

---

NAMA : Lisa Hanifatul Khasanah  
NIM : 225150401111038  
KELAS : B  
BAB : IV  
ASISTEN : Adin Rama Ariyanto Putra dan Fahru Setiawan Iskandar

---

## 1. Data dan Analisis hasil percobaan

### A. Encapsulation 1

1.) Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

#### Jawab :

Terdapat tiga kesalahan kode pada kelas tes, yaitu :

- Pada method `setMark(int m)`, parameter `m` harus berupa tipe data `int` sedangkan pada method `setMark(String m)`, parameter `m` diberikan sebagai tipe data `String`. Oleh karena itu, pada bagian `s1.setMark("90")` perlu diubah menjadi `s1.setMark(90)`.
- Pada bagian `System.out.println("s1Name is "+s1.setName())`, seharusnya dipanggil method `getName()` untuk mendapatkan nama objek `s1` bukan `setName()`. Hal yang sama berlaku untuk `s1.setMark()`. Perlu diubah menjadi `s1.getName()` dan `s1.getMark()`.
- Variabel `name` dan `mark` tidak dapat diakses pada method `main` karena dideklarasikan sebagai variabel `private` pada kelas `Student`. Untuk mendapatkan nilai dari kedua variabel tersebut, perlu dipanggil method `getName()` dan `getMark()` pada objek `s1`. Oleh karena itu, perlu ditambahkan `s1.getName()` dan `s1.getMark()` pada bagian `System.out.println("name dan mark "+name+" "+mark)`.

Berikut merupakan kode yang telah dibenahi :

```
public class Student {  
    private String name;  
    private int mark;  
  
    public void setName(String n) {  
        name = n;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setMark(int m) {
```

```
        mark = m;
    }

    public int getMark() {
        return mark;
    }
}

public class Test {
    public static void main(String [] args) {
        Student s1=new Student(); s1.setName("Enkapsulasi");
        s1.setMark(90);
        System.out.println("name dan mark " + s1.getName() + " " +
s1.getMark());
    }
}
```

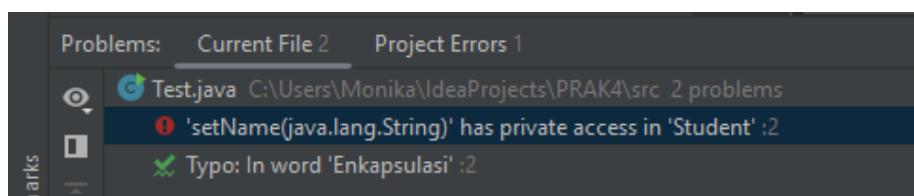
- 2.) Jika pada baris 6 `s1.setName` diubah menjadi `s1.getName` apa yang terjadi? jelaskan!

**Jawab :**

Program akan mencetak nama objek `s1` yang disimpan di dalam variabel `name`. Pada kode program, penggunaan `System.out.println("s1Name is "+s1.setName());` akan menjalankan method `setName()` terlebih dahulu tanpa mengembalikan nilai apa pun, sehingga yang dicetak hanya string "`s1Name is null`" atau "`s1Name is`", tergantung pada implementasi di dalam method `setName()`. Hal ini tidak sesuai dengan tujuan yang diinginkan. Sebaliknya, penggunaan `System.out.println("s1Name is "+s1.getName());` akan mengambil nilai variabel `name` dari objek `s1` dan menggabungkannya dengan string "`s1Name is`" untuk dicetak pada layar. Hasilnya sesuai dengan tujuan yaitu mencetak nama objek `s1` yang telah di set sebelumnya menjadi "`Enkapsulasi`".

- 3.) Setelah diperbaiki, ubahlah hak akses pada baris 4 (pada class `Student`) menjadi `private` apa yang terjadi jika class `Test` dijalankan? Jelaskan!

**Jawab :**



Jika kita mengubah baris 6 kelas Test dari `s1.setName("Enkapsulasi")` menjadi `s1.getName()`, maka program akan mengalami kesalahan kompilasi. Hal ini terjadi karena method `getName()` mengembalikan nilai String sedangkan parameter yang dibutuhkan oleh method `setName(String n)` juga harus berupa tipe data String. Sehingga, jika kita menggunakan `s1.getName()` sebagai parameter untuk method `setName()`, maka program tidak akan bisa dikompilasi dengan benar.

