

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : STATIC MODIFIER DAN KONSTANTA FINAL
NAMA : Dani Adrian
NIM : 225150201111009
ASISTEN : Tengku Muhammad Rafi Rahardiansyah
Muhammad Bin Djafar Almasyhur
TGL PRAKTIKUM : 3 Mei 2023

A. Static Method**Data dan Hasil Percobaan****Pertanyaan**

1. Apakah yang disebut dengan static variable? Dan apa fungsi dari static variable serta kapan kita dapat menggunakan static variable?

Source Code

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

Output**Penjelasan**

Static variable adalah variabel yang dideklarasikan di dalam sebuah fungsi atau blok kode, tetapi nilainya tetap sama selama program berjalan. Fungsi dari static variable adalah untuk menyimpan informasi yang dapat diakses oleh fungsi yang sama setiap kali fungsi tersebut dipanggil.

Kita dapat menggunakan static variable pada saat kita ingin menghitung atau menyimpan informasi yang berkaitan dengan fungsi tertentu, misalnya ketika kita ingin menghitung berapa kali suatu fungsi dipanggil atau menyimpan data yang bersifat global di dalam fungsi.

Pertanyaan

2. Mengapa pada main method harus dituliskan static? Jelaskan jawaban anda beserta dengan alasan!

Source Code

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

Output**Penjelasan**

Untuk Memastikan bahwa metode utama dapat dipanggil oleh main java tanpa membuat objek terlebih dahulu agar dapat dipanggil dari class lain atau main method java harus dituliskan sebagai static.

Pertanyaan

3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Source Code

Aritmatika.java	
1.	public class Aritmatika {
2.	public void hitungPenjumlahan (int a, int b) {
3.	int nilai = a+b;
4.	System.out.println("nilai penjumlahan adalah : "+nilai);
5.	}
6.	public static void hitungPerkalian(int a, int b){
7.	int nilai = a*b;
8.	System.out.println("nilai perkalian adalah : "+nilai);
9.	}
10.	
11.	public static void hitungPengurangan(int a, int b){
12.	int nilai = a-b;
13.	System.out.println("nilai pengurangan adalah : "+nilai);
14.	}
15.	}

Ouput

```

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\praktikumkedelapan> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\worksp
aceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'MainAritmatika'
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai pengurangan adalah : -1
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai perkalian adalah : 2
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai penjumlahan adalah : 3
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>

```

Dani Adrian.txt

Dani Adrian
225150201111009

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

Penjelasan

Program tidak mengalami error

Pertanyaan

4. Jika pada tubuh method hitungPenjumlahan ditambahkan syntax *hitungPerkalian(a,b)* apa yang terjadi? Jelaskan?

Source code

Aritmatika.java	
1.	public class Aritmatika {
2.	public void hitungPenjumlahan (int a, int b) {
3.	int nilai = a+b;
4.	System.out.println("nilai penjumlahan adalah : "+nilai);
5.	hitungPerkalian (a, b);
6.	}
7.	public static void hitungPerkalian(int a, int b){
8.	int nilai = a*b;
9.	System.out.println("nilai perkalian adalah : "+nilai);
10.	}
11.	public static void hitungPengurangan(int a, int b){
12.	int nilai = a-b;
13.	System.out.println("nilai pengurangan adalah : "+nilai);

14.	}
15.	}

Output

The screenshot shows a terminal window with the following output:

```
> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java
Aritmatika'
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai pengurangan adalah : -1
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai perkalian adalah : 2
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai penjumlahan adalah : 3
nilai perkalian adalah : 2
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>
```

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

```
Dani Adrian
225150201111009
```

Penjelasan

Dikarenakan pada hitungPenjumlahan terdapat instruksi untuk memerintahkan hitungPerkalian maka method hitungPerkalian akan dipanggil saat method hitungPenjumlahan dipanggil di main.

