TUGAS #3

PEMROGRAMAN BERORIENTASI OBJEK

DOSEN PENGAJAR

Alun Sujjada, S.Kom, M.T

NAMA : ARI PERDIAN
NIM : 20220040072

Kelas :

TI22J

Kelas Account:

```
public class Account {
    protected double balance;

    public Account(double balance) {
        this.balance = balance;
    }

    public double getBalance() {
        return balance;
    }

    public boolean deposit(double amount) {
        if(amount > 0) {
            balance += amount;
            return true;
        } else {
            return false;
        }
    }

    public boolean withdraw(double amount) {
        if(balance - amount >= 0.0) {
            balance -= amount;
            return true;
        } else {
            return false;
        }
```

- Pada kelas Account, terdapat atribut balance yang bersifat protected. Hal ini memungkinkan kelas turunan untuk mengakses atribut tersebut secara langsung.
- Konstruktor Account digunakan untuk memberikan nilai awal pada saldo (balance).
- Metode getBalance digunakan untuk mendapatkan nilai saldo.
- Metode deposit digunakan untuk menambah nilai saldo. Operasi ini hanya akan berhasil jika jumlah yang disetorkan lebih besar dari nol.
- Metode withdraw digunakan untuk mengurangi nilai saldo. Operasi ini hanya akan berhasil jika saldo mencukupi dan jumlah yang ditarik lebih besar dari nol.

Subkelas SavingAccount:

```
//NAMA  : ARI PERDIAN
//KELAS : TI22J
//NIM   : 20220040072

public class SavingAccount extends Account {
    private double interestRate;

    public SavingAccount(double balance, double interestRate) {
        super(balance);
        this.interestRate = interestRate;
    }
}
```

- Subkelas SavingAccount merupakan turunan dari Account dan memiliki tambahan atribut interestRate, yang digunakan untuk menangani bunga.
- Konstruktor SavingAccount menerima parameter balance dan interestRate. Nilai balance diset melalui konstruktor induk menggunakan super(balance), sedangkan nilai interestRate di-set langsung.
- Dalam implementasi ini, belum ada logika khusus untuk menangani bunga (yang umumnya diterapkan dalam perhitungan bunga berkala atau ketika melakukan operasi tertentu).

Subkelas CheckingAccount:

```
5   public class CheckingAccount extends Account {
6       private double overdraftProtection;
7
8       public CheckingAccount(double balance, double protect) {
9           super(balance);
10          this.overdraftProtection = protect;
11      }
12
13      public CheckingAccount(double balance) {
14          this(balance, -1.0); // Default overdraftProtection is -1.0
15      }
16
17      @Override
        public boolean withdraw(double amount) {
19          double overdraftNeeded = amount - balance;
20
21          if(balance - amount >= 0.0) {
22              balance -= amount;
23              return true;
24          } else if(overdraftProtection != -1.0 && overdraftProtection >= overdraftNeeded) {
25              balance = 0.0;
26              overdraftProtection -= overdraftNeeded;
27              return true;
28          } else {
29              return false;
30          }
31      }
```

- Subkelas CheckingAccount juga merupakan turunan dari Account dan memiliki atribut tambahan overdraftProtection untuk menangani proteksi cerukan.
- Terdapat dua konstruktor: satu menerima balance dan protect, sedangkan yang lain hanya menerima balance saja. Konstruktor pertama digunakan untuk menginisialisasi saldo dan proteksi cerukan, sedangkan konstruktor kedua mengatur proteksi cerukan default ke -1.0 (tidak ada proteksi cerukan).
- Metode withdraw di-overriding untuk menangani logika penarikan dana sesuai dengan deskripsi yang diberikan. Jika saldo mencukupi, penarikan akan berhasil. Namun, jika tidak cukup saldo, penarikan akan berhasil jika ada proteksi cerukan yang mencukupi. Jika tidak ada proteksi cerukan atau tidak cukup proteksi cerukan, penarikan akan gagal.