

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : STATIC MODIFIER DAN KONSTANTA FINAL
NAMA : Rochmanu Purnomohadi Erfitra
NIM : 225150200111018
ASISTEN : Tengku Muhammad Rafi Rahardiansyah
Muhammad Bin Djafar Almasyhur
TGL PRAKTIKUM : 03 Mei 2023

A. Static Method

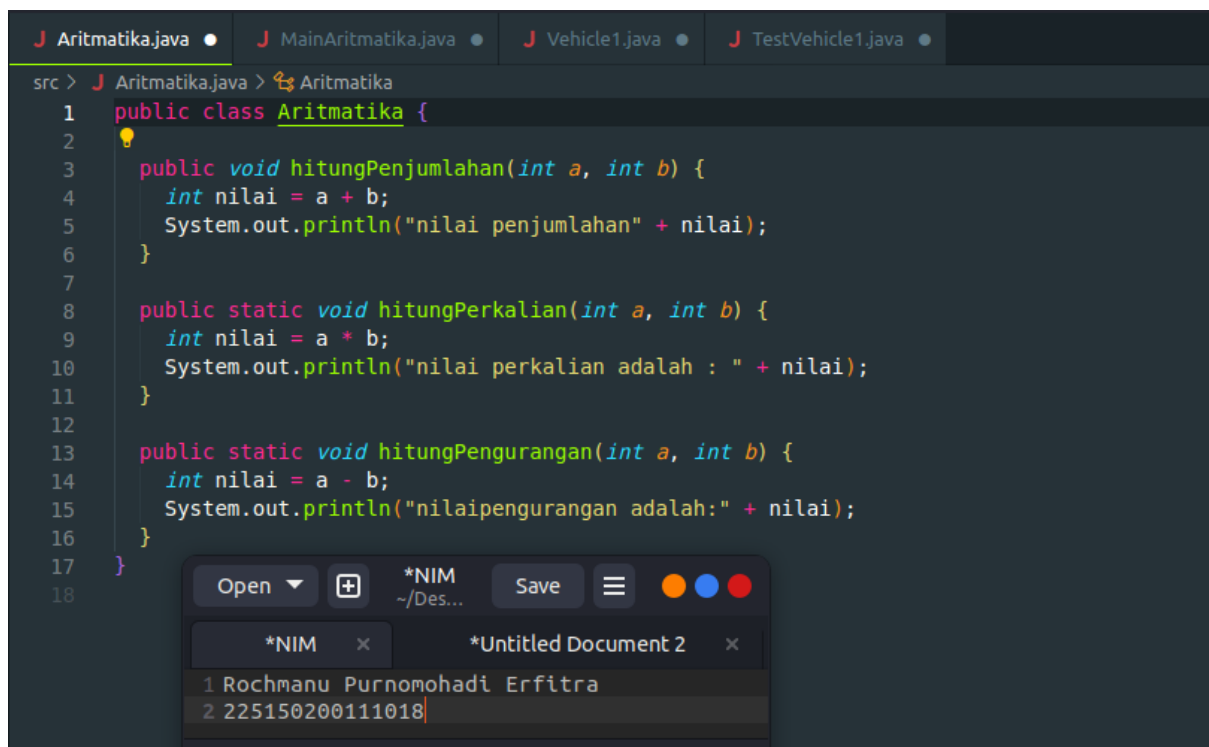
Pertanyaan

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

Syntax

```
public class Aritmatika {  
    static int abc = 19;
```

Screenshot



```
src > J Aritmatika.java > Aritmatika  
1 public class Aritmatika {  
2  
3     public void hitungPenjumlahan(int a, int b) {  
4         int nilai = a + b;  
5         System.out.println("nilai penjumlahan" + nilai);  
6     }  
7  
8     public static void hitungPerkalian(int a, int b) {  
9         int nilai = a * b;  
10        System.out.println("nilai perkalian adalah : " + nilai);  
11    }  
12  
13    public static void hitungPengurangan(int a, int b) {  
14        int nilai = a - b;  
15        System.out.println("nilaipengurangan adalah:" + nilai);  
16    }  
17 }  
18
```

