C:\Users\Lenovo\.vscode\johanbernardd>

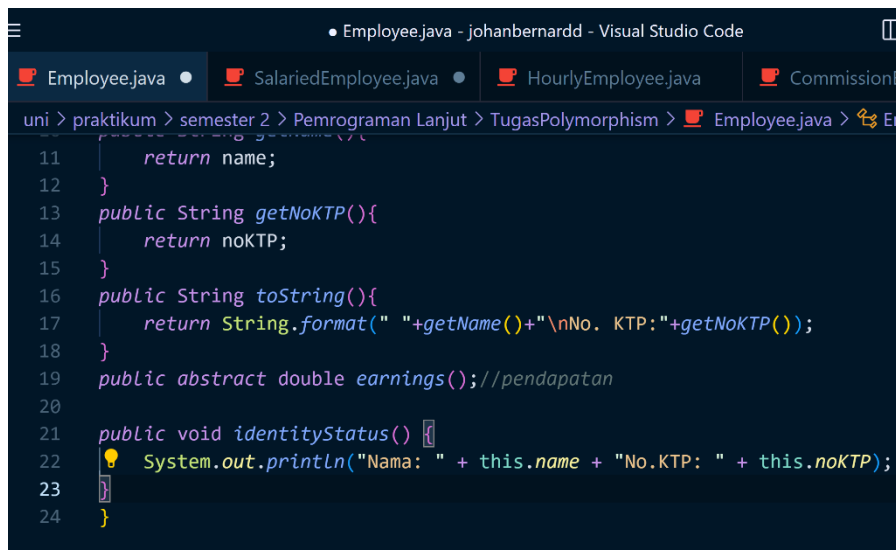
```

- Ketika class Main diatas dijalankan maka akan menghasilkan output sebagai berikut. Di awal class Main, masing-masing class yang merupakan turunan/subclass dari superclass Employee diinstansiasikan menjadi objek serta diberikan parameternya secara terpisah dan objek tersebut diproses dengan memanggil method earnings yang ada pada kelas Employee. Lalu, diberikan kode

untuk menghasilkan output yang sama namun memiliki konsep yang berbeda.

Kode dibawahnya menggunakan konsep polimorfisme untuk memproses objek-objek yang telah diinstansiasikan ke dalam array 'employees'. Objek-objek yang terdapat dalam array ini kemudian akan dilakukan iterasi for each loop sebagai 'current employees' serta memanggil isi dari method earnings yang telah dibuat untuk menghasilkan keluaran pendapatan.

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



```
11     return name;
12 }
13 public String getNoKTP(){
14     return noKTP;
15 }
16 public String toString(){
17     return String.format(" "+getName()+"\nNo. KTP:"+getNoKTP());
18 }
19 public abstract double earnings();//pendapatan
20
21 public void identityStatus() {
22     System.out.println("Nama: " + this.name + "No.KTP: " + this.noKTP);
23 }
24 }
```

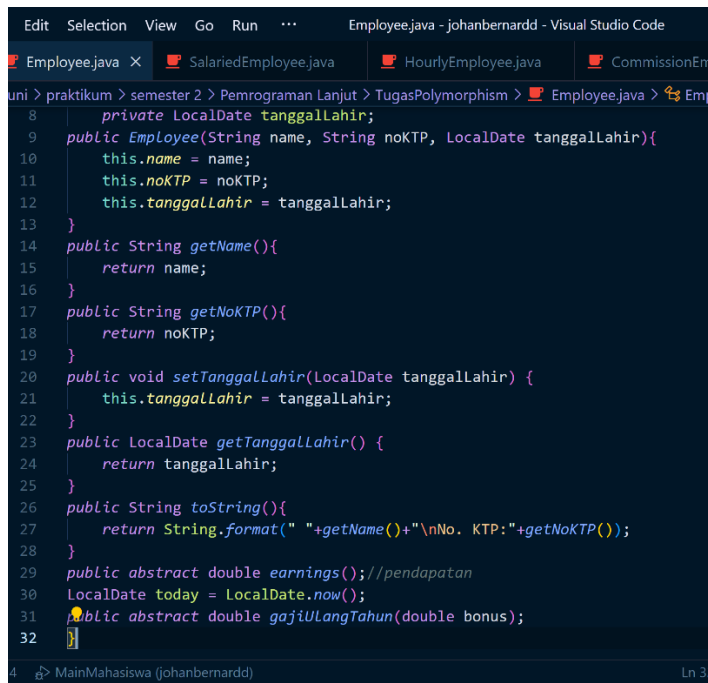


