

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 2**



CLASS DAN OBJEK

Oleh:

Ryan Muhammad Irfan NIM. 2210817310013

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
OKTOBER 2023**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 2

Laporan Praktikum Pemrograman II Modul 2: Class dan Objek ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Ryan Muhammad Irfan
NIM : 2210817310013

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Bachrul Uluum
NIM. 2010817210025

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 1 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program.....	9
C. Pembahasan.....	10
D. Tautan GIT.....	15
SOAL 2.....	16
A. Source Code	17
B. Output Program.....	18
C. Pembahasan.....	18
D. Tautan GIT.....	22
SOAL 3.....	23
A. Source Code	25
B. Output Program.....	26
C. Pembahasan.....	27
D. Tautan GIT.....	30

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	9
Gambar 2. Screenshot Hasil Jawaban Soal 2	18
Gambar 3. Screenshot Hasil Jawaban Soal 3	26

DAFTAR TABEL

Tabel 1. Soal 1 (Output)	6
Tabel 2. Source Code Soal 1(Buah.java).....	7
Tabel 3. Source Code Soal 1 (Soal1Main.java).....	8
Tabel 4. Soal 2 (Code).....	16
Tabel 5. Soal 2 (Output)	16
Tabel 6. Source Code Soal 2 (Kopi.java)	17
Tabel 7. Source Code Soal 2 (Soal2Main.java).....	17
Tabel 8. Soal 3 (Pegawai.java)	23
Tabel 9. Soal 3 (Soal3Main.java)	24
Tabel 10. Soal 3 (Output)	24
Tabel 11. Source Code Soal 3 (Pegawai.java).....	25
Tabel 12. Source Code Soal 3 (Soal3Main.java).....	26

SOAL 1

Reno membeli buah-buahan di toko buah. Setiap buah memiliki diskon atau potongan harga 2% per 4kg. Apel memiliki harga Rp7.000 per 0.4kg. Mangga memiliki harga Rp3500 per 0.2kg. Alpukat memiliki harga Rp10.000 per 0.25kg. Dari informasi tersebut, buatlah sebuah program dalam bahasa java yang mengimplementasikan konsep class dan object. Ketentuan : Di class main terdapat 3 buah instansiasi objek dan tidak boleh ada operasi apapun selain instansiasi objek dan memanggil method

Output
Nama Buah: Apel Berat: 0.4 Harga: 7000.0 Jumlah Beli: 40.0kg Harga Sebelum Diskon: Rp700000.00 Total Diskon: Rp5600.00 Harga Setelah Diskon: Rp694400.00
Nama Buah: mangga Berat: 0.2 Harga: 3500.0 Jumlah Beli: 15.0kg Harga Sebelum Diskon: Rp262500.00 Total Diskon: Rp840.00 Harga Setelah Diskon: Rp261660.00
Nama Buah: alpukat Berat: 0.25 Harga: 10000.0 Jumlah Beli: 12.0kg Harga Sebelum Diskon: Rp480000.00 Total Diskon: Rp2400.00 Harga Setelah Diskon: Rp477600.00

Tabel 1. Soal 1 (Output)

Simpan coding anda dengan nama FOLDER: **PRAK201-NIM-Nama**

A. Source Code

Buah.java	
1	public class Buah {
2	String nama;
3	double berat;
4	double harga;
5	double jumlah;
6	double diskon;
7	double harga_setelah_diskon;
8	double harga_sebelum_diskon;
9	
10	public Buah(String nama, double berat, double harga ,
	double jumlah){
11	this.nama = nama;
12	this.berat = berat;
13	this.harga = harga;
14	this.jumlah = jumlah;
15	}
16	
17	public void menghitung_harga_buah(){
18	harga_sebelum_diskon = (harga/berat) * jumlah;
19	diskon = Math.floor(jumlah/4) * 0.02 * (4 *
	harga);
20	harga_setelah_diskon = harga_sebelum_diskon -
	diskon;
21	System.out.printf("Harga Sebelum Diskon:
	Rp%.2f\n", harga_sebelum_diskon);
22	System.out.printf("Total Diskon: Rp%.2f\n",
	diskon);
23	System.out.printf("Harga Setelah Diskon:
	Rp%.2f\n\n", harga_setelah_diskon);
24	}
25	
26	public void info(){
27	System.out.println("Nama buah: " +nama);
28	System.out.println("Berat: " +berat);
29	System.out.println("Harga: " +harga);
30	System.out.println("Jumlah Beli: " +jumlah+"kg");
31	menghitung_harga_buah();
32	}
33	}