Open [icon] *NIM ~/Des... Save [icon] [icon] [icon] [icon]
*NIM x *Untitled Document 2 x
1 Rochmanu Purnomohadi Erfitra
2 225150200111018

Penjelasan

Static variable (variabel statis) adalah variabel yang dideklarasikan menggunakan kata kunci "static" pada saat variabel tersebut dideklarasikan di dalam kelas. Kita dapat menggunakan static variable ketika kita ingin menyimpan data yang sama untuk semua instance objek dari

kelas tersebut, atau ketika kita ingin mengakses variabel atau method dari kelas tersebut tanpa harus membuat instance objek terlebih dahulu.

Pertanyaan

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

Syntax

```
public class MainAritmatika {  
  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
  
        System.out.print("masukkan nilai 1 : ");  
        int nil1 = in.nextInt();
```

Screenshot

The screenshot shows an IDE with several tabs: Aritmatika.java, MainAritmatika.java (active), Vehicle1.java, and TestVehicle1.java. The code in MainAritmatika.java is as follows:

```
src > J MainAritmatika.java > MainAritmatika > main(String[])  
1  import java.util.Scanner;  
2  
3  public class MainAritmatika {  
4  
5      public static void main(String[] args) {  
6          Scanner in = new Scanner(System.in);  
7  
8          System.out.print(s:"masukkan nilai 1 : ");  
9          int nil1 = in.nextInt();  
10  
11         System.out.print(s:"masukkan nilai 2 : ");  
12         int nil2 = in.nextInt();  
13  
14         // memanggil method static  
15         Aritmatika.hitungPengurangan(nil1, nil2);  
16  
17  
18  
19  
20  
21  
22  
23  
24
```

A dialog box is open over the code, showing a form with fields for Name and NIM. The Name field contains "1 Rochmanu Purnomohadi Erfitra" and the NIM field contains "2 225150200111018". The dialog box also has buttons for "Open", "Save", and "Cancel".

Penjelasan

Menuliskan static pada method main memungkinkan program dapat dijalankan secara langsung tanpa perlu membuat objek dari kelas MainAritmatika terlebih dahulu. Jika method

main tidak memiliki modifier static, program akan menghasilkan error saat mencoba menjalankan program karena tidak bisa memanggil method main tanpa membuat objek terlebih dahulu.

Pertanyaan

3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Syntax

```
public class Aritmatika {

    public void hitungPenjumlahan(int a, int b) {
        int nilai = a + b;
        System.out.println("nilai penjumlahan adalah : " +
nilai);
    }

    public static void hitungPerkalian(int a, int b) {
        int nilai = a * b;
        System.out.println("nilai perkalian adalah : " +
nilai);
    }

    public static void hitungPengurangan(int a, int b) {
        int nilai = a - b;
        System.out.println("nilai pengurangan adalah : " +
nilai);
    }
}
```

Screenshot

The screenshot shows an IDE with four tabs: `Aritmatika.java`, `MainAritmatika.java` (active), `Vehicle1.java`, and `TestVehicle1.java`. The `MainAritmatika.java` file contains the following code:

```
src > J MainAritmatika.java > MainAritmatika > main(String[])
11 System.out.print(s:"masukkan nilai 2 : "):
```

The **TERMINAL** tab shows the following output:

```
rchmn@rchmn-comp:~/Desktop/Kampus/Pemrograman berbasis obyek/Modul 8 PB0/src$ cd "/home/rchmn/Des
vac MainAritmatika.java && java MainAritmatika
masukkan nilai 1 : 10
masukkan nilai 2 : 2
nilai pengurangan adalah : 8
masukkan nilai 1 : 10
masukkan nilai 2 : 2
nilai perkalian adalah : 20
masukkan nilai 1 : 10
masukkan nilai 2 : 2
nilai penjumlahan adalah : 12
rchmn@rchmn-comp:~/Desktop/Kampus/Pemrograman berbasis obyek/Modul 8 PB0/src$
```

Overlaid on the terminal is a text editor window titled `*NIM` with the following content:

```
1 Rochmanu Purnomohadi Erfitra
2 225150200111018
```

The editor window also shows a status bar with `Tab Width: 8`, `Ln 2, Col 16`, and `INS`.

