

Nama : Irma Nurmalasari  
Nim : 20220040127  
Kelas : Teknik Informatika/TI22C  
Mata Kuliah : Pemrograman Berorientasi Object

### TUGAS SESI 3

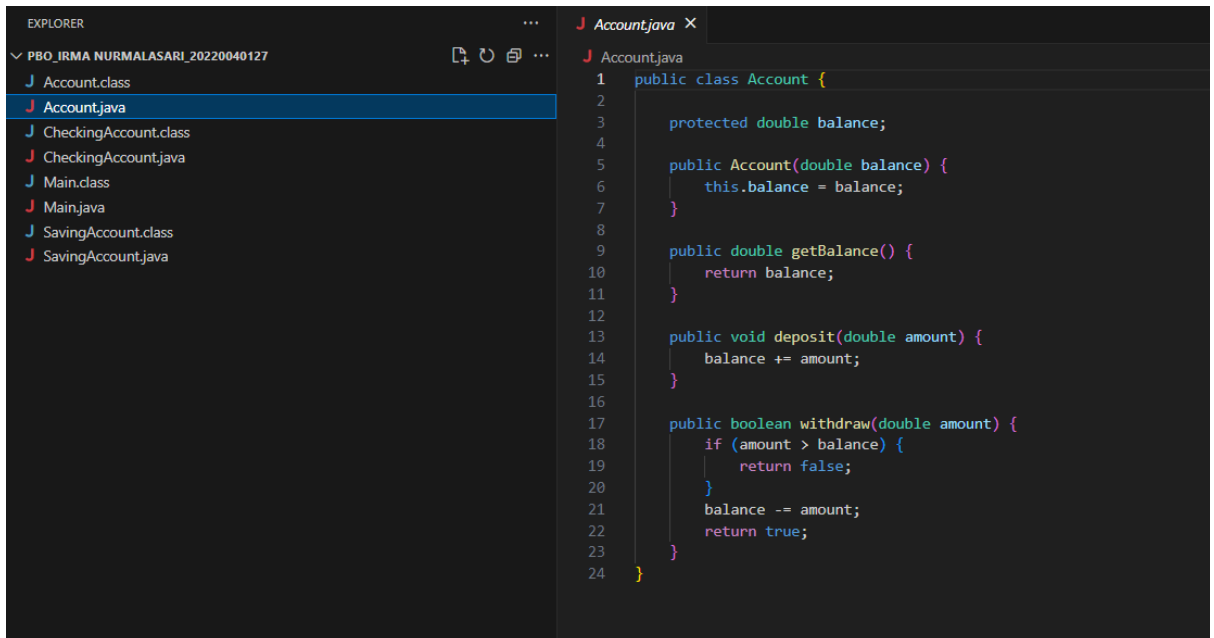
1. Kode Program diupload ke github dengan nama repository praktikum-inheritance
2. Berikan analisa setiap percobaan dalam bentuk File teks pdf dan upload juga ke github praktikum\_inheritance

### JAWAB

1. <https://github.com/AbrarRamadava/Praktikum-Inheritance>
2. Analisa

#### a) Account.java

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



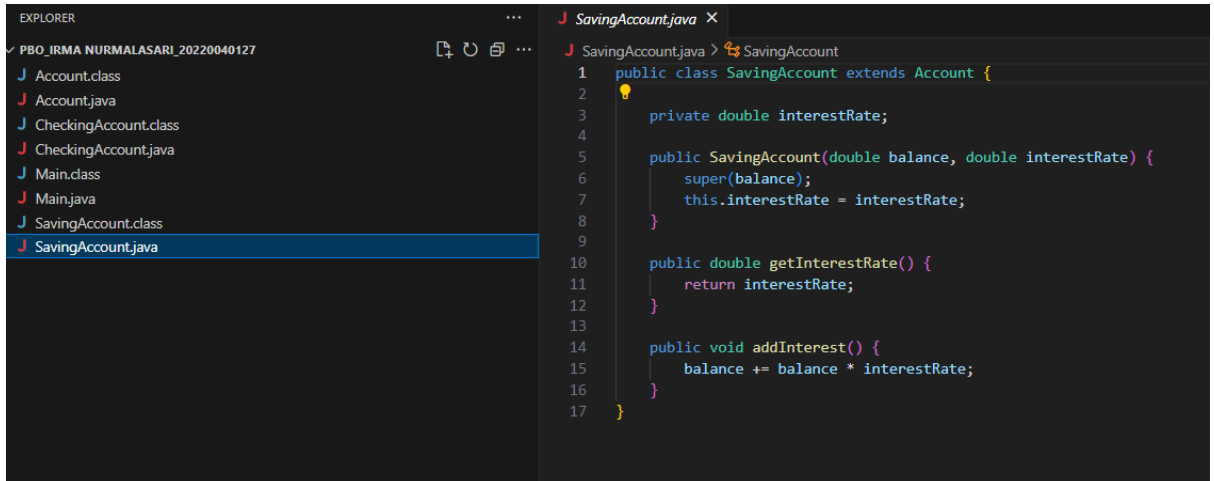
```
1 public class Account {  
2  
3     protected double balance;  
4  
5     public Account(double balance) {  
6         this.balance = balance;  
7     }  
8  
9     public double getBalance() {  
10        return balance;  
11    }  
12  
13    public void deposit(double amount) {  
14        balance += amount;  
15    }  
16  
17    public boolean withdraw(double amount) {  
18        if (amount > balance) {  
19            return false;  
20        }  
21        balance -= amount;  
22        return true;  
23    }  
24 }
```

