TUGAS SESI 3

Dibuat oleh:

Nama : **Mochamad Naufal Rizky**

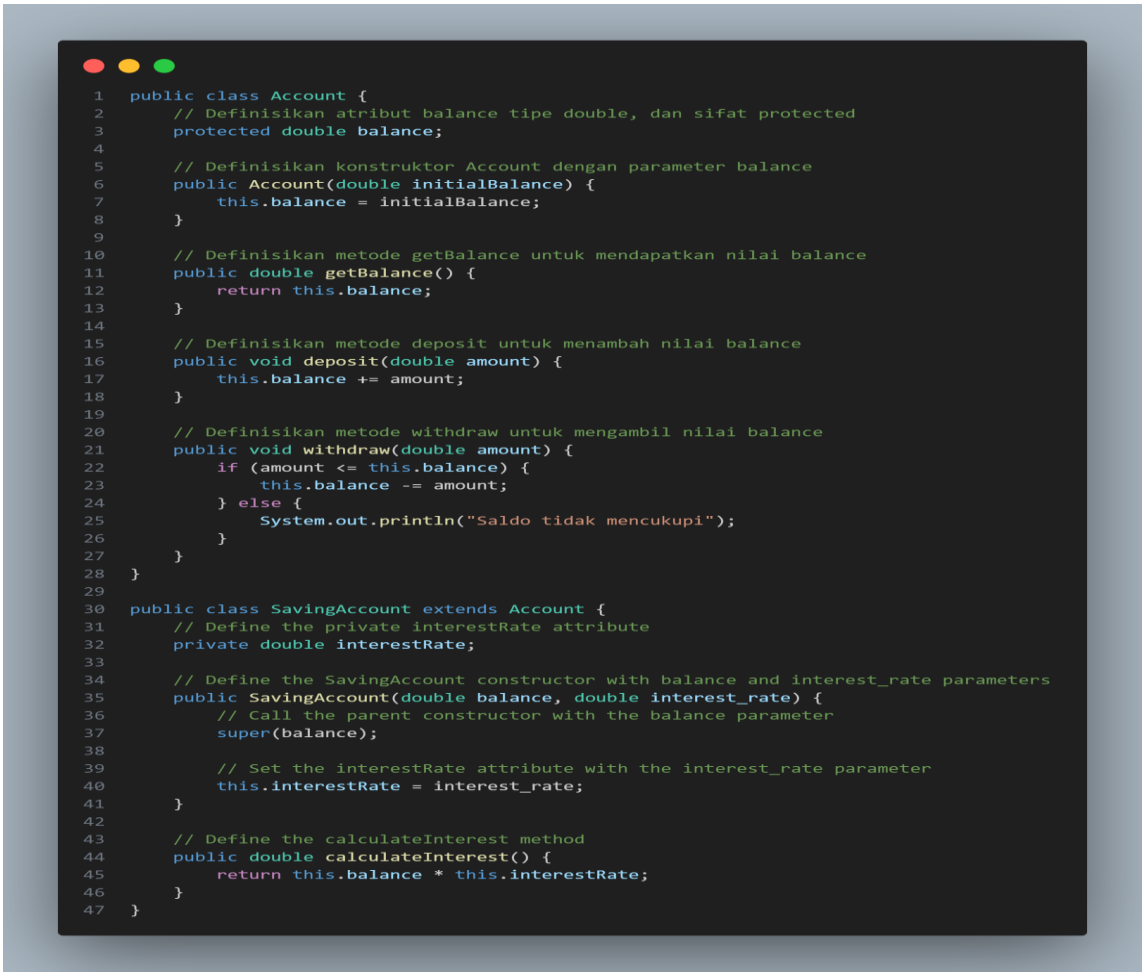
Kelas : **TI22C**

NIM : **20220040210**

MK : **PBO**

Penganalisaan setiap percobaan

1.



```java
public class Account {
    // Definisikan atribut balance tipe double, dan sifat protected
    protected double balance;

    // Definisikan konstruktor Account dengan parameter balance
    public Account(double initialBalance) {
        this.balance = initialBalance;
    }

    // Definisikan metode getBalance untuk mendapatkan nilai balance
    public double getBalance() {
        return this.balance;
    }

    // Definisikan metode deposit untuk menambah nilai balance
    public void deposit(double amount) {
        this.balance += amount;
    }

    // Definisikan metode withdraw untuk mengambil nilai balance
    public void withdraw(double amount) {
        if (amount <= this.balance) {
            this.balance -= amount;
        } else {
            System.out.println("Saldo tidak mencukupi");
        }
    }
}

public class SavingAccount extends Account {
    // Define the private interestRate attribute
    private double interestRate;

    // Define the SavingAccount constructor with balance and interest_rate parameters
    public SavingAccount(double balance, double interest_rate) {
        // Call the parent constructor with the balance parameter
        super(balance);

        // Set the interestRate attribute with the interest_rate parameter
        this.interestRate = interest_rate;
    }

    // Define the calculateInterest method
    public double calculateInterest() {
        return this.balance * this.interestRate;
    }
}
```

Berikut adalah analisa setiap percobaan dalam kodingan Account class di atas:

Atribut balance: Atribut ini bertipe double dan memiliki sifat protected. Hal ini berarti bahwa atribut ini dapat diakses oleh kelas yang sama dan kelas turunannya. Atribut ini digunakan untuk menyimpan saldo akun.

Konstruktor Account: Konstruktor ini memiliki parameter initialBalance yang digunakan untuk memberi nilai awal atribut balance. Konstruktor ini dipanggil saat membuat objek baru dari kelas Account.