Penjelasan

Tidak ada kesalahan dalam program aritmatika.

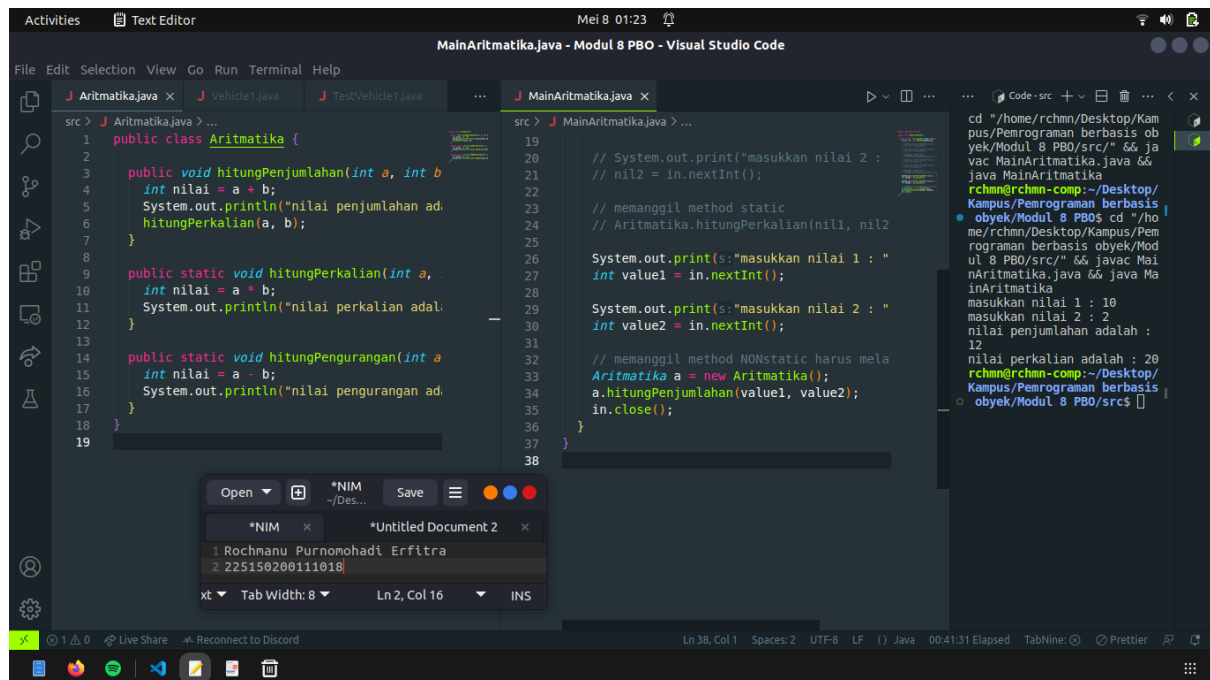
Pertanyaan

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

Syntax

	<pre>public void hitungPenjumlahan(int a, int b) { int nilai = a + b; System.out.println("nilai penjumlahan adalah : " + nilai); hitungPerkalian(a, b); }</pre>
--	---

Screenshot



Penjelasan

Tidak akan terjadi error karena seperti sebelumnya, method `hitungPerkalian()` adalah static method dan tidak perlu menambahkan objek aritmatika terlebih dahulu saat ingin memanggil method tersebut dari `hitungPenjumlahan()`.