Pertanyaan

5. Jika pada tubuh method hitungPerkalian ditambahkan syntax *hitungPenjumlahan(a,b)* apa yang terjadi? Jelaskan?

Source Code

Aritmatika.java	
1.	public class Aritmatika {
2.	public void hitungPenjumlahan (int a, int b) {
3.	int nilai = a+b;
4.	System.out.println("nilai penjumlahan adalah : "+nilai);
5.	hitungPerkalian (a, b);
6.	}
7.	public static void hitungPerkalian(int a, int b){
8.	int nilai = a*b;
9.	System.out.println("nilai perkalian adalah : "+nilai);
10.	hitungPenjumlahan (a, b);
11.	}
12.	public static void hitungPengurangan(int a, int b){
13.	int nilai = a-b;
14.	System.out.println("nilai pengurangan adalah : "+nilai);

15.	}
16.	}

Output

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan> cd "C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan"; & "C:\Program Files\Java\jdk1.8.0_111\bin\java
.exe" "-cp" "C:\Users\USER\AppData\Roaming\Code\User\workspacestorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdk_ws\praktikumkedelapan_9b8ae742\bin" "Main
Aritmatika"
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai pengurangan adalah : -1
masukkan nilai 1 : 1
masukkan nilai 2 : 2
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
  Cannot make a static reference to the non-static method hitungPenjumlahan(int, int) from the type Aritmatika

  at Aritmatika.hitungPerkalian(Aritmatika.java:10)
  at MainAritmatika.main(MainAritmatika.java:16)PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>
```

Penjelasan

Terjadi error, dikarenakan jika method hitungPenjumlahan bukan static method. Jika ingin memanggil method di dalam method yang static maka harus mengubah method tersebut menjadi static method.

Pertanyaan

6. Tambahkan method non static dengan nilai balikan double untuk menghitung pembagian dengan parameter String nil dan String nil2, dan panggil method tersebut pada method main!

Source Code

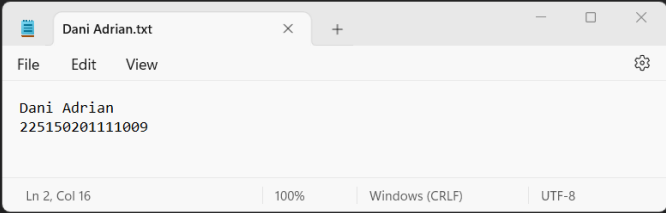
Aritmatika.java	
1.	public class Aritmatika {
2.	public void hitungPenjumlahan (int a, int b) {
3.	int nilai = a+b;
4.	System.out.println("nilai penjumlahan adalah : "+nilai);
5.	hitungPerkalian (a, b);
6.	}
7.	public static void hitungPerkalian(int a, int b){
8.	int nilai = a*b;
9.	System.out.println("nilai perkalian adalah : "+nilai);
10.	//hitungPenjumlahan (a, b);
11.	}
12.	public static void hitungPengurangan(int a, int b){
13.	int nilai = a-b;

14.	System.out.println("nilai pengurangan adalah :
15.	"+nilai);
16.	}
17.	public double hitungPembagian (String nil, String
18.	nil2) {
19.	return
20.	Double.parseDouble(nil)/Double.parseDouble(nil2);
	}
	}

MainAritmatika.java	
1.	import java.util.Scanner;
2.	public class MainAritmatika {
3.	public static void main(String[] args) {
4.	Scanner in = new Scanner(System.in);
5.	System.out.print("masukkan nilai 1 : ");
6.	int nil1 = in.nextInt();
7.	System.out.print("masukkan nilai 2 : ");
8.	int nil2 = in.nextInt();
9.	//memanggil method static
10.	Aritmatika.hitungPengurangan(nil1, nil2);
11.	System.out.print("masukkan nilai 1 : ");
12.	nil1 = in.nextInt();
13.	System.out.print("masukkan nilai 2 : ");
14.	nil2 = in.nextInt();
15.	//memanggil method static
16.	Aritmatika.hitungPerkalian(nil1, nil2);
17.	System.out.print("masukkan nilai 1 : ");
18.	int value1 = in.nextInt();
19.	System.out.print("masukkan nilai 2 : ");
20.	int value2 = in.nextInt();
21.	//memanggil method NONstatic harus melalui objek
22.	Aritmatika a = new Aritmatika();
23.	a.hitungPenjumlahan(value1, value2);
24.	System.out.print("masukkan nilai 1: ");
25.	String lol1 = in.next();
26.	System.out.print("masukkan nilai 2: ");
27.	String lol12 = in.next();
28.	System.out.println("hasil pembagian adalah :
29.	"+a.hitungPembagian(lol1, lol12));
30.	}
	}

Output

