

UTS

Nama: Rizqi Bagus Andrean

Kelas: TI-1D

Absen: 25

Kode Soal: A

Link Github: <https://github.com/bagusok/Tugas-Kuliah/tree/main/Semester%202/Praktek%20Algoritma/uts/bank>

1. Diagram Class Rekening

| Rekening |
|---|
| nama string norek: string tlp: string email: string saldo: double |
| print(): void setor(jumlah: double): void tarik(jumlah: double): void |

Penjelasan Kode Program

```
public class Rekening {  
    String norek, nama, tlp, email;  
    double saldo;  
  
    public Rekening(String norek, String nama, String tlp, String email, double saldo) {  
        this.tlp = tlp;  
        this.email = email;  
        this.norek = norek;  
        this.nama = nama;  
        this.saldo = saldo;  
    }  
}
```

Pada class Rekening terdapat atribut untuk menyimpan norek, nama, tlp, email, dan saldo. Terdapat juga konstruktor untuk memasukkan data awal saat instansiasi.

```

💡 public void print() {
    System.out.println(x:"==== Inforasi Rekening =====");
    System.out.println("No Rekening : " + norek);
    System.out.println("Nama      : " + nama);
    System.out.println("Telepon   : " + tlp);
    System.out.println("Email     : " + email);
    System.out.println("Saldo     : " + saldo);
    System.out.println();
}

```

Fungsi print digunakan untuk menampilkan data pemilik rekening beserta saldonya.

```

public void setor(double jumlah) {
    System.out.println("Nama: " + nama);
    System.out.println("No Rekening: " + norek);
    System.out.println(x:"=====");
    System.out.println("Setor "+ jumlah +" Berhasil");
    System.out.println("Saldo Awal : " + saldo);
    saldo += jumlah;
    System.out.println("Saldo Sekarang : " + saldo);
    System.out.println();
}

```

Fungsi Setor untuk menambah saldo nasabah, fungsi ini membutuhkan parameter jumlah. Pada saat dipanggil akan menampilkan nama dan norek lalu menambah saldo.

```

public void tarik(double jumlah) {

    System.out.println("Nama: " + nama);
    System.out.println("No Rekening: " + norek);
    System.out.println(x:"=====");

    if (jumlah > saldo) {
        System.out.println(x:"Saldo tidak mencukupi");
        System.out.println("Sisa Saldo Anda : " + saldo);
        System.out.println();
    }else{
        System.out.println("Tarik "+ jumlah +" Berhasil");
        System.out.println("Saldo Awal : " + saldo);
        saldo -= jumlah;
        System.out.println("Saldo Sekarang : " + saldo);
        System.out.println();
    }

}
}

```

Fungsi Tarik digunakan untuk menarik saldo, saat fungsi ini dijalankan akan menampilkan nama dan norek nasabah lalu melakukan pengecekan apakah jumlah yang ditarik lebih besar dari saldo. Apabila iya maka penarikan gagal dan akan menampilkan "Saldo tidak mencukupi".

Apabila tidak maka saldo akan dikurangi sejumlah jumlah yang ditarik dan akan menampilkan saldo awal dan saldo akhir.

2. Diagram Class Bank

| |
|--|
| Bank |
| rekening: Rekening[] top: int |
| findRekByName(name: string): void findIndexRekByNorek(norek: string): int showAllRek(): void addRek(rek: Rekening): void sortRekByDesc(): void showRekUnder50: void addSaldo(norek: string, jumlah: double): void tarikSaldo(norek: string, jumlah: double): void |

Penjelasan Kode Program

```

public class Bank {
    private Rekening[] rekening;
    private int top;

    public Bank(int jumlahRekening) {
        rekening = new Rekening[jumlahRekening];
        top = 0;
    }
}

```

Class Bank memiliki atribut rekening dengan tipe data array dari class Rekening digunakan untuk menampung list rekening dari nasabah.

Attribute top digunakan untuk menampung index array terakhir dari rekening yang sudah diisi.

Konstruktor Bank berisi parameter jumlah rekening yang digunakan untuk menampung maksimal rekening yang bisa disimpan oleh bank.

```

public void findRekByName(String name){
    for (int i = 0; i < rekening.length; i++) {
        if (rekening[i].nama.equalsIgnoreCase(name)) {
            rekening[i].print();
        }
    }
}

```

Fungsi findRekByName membutuhkan parameter name. fungsi ini akan melakukan looping sesuai jumlah rekening yang disimpan di bank dan setiap loopnya akan mencocokkan apakah nama yang diinputkan ada yang sama dengan list rekening yang disimpan di bank. Jika ada yang sama maka akan ditampilkan detail rekeningnya dengan memanggil fungsi print yang ada di instance rekening yang disimpan di array rekening yang ditemukan.

```

int findIndexRekByNorek(String norek){
    for (int i = 0; i < rekening.length; i++) {
        if (rekening[i].norek.equalsIgnoreCase(norek)) {
            return i;
        }
    }
    return -1;
}

