

NAMA : RISKI ABDILAH PRATAMA

NIM : 20220040091

KELAS : TI22C

TUGAS : PBO SESI 3

Penganalisaan setiap percobaan 1.

```
1 public class Account {
2     // Definisikan atribut balance tipe double, dan sifat protected
3     protected double balance;
4
5     // Definisikan konstruktor Account dengan parameter balance
6     public Account(double initialBalance) {
7         this.balance = initialBalance;
8     }
9
10    // Definisikan metode getBalance untuk mendapatkan nilai balance
11    public double getBalance() {
12        return this.balance;
13    }
14
15    // Definisikan metode deposit untuk menambah nilai balance
16    public void deposit(double amount) {
17        this.balance += amount;
18    }
19
20    // Definisikan metode withdraw untuk mengambil nilai balance
21    public void withdraw(double amount) {
22        if (amount <= this.balance) {
23            this.balance -= amount;
24        } else {
25            System.out.println("Saldo tidak mencukupi");
26        }
27    }
28 }
29
30 public class SavingAccount extends Account {
31     // Define the private interestRate attribute
32     private double interestRate;
33
34     // Define the SavingAccount constructor with balance and interest_rate parameters
35     public SavingAccount(double balance, double interest_rate) {
36         // Call the parent constructor with the balance parameter
37         super(balance);
38
39         // Set the interestRate attribute with the interest_rate parameter
40         this.interestRate = interest_rate;
41     }
42
43     // Define the calculateInterest method
44     public double calculateInterest() {
45         return this.balance * this.interestRate;
46     }
47 }
```

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 mengembalikan 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.

```
1 public class CheckingAccount extends Account {
2     private double overdraftProtection;
3
4     public CheckingAccount(double balance, double protect) {
5         super(balance + protect);
6         this.overdraftProtection = protect;
7     }
8
9     public CheckingAccount(double balance) {
10        this(balance, -1.0);
11    }
12
13    @Override
14    public boolean withdraw(double amount) {
15        double overdraftNeeded = amount - this.balance;
16        if (overdraftNeeded > 0) {
17            if (this.overdraftProtection < 0 || this.overdraftProtection < overdraftNeeded) {
18                return false;
19            } else {
20                this.balance = 0.0;
21                this.overdraftProtection -= overdraftNeeded;
22                return true;
23            }
24        } else {
25            if (this.balance < amount) {
26                return false;
27            } else {
28                this.balance -= amount;
29                return true;
30            }
31        }
32    }
33 }
```

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`.