- 4.) Jika kedua kelas diatas terdapat dalam package yang sama apakah konsep enkapsulasi tetap berfungsi? jelaskan!

**Jawab :**

Meskipun kedua kelas berada dalam package yang sama, konsep enkapsulasi masih berfungsi dengan baik. Hal ini dikarenakan adanya modifier `private` pada variabel `name` dan `mark` pada kelas `Student`, sehingga akses langsung dari luar kelas terbatas. Dalam kata lain, variabel `name` dan `mark` hanya bisa diakses melalui method `getName()` dan `getMark()` yang telah disediakan oleh kelas `Student`. Dengan demikian, kelas `Test` tidak dapat mengubah nilai variabel `name` dan `mark` secara langsung, kecuali melalui method yang telah disediakan.

## **B. Encapsulation 2**

- 1.) Method apakah yang menjadi accessor (getter)?

**Jawab :**

Method yang menjadi accessor (getter) pada kode program di bawah ini adalah `public double getLoad()`. Method tersebut memungkinkan untuk mengakses nilai variabel `private load` pada kelas `Vehicle1` dari kelas lain, dengan cara memperbolehkan kelas lain untuk membaca nilai variabel `load` tanpa memperbolehkan kelas lain untuk memodifikasi nilai tersebut. Dalam kasus ini, method `getLoad()` digunakan untuk mengembalikan nilai variabel `load` yang telah di-set sebelumnya. Sehingga, method `getLoad()` adalah accessor (getter) yang berfungsi untuk membaca nilai variabel `private` dari kelas `Vehicle1`.

2.) Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

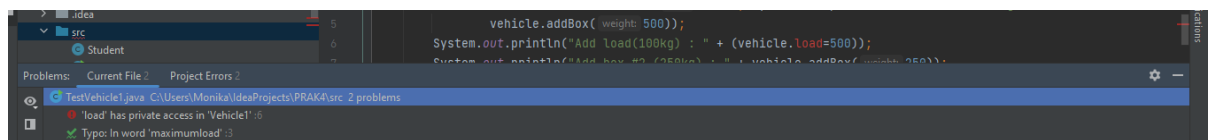
Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

**Jawab :**

Saat kode ditambahkan System.out.println("Add load(100kg) : " +

(vehicle.load=500)); pada kelas testvehicle1, terjadinya error pada program karena tipe data load merupakan private sehingga tidak dapat diakses pada kelas test.



3.) Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi public. Jalankan program, apakah output dari program tersebut?

a. Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.

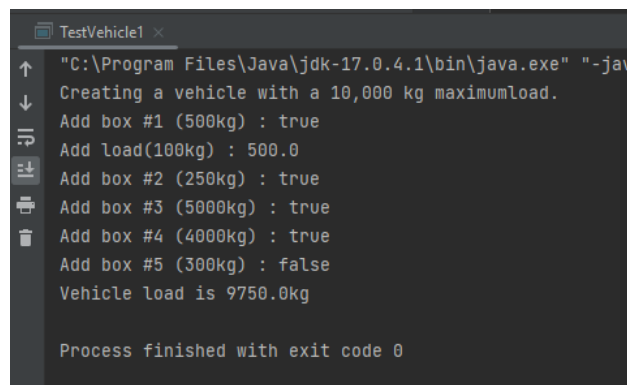
b. Tambahkan source code berikut dibawah baris ke 12 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.

**Jawab :**

a. Output



b. Output

```
TestVehicle1 x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-
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 load(100kg) : 500.0
Add box #5 (300kg) : true
Vehicle load is 800.0kg

Process finished with exit code 0
```

- 4.) Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

**Jawab :**

a. Output

```
TestVehicle1 x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "
Creating a vehicle with a 10,000 kg maximumload.
Add box #1 (500kg) : true
Add load(100kg) : 500.0
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

Process finished with exit code 0
```

b. Output

```
TestVehicle1 x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "
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 load(100kg) : 500.0
Add box #5 (300kg) : true
Vehicle load is 800.0kg

Process finished with exit code 0
```

- 5.) Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi default.