```
25     System.out.printf(format:"Gaji pokok setelah dinaikkan 11% : $%,.2f\n",
26     }
27     System.out.printf(format:"pendapatan: $%,.2f\n\n", currentEmployee.earnings());
28 }
29 for (int j = 0; j < employees.Length; j++) {
30     System.out.printf(format:"Employee %d = %s\n", j, employees[j].getClass().ge
31 }
32 }
33
34 Employee karyawan = new Employee("Johanes", "123456789");
35 karyawan.identityStatus();
36 }
```

- Untuk membuat objek dari method Employee tidak dapat dilakukan. Hal ini karena method yang dibuat pada class Employee merupakan method yang bersifat non-abstract padahal class Employee tersebut merupakan abstract class dimana hanya dapat dibuat method yang bersifat abstract sehingga kemudian akan digunakan oleh class child-nya melalui override. Untuk contoh dari kasus diatas, dibuat method void bernama identityStatus yang akan menghasilkan keluaran

untuk menampilkan informasi nama dan no.ktp dan ketika akan dibuat menjadi objek untuk dijalankan pada class Main menghasilkan error.

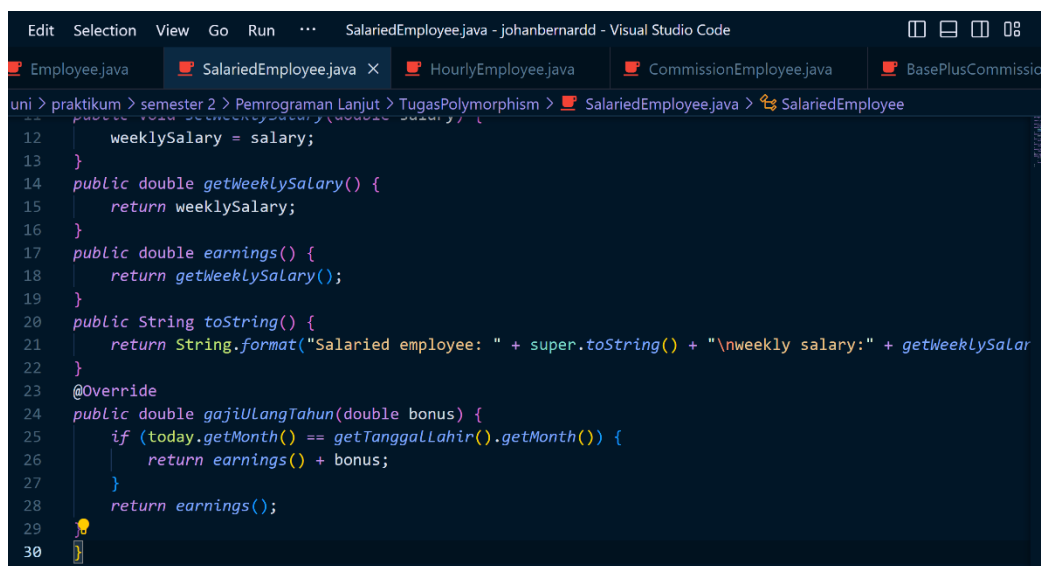
4. Tambahkan atribut tanggal lahir di Kelas Employee, serta tambahkan method pendukungnya (accesor dan mutator). Modifikasi program agar sesuai. Asumsikan gaji yang diterima adalah per bulan, buat kelas uji untuk menguji program yang sudah anda modifikasi, kemudian buat objek dari semua class (salariedEmployee,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



```

8      private LocalDate tanggalLahir;
9      public Employee(String name, String noKTP, LocalDate tanggalLahir){
10         this.name = name;
11         this.noKTP = noKTP;
12         this.tanggalLahir = tanggalLahir;
13     }
14     public String getName(){
15         return name;
16     }
17     public String getNoKTP(){
18         return noKTP;
19     }
20     public void setTanggalLahir(LocalDate tanggalLahir) {
21         this.tanggalLahir = tanggalLahir;
22     }
23     public LocalDate getTanggalLahir() {
24         return tanggalLahir;
25     }
26     public String toString(){
27         return String.format("%s\nNo. KTP:%s", getName(), getNoKTP());
28     }
29     public abstract double earnings(); //pendapatan
30     LocalDate today = LocalDate.now();
31     public abstract double gajiUlangTahun(double bonus);
32

```



```

12     weeklySalary = salary;
13 }
14 public double getWeeklySalary() {
15     return weeklySalary;
16 }
17 public double earnings() {
18     return getWeeklySalary();
19 }
20 public String toString() {
21     return String.format("Salaried employee: " + super.toString() + "\nweekly salary:" + getWeeklySalary());
22 }
23 @Override
24 public double gajiUlangTahun(double bonus) {
25     if (today.getMonth() == getTanggalLahir().getMonth()) {
26         return earnings() + bonus;
27     }
28     return earnings();
29 }
30

```

```

1
2
3
4
5
Run | Debug
6
7 ew SalariedEmployee(name:"Daniel", noKTP:"135", LocalDate.of(year:1997, month:05, dayOfMonth:19), salary:800.00);
8 ourLyEmployee(name:"Karina", noKTP:"234", LocalDate.of(year:1997, month:04, dayOfMonth:20), hourlyWage:16.75, hoursWorked:40);
9 = new CommissionEmployee(name:"Keanu", noKTP:"145", LocalDate.of(year:1997, month:05, dayOfMonth:21), sales:10000, rate:0.06);
10 mmissionEmployee = new BasePlusCommissionEmployee(name:"Bondan", noKTP:"234", LocalDate.of(year:1997, month:06, dayOfMonth:22))

```

```

35
36 Employee[] employees = new Employee[4];
37 employees[0] = salariedEmployee;
38 employees[1] = hourlyEmployee;
39 employees[2] = commissionEmployee;
40 employees[3] = basePlusCommissionEmployee;
41 System.out.println("Employees diproses secara polimorfisme:\n");
42 for (Employee currentEmployee : employees) {
43     System.out.println(currentEmployee);
44     if (currentEmployee instanceof BasePlusCommissionEmployee) {
45         BasePlusCommissionEmployee employee = (BasePlusCommissionEmployee) currentEmployee;
46         employee.setBaseSalary(1.10 * employee.getBaseSalary());
47         System.out.printf("Gaji pokok setelah dinaikkan 11% : $%.2f\n", employee.getBaseSalary());
48     }
49     //System.out.printf("pendapatan: $%.2f\n", currentEmployee.earnings());
50     double bonus = 100000.0;
51     double earningsWithBonus = currentEmployee.gajiUlangTahun(bonus);
52     System.out.printf("pendapatan: $%.2f\n", earningsWithBonus);
53 }
54 for (int j = 0; j < employees.Length; j++) {
55     System.out.printf("Employee %d = %s\n", j, employees[j].getClass().getName());
56 }
57 }
58

```