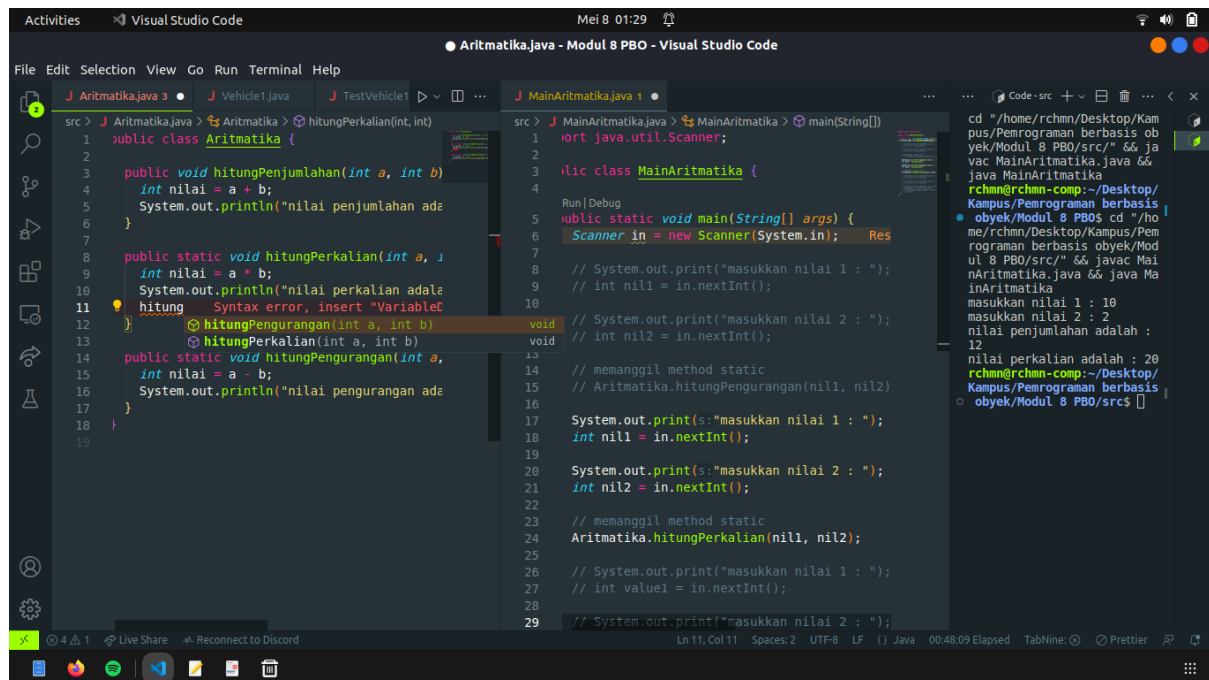
Pertanyaan

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

Syntax

	<pre>public static void hitungPerkalian(int a, int b) { int nilai = a * b; System.out.println("nilai perkalian adalah : " + nilai); hitungPenjumlahan(a, b); }</pre>
--	--

Screenshot



Penjelasan

Akan terjadi error, karena hitungPenjumlahan() bukan merupakan suatu static method jadi tidak bisa langsung memanggil method tanpa menginisialisasi objek terlebih dahulu.