**Jawab :**

Kode program yang diubah pada kelas Vehicle1 :

```
public class Vehicle1 {
    double load, maxLoad;

    public Vehicle1(double max) {
        this.maxLoad = max;
    }

    public double getLoad() {
        return this.load;
    }

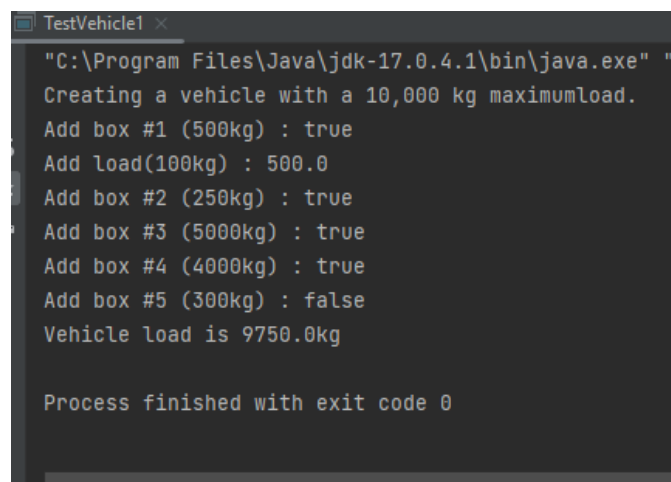
    public double getMaxLoad() {
        return this.maxLoad;
    }

    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;
        }
    }
}
```

**Output**

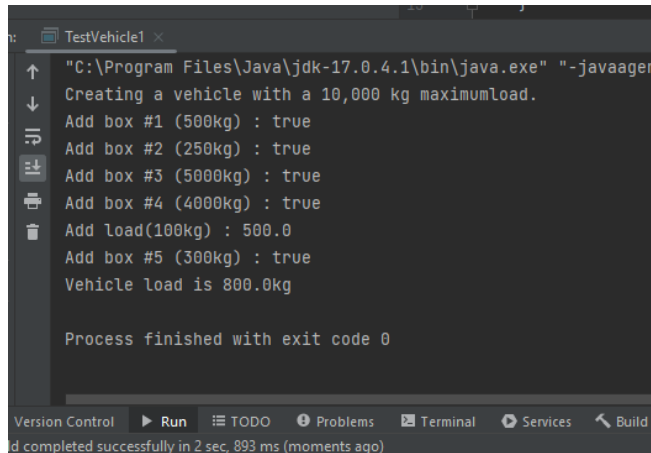
- Tambahan source code `System.out.println("Add load(100kg) : " + (vehicle.load=500));` pada baris 6



```
TestVehicle1 x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-
Creating a vehicle with a 10,000 kg maximumload.
Add box #1 (500kg) : true
Add load(100kg) : 500.0
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

Process finished with exit code 0
```

- Tambahan source code `System.out.println("Add load(100kg) : " + (vehicle.load=500));` pada baris 12



```
TestVehicle1 x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent
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 load(100kg) : 500.0
Add box #5 (300kg) : true
Vehicle load is 800.0kg

Process finished with exit code 0

Version Control Run TODO Problems Terminal Services Build
d completed successfully in 2 sec, 893 ms (moments ago)
```

## 2. Tugas Praktikum

### 2.1 Source code

\*Kode ini memiliki 4 kelas, bila ingin mencoba harus membuat 4 file untuk tiap kelasnya terlebih dahulu agar proses kompilasinya tidak error.