```

PS C:\Users\Lenovo\vscode\johanbernardd> & 'C:\Program Files\Java\jdk-17.0.5\bin\java.exe'
rs\Lenovo\AppData\Roaming\Code\User\workspaceStorage\b60195e65e2688377094ded7d4ac6920\red
ism.Main'
Employees diproses secara polimorfisme:

Salaried employee: Daniel
No. KTP:135
weekly salary:800.0
pendapatan: $100,800.00

Hourly employee: Karina
No. KTP:234
hourly wage: 16.75
hours worked: 40.0
pendapatan: $670.00

Commission employee: Keanu
No. KTP:145
gross sales: 10000.0
commission rate: 0.06
pendapatan: $100,600.00

Base-Salaried Commision employee: Bondan
No. KTP:234
gross sales: 5000.0
commission rate: 0.04
base salary 300.0
Gaji pokok setelah dinaikkan 11% : $330.00
pendapatan: $530.00

Employee 0 = TugasPolymorphism.SalariedEmployee
Employee 1 = TugasPolymorphism.HourlyEmployee

```

- Untuk dapat menghitung besar gaji setelah ditambahkan sebesar 100.000 jika bulan tersebut merupakan bulan ulang tahun seorang Employee maka perlu dibuatkan atribut tanggal lahir dengan accessor serta mutatornya di kelas

Employee serta menginstansiasi objek `LocalDate` dengan nilai `now()` dan juga membuat method abstract bertipe `double` untuk menambahkan gaji Employee jika ia berulang tahun dengan parameter `bonus`. Abstract method yang telah dibuat di kelas `Employee` kemudian di-override di kelas-kelas turunannya dengan penambahan kode jika ia berulang tahun maka gaji awalnya akan ditambahkan dengan besaran `bonus` kemudian di-return pendapatannya. Pada kelas `Main`, hanya dibuat 2 orang saja yang berulang tahun dan 2 orang lain yang tidak sebagai perbandingan hasil keluaran nanti serta dibawahnya terdapat penambahan kode dengan membuat variabel baru `earningWithBonus` untuk menambahkan return value dari method `gajiUlangTahun` ke objek `currentEmployee` sehingga dapat dibuat kode keluaran dibawahnya.

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.

a) Proses modifikasi program dimulai dari pemilihan jenis class pada class `Employee` dengan dibuat suatu abstract method bernama `gajiBasedProduction` berparameter `double upahPerProduksi` yang ditentukan besarnya pada class `Main`. Tidak lupa juga untuk menambahkan variabel `barangProduksi` bertipe data `double` pada constructor `Employee` untuk menentukan setiap `Employee` memproduksi berapa banyak barang. Kemudian pada tahap perancangan class, abstract class yang telah dibuat sebelumnya akan di-override pada class turunannya dengan mengembalikan return value bernilai pendapatan yang telah dimiliki sebelumnya (`earnings`) dan dijumlahkan dengan perkalian dari banyaknya barang yang diproduksi oleh setiap `Employee` dan `upahPerProduksi`-nya. Lalu, pada tahap penempatan class, class `Main`-nya akan dibuat besaran dari `upahPerProduksi`-nya dan membuat variabel untuk menampung nilai dari penambahan `upahPerProduksi` per masing-masing `Employee` yang merupakan iterasi dari `currentEmployee` yang telah dibuat

sebelumnya. Pada baris kode dibawahnya, akan dibuat variabel baru totalGaji yang merupakan penjumlahan dari pendapatan jika berulang tahun pada bulan tersebut dengan pendapatan dari banyaknya barang yang telah diproduksi.

b) Implementasi hasil analisis:

```
Employee.java - johanbernardd - Visual Studio Code
Employee.java X SalariedEmployee.java HourlyEmployee.java CommissionEmployee.java Ba
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism > Employee.java > Employee
1 package TugasPolymorphism;
2
3 import java.time.LocalDate;
4
5 public abstract class Employee {
6     private String name;
7     private String noKTP;
8     private LocalDate tanggalLahir;
9     private double barangProduksi;
10    public Employee(String name, String noKTP, LocalDate tanggalLahir, double barangProduksi){
11        this.name = name;
12        this.noKTP = noKTP;
13        this.tanggalLahir = tanggalLahir;
14        this.barangProduksi = barangProduksi;
15    }
```

```
Employee.java - johanbernardd - Visual Studio Code
Employee.java X SalariedEmployee.java HourlyEmployee.java Commission
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism > Employee.java > E
24 }
25 public LocalDate getTanggalLahir() {
26     return tanggalLahir;
27 }
28 public double getBarangProduksi() {
29     return barangProduksi;
30 }
31 public String toString(){
32     return String.format(" "+getName()+"\nNo. KTP:"+getNoKTP());
33 }
34 public abstract double earnings();//pendapatan
35 LocalDate today = LocalDate.now();
36 public abstract double gajiUlangTahun(double bonus);
37 public abstract double gajiBasedProduction(double upahPerProduksi);
38 }
```

```
SalariedEmployee.java - johanbernardd - Visual Studio
Employee.java X SalariedEmployee.java X HourlyEmployee.java Comm
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism > SalariedEmployee
23 @Override
24 public double gajiUlangTahun(double bonus) {
25     if (today.getMonth() == getTanggalLahir().getMonth()) {
26         return earnings() + bonus;
27     }
28     return earnings();
29 }
30 @Override
31 public double gajiBasedProduction(double upahPerProduksi) {
32     return earnings() + (getBarangProduksi() * upahPerProduksi);
33 }
34 }
```