```
.exe' -cp 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdk_ws\praktikumkedelapan_9b8ae742\bin' 'Main
Aritmatika'
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai pengurangan adalah : -1
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai perkalian adalah : 2
masukkan nilai 1 : 1
masukkan nilai 2 : 2
nilai penjumlahan adalah : 3
nilai perkalian adalah : 2
masukkan nilai 1: 1
masukkan nilai 2: 2
hasil pembagian adalah : 0.5
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>
```



Penjelasan

Method berupa nonstatic, maka tidak menggunakan kata static pada method. Isi method adalah mereturn variable string nil dan nil2 yang terlebih dahulu diubah menjadi double menggunakan Double.parseDouble lalu dibagikan. Untuk menampilkan output pada main kita gunakan instansiasi objek Aritmatika bernama a dan memanggil method hitungPembagian.

B. Konstanta Final

Pertanyaan

1. Benahi Kode Vehicle1 dan TestVehicle1 dan perbaiki jika menemui kesalahan!

Source Code

Vehicle1.java	
1.	public class Vehicle1 {
2.	private double load;
3.	private final double maxLoad = 10000;
4.	
5.	// public Vehicle1 (double max){
6.	// this.maxLoad = max;
7.	// }
8.	public double getLoad() {
9.	return this.load;
10.	}
11.	
12.	public double getMaxLoad() {
13.	return this.maxLoad;
14.	}
15.	
16.	public boolean addBox(double weight) {
17.	double temp = 0.0D;
18.	temp = this.load + weight;
19.	if (temp <= maxLoad) {

20.	this.load = this.load + weight;
21.	return true;
22.	} else {
23.	return false;
24.	}
25.	}
26.	}

TestVehicle.java	
1.	public class TestVehicle {
2.	public static void main(String[] args) {
3.	System.out.println("Creating a vehicle with a
	10,000 kg maximumload.");
4.	
5.	Vehicle1 vehicle = new Vehicle1 ();
6.	// Vehicle1 vehicle2 = new Vehicle1(1000);
7.	
8.	System.out.println("Add box #1 (500kg) : " +
	vehicle.addBox(500));
9.	System.out.println("Add box #2 (250kg) : " +
	vehicle.addBox(250));
10.	System.out.println("Add box #3 (5000kg) : " +
	vehicle.addBox(5000));
11.	System.out.println("Add box #4 (4000kg) : " +
	vehicle.addBox(4000));
12.	System.out.println("Add box #5 (300kg) : " +
	vehicle.addBox(300));
13.	System.out.println("Vehicle load is " +
	vehicle.getLoad() + "kg");
14.	}
15.	}

Output