Kode dengan Kelas Pelanggan :

```
public class Pelanggan {
    private String nomorPelanggan;
    private String nama;
    private int jenisRekening;
    private int saldo;
    private int salahAutentikasiCount;

    public Pelanggan(String nomorPelanggan, String nama, int
jenisRekening, int saldo) {
        this.nomorPelanggan = nomorPelanggan;
        this.nama = nama;
        this.jenisRekening = jenisRekening;
        this.saldo = saldo;
        this.salahAutentikasiCount = 0;
    }

    public boolean autentikasi(int pin) {
        // contoh sederhana autentikasi dengan PIN
        if (pin == 1234) {
            this.salahAutentikasiCount = 0;
            return true;
        } else {
            this.salahAutentikasiCount++;
            if (this.salahAutentikasiCount >= 3) {
                // jika sudah salah autentikasi 3 kali, blokir
akun
                this.saldo = 0;
            }
            return false;
        }
    }
}
```

```
}

public String getNomorPelanggan() {
    return nomorPelanggan;
}

public String getName() {
    return nama;
}

public int getJenisRekening() {
    return jenisRekening;
}

public int getSaldo() {
    return saldo;
}

public void setSaldo(int saldo) {
    this.saldo = saldo;
}

public boolean beli(int nominal) {
    if (this.saldo - nominal >= 10000) {
        int cashback = 0;
        switch (this.jenisRekening) {
            case 38:
                if (nominal > 1000000) {
                    cashback = (int) (nominal * 0.05);
                }
                break;
            case 56:
                if (nominal > 1000000) {
                    cashback = (int) (nominal * 0.07);
                } else {
                    cashback = (int) (nominal * 0.02);
                }
                break;
            case 74:
                if (nominal > 1000000) {
                    cashback = (int) (nominal * 0.1);
                } else {
                    cashback = (int) (nominal * 0.05);
                }
                break;
        }
        this.saldo = this.saldo - nominal + cashback;
        return true;
    } else {
        return false;
    }
}

public boolean topUp(int nominal) {
    if (nominal > 0) {
        this.saldo = this.saldo + nominal;
        return true;
    }
}
```



```
    } else {  
        return false;  
    }  
}  
}
```

Kode dengan Kelas Transaksi :

```
public class TransaksiPembelian {  
    private Pelanggan pelanggan;  
    private int nominal;  
  
    public TransaksiPembelian(Pelanggan pelanggan, int nominal)  
    {  
        this.pelanggan = pelanggan;  
        this.nominal = nominal;  
    }  
  
    public boolean beli() {  
        return pelanggan.beli(nominal);  
    }  
  
    public Pelanggan getPelanggan() {  
        return pelanggan;  
    }  
  
    public int getNominal() {  
        return nominal;  
    }  
}  
  
public class TransaksiTopUp {  
    private Pelanggan pelanggan;  
    private int nominal;  
  
    public TransaksiTopUp(Pelanggan pelanggan, int nominal) {  
        this.pelanggan = pelanggan;  
        this.nominal = nominal;  
    }  
  
    public void topUp() {  
        this.pelanggan.setSaldo(this.pelanggan.getSaldo() +  
this.nominal);  
    }  
  
    public Pelanggan getPelanggan() {  
        return pelanggan;  
    }  
  
    public int getNominal() {  
        return nominal;  
    }  
}
```

Kode dengan kelas topUp untuk dapat diakses di kelas Transaksi :

```
public class TransaksiTopUp {
    private Pelanggan pelanggan;
    private int nominal;

    public TransaksiTopUp(Pelanggan pelanggan, int nominal) {
        this.pelanggan = pelanggan;
        this.nominal = nominal;
    }

    public void topUp() {
        this.pelanggan.setSaldo(this.pelanggan.getSaldo() +
this.nominal);
    }

    public Pelanggan getPelanggan() {
        return pelanggan;
    }

    public int getNominal() {
        return nominal;
    }
}
```