Pertanyaan

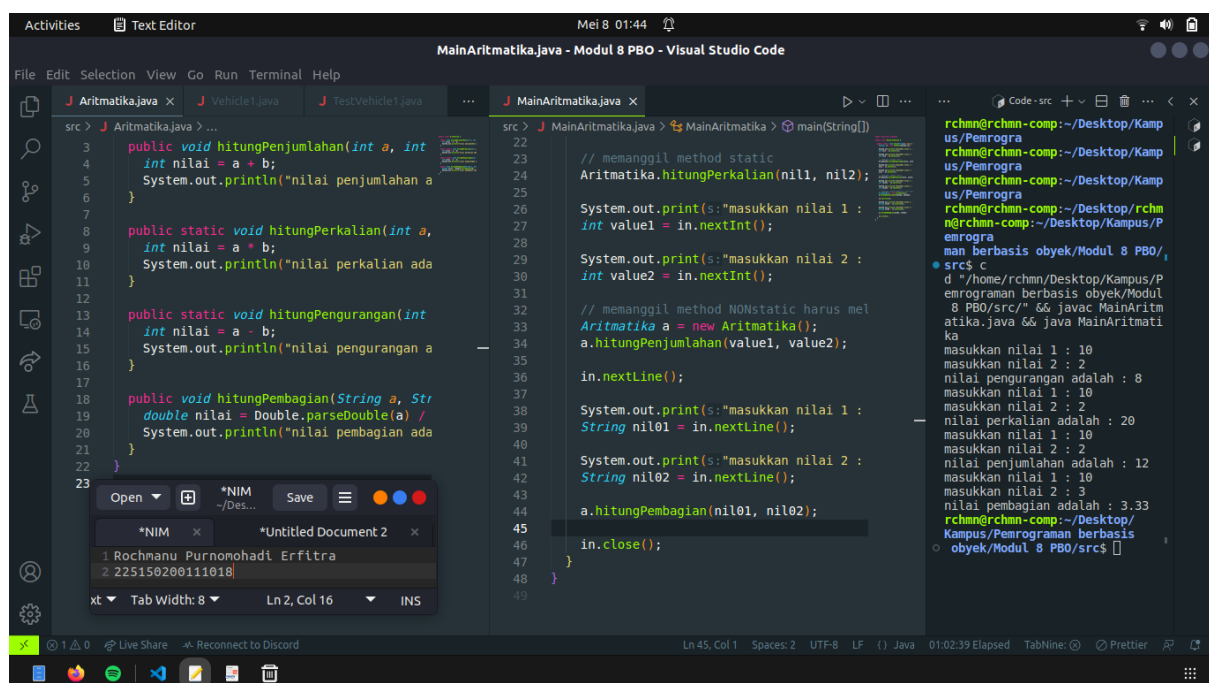
6. Tambahkan method non static dengan nilai balikan double untuk menghitung pembagian dengan parameter String nil dan String nil2, dan panggil method tersebut pada method main!

Syntax

	Aritmatika.java
	<pre>public void hitungPembagian(String a, String b) { double nilai = Double.parseDouble(a) / Double.parseDouble(b); System.out.println("nilai pembagian adalah : " + String.format("%.2f", nilai)); }</pre>
	AritmatikaMain.java
	<pre>in.nextLine(); System.out.print("masukkan nilai 1 : ");</pre>

	<pre>String nil01 = in.nextLine(); System.out.print("masukkan nilai 2 : "); String nil02 = in.nextLine(); a.hitungPembagian(nil01, nil02);</pre>
--	---

Screenshot



Penjelasan

Karena input menggunakan String maka mengunaka parse double dan untuk mengeluarkan output yang lebih rapi menggunakan format string pada syntax print, dan karena ini adalah method non-static maka diperlukan deklarasi objek aritmatika sebelumnya.

B. Konstanta Final

Pertanyaan

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

Syntax

```
public class Vehicle1 {  
    private double load;  
    private final double maxLoad = 10000;  
  
    // public Vehicle1 (double max){  
    // this.maxLoad=max;  
    // }  
  
    public double getLoad() {  
        return this.load;  
    }  
  
    public double getMaxLoad() {  
        return this.maxLoad;  
    }  
  
    public boolean addBox(double weight) {  
        double temp = 0.0D;  
        temp = this.load + weight;  
        if (temp <= maxLoad) {  
            this.load = this.load + weight;  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```


Screenshot

The screenshot shows the Visual Studio Code editor with the file `TestVehicle1.java` open. The code defines a `Vehicle1` class with a `load` attribute and a `getLoad()` method. The `main` method creates a `Vehicle1` object and adds five boxes with weights of 500kg, 250kg, 5000kg, 4000kg, and 300kg. The terminal output shows the execution of the program, which prints the load of each box and the total vehicle load of 9750.0kg.

