

LAPORAN PRAKTIKUM 1
ALGORITMA PEMOGRAMAN DAN
KOMPUTER 2 “Class and Object”



Disusun oleh

Jiryan Farokhi/5002221102

Asisten Laboratorium

Muhammad Andhika Reswara/5002211086 dan Komang
Ryaandhi Suandita/5002211109

Dosen Pengampu

Dr. Budi Setiyono, S.Si, MT / 19720207 199702 1 001

DEPARTEMEN
MATEMATIKA FAKULTAS
SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH
NOPEMBER SURABAYA
2024

BAB 1

DESKRIPSI

1. Diberikan sebuah pertanyaan untuk membuat sebuah program java main class dan class untuk menghitung saldo dari tabungan setiap kali transaksi dilaksanakan dengan saldo awal yang telah ditentukan. Pertama program dimulai dengan membuat objek tabungan dengan saldo awal sebesar 5000. Objek tabungan dibuat dengan menggunakan konstruktor yang menerima saldo awal. program mencetak saldo awal user. Dilakukan transaksi pertama, yaitu menyimpan uang sebesar 3000. Metode simpanUang dijalankan dengan argumen 3000, sehingga saldo bertambah menjadi 8000. Pesan "Jumlah uang yang disimpan: 3000" dicetak. Dilakukan transaksi kedua, yaitu mengambil uang sebesar 6000. Metode ambilUang dijalankan dengan argumen 6000. Karena saldo saat ini (8000) lebih dari jumlah yang akan diambil (6000), saldo dikurangi menjadi 2000. Pesan "Jumlah uang yang diambil: 6000" dicetak. Dilakukan transaksi ketiga, yaitu menyimpan uang sebesar 3500. Metode simpanUang dijalankan dengan argumen 3500, sehingga saldo bertambah menjadi 5500. Pesan "Jumlah uang yang disimpan: 3500" dicetak. Dilakukan transaksi keempat, yaitu mengambil uang sebesar 4000. Metode ambilUang dijalankan dengan argumen 4000. Karena saldo saat ini (5500) lebih dari jumlah yang akan diambil (4000), saldo dikurangi menjadi 1500. Pesan "Jumlah uang yang diambil: 4000" dicetak. Dilakukan transaksi kelima, yaitu mengambil uang sebesar 1600. Metode ambilUang dijalankan dengan argumen 1600, sehingga saldo didapatkan saldo saat ini (1500) tidak lebih dari atau sama dengan 1600 sehingga pesan "Saldo tidak mencukupi untuk melakukan penarikan sejumlah 1600" dilakukan. Operasi akan diskip dan dilanjutkan ke operasi selanjutnya. Dilakukan transaksi keenam, yaitu menyimpan uang sebesar 1600. Metode simpanUang dijalankan dengan argumen 1600 sehingga saldo bertambah menjadi 3500. Pesan "Jumlah uang yang disimpan: 1600" dicetak. Setelah semua transaksi selesai, saldo akhir user(3500) dicetak sebagai output.
2. Diberikan sebuah pertanyaan untuk membuat sebuah program java main class dan class untuk melakukan transfer antara user1 dengan user2 sebesar 4000 dengan saldo awal user1 sebesar 5000 dan user2 sebesar 2000. Alur program dimulai dengan definisi kelas User, yang bertanggung jawab untuk merepresentasikan pengguna dengan atribut seperti nama pengguna (username) dan saldo (saldo). Setiap objek User memiliki metode untuk mendapatkan dan mengubah nilai saldo, serta metode untuk melakukan transfer saldo ke pengguna lain. Kemudian, kelas Main digunakan sebagai kelas utama yang berisi metode main(). Di dalam metode main(), dua objek User yaitu user1 dan user2 dibuat dengan saldo awal yang telah ditentukan. Saldo masing-masing pengguna kemudian ditampilkan untuk memberikan gambaran tentang status saldo awal. Program kemudian menentukan jumlah transfer yang akan dilakukan antara dua pengguna. Metode transfer() dipanggil untuk melakukan transfer saldo dari user1 ke user2 dengan jumlah yang telah ditentukan sebelumnya. Setelah transfer selesai, saldo masing-masing pengguna diperbarui dan ditampilkan kembali untuk memverifikasi bahwa transfer telah berhasil dilakukan. Dengan menggunakan pendekatan objek dan kelas, program ini memungkinkan pengelolaan saldo antar pengguna secara efisien dan terstruktur.