Kode dengan Kelas TestSwalayan untuk menjalankan program :

```
import java.util.Scanner;

public class TestSwalayan {
    public static void main(String[] args) {
        // inisialisasi pelanggan
        Pelanggan pelanggan = new Pelanggan("5601234567", "Hamba
Allah", 56, 5000000);

        Scanner input = new Scanner(System.in);

        // autentikasi pelanggan
        int salahAutentikasiCount = 0;
        boolean autentikasiSukses = false;
        while (!autentikasiSukses && salahAutentikasiCount < 3)
        {
            System.out.print("Masukkan nomor pelanggan : ");
            String nomorPelanggan = input.next();
            System.out.print("Masukkan PIN : ");
            int pin = input.nextInt();

            if
(nomorPelanggan.equals(pelanggan.getNomorPelanggan()) &&
pelanggan.autentikasi(pin)) {
                autentikasiSukses = true;
                System.out.println("Autentikasi berhasil!");
            } else {
                salahAutentikasiCount++;
                if (salahAutentikasiCount == 3) {
                    System.out.println("Anda sudah salah
autentikasi sebanyak 3 kali. Akun Anda akan diblokir.");
                }
            }
        }
    }
}
```

```
        System.exit(0);
    } else {
        System.out.println("Nomor pelanggan atau PIN
salah. Silakan coba lagi.");
    }
}

    System.out.println("Selamat datang kembali, " +
pelanggan.getNama());
    System.out.println("Saldo anda saat ini : " +
pelanggan.getSaldo());
    System.out.println("Silahkan pilih jenis transaksi : ");
    System.out.println("1. Pembelian barang");
    System.out.println("2. Top Up saldo");

    int pilihan = input.nextInt();

    if (pilihan == 1) {
        int nominalTopUp = 0;
        if (autentikasiSukses) {
            // melakukan transaksi pembelian
            System.out.print("Masukkan nominal pembelian :
");

            int nominalPembelian = input.nextInt();

            // menambahkan kondisi untuk memeriksa
            apakah saldo mencukupi
            if (nominalPembelian >
pelanggan.getSaldo()) {
                System.out.println("Transaksi pembelian
gagal. Saldo Anda tidak mencukupi.");
            } else {
                TransaksiPembelian transaksiPembelian =
new TransaksiPembelian(pelanggan, nominalPembelian);
                transaksiPembelian.beli();
                System.out.println("Transaksi pembelian
berhasil!");

                System.out.println("Saldo Anda sekarang
: Rp" + pelanggan.getSaldo());
            }
        }
    } else if (pilihan == 2) {
        System.out.print("Masukkan nominal topUp : ");
        int nominalTopUp = input.nextInt();

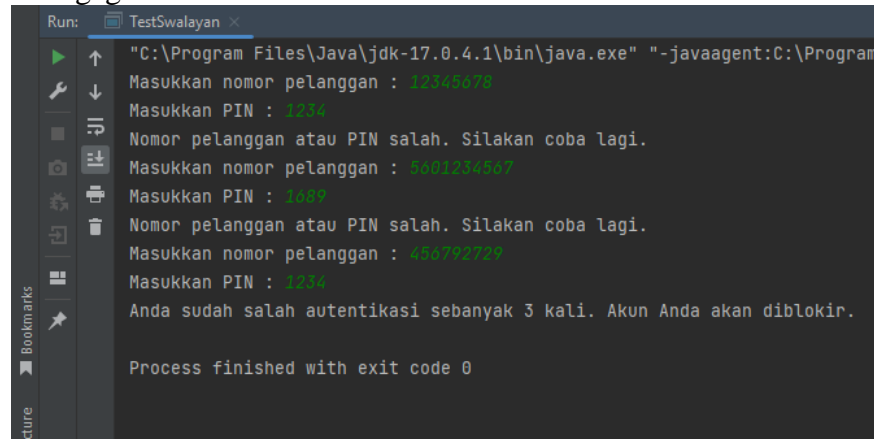
        TransaksiTopUp transaksiTopUp = new
TransaksiTopUp(pelanggan, nominalTopUp);
        transaksiTopUp.topUp();
        System.out.println("Transaksi top up
berhasil!");

        System.out.println("Saldo Anda sekarang : Rp" +
pelanggan.getSaldo());
    }
    else
        System.out.println("Pilihan yang anda masukan tidak
valid!");
}
```

```
}  
}
```