```

Edit Selection View Go Run ... Main.java - johanbernardd - Visual Studio Code
Employee.java HourlyEmployee.java CommissionEmployee.java BasePlusCommissionEmployee.java Main.java
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism > Main.java
1
2
3
4
5
Run | Debug
6
7   LocalDate.of(year:1997, month:05, dayOfMonth:19), barangProduksi:10, salary:800.00);
8   .Date.of(year:1997, month:04, dayOfMonth:20), barangProduksi:15, hourlyWage:16.75, hoursWorked:40);
9   '145", LocalDate.of(year:1997, month:05, dayOfMonth:21), barangProduksi:12, sales:10000, rate:.06);
10  'ee(name:"Bondan", noKTP:"234", LocalDate.of(year:1997, month:06, dayOfMonth:22), barangProduksi:11, sa

```

```

Edit Selection View Go Run ... Main.java - johanbernardd - Visual Studio Code
Employee.java HourlyEmployee.java CommissionEmployee.java BasePlusCommissionEmployee.java Main.java
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism > Main.java
39     employees[2] = commissionEmployee;
40     employees[3] = basePlusCommissionEmployee;
41     System.out.println(x;"Employees diproses secara polimorfisme:\n");
42     for (Employee currentEmployee : employees) {
43         System.out.println(currentEmployee);
44         if (currentEmployee instanceof BasePlusCommissionEmployee) {
45             BasePlusCommissionEmployee employee = (BasePlusCommissionEmployee) currentEmployee;
46             employee.setBaseSalary(1.10 * employee.getBaseSalary());
47             System.out.printf(format:"Gaji pokok setelah dinaikkan 11% : $%,.2f\n", employee.getBaseSa
48         }
49         //System.out.printf("pendapatan: $%,.2f\n", currentEmployee.earnings());
50         double bonus = 10000.0;
51         double earningsWithBonus = currentEmployee.gajiULangTahun(bonus);
52
53         double upahPerProduksi = 70.0;
54         double earningWithUpahProduksi = currentEmployee.gajiBasedProduction(upahPerProduksi);
55
56         double totalGaji = earningsWithBonus + earningWithUpahProduksi;
57         System.out.printf(format:"pendapatan: $%,.2f\n\n", totalGaji);
58     }

```

```

File Edit Selection View Go Run Terminal Help Main.java
Employee.java SalariedEmployee.java HourlyEmployee.java
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPolymorphism >
1   nackae TugasPolymorphism:
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL
Employees diproses secara polimorfisme:
Salaried employee: Daniel
No. KTP:135
weekly salary:800.0
pendapatan: $102,300.00
Hourly employee: Karina
No. KTP:234
hourly wage: 16.75
hours worked: 40.0
pendapatan: $2,390.00
Commision employee: Keanu
No. KTP:145
gross sales: 10000.0
commission rate: 0.06
pendapatan: $102,040.00
Base-Salaried Commision employee: Bondan
No. KTP:234
gross sales: 5000.0
commission rate: 0.04
base salary 300.0
Gaji pokok setelah dinaikkan 11% : $330.00
pendapatan: $1,830.00
Employee 0 = TugasPolymorphism.SalariedEmployee
Employee 1 = TugasPolymorphism.HourlyEmployee
Employee 2 = TugasPolymorphism.CommissionEmployee
Employee 3 = TugasPolymorphism.BasePlusCommissionEmployee
PS C:\Users\Lenovo\.vscode\johanbernardd>
14 MainMahasiswa (johanbernardd)

```


2. Tugas Praktikum

2.1 Source code

```
package TugasPrakPolymorphism;

public abstract class Kue {
    private String nama;
    private double harga;
    public Kue(String nama, double harga) {
        this.nama = nama;
        this.harga = harga;
    }
    public String getNama() {
        return nama;
    }
    public double getHarga() {
        return harga;
    }
    public String toString() {
        return String.format(getNama() + ", Harga: " + getHarga() +
"/kue");
    }
    public abstract double hitungHarga();
}
```

```
package TugasPrakPolymorphism;

public class KuePesanan extends Kue {
    private double berat;
    public KuePesanan(String nama, double harga, double berat) {
        super(nama, harga);
        this.berat = berat;
    }
    public double getBerat() {
        return berat;
    }
    public double hitungHarga() {
        return getHarga() * getBerat();
    }
}
```

```
package TugasPrakPolymorphism;

public class KueJadi extends Kue {
    private double jumlah;
```

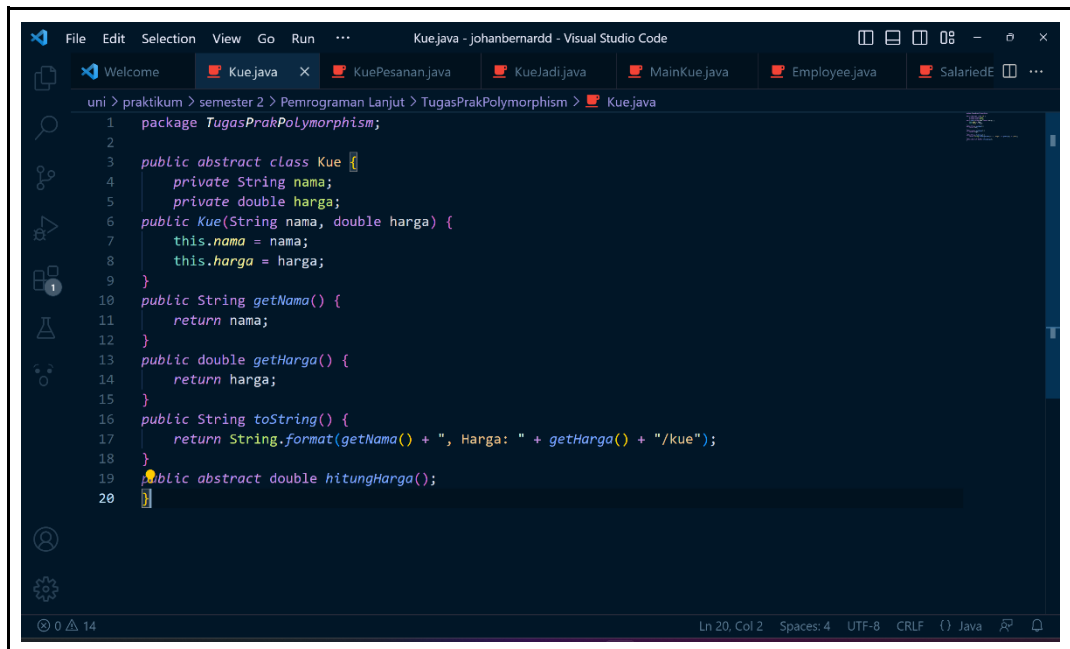
```
public KueJadi(String nama, double harga, double jumlah) {  
    super(nama, harga);  
    this.jumlah = jumlah;  
}  
  