Tabel 2. Source Code Soal 1(Buah.java)

Soal1Main.java	
1	public class Soal1Main {
2	public static void main(String[] args) {
3	Buah buah1 = new Buah("Apel", 0.4, 7000, 40);
4	buah1.info();
5	
6	Buah buah2 = new Buah("Mangga", 0.2, 3500, 15);
7	buah2.info();
8	
9	Buah buah3 = new Buah("Alpukat", 0.25, 10000, 12);
10	buah3.info();
11	}
12	}

Tabel 3. Source Code Soal 1 (Soal1Main.java)

B. Output Program



```
Run Soal1Main x
"C:\Users\Ryan muhammad irfan\.jdk\open
Nama buah: Apel
Berat: 0.4
Harga: 7000.0
Jumlah Beli: 40.0kg
Harga Sebelum Diskon: Rp700000.00
Total Diskon: Rp5600.00
Harga Setelah Diskon: Rp694400.00

Nama buah: Mangga
Berat: 0.2
Harga: 3500.0
Jumlah Beli: 15.0kg
Harga Sebelum Diskon: Rp262500.00
Total Diskon: Rp840.00
Harga Setelah Diskon: Rp261660.00

Nama buah: Alpukat
Berat: 0.25
Harga: 10000.0
Jumlah Beli: 12.0kg
Harga Sebelum Diskon: Rp480000.00
Total Diskon: Rp2400.00
Harga Setelah Diskon: Rp477600.00

Process finished with exit code 0
```

Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Buah.java

Pada baris [1-33], `public class Buah`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Buah.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2], `String nama;`

“String” merupakan tipe data dari atribut nama.

“nama” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `double berat;`

“double” merupakan tipe data dari atribut berat.

“berat” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `double harga;`

“double” merupakan tipe data dari atribut harga.

“harga” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5], `double jumlah;`

“double” merupakan tipe data dari atribut jumlah.

“jumlah” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `double diskon;`

“double” merupakan tipe data dari atribut diskon.

“diskon” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `double harga_setelah_diskon;`

“double” merupakan tipe data dari atribut harga_setelah_diskon.

“harga_setelah diskon” merupakan atribut dari sebuah class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8], `double harga_sebelum_diskon;`
“double” merupakan tipe data dari atribut `harga_sebelum_diskon`.

“`harga_sebelum_diskon`” merupakan atribut dari sebuah class Buah.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10-15], `public Buah(String nama, double berat, double harga , double jumlah){...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“`Buah(String nama, double berat, double harga , double jumlah)`” merupakan sebuah constructor dari class buah. Constructor ini menerima parameter String nama, double berat, double harga, dan double jumlah.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [11], `this.nama = nama;`

“`this.nama = nama`” berfungsi untuk menginisialisasi atribut “nama” dengan nilai yang diterima oleh parameter “nama”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12], `this.berat = berat;`

“`this.berat = berat`” berfungsi untuk menginisialisasi atribut “berat” dengan nilai yang diterima melalui parameter “berat”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13], `this.harga = harga;`

“`this.harga = harga`” berfungsi untuk menginisialisasi atribut “harga” dengan nilai yang diterima melalui parameter “harga”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `this.jumlah = jumlah;`

“`this.jumlah = jumlah`” berfungsi untuk menginisialisasi atribut “jumlah” dengan nilai yang diterima melalui parameter “jumlah”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17-24], `public void menghitung_harga_buah() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“`menghitung_harga_buah()`” merupakan sebuah method dengan nama `menghitung_harga_buah`. Di dalam method ini terdapat kalkulasi dan print yang dapat dipanggil pada method yang lain atau pada class lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [18], `harga_sebelum_diskon = (harga/berat) * jumlah;`
Berfungsi untuk menyatakan value dari `harga_sebelum_diskon = (value harga/ value berat) * value jumlah`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [19], `diskon = Math.floor(jumlah/4) * 0.02 * (4 * harga);`
Berfungsi untuk menyatakan value dari `diskon = Math.floor (value jumlah / 4) * 0.02 (4 * value harga)`.