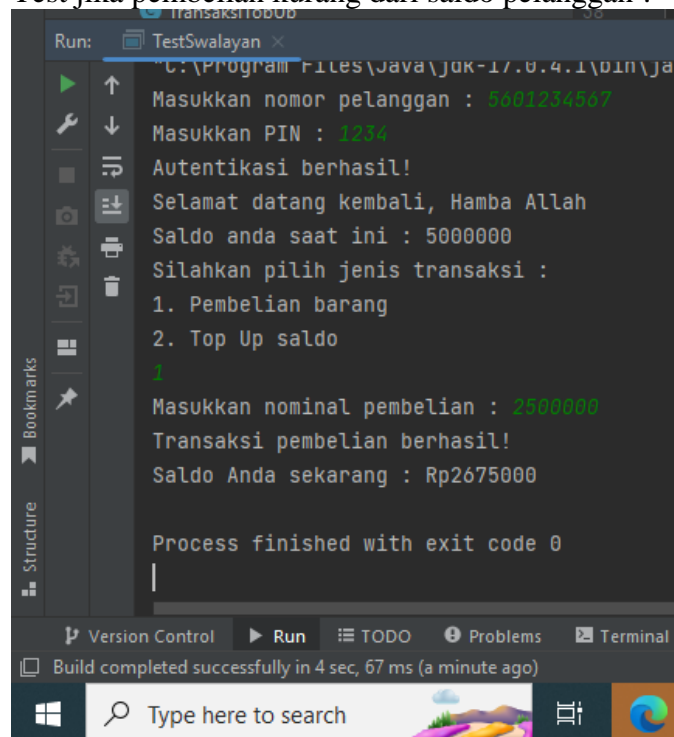
## 2.2 Screenshot hasil

Test gagal autentikasi :



```
Run: TestSwalayan x  
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program  
Masukkan nomor pelanggan : 12345678  
Masukkan PIN : 1234  
Nomor pelanggan atau PIN salah. Silakan coba lagi.  
Masukkan nomor pelanggan : 8801234567  
Masukkan PIN : 1234  
Nomor pelanggan atau PIN salah. Silakan coba lagi.  
Masukkan nomor pelanggan : 456792729  
Masukkan PIN : 1234  
Anda sudah salah autentikasi sebanyak 3 kali. Akun Anda akan diblokir.  
  
Process finished with exit code 0
```

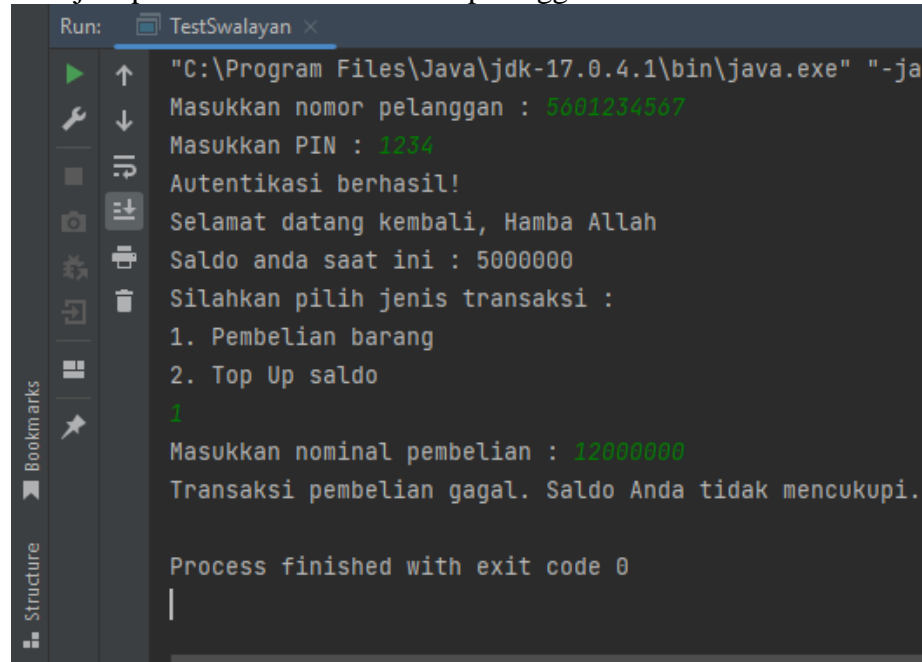
Test jika pembelian kurang dari saldo pelanggan :



```
Run: TestSwalayan x  
"C:\Program Files\Java\jdk-17.0.4.1\bin\jav  
Masukkan nomor pelanggan : 8801234567  
Masukkan PIN : 1234  
Autentikasi berhasil!  
Selamat datang kembali, Hamba Allah  
Saldo anda saat ini : 5000000  
Silahkan pilih jenis transaksi :  
1. Pembelian barang  
2. Top Up saldo  
1  
Masukkan nominal pembelian : 2500000  
Transaksi pembelian berhasil!  
Saldo Anda sekarang : Rp2675000  
  
Process finished with exit code 0  
|
```