```

PS C:\> c:\> cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roami
ng\Code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'TestVehicle'
Creating a vehicle with a 10,000 kg maximumload.
Add box #1 (500kg) : true
Add box #2 (250kg) : true
Add box #3 (5000kg) : true
Add box #4 (4000kg) : true
Add box #5 (300kg) : false
Vehicle load is 9750.0kg
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>
  
```

Dani Adrian
225150201111009

Ln 2, Col 16 100% Windows (CRLF) UTF-8

Penjelasan

Tidak terjadi error pada program

Pertanyaan

2. Hapus separator “/” pada file Vehicle1.java pada baris 4-6 serta pada file TestVehicle1.java pada baris 6, apa yang terjadi dan jelaskan!

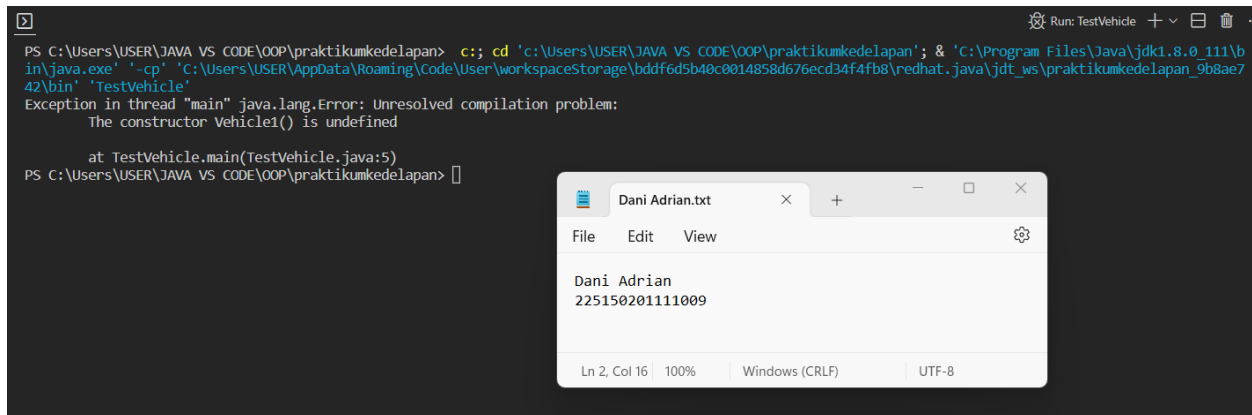
Source Code

Vehicle1.java	
1.	public class Vehicle1 {
2.	private double load;
3.	private final double maxLoad = 10000;
4.	
5.	public Vehicle1 (double max) {
6.	this.maxLoad = max;
7.	}
8.	public double getLoad() {
9.	return this.load;
10.	}
11.	
12.	public double getMaxLoad() {
13.	return this.maxLoad;
14.	}
15.	
16.	public boolean addBox(double weight) {
17.	double temp = 0.0D;
18.	temp = this.load + weight;
19.	if (temp <= maxLoad) {
20.	this.load = this.load + weight;
21.	return true;
22.	} else {
23.	return false;
24.	}
25.	}
26.	}

TestVehicle.java	
1.	public class TestVehicle {
2.	public static void main(String[] args) {
3.	System.out.println("Creating a vehicle with a 10,000 kg maximumload.");
4.	
5.	Vehicle1 vehicle = new Vehicle1 ();
6.	Vehicle1 vehicle2 = new Vehicle1(1000);
7.	

8.	System.out.println("Add box #1 (500kg) : " + vehicle.addBox(500));
9.	System.out.println("Add box #2 (250kg) : " + vehicle.addBox(250));
10.	System.out.println("Add box #3 (5000kg) : " + vehicle.addBox(5000));
11.	System.out.println("Add box #4 (4000kg) : " + vehicle.addBox(4000));
12.	System.out.println("Add box #5 (300kg) : " + vehicle.addBox(300));
13.	System.out.println("Vehicle load is " + vehicle.getLoad() + "kg");
14.	}
15.	}

Output



Penjelasan

Terjadi error, dikarenakan pada TestVehicle.java terdapat pemanggilan konstruktor Vehicle dengan parameter int, namun konstruktor tersebut tidak didefinisikan di kelas Vehicle. Oleh karena itu, Java tidak dapat menyelesaikan kompilasi kode tersebut dan menghasilkan error "The constructor Vehicle(int) is undefined".

Pertanyaan

3. Pada file Vehicle1.java variabel load ubah menjadi konstanta final, apa yang terjadi, jelaskan!

Source Code

Vehicle1.java	
1.	public class Vehicle1 {
2.	private final double load;
3.	Private final double maxLoad = 10000;

4.	
5.	// public Vehicle1 (double max){
6.	// this.maxLoad = max;
7.	// }
8.	public double getLoad() {
9.	return this.load;
10.	}
11.	
12.	public double getMaxLoad() {
13.	return this.maxLoad;
14.	}
15.	
16.	public boolean addBox(double weight) {
17.	double temp = 0.0D;
18.	temp = this.load + weight;
19.	if (temp <= maxLoad) {
20.	this.load = this.load + weight;
21.	return true;
22.	} else {
23.	return false;
24.	}
25.	}
26.	}

Output

The screenshot shows a terminal window with the following output:

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan> c++; cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' -cp 'C:\Users\USER\AppData\Roaming\code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'TestVehicle'
Creating a vehicle with a 10,000 kg maximumload.
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
The blank final field load may not have been initialized
The final field Vehicle1.load cannot be assigned
    at Vehicle1.<init>(Vehicle1.java:2)
    at TestVehicle.main(TestVehicle.java:5)
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>
```