BAB 2

SOURCE CODE

Soal 1

- Main Class

```
package Praktikumclass;

public class bankoperation {

    public static void main(String[] args) {

        tabungan user = new tabungan(5000);

        System.out.println("Saldo awal : " + user.getSaldo());

        user.simpanUang(3000);
        user.ambilUang(6000);
        user.simpanUang(3500);
        user.ambilUang(4000);
        user.ambilUang(1600);
        user.simpanUang(2000);

        System.out.println("Saldo sekarang : " + user.getSaldo());
    }
}
```

- Class

```
package Praktikumclass;

public class tabungan {
    public int saldo;
    public int simpan;
    public int ambil;

    public tabungan(int saldoAwal){
        saldo = saldoAwal;
    }

    public void simpanUang(int jumlah) {
        saldo += jumlah;
        System.out.println("Jumlah uang yang disimpan : " + jumlah);
    }

    public void ambilUang(int jumlah){
        if(saldo >= jumlah){
            saldo -= jumlah;
            System.out.println("Jumlah uang yang diambil : " + jumlah);
        }else{
            System.out.println("Saldo tidak mencukupi untuk melakukan penarikan sejumlah " + jumlah);
        }
    }
}
```

```

        public int getSaldo(){
            return saldo;
        }
    }
}

```

Soal 2

- Main Class

```

package Praktikumclass;

public class bankoperation {

    public static void main(String[] args) {

        tabungan user1 = new tabungan("User 1", 5000);
        tabungan user2 = new tabungan("User 2", 2000);

        System.out.println("Saldo " + user1.getUsername() + ": " +
user1.getSaldo());
        System.out.println("Saldo " + user2.getUsername() + ": " +
user2.getSaldo());

        int transferAmount = 4000;
        user1.transfer(user2, transferAmount);

        System.out.println("Saldo " + user1.getUsername() + ": " +
user1.getSaldo());
        System.out.println("Saldo " + user2.getUsername() + ": " +
user2.getSaldo());
    }
}

```

- Class

```

package Praktikumclass;

public class tabungan {
    private String username;
    private int saldo;

    public tabungan(String username, int saldo) {
        this.username = username;
        this.saldo = saldo;
    }

    public String getUsername() {
        return username;
    }

    public int getSaldo() {
        return saldo;
    }

    public void setSaldo(int saldo) {

```

```
        this.saldo = saldo;
    }

    public void transfer(tabungan penerima, int jumlah) {
        if (jumlah > saldo) {
            System.out.println("Transfer gagal. Saldo tidak mencukupi.");
        } else {
            saldo -= jumlah;
            penerima.setSaldo(penerima.getSaldo() + jumlah);
            System.out.println("Transfer berhasil dilakukan ");
        }
    }
}
```

BAB 3

OUTPUT PROGRAM

Soal 1

The screenshot shows an IDE with the `bankoperation.java` file open. The code defines a `bankoperation` class with a `main` method. Inside `main`, a `tabungan` object is created with an initial balance of 5000. The program then performs a series of transactions: depositing 3000, withdrawing 6000, depositing 3500, withdrawing 4000, and depositing 2000. The output window shows the results of these operations.

```
9  * @author jirya
10 */
11 public class bankoperation {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17
18         tabungan user = new tabungan(5000);
19
20         System.out.println("Saldo awal : " + user.getSaldo());
21
22         user.simpanUang(3000);
23         user.ambilUang(6000);
24         user.simpanUang(3500);
25         user.ambilUang(4000);
26         user.simpanUang(2000);
27
28         System.out.println("Saldo sekarang : " + user.getSaldo());
29     }
30 }
31
32 }
```

Output:

```
run:
Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000
Saldo tidak mencukupi untuk melakukan penarikan sejumlah 1600
Jumlah uang yang disimpan : 2000
Saldo sekarang : 3500
```

