
Nama	: Muhammad Hazel Zahran Saleh
NIM	: 245150707111004
Kelas	: TI - A
Bab	: 4/Encapsulation
Asisten	: 1. Ketut Bagus Wedanta Ananda Murti 2. Gantang Satria Yudha.

LANGKAH 1

A. Soal

Tugas Praktikum

Anda dan tim anda mendapat sebuah proyek untuk merancang sistem transaksi pada sebuah swalayan Tiny. Anda ditugasi oleh tim untuk membuat programnya berdasarkan hasil analisis tim anda :

1. Informasi akun seorang pelanggan (saldo, nomor pelanggan, nama) tidak bias diubah oleh pelanggan secara langsung.
2. Nomor pelanggan terdiri dari 10 digit, dimana 2 digit awal adalah jenis rekening
 - o 38 : Pelanggan jenis silver; setiap pembelian diatas 1 jt maka mendapat cashback sebesar 5%
 - o 56 : Pelanggan jenis gold; setiap pembelian diatas 1 jt maka mendapat cashback sebesar 7%, selain itu cashback 2% (cashback kembali ke saldo)
 - o 74 : Pelanggan jenis platinum; setiap pembelian diatas 1 jt maka mendapat cashback sebesar 10%, selain itu cashback 5% (cashback kembali ke saldo)
3. Pelanggan harus memiliki saldo minimal Rp10.000, jika saldo pasca transaksi kurang dari batas minimal tadi, maka transaksi pembelian dianggap gagal
4. Buatlah sistem transaksi swalayan ini terbatas pada pembelian dan top up saja dan menggunakan PIN dan nomor pelanggan sebagai syarat transaksi pembelian atau top up.
5. Apabila pelanggan melakukan 3x kesalahan dalam autentifikasi, maka akun pelanggan akan defreeze / diblokir sehingga tidak bisa digunakan lagi.

B. Screenshot

```
1  import java.util.Scanner;   PraktikumPemlan4.java is a non-project file, only syntax errors are reported
2
3  Codeium: Refactor | Explain
4  class Customer {
5      private String customerNumber;   Field customerNumber can be final
6      private String name;             Field name can be final
7      private double balance;
8      private String pin;              Field pin can be final
9
10     public Customer(String customerNumber, String name, double balance, String pin) {
11         this.customerNumber = customerNumber;
12         this.name = name;
13         this.balance = balance;
14         this.pin = pin;
15     }
16
17     Codeium: Refactor | Explain | Generate Javadoc | X
18     public String getCustomerNumber() {
19         return customerNumber;
20     }
21
22     Codeium: Refactor | Explain | Generate Javadoc | X
23     public String getName() {
24         return name;
25     }
26
27     Codeium: Refactor | Explain | Generate Javadoc | X
28     public double getBalance() {
29         return balance;
30     }
31
32     Codeium: Refactor | Explain | Generate Javadoc | X
33     public boolean authenticate(String pin) {
34         return this.pin.equals(pin);
35     }
36
37     Codeium: Refactor | Explain | Generate Javadoc | X
38     public void topUp(double amount) {
39         balance += amount;
40     }
41
42     Codeium: Refactor | Explain | Generate Javadoc | X
43     public boolean purchase(double amount) {
44         if (balance - amount < 10000) {
45             System.out.println(x:"Transaksi gagal: Saldo setelah transaksi kurang dari Rp10.000");
46             return false;
47         }
48         double cashback = calculateCashback(amount);
49         balance -= amount;
50         balance += cashback;
51         return true;
52     }
53 }
```

```

Codeium: Refactor | Explain | Generate Javadoc | X
47 private double calculateCashback(double amount) {
48     double cashback = 0;
49     if (amount > 1000000) {
50         switch (customerNumber.substring(beginIndex:0, endIndex:2)) { Convert switch to rule switch
51             case "38": // Silver
52                 cashback = amount * 0.05;
53                 break;
54             case "56": // Gold
55                 cashback = amount * 0.07;
56                 break;
57             case "74": // Platinum
58                 cashback = amount * 0.10;
59                 break;
60         }
61     }
62     // Cashback tambahan
63     switch (customerNumber.substring(beginIndex:0, endIndex:2)) { Convert switch to rule switch
64         case "56": // Gold
65             cashback += amount * 0.02;
66             break;
67         case "74": // Platinum
68             cashback += amount * 0.05;
69             break;
70     }
71     return cashback;
72 }
73
Codeium: Refactor | Explain
74
75 public class PraktikumPemlana4 {
76     Run | Debug | Run main | Debug main | Codeium: Refactor | Explain | Generate Javadoc | X
77     public static void main(String[] args) {
78         Scanner scanner = new Scanner(System.in); Resource leak: 'scanner' is never closed
79         Customer customer = new Customer(customerNumber: "812345678", name: "John Doe", balance: 500000, pin: "1234");
80
81         int attempts = 0;
82         while (attempts < 3) {
83             System.out.print(s: "Masukkan PIN: ");
84             String pin = scanner.nextLine();
85             if (customer.authenticate(pin)) {
86                 System.out.println(x: "Autentikasi berhasil!");
87                 break;
88             } else {
89                 attempts++;
90                 System.out.println(x: "PIN salah. Coba lagi.");
91             }
92         }
93
94         if (attempts == 3) {
95             System.out.println(x: "Akun terkunci setelah 3 kali kesalahan.");
96             return;
97         }
98
99         while (true) {
100             System.out.println(x: "\nMenu:");
101             System.out.println(x: "1. Top Up");
102             System.out.println(x: "2. Pembelian");
103             System.out.println(x: "3. Cek Saldo");
104             System.out.println(x: "4. Keluar");
105             System.out.print(s: "Pilih opsi: ");
106             int option = scanner.nextInt();
107
108             switch (option) { Convert switch to rule switch
109                 case 1:
110                     System.out.print(s: "Masukkan jumlah top up: ");
111                     double topUpAmount = scanner.nextDouble();
112                     customer.topUp(topUpAmount);
113                     System.out.println("Top up berhasil. Saldo sekarang: Rp" + customer.getBalance());
114                     break;
115                 case 2:
116                     System.out.print(s: "Masukkan jumlah pembelian: ");
117                     double purchaseAmount = scanner.nextDouble();
118                     if (customer.purchase(purchaseAmount)) {
119                         System.out.println("Pembelian berhasil. Saldo sekarang: Rp" + customer.getBalance());
120                     }
121                     break;
122                 case 3:
123                     System.out.println("Saldo Anda: Rp" + customer.getBalance());
124                     break;
125                 case 4:
126                     System.out.println(x: "Terima kasih! Sampai jumpa.");
127                     return;
128                 default:
129                     System.out.println(x: "Opsi tidak valid.");
130             }
131         }
132     }
133 }