public double getJumlah() {  
    return jumlah;  
}  
  
public double hitungHarga() {  
    return getHarga() * getJumlah() * 2;  
}  
}
```

```
package TugasPrakPolymorphism;  
  
public class MainKue {  
    public static void main(String[] args) {  
        Kue[] kue = new Kue[20];  
        kue[0] = new KuePesanan("Kue Lapis", 8000.0, 2);  
        kue[1] = new KuePesanan("Kue Klepon", 6000.0, 1);  
        kue[2] = new KuePesanan("Kue Onde-onde", 5000.0, 2);  
        kue[3] = new KuePesanan("Kue Lumpur", 7000.0, 1);  
        kue[4] = new KuePesanan("Kue Ketan Hitam", 4000.0, 1);  
        kue[5] = new KuePesanan("Kue Putu Ayu", 7000.0, 1);  
        kue[6] = new KuePesanan("Kue Nagasari", 10000.0, 2);  
        kue[7] = new KuePesanan("Kue Serabi", 10000.0, 1);  
        kue[8] = new KuePesanan("Kue Ubi Ungu", 7000.0, 1);  
        kue[9] = new KuePesanan("Kue Lepet", 8000.0, 1);  
  
        kue[10] = new KueJadi("Kue Talam", 3000.0, 2);  
        kue[11] = new KueJadi("Kue Lupis", 4000.0, 1);  
        kue[12] = new KueJadi("Kue Getuk", 3000.0, 1);  
        kue[13] = new KueJadi("Kue Bingka Ubi", 6000.0, 2);  
        kue[14] = new KueJadi("Kue Cucur", 4000.0, 3);  
        kue[15] = new KueJadi("Kue Pancong", 7000.0, 3);  
        kue[16] = new KueJadi("Kue Pukis", 8000.0, 2);  
        kue[17] = new KueJadi("Kue Tape", 6000.0, 1);  
        kue[18] = new KueJadi("Kue Dadar Gulung", 4000.0, 2);  
        kue[19] = new KueJadi("Kue Mendut", 5000.0, 1);  
  
        System.out.println("Daftar Kue:");  
        for (Kue instKue : kue) {  
            if (instKue != null) {  
                if (instKue instanceof KuePesanan) {  
                    System.out.println("Kue Pesanan -> " +  
instKue);
```

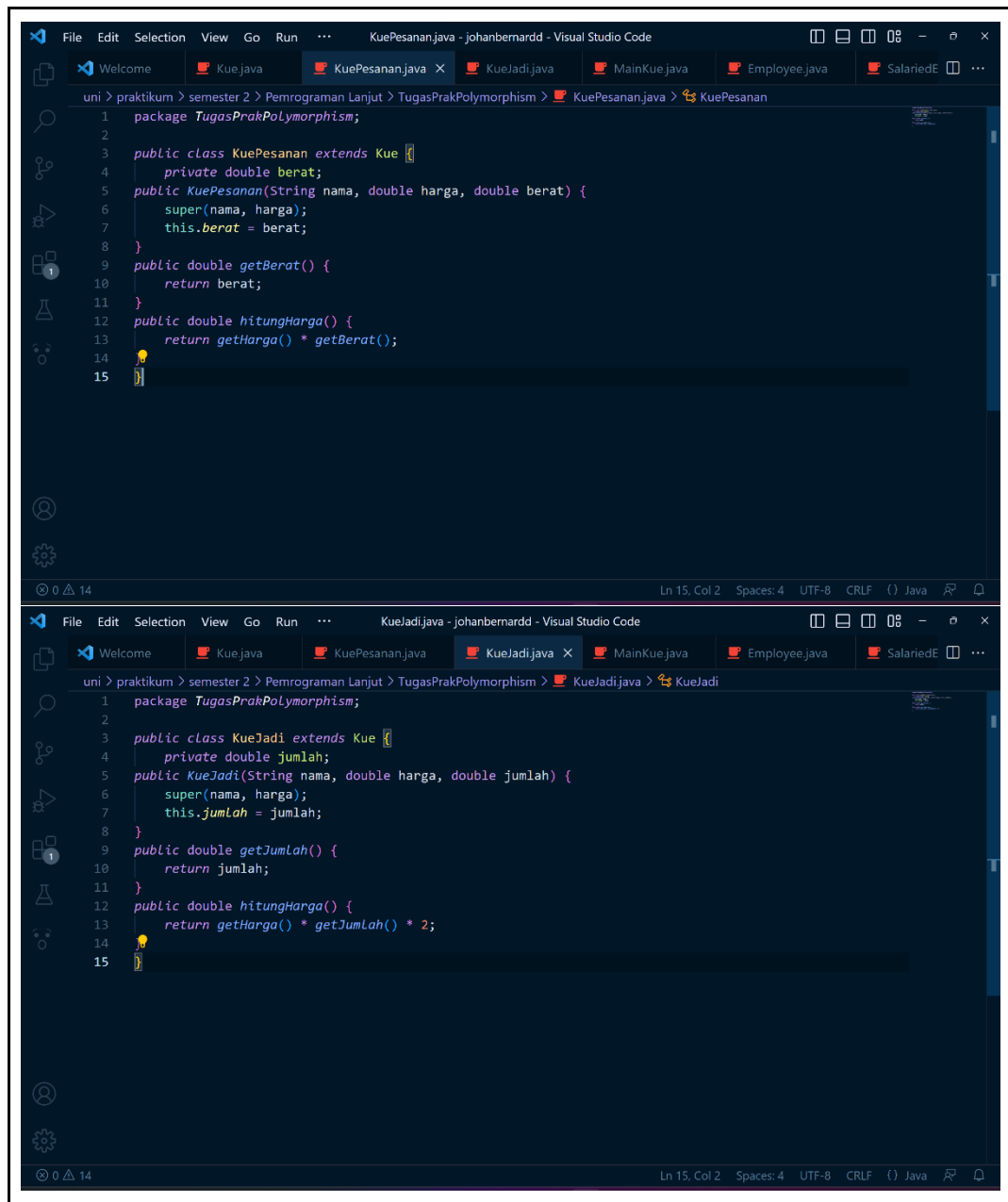
```
        }  
        else if (instKue instanceof KueJadi) {  
            System.out.println("Kue Jadi -> " + instKue);  
        }  
    }  
}  
  
