

Nama : Rustiani

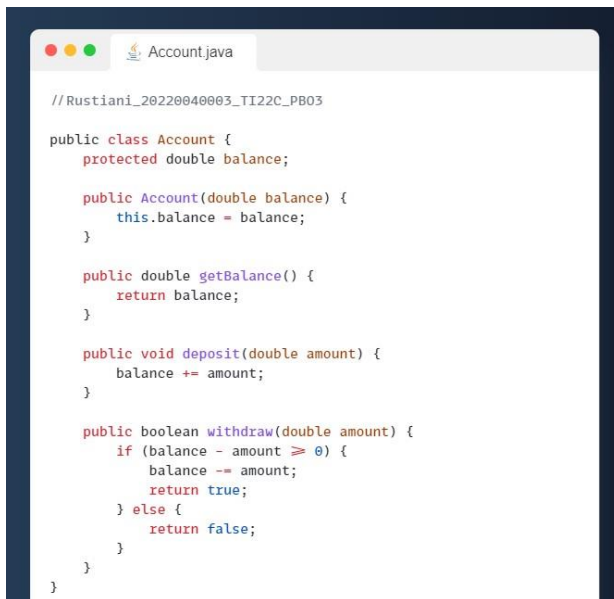
NIM : 2022040003

Kelas : TI22C

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK SESI 3

ANALISA SETIAP PERCOBAAN PRAKTIKUM

➤ Account.java



```
// Rustiani_2022040003_TI22C_PB03

public class Account {
    protected double balance;

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

    public double getBalance() {
        return balance;
    }

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

    public boolean withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            return true;
        } else {
            return false;
        }
    }
}
```

Account.java merupakan kelas dasar yang mewakili akun umum dengan saldo, dan menyediakan fungsi dasar seperti deposit dan withdraw.

- Kelas ini memiliki atribut balance yang diatur sebagai protected, yang berarti hanya kelas turunan yang dapat mengaksesnya secara langsung
- Konstruktor menerima parameter saldo awal dan menginisialisasi saldo
- Method getBalance() mengembalikan nilai saldo.
- Method deposit(double amount) menambahkan jumlah ke saldo.
- Method withdraw(double amount) mencoba untuk menarik jumlah tertentu dari saldo, mengembalikan true jika penarikan berhasil dan false jika tidak.

➤ SavingAccount.java

```
//Rustiani_20220040003_TI22C_PB03

public class SavingAccount extends Account {
    public SavingAccount(double balance, double interest_rate) {
        super(balance);
    }
}
```

SavingAccount.java merupakan kelas turunan dari Account yang mewakili akun tabungan dengan tambahan atribut tingkat bunga.

- Kelas ini memiliki atribut tambahan interestRate.
- Konstruktor menerima saldo awal dan tingkat bunga, kemudian memanggil konstruktor kelas induk dengan menggunakan super (balance).
- Tidak ada tambahan method pada kelas ini karena hanya memiliki atribut tambahan dan mewarisi fungsi dari kelas induk.

➤ CheckingAccount.java

```
// put your code here Rustiani_20220040003_TI22C_PB03

public class CheckingAccount extends Account {
    private double overdraftProtection;

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

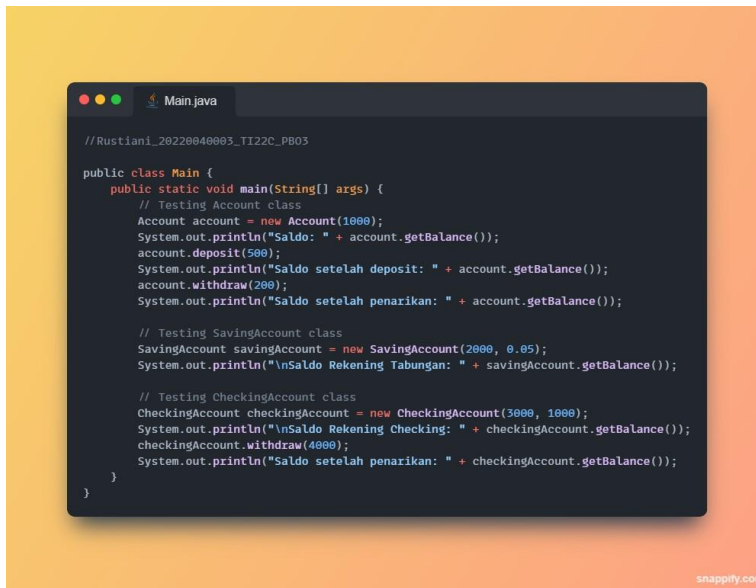
    public CheckingAccount(double balance) {
        this(balance, -1.0); // Default overdraftProtection is -1.0
        this.balance += overdraftProtection;
    }

    @Override
    public boolean withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            return true;
        } else {
            if (overdraftProtection == -1.0 || overdraftProtection < (amount - balance)) {
                return false;
            } else {
                balance = 0.0;
                overdraftProtection -= (amount - balance);
                return true;
            }
        }
    }
}
```

CheckingAccount.java merupakan kelas turunan dari account yang mewakili akun check dengan tambahan fitur proteksi overdraft.

- Kelas ini mewakili atribut tambahan overdraftProtection
- Ada dua konstruktor yaitu, satu menerima saldo awal dan jumlah proteksi overdraft, dan yang lain menerima hanya saldo awal dengan proteksi overdraft default.
- Method withdraw (double amount) dioverride untuk menangani logika penarikan khusus akun check, yang memeriksa saldo dan proteksi overdraft.

➤ Main.java



Main.java merupakan tempat pengujian dilakukan untuk semua kelas.

- Membuat objek dari kelas Account, SavingAccount dan CheckingAccount untuk menguji fungsionalitas mereka.
- Menggunakan method yang telah didefinisikan untuk memeriksa deposit, penarikan dan saldo setelah operasi tersebut dilakukan.