- 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.

## b) SavingAccount.java

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



```

1 public class SavingAccount extends Account {
2
3     private double interestRate;
4
5     public SavingAccount(double balance, double interestRate) {
6         super(balance);
7         this.interestRate = interestRate;
8     }
9
10    public double getInterestRate() {
11        return interestRate;
12    }
13
14    public void addInterest() {
15        balance += balance * interestRate;
16    }
17 }

```

- 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.

## c) CheckingAccount.java

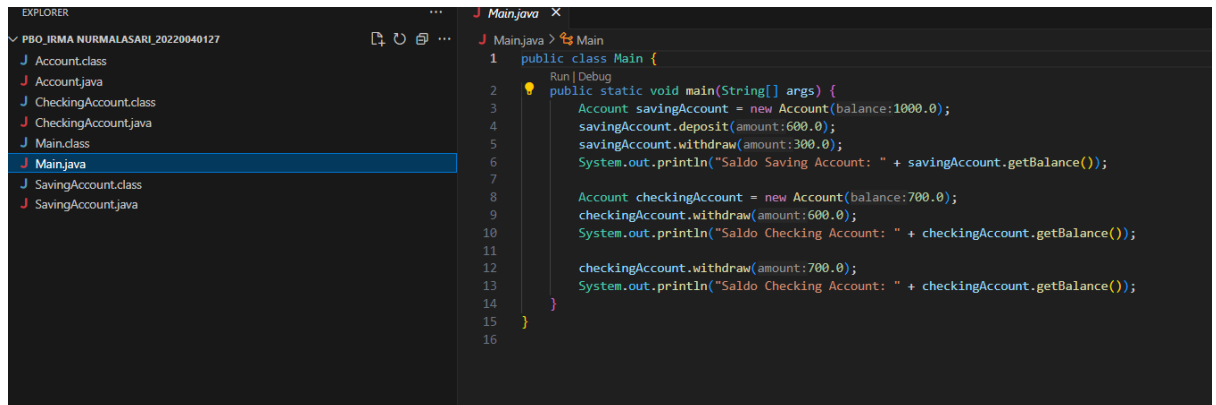
`CheckingAccount.java` adalah kelas turunan dari `Account` yang mewakili akun cek dengan tambahan fitur

proteksi overdraft.

- Kelas ini memiliki atribut tambahan `overdraftProtection`.
- Ada dua konstruktor: 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 cek, yang memeriksa saldo dan proteksi overdraft.

#### d) Main.java

Main.java adalah kelas tempat pengujian dilakukan untuk semua kelas.



```
1 public class Main {
2     public static void main(String[] args) {
3         Account savingAccount = new Account(balance:1000.0);
4         savingAccount.deposit(amount:600.0);
5         savingAccount.withdraw(amount:300.0);
6         System.out.println("Saldo Saving Account: " + savingAccount.getBalance());
7
8         Account checkingAccount = new Account(balance:700.0);
9         checkingAccount.withdraw(amount:600.0);
10        System.out.println("Saldo Checking Account: " + checkingAccount.getBalance());
11
12        checkingAccount.withdraw(amount:700.0);
13        System.out.println("Saldo Checking Account: " + checkingAccount.getBalance());
14    }
15 }
16
```

- 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.