double totalHarga = 0;  
for (Kue instKue : kue) {  
    if (instKue != null) {  
        totalHarga += instKue.hitungHarga();  
    }  
}  
System.out.println("Total Harga semua jenis kue adalah " +  
totalHarga);  
  
double totalHargaKuePesanan = 0;  
double totalBeratKuePesanan = 0;  
for (Kue instKue : kue) {  
    if (instKue instanceof KuePesanan) {  
        totalHargaKuePesanan += instKue.hitungHarga();  
        totalBeratKuePesanan += ((KuePesanan)  
instKue).getBerat();  
    }  
}  
System.out.println("Total Harga Kue Pesanan: " +  
totalHargaKuePesanan);  
System.out.println("Total Berat Kue Pesanan: " +  
totalBeratKuePesanan);  
  
double totalHargaKueJadi = 0;  
double totalJumlahKueJadi = 0;  
for (Kue instKue : kue) {  
    if (instKue instanceof KueJadi) {  
        totalHargaKueJadi += instKue.hitungHarga();  
        totalJumlahKueJadi += ((KueJadi)  
instKue).getJumlah();  
    }  
}  
System.out.println("Total Harga Kue Jadi: " +  
totalHargaKueJadi);  
System.out.println("Total Jumlah Kue Jadi: " +  
totalJumlahKueJadi);  
  
double hargaMax = 0;  
Kue kueTerbesar = null;
```

```
        for (Kue instKue : kue) {  
            if (instKue != null && instKue.hitungHarga() >  
hargaMax) {  
                hargaMax = instKue.hitungHarga();  
                kueTerbesar = instKue;  
            }  
        }  
        if (kueTerbesar != null) {  
            System.out.println("Kue dengan harga terbesar: \n" +  
kueTerbesar);  
        }  
    }  
}
```

2.2 Screenshot hasil

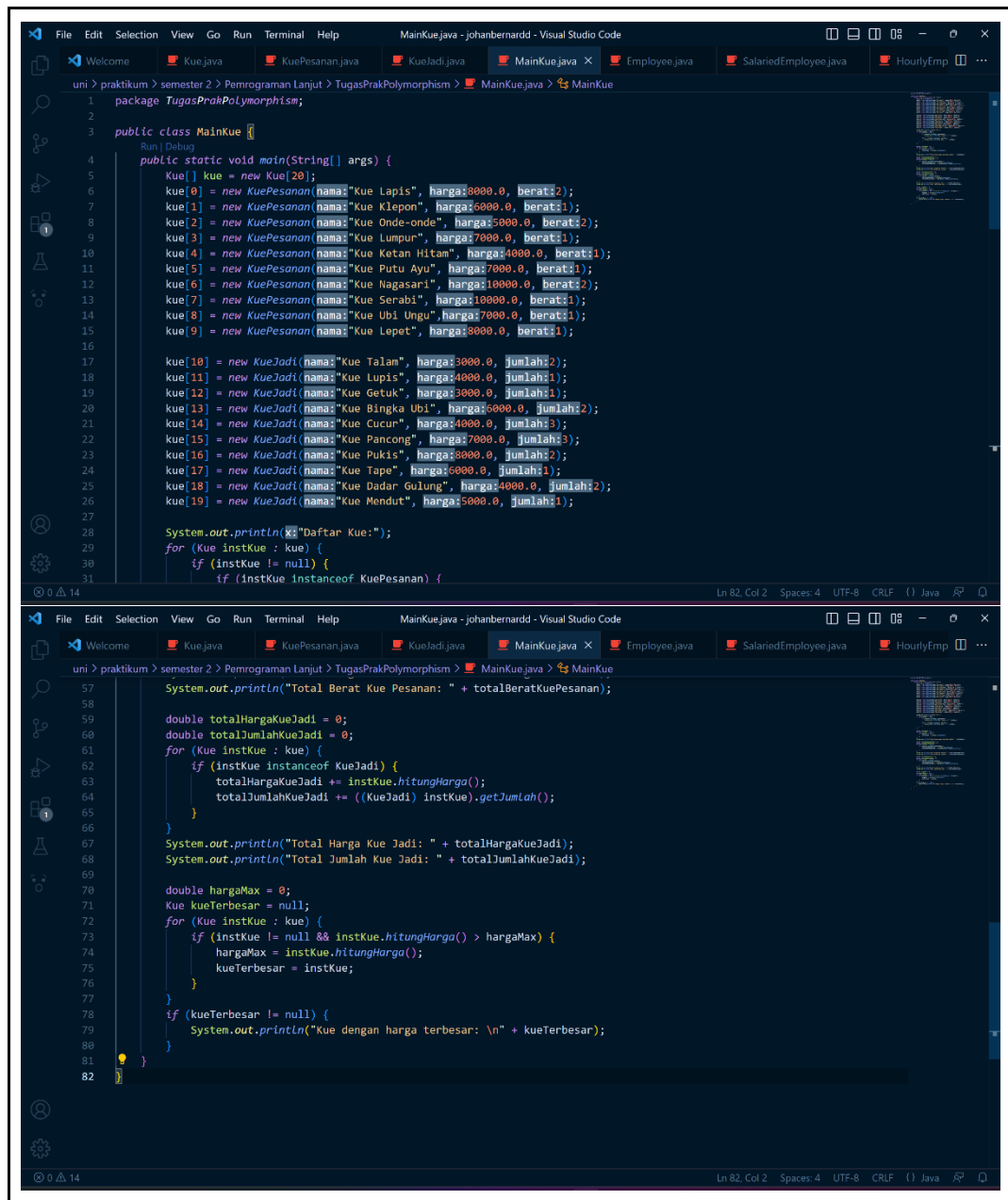