```
src > J TestVehicle1.java > TestVehicle1 > main(String[])
// VEHICLE1 VEHICLE2 = new VEHICLE1(10000);
6 System.out.println(x:"Add box #1 (500kg)");
7 vehicle.addBox(weight:500);
8 System.out.println(x:"Add box #2 (250kg)");
9 vehicle.addBox(weight:250);
10 System.out.println(x:"Add box #3 (5000kg)");
11 vehicle.addBox(weight:5000);
12 System.out.println(x:"Add box #4 (4000kg)");
13 vehicle.addBox(weight:4000);
14 System.out.println(x:"Add box #5 (300kg)");
15 vehicle.addBox(weight:300);
16 System.out.println("Vehicle load is " + vehicle.getLoad() + "kg");
17 }
18 }
19 }
```

```
● rchmn@rchmn-comp:~/Desktop/Kampus/Pemrograman berbasis obyek/Modul 8 PB0/src$ cd "/home/rchmn/Desktop/Kampus/Pemrograman berbasis obyek/Modul 8 PB0/src/" && javac TestVehicle1.java && java TestVehicle1
Creating a vehicle with 10,000 kg maximum load.
Add box #1 (500kg)
Add box #2 (250kg)
Add box #3 (5000kg)
Add box #4 (4000kg)
Add box #5 (300kg)
Vehicle load is 9750.0kg
● rchmn@rchmn-comp:~/Desktop/Kampus/Pemrograman berbasis obyek/Modul 8 PB0/src$
```

Penjelasan

Tidak ada kesalahan pada kode, dan kode berjalan dengan semestinya.

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!

Syntax

```
public class Vehicle1 {

    private double load;

    private final double maxLoad = 10000;

    public Vehicle1 (double max){

        this.maxLoad=max;

    }

    public double getLoad() {

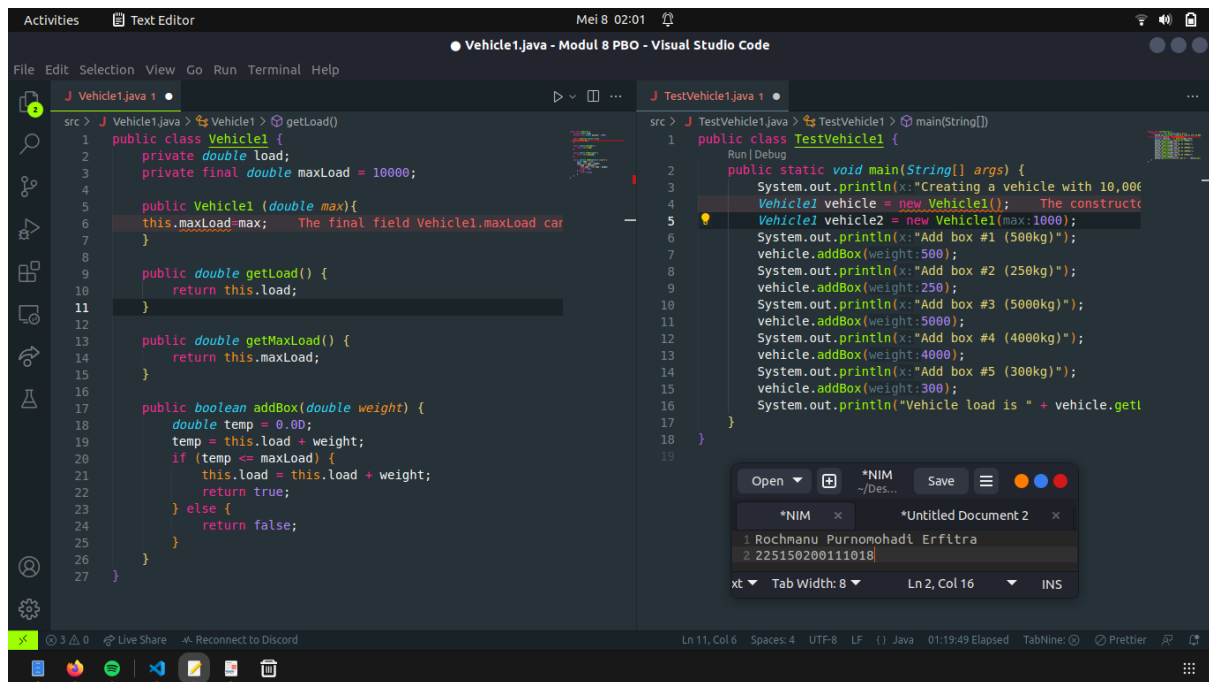
        return this.load;