“Math.floor()” berfungsi membulatkan suatu bilangan desimal ke bawah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [20], `harga_setelah_diskon = harga_sebelum_diskon - diskon;`
Berfungsi untuk menyatakan value dari `harga_setelah_diskon = value harga_sebelum_diskon – value diskon;`

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21], `System.out.printf("Harga Sebelum Diskon: Rp%.2f\n", harga_sebelum_diskon);`

“System.out.printf(…)” berfungsi untuk mencetak angka atau karakter dengan format tertentu. Dalam baris ini mencetak "Harga Sebelum Diskon: Rp%.2f\n", value `harga_sebelum_diskon`

“%.2f” berfungsi untuk mengatur tampilan bilangan desimal 2 angka dibelakang koma.

“\n” berfungsi untuk pindah ke baris berikutnya (newline).

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22], `System.out.printf("Total Diskon: Rp%.2f\n", diskon);`

“System.out.printf(…)” berfungsi untuk mencetak angka atau karakter dengan format tertentu. Dalam baris ini mencetak "Total Diskon: Rp%.2f\n", value `diskon`

“%.2f” berfungsi untuk mengatur tampilan bilangan desimal 2 angka dibelakang koma.

“\n” berfungsi untuk pindah ke baris berikutnya (newline).

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [23], `System.out.printf("Harga Setelah Diskon: Rp%.2f\n\n", harga_setelah_diskon);`

“System.out.printf(…)” berfungsi untuk mencetak angka atau karakter dengan format tertentu. Dalam baris ini mencetak "Harga Setelah Diskon: Rp%.2f\n", value `harga_setelah_diskon`

“%.2f” berfungsi untuk mengatur tampilan bilangan desimal 2 angka dibelakang koma.

“\n” berfungsi untuk pindah ke baris berikutnya (newline).

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26-32], `public void info() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“info()” merupakan sebuah method dengan nama info. Di dalam method ini terdapat print dan pemanggilan dari method yang lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [27], `System.out.println("Nama buah: " +nama);`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Nama buah: " + value dari atribut nama.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [28], `System.out.println("Berat: " +berat);`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "berat: " + value dari atribut berat.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [29], `System.out.println("harga: " +harga);`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "harga: " + value dari atribut harga.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [30], `System.out.println("Jumlah Beli: " +jumlah+"kg");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Jumlah Beli: " + value dari atribut jumlah + “kg”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [31], `menghitung_harga_buah();`

“menghitung_harga_buah()” berfungsi untuk memanggil method menghitung_harga_buah

“;” berfungsi untuk mengakhiri sebuah perintah.

Soal1Main.java

Pada baris [1-12], `public class Soal1Main`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Soal1Main.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2-11], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [3], `Buah buah1 = new Buah("Apel", 0.4, 7000, 40);`

“Buah” merupakan sebuah class.

“buah1” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Buah.

“new Buah” membuat objek baru dari kelas buah

“("Apel", 0.4, 7000, 40)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter nama: “Apel”, berat: 0.4 , harga: 7000, jumlah: 40.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `buah1.info();`

“buah1.info” berfungsi untuk memanggil method info pada objek “buah1” dari class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `Buah buah2 = new Buah("Mangga", 0.2, 3500, 15);`

“Buah” merupakan sebuah class.

“buah2” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Buah.

“new Buah” membuat objek baru dari kelas buah

“("Mangga", 0.2, 3500, 15)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter nama: “Mangga”, berat: 0.2 , harga: 3500, jumlah: 15 .

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `buah2.info()` ;

“buah2.info” berfungsi untuk memanggil method info pada objek “buah2” dari class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9], `Buah buah3 = new Buah("Alpukat", 0.25, 10000, 12);`

“Buah” merupakan sebuah class.

“buah3 ” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Buah.

“new Buah” membuat objek baru dari kelas buah

“(“Alpukat”, 0.25, 10000, 12)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter nama: “Alpukat”, berat: 0.25 , harga: 10000, jumlah: 12 .

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10], `buah3.info()` ;

“buah3.info” berfungsi untuk memanggil method info pada objek “buah3” dari class Buah.

“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%202/PRAK201-2210817310013-RyanMuhammadIrfan>

SOAL 2

Diberikan sebuah class Soal2Main. Soal ini berisi sebuah kode program dalam bahasa java yang digunakan untuk menentukan jumlah pajak yang harus dibayar oleh pelanggan ketika berbelanja di coffe shop. Lengkapilah kode pada program berikut sehingga dapat menghasilkan keluaran seperti yang ditunjukkan pada tabel di bawah.