```
File Edit Selection View Go Run ... Kue.java - johanbernardd - Visual Studio Code  
Welcome Kue.java X KuePesanan.java KueJadi.java MainKue.java Employee.java SalariedE ...  
uni > praktikum > semester 2 > Pemrograman Lanjut > TugasPrakPolymorphism > Kue.java  
1 package TugasPrakPolymorphism;  
2  
3 public abstract class Kue {  
4     private String nama;  
5     private double harga;  
6     public Kue(String nama, double harga) {  
7         this.nama = nama;  
8         this.harga = harga;  
9     }  
10    public String getNama() {  
11        return nama;  
12    }  
13    public double getHarga() {  
14        return harga;  
15    }  
16    public String toString() {  
17        return String.format(getNama() + ", Harga: " + getHarga() + "/kue");  
18    }  
19    public abstract double hitungHarga();  
20 }
```



```
1 package TugasPrakPolymorphism;
2
3 public class KuePesanan extends Kue {
4     private double berat;
5     public KuePesanan(String nama, double harga, double berat) {
6         super(nama, harga);
7         this.berat = berat;
8     }
9     public double getBerat() {
10        return berat;
11    }
12    public double hitungHarga() {
13        return getHarga() * getBerat();
14    }
15 }
```

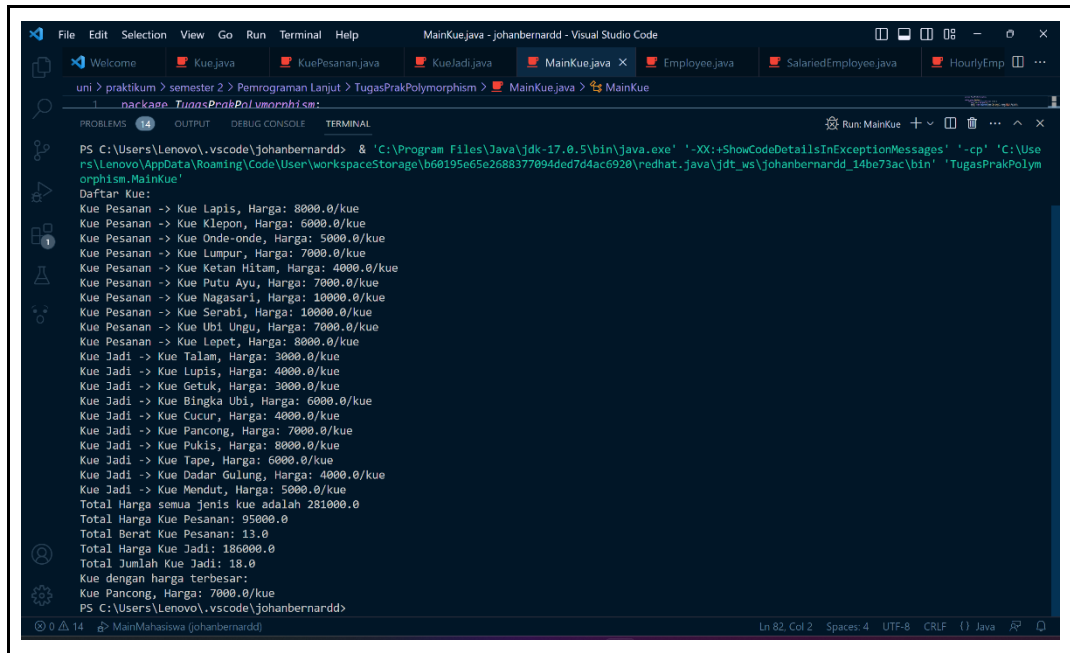
```
1 package TugasPrakPolymorphism;
2
3 public class KueJadi extends Kue {
4     private double jumlah;
5     public KueJadi(String nama, double harga, double jumlah) {
6         super(nama, harga);
7         this.jumlah = jumlah;
8     }
9     public double getJumlah() {
10        return jumlah;
11    }
12    public double hitungHarga() {
13        return getHarga() * getJumlah() * 2;
14    }
15 }
```