    }

}
```

	<pre>} public double getMaxLoad() { return this.maxLoad; } public boolean addBox(double weight) { double temp = 0.0D; temp = this.load + weight; if (temp <= maxLoad) { this.load = this.load + weight; return true; } else { return false; } } }</pre>
--	---

Screenshot



Penjelasan

Terjadi error karena variabel maxLoad merupakan variabel final, jadi value nya tidak akan bisa diubah.

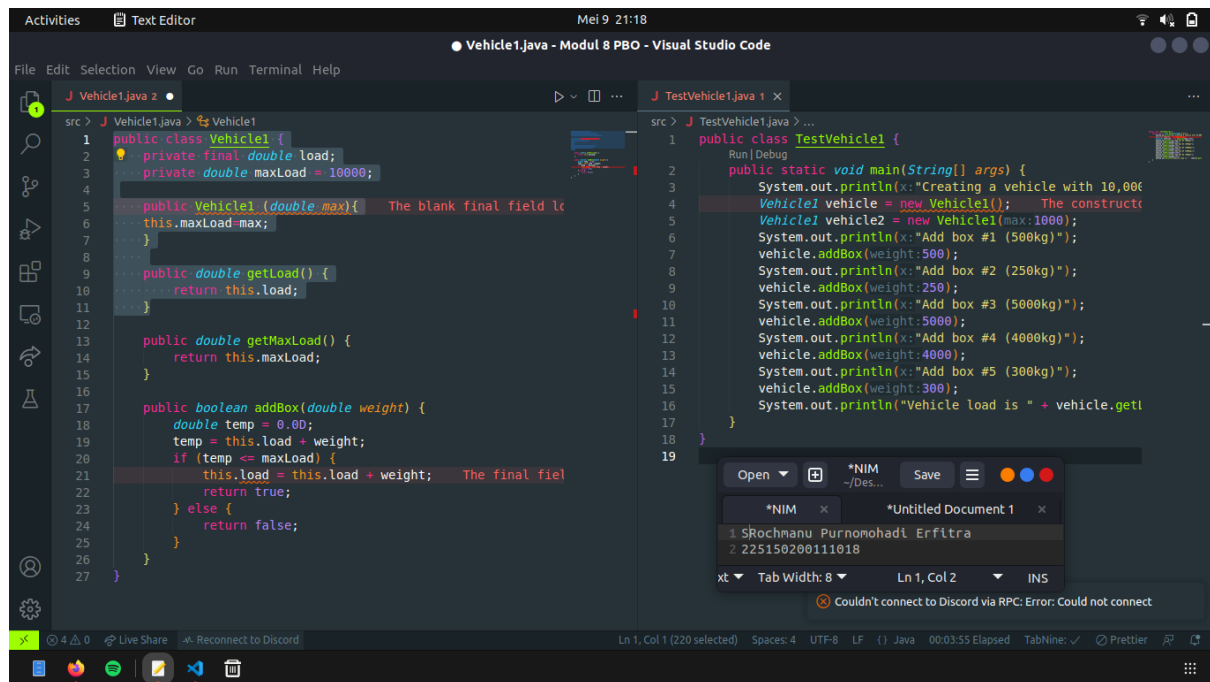
Pertanyaan

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

Syntax

	<pre>public class Vehicle1 { private final double load; private double maxLoad = 10000; public Vehicle1 (double max){ this.maxLoad=max; } public double getLoad() { return this.load; } }</pre>
--	---

Screenshot



Penjelasan