Ketentuan : Class Soal2Main **tidak boleh** diubah isinya. Pajak dari tiap pembelian adalah 11%.

File: Soal2Main.java
<pre>package praktikum2.soal2; public class Soal2Main { public static void main(String[] args) { Kopi kopi1 = new Kopi(); kopi1.namaKopi = "Espresso"; kopi1.ukuran = "Medium"; kopi1.harga = 25000; kopi1.info(); kopi1.setPembeli("Alice"); System.out.println("Pembeli Kopi: " + kopi1.getPembeli()); System.out.println("Pajak Kopi: Rp. " + kopi1.getPajak()); } }</pre>

Tabel 4. Soal 2 (Code)

Output
Nama Kopi: Espresso Ukuran: Medium Harga: Rp. 25000.0 Pembeli Kopi: Alice Pajak Kopi: Rp. 2750.0

Tabel 5. Soal 2 (Output)

Simpan coding anda dengan nama FOLDER: **PRAK202-NIM-Nama**

A. Source Code

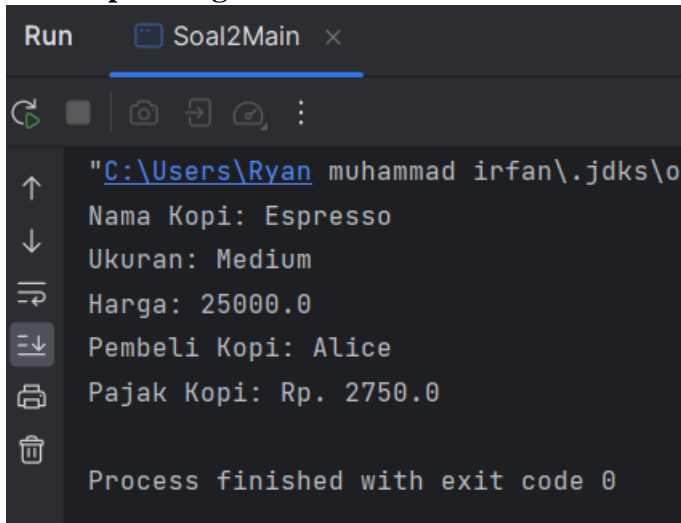
Kopi.java	
1	public class Kopi {
2	public String namaKopi;
3	public String ukuran;
4	public double harga;
5	private double pajak;
6	private String Pembeli;
7	
8	public String getPembeli() {
9	return Pembeli;
10	}
11	
12	public void setPembeli(String pembeli){
13	Pembeli = pembeli;
14	}
15	
16	public double getPajak() {
17	pajak = harga * 0.11;
18	return pajak;
19	}
20	
21	public void info() {
22	System.out.println("Nama Kopi: " + namaKopi);
23	System.out.println("Ukuran: " + ukuran);
24	System.out.println("Harga: " + harga);
25	}
26	
27	}
28	

Tabel 6. Source Code Soal 2 (Kopi.java)

Soal2Main.java	
1	public class Soal2Main {
2	public static void main(String[] args) {
3	Kopi kopil = new Kopi();
4	kopil.namaKopi = "Espresso";
5	kopil.ukuran = "Medium";
6	kopil.harga = 25000;
7	kopil.info();
8	kopil.setPembeli("Alice");
9	System.out.println("Pembeli Kopi: " +
10	kopil.getPembeli());
10	System.out.println("Pajak Kopi: Rp. " +
11	kopil.getPajak());
11	}
12	}

Tabel 7. Source Code Soal 2 (Soal2Main.java)

B. Output Program



```
Run Soal2Main x
"C:\Users\Ryan muhammad irfan\.jdk\src\main\java\com\example\Soal2Main.java:1: error: package org.example does not exist
Nama Kopi: Espresso
Ukuran: Medium
Harga: 25000.0
Pembeli Kopi: Alice
Pajak Kopi: Rp. 2750.0
Process finished with exit code 0
```

Gambar 2. Screenshot Hasil Jawaban Soal 2

C. Pembahasan

Kopi.java

Pada baris [1-27], `public class Kopi`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Kopi.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2], `public String namaKopi;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“String” merupakan tipe data dari atribut namaKopi.