```

C. Syntax

```

import java.util.Scanner;

class Customer {
    private String customerNumber;

```

```

private String name;
private double balance;
private String pin;

public Customer(String customerNumber, String name,
double balance, String pin) {
    this.customerNumber = customerNumber;
    this.name = name;
    this.balance = balance;
    this.pin = pin;
}

public String getCustomerNumber() {
    return customerNumber;
}

public String getName() {
    return name;
}

public double getBalance() {
    return balance;
}

public boolean authenticate(String pin) {
    return this.pin.equals(pin);
}

public void topUp(double amount) {
    balance += amount;
}

public boolean purchase(double amount) {
    if (balance - amount < 10000) {
        System.out.println("Transaksi gagal: Saldo
setelah transaksi kurang dari Rp10.000");
        return false;
    }
    double cashback = calculateCashback(amount);
    balance -= amount;
    balance += cashback;
    return true;
}

private double calculateCashback(double amount) {
    double cashback = 0;
    if (amount > 1000000) {
        switch (customerNumber.substring(0, 2)) {
            case "38": // Silver
                cashback = amount * 0.05;
                break;
            case "56": // Gold
                cashback = amount * 0.07;
                break;
            case "74": // Platinum
                cashback = amount * 0.10;
                break;
        }
    }
    // Cashback tambahan
    switch (customerNumber.substring(0, 2)) {
        case "56": // Gold
            cashback += amount * 0.02;
    }
}