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

```
Dani Adrian
225150201111009
```

The text editor shows the cursor at Line 2, Column 16, with a 100% zoom level, Windows (CRLF) line endings, and UTF-8 encoding.

Penjelasan

Terjadi error, dikarenakan Karena variabel final harus diinisialisasi saat deklarasi atau di dalam konstruktor, maka untuk mengatasi error tersebut perlu memastikan bahwa variabel load diberi nilai awal saat deklarasi atau di dalam konstruktor.

Pertanyaan

4. Pada file Vehicle1.java variabel load ubah menjadi konstanta final, apa yang terjadi, jelaskan!

Source Code

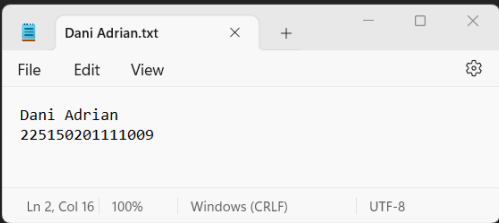
Vehicle1.java	
1.	public class Vehicle1 {
2.	private double load;
3.	Private static final double maxLoad = 10000;
4.	
5.	// public Vehicle1 (double max){
6.	// this.maxLoad = max;
7.	// }
8.	public double getLoad() {
9.	return this.load;
10.	}
11.	
12.	public double getMaxLoad() {
13.	return this.maxLoad;
14.	}
15.	
16.	public boolean addBox(double weight) {
17.	double temp = 0.0D;
18.	temp = this.load + weight;
19.	if (temp <= maxLoad) {
20.	this.load = this.load + weight;
21.	return true;
22.	} else {
23.	return false;
24.	}
25.	}
26.	}

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'TestVehicle'
Creating a vehicle with a 10,000 kg maximumload.
Add box #1 (500kg) : true
Add box #2 (250kg) : true
Add box #3 (5000kg) : true
Add box #4 (4000kg) : true
Add box #5 (300kg) : false
Vehicle load is 9750.0kg
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>

```



Penjelasan

Program tereksekusi tanpa ada error. Namun, pada getMaxLoad, kita bisa mengaksesnya tanpa keyword this. karena sudah berupa static.

Tugas Praktikum

1. Buatlah program dengan menggunakan class untuk menghitung penjumlahan, pengurangan, perkalian dan pembagian. Method penjumlahan dan pengurangan menggunakan static method sedangkan perkalian dan pembagian menggunakan method non static. Tambahkan method bertipe non static bernama Sederhana untuk menyederhanakan sebuah pecahan.

Source Code

MainPraktikum1.java	
1.	public class MainPraktikum1 {
2.	public static void main(String[] args) {
3.	Praktikum1 prk1 = new Praktikum1();
4.	System.out.println("Hasil dari operasi angka 100 dan 2");
5.	Praktikum1.Penjumlahan(100, 2);
6.	Praktikum1.Pengurangan(100, 2);
7.	prk1.Perkalian(100, 2);
8.	prk1.Pembagian(100, 2);
9.	prk1.Sederhana(100, 2);
10.	}
11.	}

Praktikum1.java	
1.	public class Praktikum1 {
2.	public static void Penjumlahan(int a,int b) {
3.	int nilai = a+b;
4.	System.out.println("Hasil Penjumlahan : "+nilai);
5.	}
6.	public static void Pengurangan(int a, int b){
7.	int nilai = a-b;
8.	System.out.println("Hasil Pengurangan : "+nilai);
9.	}
10.	public void Perkalian(int a, int b){
11.	int nilai = a*b;
12.	System.out.println("Hasil Perkalian : "+nilai);
13.	}
14.	public void Pembagian(double bagi, double bagi2) {
15.	System.out.println("Hasil Pembagian : "+bagi/bagi2);
16.	}