“namaKopi” merupakan atribut dari sebuah class Kopi.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `public String ukuran;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“String” merupakan tipe data dari atribut ukuran.

“ukuran” merupakan atribut dari sebuah class Kopi.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `public double harga;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“double” merupakan tipe data dari atribut harga.

“harga” merupakan atribut dari sebuah class Kopi.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5], `private double pajak;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“double” merupakan tipe data dari atribut pajak.

“pajak” merupakan atribut dari sebuah class Kopi.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `private double pembeli;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“double” merupakan tipe data dari atribut pembeli.

“pembeli” merupakan atribut dari sebuah class Kopi.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8-10], `public String getPembeli() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getPembeli()`.

“`getPembeli()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut Pembeli, yang dideklarasikan sebagai atribut private.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [9], `return Pembeli;`

“`return Pembeli`” berfungsi untuk mengembalikan nilai atribut Pembeli ketika method `getPembeli()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12-14], `public void setPembeli(String pembeli){...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak mengembalikan nilai.

“`setPembeli()`” merupakan method setter. Digunakan untuk mengatur atau mengubah nilai atribut yang private.

“(String pembeli)” merupakan parameter dari method setter. String adalah tipe datanya. Parameter ini digunakan untuk mengatur nilai atribut “Pembeli”

“{...}” berfungsi untuk memulai dan mengakhiri blok kode

Pada baris [13], `Pembeli = pembeli;`

“`Pembeli = pembeli`” berfungsi untuk menyatakan nilai atribut `Pembeli` diberikan melalui parameter `pembeli`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16-19], `public String getPajak() {...}`

“`public`” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“`String`” merupakan tipe data yang dikembalikan oleh method `getPembeli()`.

“`getPajak()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut `pajak`, yang dideklarasikan sebagai atribut `private`.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [17], `pajak = harga * 0.11;`

“`pajak = harga * 0.11`” Berfungsi untuk menyatakan value dari `pajak = value harga * 0.11`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [18], `return pajak;`

“`return pajak`” berfungsi untuk mengembalikan nilai atribut `pajak` ketika method `getPembeli()` dipanggil.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21-25], `public void info() {...}`

“`public`” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“`void`” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“`info()`” merupakan sebuah method dengan nama `info`. Di dalam method ini terdapat `print`.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [22], `System.out.println("Nama Kopi: " + namaKopi);`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak “`Nama Kopi:` ” + value dari atribut `namaKopi`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [23], `System.out.println("Ukuran: " + ukuran);`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak “`Ukuran:` ” + value dari atribut `ukuran`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [24], `System.out.println("Harga: " + harga);`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Harga: " + value dari atribut harga.

“;” berfungsi untuk mengakhiri sebuah perintah.

Soal2Main.java

Pada baris [1-12], `Soal2Main`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama `Soal2Main`.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2-11], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [3], `Kopi kopi1 = new Kopi();`

“Kopi” merupakan sebuah class.

“Kopi1” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class `Kopi`.

“new Kopi” membuat objek baru dari kelas `Kopi`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `kopi1.namaKopi = "Espresso";`

“kopi1.namaKopi” Berfungsi untuk mengatur nilai atribut `namaKopi` dengan objek `kopi1` Diatur dengan nilai “Espresso”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5], `kopi1.harga = "25000";`

“kopi1.harga” Berfungsi untuk mengatur nilai atribut `harga` dengan objek `kopi1` Diatur dengan nilai “2500”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `kopi1.info()` ;

“`kopi1.info`” berfungsi untuk memanggil method `info` pada objek “`kopi1`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `kopi1.setPembeli("Alice")` ;

“`kopi1.setPembeli`” Berfungsi untuk mengatur nilai atribut `private Pembeli` dengan pemanggilan method setter “`setPembeli(String pembeli)`” pada objek `kopi1` Diatur dengan nilai “`Alice`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8],

`System.out.println("Pembeli Kopi: " + kopi1.getPembeli());`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak “`Pembeli:` ” + value `private Pembeli` dengan objek `kopi1` diambil melalui method getter “`getPembeli()`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9],

`System.out.println("Pajak Kopi: Rp. " + kopi1.getPajak());`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak “`Pajak Kopi: Rp.` ” + value `private pajak` dengan objek `kopi1` diambil melalui method getter “`getPajak()`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%202/PRAK202-2210817310013-RyanMuhammadIrfan>