Version Control Run TODO Problems Terminal  
Build completed successfully in 4 sec, 67 ms (a minute ago)

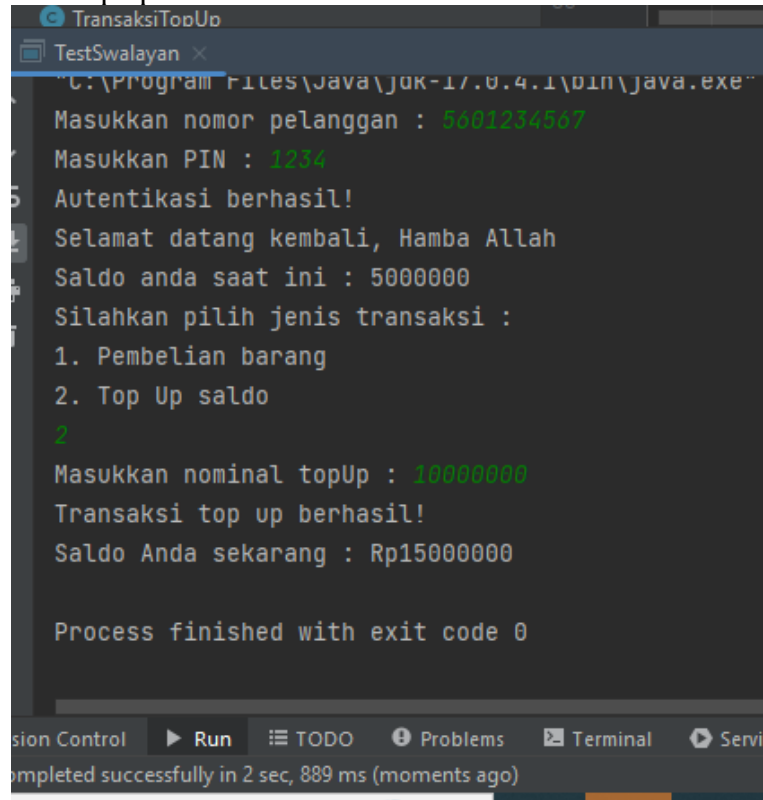
Test jika pembelian melebihi saldo pelanggan :



```
Run: TestSwalayan x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-ja
Masukkan nomor pelanggan : 5601234567
Masukkan PIN : 1234
Autentikasi berhasil!
Selamat datang kembali, Hamba Allah
Saldo anda saat ini : 5000000
Silahkan pilih jenis transaksi :
1. Pembelian barang
2. Top Up saldo
1
Masukkan nominal pembelian : 12000000
Transaksi pembelian gagal. Saldo Anda tidak mencukupi.

Process finished with exit code 0
```

Test topUp :



```
TransaksiTopUp
TestSwalayan x
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe"
Masukkan nomor pelanggan : 5601234567
Masukkan PIN : 1234
Autentikasi berhasil!
Selamat datang kembali, Hamba Allah
Saldo anda saat ini : 5000000
Silahkan pilih jenis transaksi :
1. Pembelian barang
2. Top Up saldo
2
Masukkan nominal topUp : 10000000
Transaksi top up berhasil!
Saldo Anda sekarang : Rp15000000

Process finished with exit code 0
```

Run | Run | TODO | Problems | Terminal | Service |  
Completed successfully in 2 sec, 889 ms (moments ago)