```

17. public void Sederhana(int fpb1, int fpb2) {
18.     int FPB = searchforFPB(fpb1, fpb2);
19.     int a = fpb1/FPB;
20.     int b = fpb2/FPB;
21.     System.out.println("Hasil penyederhanaan : "+ a +
    "/" + b);
22. }
23. private int searchforFPB(int a, int b) {
24.     if (a%b == 0) {
25.         return b;
26.     }
27.     return searchforFPB(a, a % b);
28. }
29. }

```

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\praktikumkedelapan> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\bddfd5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'MainPraktikum1'
Hasil dari operasi angka 100 dan 2
Hasil Penjumlahan : 102
Hasil Pengurangan : 98
Hasil Perkalian : 200
Hasil Pembagian : 50.0
Hasil penyederhanaan : 50/1
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>

```

Penjelasan

Berikut adalah penjelasan mengenai kelas MainPraktikum1.java dan Praktikum1.java :

1. Penjumlahan(int a, int b) : Static method yang menghitung penjumlahan dari dua buah bilangan bulat (int) dan menampilkan outputnya.
2. Pengurangan(int a, int b) : Static method yang menghitung pengurangan dari dua buah bilangan bulat (int) dan menampilkan outputnya.
3. Perkalian(int a, int b) : Instance method yang menghitung perkalian dari dua buah bilangan bulat (int) dan menampilkan outputnya.
4. Pembagian(double bagi, double bagi2) : Instance method menghitung pembagian dari dua buah bilangan riil (double) dan menampilkan outputnya.
5. Sederhana(int fpb1, int fpb2) : Instance method untuk menyederhanakan pecahan dengan cara mencari faktor persekutuan terbesar (FPB) dari pembilang dan penyebut, selanjutnya membagi pembilang dan penyebut dengan FPB tersebut. Hasil penyederhanaan kemudian akan ditampilkan.

6. `searchforFPB(int a, int b)` : Private method yang digunakan method sederhana untuk mencari FPB dari dua buah bilangan bulat. Method akan mengembalikan sebuah bilangan bulat yang merupakan FPB dari a dan b.

Dalam kelas `MainPraktikum1`, ada method `main`, sebuah method utama untuk dieksekusi oleh JVM pada saat program dijalankan. Di method `main`, dilakukan instansiasi objek dari kelas `Praktikum 1` dengan dipanggil beberapa method pada class tersebut. Untuk method `Penjumlahan` dan `Pengurangan` dipanggil secara langsung menggunakan nama dari class (`Praktikum1`) karena keduanya merupakan method static. Namun method `Perkalian`, `Pembagian`, dan `Sederhana` dipanggil melalui objek `prk1` karena ketiga method tersebut adalah method instance.

2. CV. Labkomdas adalah perusahaan yang bergerak di bidang produksi jaket, terdapat 3 tipe jaket yang memiliki harga yang tetap (final) yaitu jaket dengan bahan A, B dan C yang masing-masing memiliki harga persatuannya adalah Rp 100.000, Rp 125.000, Rp 175.000. Karena produksi jaket tersebut terbilang masih baru pihak perusahaan melakukan strategi dengan memberi diskon harga bila pembeli membeli banyak jenis jaket, diskon di berikan apabila :
 - a. Jaket A di beli sebanyak lebih dari 100 buah maka harga menjadi 95.000 per biji
 - b. Jaket B di beli sebanyak lebih dari 100 buah maka harga menjadi 120.000 per biji
 - c. Jaket C di beli sebanyak lebih dari 100 buah maka harga menjadi 160.000 per biji