SOAL 3

Diberikan 2 buah file yaitu Soal3Main.java dan Pegawai.java. Temukan dan Perbaiki kesalahan yang terdapat pada kedua file tersebut sehingga program dapat dijalankan dan menghasilkan keluaran seperti yang diminta. Baris kode yang terindikasi memiliki error dijadikan komentar dan berikan penjelasan diatas baris kode yang terdapat kesalahan. Isi penjelasan berupa penyebab/alasan kenapa terjadi error. Baris berikutnya merupakan hasil perbaikan untuk mengatasi error yang terjadi. Misal pada salah satu baris kode,

```
//Pada baris ini terjadi error karena kurangnya titik koma (;)
//public String nama
public String nama;
```

Catatan: **Jika tidak ada komentar/penjelasan tentang** error yang terjadi, meskipun program berjalan lancar dan menghasilkan keluaran sesuai yang diminta, maka **tidak akan dinilai**.

File: Pegawai.java
<pre>package praktikum2.soal3; public class Employee { public String nama; public char asal; public String jabatan; public int umur; public String getNama() { return nama; } public String getAsal() { return asal; } public void setJabatan() { this.jabatan = j; } }</pre>

Tabel 8. Soal 3 (Pegawai.java)

File: Soal3Main.java

```
package praktikum2.soal3;

public class Soal3Main {
    public static void main(String[] args) {
        Pegawai p1 = new Pegawai();
        p1.nama = "Roi"
        p1.asal = "Kingdom of Orvel";
        p1.setJabatan("Assasin");
        System.out.println("Nama Pegawai: " + p1.getNama());
        System.out.println("Asal: " + p1.getAsal());
        System.out.println("Jabatan: " + p1.jabatan);
        System.out.println("Umur: " + p1.umur);
    }
}
```

*Tabel 9. Soal 3 (Soal3Main.java)***Output**

Nama: Roi
Asal: Kingdom of Orvel
Jabatan: Assasin
Umur: 17 tahun

Tabel 10. Soal 3 (Output)

Simpan coding anda dengan nama FOLDER: **PRAK203-NIM-Nama**

A. Source Code

Pegawai.java	
1	public class Pegawai {
2	public String nama;
3	//Pada baris ini terjadi error karena variabel asal bertipe data char tidak sesuai dengan method. Sehingga variabel asal di ubah menjadi String.
4	// public char asal;
5	public String asal;
6	public String jabatan;
7	public int umur;
8	public String getNama() {
9	return nama;
10	}
11	public String getAsal() {
12	return asal;
13	}
14	//Pada baris ini terjadi error karena tidak menerima argumen atau parameter . Sehingga perlu ditambahkan argumen yang sesuai dengan tipe data dari parameter yaitu String.
15	//public void setJabatan()
16	public void setJabatan(String j) {
17	this.jabatan = j;
18	}
19	}

Tabel 11. Source Code Soal 3 (Pegawai.java)

Soal3Main.java	
1	public class Soal3Main {
2	public static void main(String[] args) {
3	Pegawai p1 = new Pegawai();
4	//Pada baris ini terjadi error karena kurangnya titik koma (;)
5	//p1.nama = "Roi"
6	p1.nama = "Roi";
7	p1.asal = "Kingdom of Orvel";
8	p1.setJabatan("Assasin");
9	//p1.umur tidak memiliki nilai sehingga harus diisi 17 untuk menghasilkan output yang sesuai
10	p1.umur = 17;
11	//Pada baris ini harus nya mengasilkan output Nama bukan Nama Pegawai. "Nama Pegawai" idubah menjadi "Nama"
12	//System.out.println("Nama Pegawai: " + p1.getNama());
13	System.out.println("Nama: " + p1.getNama());
14	System.out.println("Asal: " + p1.getAsal());
15	System.out.println("Jabatan: " + p1.jabatan);
16	//Menambahkan text "tahun" dalam print setelah nilai p1.umur agar sesuai dengan output
17	//System.out.println("Umur: " + p1.umur);
18	System.out.println("Umur: " + p1.umur + " Tahun");
19	}
20	}

Tabel 12. Source Code Soal 3 (Soal3Main.java)

B. Output Program

```

Run    Soal3Main x
C:\Users\Ryan muhammad irfan\.jdk\openjdk-20.0.2\bin\java.exe
Nama: Roi
Asal: Kingdom of Orvel
Jabatan: Assasin
Umur: 17 Tahun
Process finished with exit code 0

```

Gambar 3. Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Pegawai.java

Pada baris [1-19], `public class Pegawai`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Pegawai.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2], `public String nama;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“String” merupakan tipe data dari atribut nama.