The screenshot shows an IDE with the `tabungan.java` file open. The code defines a `tabungan` class with methods for depositing and withdrawing money, and a method to get the current balance. The output window shows the results of these operations.

```
13
14 public tabungan(int saldoAwal){
15     saldo = saldoAwal;
16 }
17
18 public void simpanUang(int jumlah) {
19     saldo += jumlah;
20     System.out.println("Jumlah uang yang disimpan : " + jumlah);
21 }
22
23 public void ambilUang(int jumlah){
24     if(saldo >= jumlah){
25         saldo -= jumlah;
26         System.out.println("Jumlah uang yang diambil : " + jumlah);
27     }else{
28         System.out.println("Saldo tidak mencukupi untuk melakukan penarikan sejumlah " + jumlah);
29     }
30 }
31
32 public int getSaldo(){
33     return saldo;
34 }
35
36 }
```

Output:

```
run:
Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000
Saldo tidak mencukupi untuk melakukan penarikan sejumlah 1600
Jumlah uang yang disimpan : 2000
Saldo sekarang : 3500
```

Soal 2

The screenshot shows an IDE with the `bankoperation.java` file open. The code defines a `bankoperation` class with a `main` method that creates two `tabungan` objects, prints their initial balances, performs a transfer of 4000 from user1 to user2, and prints the updated balances. The output window shows the execution results, confirming the successful transfer.

```
10  /**
11  public class bankoperation {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17
18          tabungan user1 = new tabungan("User 1", 5000);
19          tabungan user2 = new tabungan("User 2", 2000);
20
21          System.out.println("Saldo " + user1.getUsername() + ": " + user1.getSaldo());
22          System.out.println("Saldo " + user2.getUsername() + ": " + user2.getSaldo());
23
24          int transferAmount = 4000;
25          user1.transfer(user2, transferAmount);
26
27          System.out.println("Saldo " + user1.getUsername() + ": " + user1.getSaldo());
28          System.out.println("Saldo " + user2.getUsername() + ": " + user2.getSaldo());
29      }
30  }
31
32  //      tabungan user1 = new tabungan(5000);
33  //      tabungan user2 = new tabungan(2000);
```

bankoperation.java - Properties

Properties	
Name	bankoperation
Extension	java
File Size	1468
Modification Time	May 6, 2024, 1:13:49 PM
All Files	D:\Semester_4\ALPROG_2\rakti...
Classpaths	
Compile Classpath	...
Runtime Classpath	D:\Semester_4\ALPROG_2\rakti...
Boot Classpath	nbjrtfile/C:/Program%20Files/...

Output - praktikumclass (run)

```
run:
Saldo User 1: 5000
Saldo User 2: 2000
Transfer berhasil dilakukan
Saldo User 1: 1000
Saldo User 2: 6000
BUILD SUCCESSFUL (total time: 0 seconds)
```

465 INS Windows (CRLF)

The screenshot shows an IDE with the `tabungan.java` file open. The code defines a `tabungan` class with attributes `username` and `saldo`, and methods `getUsername`, `getSaldo`, `setSaldo`, and `transfer`. The output window shows the execution results, confirming the successful transfer.

```
13  public tabungan(String username, int saldo) {
14      this.username = username;
15      this.saldo = saldo;
16  }
17
18  public String getUsername() {
19      return username;
20  }
21
22  public int getSaldo() {
23      return saldo;
24  }
25
26  public void setSaldo(int saldo) {
27      this.saldo = saldo;
28  }
29
30  public void transfer(tabungan penerima, int jumlah) {
31      if (jumlah > saldo) {
32          System.out.println("Transfer gagal. Saldo tidak mencukupi.");
33      } else {
34          saldo -= jumlah;
35          penerima.setSaldo(penerima.getSaldo() + jumlah);
36          System.out.println("Transfer berhasil dilakukan");
37      }
38  }
```

tabungan.java - Properties

Properties	
Name	tabungan
Extension	java
File Size	1643
Modification Time	May 6, 2024, 1:13:01 PM
All Files	D:\Semester_4\ALPROG_2\rakti...
Classpaths	
Compile Classpath	...
Runtime Classpath	D:\Semester_4\ALPROG_2\rakti...
Boot Classpath	nbjrtfile/C:/Program%20Files/...

Output - praktikumclass (run)

```
run:
Saldo User 1: 5000
Saldo User 2: 2000
Transfer berhasil dilakukan
Saldo User 1: 1000
Saldo User 2: 6000
BUILD SUCCESSFUL (total time: 0 seconds)
```

206 INS Windows (CRLF)