Buat program dari kasus diatas dengan materi yang sudah di ajarkan oleh asisten

Nb : nilai final adalah harga jaket A, B dan C

Source Code

MainPraktikumLabKomdas.java	
1.	<code>import java.util.*;</code>
2.	<code>public class MainPraktikumLabKomdas {</code>
3.	<code>public static void main(String[] args) {</code>
4.	<code>Scanner newJaket = new Scanner(System.in);</code>
5.	<code>System.out.println("Masukan banyak");</code>
6.	<code>System.out.print("Jaket A : ");</code>
7.	<code>int jaketA = newJaket.nextInt();</code>
8.	<code>System.out.print("Jaket B : ");</code>
9.	<code>int jaketB = newJaket.nextInt();</code>
10.	<code>System.out.print("Jaket C : ");</code>
11.	<code>int jaketC = newJaket.nextInt();</code>
12.	<code>System.out.println();</code>
13.	<code>LabKomdas jaketbaru = new LabKomdas(jaketA, jaketB,</code>
	<code>jaketC);</code>
14.	<code>}</code>
15.	<code>}</code>

LabKomdas.java	
1.	<code>public class LabKomdas {</code>

2.	<code>final int jaketA, jaketB, jaketC;</code>
3.	<code>public LabKomdas(int p, int q, int r) {</code>
4.	<code> jaketA = p > 100 ? 95000 : 100000;</code>
5.	<code> jaketB = q > 100 ? 100000 : 125000;</code>
6.	<code> jaketC = r > 100 ? 160000 : 175000;</code>
7.	<code> getJaket(jaketA, jaketB, jaketC);</code>
8.	<code> }</code>
9.	<code>public void getJaket(int p, int q, int r) {</code>
10.	<code> System.out.println("Harga tetap jaket A : " + p);</code>
11.	<code> System.out.println("Harga tetap jaket B : " + q);</code>
12.	<code> System.out.println("Harga tetap jaket C : " + r);</code>
13.	<code> }</code>
14.	<code>}</code>

Output

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

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'MainPraktikumLabKomdas'
Masukan banyak
Jaket A : 1
Jaket B : 1
Jaket C : 1

Harga tetap jaket A : 100000
Harga tetap jaket B : 125000
Harga tetap jaket C : 175000
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>

```

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

```

Dani Adrian
225150201111009

```

The text editor status bar shows 'Ln 2, Col 16 | 100% | Windows (CRLF) | UTF-8'.

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

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\bddf6d5b40c0014858d676ecd34f4fb8\redhat.java\jdt_ws\praktikumkedelapan_9b8ae742\bin' 'MainPraktikumLabKomdas'
Masukan banyak
Jaket A : 140
Jaket B : 150
Jaket C : 114

Harga tetap jaket A : 95000
Harga tetap jaket B : 100000
Harga tetap jaket C : 160000
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkedelapan>

```

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

```

Dani Adrian
225150201111009

```

The text editor status bar shows 'Ln 2, Col 16 | 100% | Windows (CRLF) | UTF-8'.

Penjelasan

Berikut penjelasan mengenai kedua class tersebut, class LabKomdas.Java dan class MainPraktikumLabKomdas.java :

Class Labkomdas memiliki variable final sebanyak tiga, yaitu jaketA, jaketB, dan jaketC. Di konstruktor class Labkomdas, nilai dari ketiga variabel diinisialisasi menggunakan operator ternary berdasarkan nilai parameter p, q, dan r. Jika nilai dari parameter tersebut lebih besar dari 100, maka nilai variabel akan diisi dengan harga diskon, namun jika tidak, maka nilai variabel akan diisi dengan harga normal. Selanjutnya, constructor memanggil metode getJaket serta memberikan sebanyak tiga variabel harga.

Pada class MainPraktikumLabKomdas, dimana sebagai class utama. Pada class tersebut terdapat objek Scanner untuk membaca inputan dari user yang dimana selanjutnya user akan diminta untuk menginputkan sejumlah jaket A, B, dan C yang akan dibeli. Setelah itu, akan dilakukan instansiasi objek LabKomdas menggunakan sejumlah jaket yang telah diinputkan user sebagai argumen. Kemudian objek LabKomdas menampilkan harga setiap jenis jaket melalui metode getJaket.