```

1 package TugasPrakPolymorphism;
2
3 public class MainKue {
4     Run | Debug
5     public static void main(String[] args) {
6         Kue[] kue = new Kue[20];
7         kue[0] = new KuePesanan(nama: "Kue Lapis", harga: 8000.0, berat: 2);
8         kue[1] = new KuePesanan(nama: "Kue Klepon", harga: 6000.0, berat: 1);
9         kue[2] = new KuePesanan(nama: "Kue Onde-onde", harga: 5000.0, berat: 2);
10        kue[3] = new KuePesanan(nama: "Kue Lumpur", harga: 7000.0, berat: 1);
11        kue[4] = new KuePesanan(nama: "Kue Ketan Hitam", harga: 4000.0, berat: 1);
12        kue[5] = new KuePesanan(nama: "Kue Putu Ayu", harga: 7000.0, berat: 1);
13        kue[6] = new KuePesanan(nama: "Kue Nagasari", harga: 10000.0, berat: 2);
14        kue[7] = new KuePesanan(nama: "Kue Serabi", harga: 10000.0, berat: 1);
15        kue[8] = new KuePesanan(nama: "Kue Ubi Ungu", harga: 7000.0, berat: 1);
16        kue[9] = new KuePesanan(nama: "Kue Lepet", harga: 8000.0, berat: 1);
17
18        kue[10] = new KueJadi(nama: "Kue Talam", harga: 3000.0, jumlah: 2);
19        kue[11] = new KueJadi(nama: "Kue Lupis", harga: 4000.0, jumlah: 1);
20        kue[12] = new KueJadi(nama: "Kue Getuk", harga: 3000.0, jumlah: 1);
21        kue[13] = new KueJadi(nama: "Kue Bingka Ubi", harga: 5000.0, jumlah: 2);
22        kue[14] = new KueJadi(nama: "Kue Cucur", harga: 4000.0, jumlah: 3);
23        kue[15] = new KueJadi(nama: "Kue Pancong", harga: 7000.0, jumlah: 3);
24        kue[16] = new KueJadi(nama: "Kue Pukis", harga: 8000.0, jumlah: 2);
25        kue[17] = new KueJadi(nama: "Kue Tape", harga: 6000.0, jumlah: 1);
26        kue[18] = new KueJadi(nama: "Kue Dadar Gulung", harga: 4000.0, jumlah: 2);
27        kue[19] = new KueJadi(nama: "Kue Mendut", harga: 5000.0, jumlah: 1);
28
29        System.out.println("Daftar Kue:");
30        for (Kue instKue : kue) {
31            if (instKue != null) {
32                if (instKue instanceof KuePesanan) {
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57        System.out.println("Total Berat Kue Pesanan: " + totalBeratKuePesanan);
58
59        double totalHargaKueJadi = 0;
60        double totalJumlahKueJadi = 0;
61        for (Kue instKue : kue) {
62            if (instKue instanceof KueJadi) {
63                totalHargaKueJadi += instKue.hitungHarga();
64                totalJumlahKueJadi += ((KueJadi) instKue).getJumlah();
65            }
66        }
67        System.out.println("Total Harga Kue Jadi: " + totalHargaKueJadi);
68        System.out.println("Total Jumlah Kue Jadi: " + totalJumlahKueJadi);
69
70        double hargaMax = 0;
71        Kue kueTerbesar = null;
72        for (Kue instKue : kue) {
73            if (instKue != null && instKue.hitungHarga() > hargaMax) {
74                hargaMax = instKue.hitungHarga();
75                kueTerbesar = instKue;
76            }
77        }
78        if (kueTerbesar != null) {
79            System.out.println("Kue dengan harga terbesar: \n" + kueTerbesar);
80        }
81    }
82 }

```

```

PS C:\Users\Lenovo\.vscode\johanbernardd> & 'C:\Program Files\Java\jdk-17.0.5\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Lenovo\AppData\Roaming\Code\User\workspaceStorage\b60195e65e2688377094ded7d4ac6920\redhat.java\jdt_ws\johanbernardd_14be73ac\bin' 'TugasPrakPolymorphism.MainKue'
Daftar Kue:
Kue Pesanan -> Kue Lapis, Harga: 8000.0/kue
Kue Pesanan -> Kue Klepon, Harga: 6000.0/kue
Kue Pesanan -> Kue Onde-onde, Harga: 5000.0/kue
Kue Pesanan -> Kue Lumpur, Harga: 7000.0/kue
Kue Pesanan -> Kue Ketan Hitam, Harga: 4000.0/kue
Kue Pesanan -> Kue Putu Ayu, Harga: 7000.0/kue
Kue Pesanan -> Kue Nagasari, Harga: 10000.0/kue
Kue Pesanan -> Kue Serabi, Harga: 10000.0/kue
Kue Pesanan -> Kue Ubi Ungu, Harga: 7000.0/kue
Kue Pesanan -> Kue Lepet, Harga: 8000.0/kue
Kue Jadi -> Kue Talam, Harga: 3000.0/kue
Kue Jadi -> Kue Lupis, Harga: 4000.0/kue
Kue Jadi -> Kue Getuk, Harga: 3000.0/kue
Kue Jadi -> Kue Bingka Ubi, Harga: 6000.0/kue
Kue Jadi -> Kue Cucur, Harga: 4000.0/kue
Kue Jadi -> Kue Pancong, Harga: 7000.0/kue
Kue Jadi -> Kue Pukis, Harga: 8000.0/kue
Kue Jadi -> Kue Tape, Harga: 6000.0/kue
Kue Jadi -> Kue Dadar Gulung, Harga: 4000.0/kue
Kue Jadi -> Kue Mendut, Harga: 5000.0/kue
Total Harga semua jenis kue adalah 281000.0
Total Harga Kue Pesanan: 95000.0
Total Berat Kue Pesanan: 13.0
Total Harga Kue Jadi: 186000.0
Total Jumlah Kue Jadi: 18.0
Kue dengan harga terbesar:
Kue Pancong, Harga: 7000.0/kue
PS C:\Users\Lenovo\.vscode\johanbernardd>
  
```