“namaKopi” merupakan atribut dari sebuah class Pegawai.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3-4], Merupakan penjelasan tentang error yang terjadi.

Pada baris [5], `public String asal;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“String” merupakan tipe data dari atribut asal.

“asal” merupakan atribut dari sebuah class Pegawai.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `public String jabatan;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“String” merupakan tipe data dari atribut jabatan.

“jabatan” merupakan atribut dari sebuah class Pegawai.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `public int umur;`

“public” berfungsi sebagai penanda bahwa atribut dapat diakses dari mana saja.

“int” merupakan tipe data dari atribut umur.

“umur” merupakan atribut dari sebuah class Pegawai.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8-10], `public String getNama() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getPembeli()`.

“getNama()” merupakan method getter. Digunakan untuk mengambil nilai dari atribut nama.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [9], `return nama;`

“return nama” berfungsi untuk mengembalikan nilai atribut nama ketika method getNama() dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11-13], `public String getAsal() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method getAsal().

“getAsal()” merupakan method getter. Digunakan untuk mengambil nilai dari atribut asal.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [12], `return asal;`

“return asal” berfungsi untuk mengembalikan nilai atribut asal ketika method getAsal () dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14-15], Merupakan penjelasan tentang error yang terjadi.

Pada baris [16-18], `public void setJabatan(String j) {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak mengembalikan nilai.

“setPembeli()” merupakan method setter. Digunakan untuk mengatur atau mengubah nilai atribut jabatan.

“(String j) merupakan parameter dari method setter. String adalah tipe datanya. Parameter ini digunakan untuk mengatur nilai atribut “jabatan”

“{...}” berfungsi untuk memulai dan mengakhiri blok kode

Pada baris [17], `this.jabatan = j;`

“this.jabatan = j” berfungsi untuk mengambil nilai yang diberikan parameter j dan mengaturnya ke dalam atribut jabatan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Soal3Main.java

Pada baris [1-20], `Soal3Main`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Soal3Main.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [2-19], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [3], `Pegawai p1 = new Pegawai();`

“Pegawai” merupakan sebuah class.

“p1” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Pegawai.

“new Pegawai” membuat objek baru dari kelas Pegawai

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4-5], Merupakan penjelasan tentang error yang terjadi

Pada baris [6], `p1.nama = "Roi";`

“p1.nama” Berfungsi untuk mengatur nilai atribut nama dengan objek p1. Diatur dengan nilai “Roi”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `p1.asal = "Kingdom of Orvel";`

“p1.asal” Berfungsi untuk mengatur nilai atribut asal dengan objek p1 Diatur dengan nilai “Kingdom of Orvel”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8], `p1.setJabatan("Assasin");`

“p1.setJabatan” Berfungsi untuk mengatur nilai atribut jabatan dengan pemanggilan method setter “setJabatan (String j)” pada objek p1 Diatur dengan nilai “Assasin”.

Pada baris [9], Merupakan penjelasan perbaikan agar sesuai dengan output soal.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10], `p1.umur = "17";`

“p1.umur” Berfungsi untuk mengatur nilai atribut umur dengan objek p1 Diatur dengan nilai “17”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11-12], Merupakan penjelasan perbaikan agar sesuai dengan output soal.

Pada baris [13], `System.out.println("Nama: " + p1.getNama());`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Nama: " + value nama dengan objek p1 diambil melalui method getter “getNama()”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `System.out.println("Asal : " + p1.getAsal());`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Asal : " + value asal dengan objek p1 diambil melalui method getter “getAsal()”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `System.out.println("Jabatan: " + p1.jabatan);`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Jabatan : " + value jabatan dengan objek p1.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16-17], Merupakan penjelasan perbaikan agar sesuai dengan output soal.

Pada baris [18], `System.out.println("Umur: " + p1.umur + " Tahun");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Umur: " + value umur dengan objek p1 + “ Tahun”.

“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%202/PRAK203-2210817310013-RyanMuhammadIrfan>