Metode getBalance: Metode ini digunakan untuk mendapatkan nilai dari atribut balance. Metode ini hanya membalikkan nilai atribut balance dan tidak melakukan apa-apa pengolahan lain.

Metode deposit: Metode ini digunakan untuk menambahkan jumlah amount ke atribut balance. Hal ini dilakukan dengan menggunakan operator +=.

Metode withdraw: Metode ini digunakan untuk mengurangi jumlah amount dari atribut balance. Namun, sebelum melakukan pengurangan, metode ini akan memeriksa apakah jumlah amount lebih kecil atau sama dengan nilai saat ini dari atribut balance. Jika amount lebih besar dari nilai saat ini dari atribut balance, maka akan ditampilkan pesan "Saldo tidak mencukupi".

Dalam kelas ini, atribut interestRate bertipe double dan memiliki sifat private. Konstruktor SavingAccount mengambil dua parameter, yaitu balance dan interest_rate. Parameter balance dipassing ke konstruktor Account dengan menggunakan super(balance), sehingga nilai awal dari atribut balance diinisialisasi. Nilai dari parameter interest_rate kemudian diset ke atribut interestRate.

Selain itu, kelas SavingAccount juga memiliki metode calculateInterest() yang digunakan untuk menghitung bunga dari saldo akun. Metode ini mengembalikan hasil perkalian antara nilai balance dan interestRate.

2.

```java
public class CheckingAccount extends Account {
    private double overdraftProtection;

    public CheckingAccount(double balance, double protect) {
        super(balance + protect);
        this.overdraftProtection = protect;
    }

    public CheckingAccount(double balance) {
        this(balance, -1.0);
    }

    @Override
    public boolean withdraw(double amount) {
        double overdraftNeeded = amount - this.balance;
        if (overdraftNeeded > 0) {
            if (this.overdraftProtection < 0 || this.overdraftProtection < overdraftNeeded) {
                return false;
            } else {
                this.balance = 0.0;
                this.overdraftProtection -= overdraftNeeded;
                return true;
            }
        } else {
            if (this.balance < amount) {
                return false;
            } else {
                this.balance -= amount;
                return true;
            }
        }
    }
}
```

Kelas CheckingAccount ini merupakan turunan dari kelas Account. Kelas ini memiliki atribut overdraftProtection dengan tipe data double dan sifat private.

Kelas ini memiliki dua constructor. Constructor pertama memiliki dua parameter, yaitu balance dan protect. Constructor ini akan memanggil constructor dari kelas Account dengan menggunakan super(balance + protect) dan akan menetapkan nilai dari atribut overdraftProtection dengan protect.

Constructor kedua memiliki satu parameter, yaitu balance. Constructor ini akan memanggil constructor lokal dengan menggunakan this(balance, -1.0). Ini berarti bahwa jika hanya

menggunakan constructor ini, maka overdraftProtection akan bernilai -1.0, yang berarti bahwa pada akun tidak terdapat overdraft protection.

Kelas ini juga meng-override method withdraw dari kelas Account. Method ini akan melakukan cek terhadap saldo (balance) apakah jumlahnya cukup jika terjadi pengambilan sejumlah uang (amount). Jika balance - amount menjadi 0.0 atau lebih besar, maka proses pengambilan uang diperbolehkan dan method akan mengembalikan nilai true. Selain itu, jika balance - amount kurang dari 0.0, maka akan dilakukan cek terhadap overdraftProtection. Jika overdraftProtection kurang dari 0 atau overdraftProtection kurang dari overdraftNeeded, maka proses pengambilan uang gagal dan method akan mengembalikan nilai false. Jika overdraftProtection lebih besar dari overdraftNeeded, maka proses pengambilan uang berhasil dan method akan mengembalikan nilai true. Selanjutnya, balance akan diset menjadi 0.0 dan overdraftProtection akan dikurangi dengan overdraftNeeded.

Selain itu, kelas ini juga memiliki atribut saldo yang merupakan jumlah dari balance dan overdraftProtection. overdraftProtection merupakan saldo minimal, yaitu saldo yang diharapkan tidak boleh diambil pada suatu rekening, kecuali konsumen ingin menutup rekening. Dalam kelas ini, kita memiliki atribut overdraftProtection dengan tipe double dan sifat private. Kita juga memiliki dua constructor:

Constructor pertama memiliki dua parameter, yaitu balance dan overdraftProtection. Constructor ini akan memanggil constructor dari kelas Account dengan balance + overdraftProtection sebagai nilai awal.

Constructor kedua memiliki satu parameter, yaitu balance. Constructor ini akan memanggil constructor lokal dengan balance dan -1.0 sebagai nilai default dari overdraftProtection, yang berarti bahwa pada akun tidak terdapat overdraft protection.

Kelas ini juga meng-override method withdraw dari kelas Account. Method ini akan melakukan cek terhadap saldo (balance) apakah jumlahnya cukup jika terjadi pengambilan sejumlah uang (amount). Jika balance cukup, maka balance akan dikurangi dengan amount. Jika balance tidak cukup, maka akan dilakukan cek terhadap overdraftProtection. Jika overdraftProtection lebih besar dari 0, maka akan dilakukan cek terhadap kelebihan penarikan dana (overdraftNeeded). Jika overdraftProtection cukup untuk mengurangi kelebihan penarikan dana, maka balance akan diset menjadi 0 dan overdraftProtection akan dikurangi dengan overdraftNeeded. Jika overdraftProtection tidak cukup, maka proses pengambilan uang gagal dan method akan mengembalikan nilai false.

Kelas ini juga memiliki getter dan setter untuk overdraftProtection.