```

```

        break;
        case "74": // Platinum
            cashback += amount * 0.05;
            break;
    }
    return cashback;
}

}

public class PraktikumPemlan4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Customer customer = new Customer("3812345678", "John
Doe", 500000, "1234");

        int attempts = 0;
        while (attempts < 3) {
            System.out.print("Masukkan PIN: ");
            String pin = scanner.nextLine();
            if (customer.authenticate(pin)) {
                System.out.println("Autentikasi berhasil!");
                break;
            } else {
                attempts++;
                System.out.println("PIN salah. Coba lagi.");
            }
        }

        if (attempts == 3) {
            System.out.println("Akun terkunci setelah 3 kali
kesalahan.");
            return;
        }

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Top Up");
            System.out.println("2. Pembelian");
            System.out.println("3. Cek Saldo");
            System.out.println("4. Keluar");
            System.out.print("Pilih opsi: ");
            int option = scanner.nextInt();

            switch (option) {
                case 1:
                    System.out.print("Masukkan jumlah top
up: ");
                    double topUpAmount =
scanner.nextDouble();
                    customer.topUp(topUpAmount);
                    System.out.println("Top up berhasil.
Saldo sekarang: Rp" + customer.getBalance());
                    break;
                case 2:
                    System.out.print("Masukkan jumlah
pembelian: ");
                    double purchaseAmount =
scanner.nextDouble();
                    if (customer.purchase(purchaseAmount)) {
                        System.out.println("Pembelian
berhasil. Saldo sekarang: Rp" + customer.getBalance());
                    }
                    break;
            }
        }
    }
}

```

	<pre> case 3: System.out.println("Saldo Anda: Rp" + customer.getBalance()); break; case 4: System.out.println("Terima kasih! Sampai jumpa."); return; default: System.out.println("Opsi tidak valid."); } } } } </pre>
--	--

D. Penjelasan

```

CodeEditor | Kotlin | Explain
class Customer {
    private String customerNumber;    Field customerNumber can be final
    private String name;              Field name can be final
    private double balance;
    private String pin;               Field pin can be final
}

```

A. Atribut

customerNumber → Nomor identitas nasabah. Dua digit pertama menentukan jenis pelanggan:

- 38 → Silver
- 56 → Gold
- 74 → Platinum

name → Nama nasabah.

balance → Saldo nasabah.

pin → PIN rahasia yang digunakan untuk autentikasi.

B. Konstruktor

```

public Customer(String customerNumber, String name, double balance, String pin) {
}

```

Digunakan untuk membuat objek baru dari Customer dengan data awal.

C. Method

1. Class Customer

- `getCustomerNumber()`, `getName()`, `getBalance()`
Getter untuk mengambil data customer seperti nomor, nama, dan saldo.

- `authenticate(String pin)`
Fungsi untuk mencocokkan PIN yang dimasukkan dengan PIN yang tersimpan.

- `public void topUp(double amount) {`
- `balance += amount;`
Menambahkan dana ke saldo nasabah.

```
- purchase(double amount)
```

Melakukan pembelian. Cek dulu apakah saldo cukup, minimal **Rp10.000** harus tersisa setelah transaksi.

- Jika saldo mencukupi, dana akan dikurangi.
- Cashback akan dihitung dan ditambahkan kembali ke saldo.

```
- CalculateCashback(double amount)
```

Fungsi privat yang menghitung total cashback berdasarkan:

- **Jumlah pembelian**
- **Jenis pelanggan**
- **Ketentuan Cashback:**
 - Jika jumlah pembelian > Rp1.000.000:
 - Silver (38): 5%
 - Gold (56): 7%
 - Platinum (74): 10%
 - Cashback tambahan:
 - Gold (56): +2%
 - Platinum (74): +5%

2. Class Main (PraktikumPemlan4)

➤ Login Menggunakan PIN

```
Customer customer = new Customer(customerNumber:"3812345678", name:"John Doe", balance:500000, pin:"1234");
```

Objek nasabah dibuat dengan data contoh (Silver, saldo Rp500.000).

Lalu dilakukan proses autentikasi:

- Maksimal 3 kali percobaan PIN.
- Jika gagal 3 kali, akun dikunci.

➤ Jika Berhasil

Setelah berhasil login, pengguna bisa memilih dari menu:

```
Menu:
1. Top Up
2. Pembelian
3. Cek Saldo
4. Keluar
Pilih opsi: █
```

Opsi 1 – Top Up:

```
customer.topUp(topUpAmount);
```

Menambahkan saldo nasabah.

Opsi 2 – Pembelian:

customer.purchase(purchaseAmount);

Melakukan pembelian dan otomatis mengatur saldo serta cashback.

Opsi 3 – Cek Saldo:

Menampilkan saldo saat ini.

Opsi 4 – Keluar:

Mengakhiri program.