Jika variable load pada Vehicle1 diubah menjadi final class maka method addBox tidak akan bisa dipakai karean variable load value nya tidak dapat diubah sebab telah menjadi konstanta final yang menyebabkan terjadinya error pada program

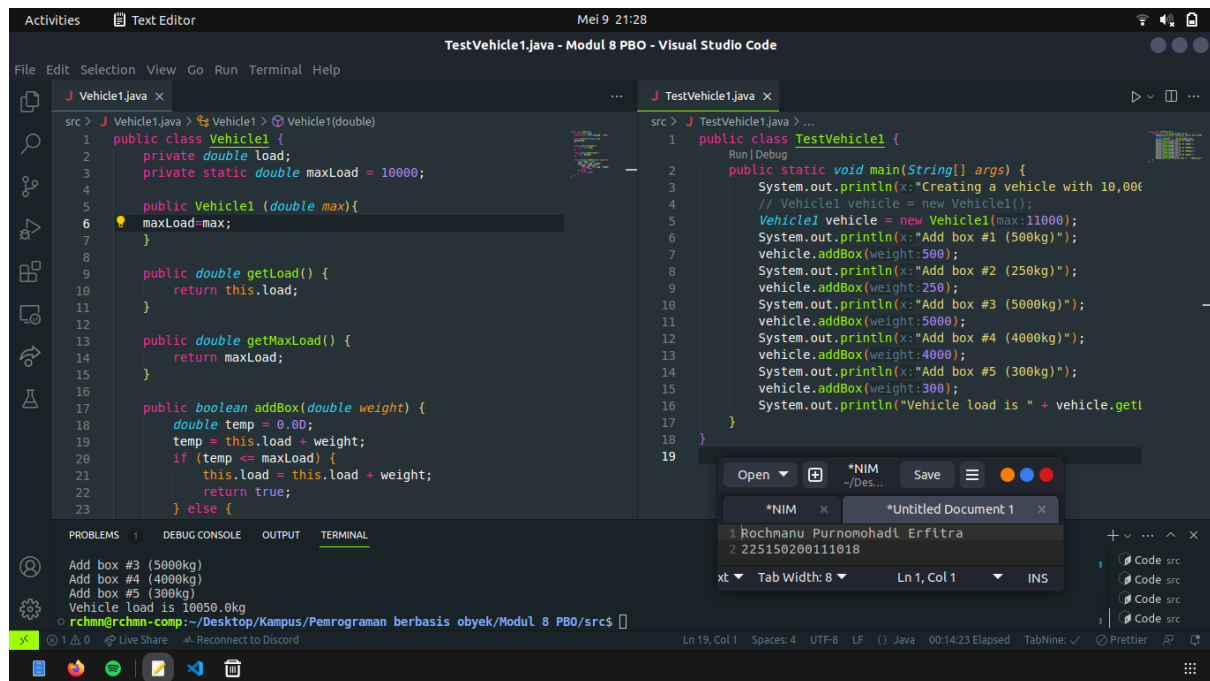
Pertanyaan

4. Tambahkan keyword “static” pada file Vehicle1.java variabel maxLoad, apa yang terjadi dan jelaskan!

Syntax

	<pre>public class Vehicle1 { private double load; private static double maxLoad = 10000;</pre>
--	--

Screenshot



Penjelasan

Maka dalam mengakses atau mengubah nilai dari maxLoad melalui constructor kita tidak perlu menyatakan menggunakan this lagi karena sifat maxLoad sudah menjadi static atau sama untuk semua variable yang ada pada object yang dibuat dari class Vehicle1