```

Fungsi findIndexRekByNorek digunakan untuk mencari rekening berdasarkan nomor rekening. Fungsi ini melakukan looping sesuai jumlah maksimal rekening yang disimpan, dan setiap loopnya akan mencocokkan apakah nomor yang diinputkan ada yang sama dengan rekening yang disimpan oleh bank, jika ada maka akan mereturn indexnya dan apabila tidak ada yang sama akan mereturn -1.

```
public void showAllRek(){  
    for(Rekening rek : rekening){  
        rek.print();  
    }  
}
```

Fungsi showAllRek digunakan untuk menampilkan seluruh rekening yang disimpan oleh bank. Didalam fungsi ini melakukan foreach dari array rekening dan memanggil fungsi print dari instance rekening.

```
public void addRek(Rekening rek) {  
    rekening[top] = rek;  
    top++;  
}
```

Fungsi addRek digunakan untuk menambah rekening dengan parameter rek yang bertipe data Rekening. Didalam fungsi ini dia akan memasukkan instance dari rekening ke array rekening dengan index yang kosong, lalu menambah atribut top indexnya dengan 1 untuk setiap data yang dimasukkan.

```

public void sortRekByDesc(){
    for (int i = 0; i < rekening.length; i++) {
        for (int j = 0; j < rekening.length; j++) {
            if (rekening[i].saldo > rekening[j].saldo) {
                Rekening temp = rekening[i];
                rekening[i] = rekening[j];
                rekening[j] = temp;
            }
        }
    }

    showAllRek();
}

```

Fungsi sortRekByDesc digunakan untuk mengurutkan list array rekening dari rekening yang memiliki saldo terbesar ke saldo terkecil. Fungsi ini menggunakan algoritma bubble sort dengan dua loop for yang saling bersarang. Loop for pertama iterasi melalui seluruh elemen array rekening dengan variabel i. Di dalam loop for kedua, metode ini membandingkan saldo elemen array rekening di indeks i dengan saldo elemen array rekening di indeks j. Jika saldo elemen array rekening di indeks i lebih besar daripada saldo elemen array rekening di indeks j, maka dua elemen tersebut ditukar menggunakan variabel temporer temp.

Setelah proses pengurutan selesai, fun showAllRek() dipanggil untuk menampilkan seluruh elemen array rekening setelah diurutkan.

```

public void showRekUnder50() {
    for(Rekening rek : rekening){
        if (rek.saldo < 50000) {
            rek.print();
        }
    }
}

```

Fungsi showRekUnder50 digunakan untuk menampilkan daftar rekening dengan saldo dibawah 50000. Fungsi ini akan melakukan foreach array rekening dan melakukan pengecekan apakah saldo dibawah 50000, jika iya maka akan memanggil fungsi print dari instance rek.

```

public void addSaldo(String norek, double jumlah) {
    int index = findIndexRekByNorek(norek);
    if (index != -1) {
        rekening[index].setor(jumlah);
    } else {
        System.out.println("Rekening tidak ditemukan");
    }
}

```

Fungsi addSaldo digunakan untuk menambah saldo, membutuhkan parameter norek dan jumlah. Sebelum melakukan penambahan maka akan memanggil fungsi findIndexRekByNorek dengan parameter norek untuk mengecek apakah rekening dengan norek yang dimasukkan ada atau tidak, jika fungsi ini mereturn -1 maka akan menampilkan "Rekening tidak ditemukan" dan apabila ditemukan maka akan mereturn indexnya, index ini akan digunakan untuk menambah saldo. Setelah itu akan memanggil fungsi setor dengan parameter jumlah yang ada di instance rekening yang tersimpan di array rekening dengan index yang ditemukan tadi.

```
public void tarikSaldo(String norek, double jumlah) {
    int index = findIndexRekByNorek(norek);
    if (index != -1) {
        rekening[index].tarik(jumlah);
    } else {
        System.out.println("Rekening tidak ditemukan");
    }
}
```

Fungsi tarikSaldo digunakan untuk menarik saldo, fungsi ini membutuhkan parameter norek dan jumlah. Isi fungsi ini kurang lebih sama dengan fungsi addSaldo, yang mencari rekening dulu, namun yang membedakan adalah saat rekening ditemukan dia akan memanggil fungsi tarik dari instance rekening yang disimpan.

Fungsi main

Didalam fungsi ini dilakukan penginputan rekening, nambah saldo, tarik, dan menampilkan list rekening.

Proses menambah rekening

```
Run | Debug
public static void main(String[] args) {

    // INPUT ALL REK
    Bank b = new Bank(jumlahRekening:11);
    b.addRek(new Rekening(norek:"160309273084", nama:"Wallace", tlp:"1-458-264-3263", email:"ligula.Nullam@tacitisociosqu.edu", saldo:10000));
    b.addRek(new Rekening(norek:"161006170573", nama:"Darius", tlp:"1-357-843-0547", email:"nec@lectusjusto.org", saldo:7000));
    b.addRek(new Rekening(norek:"162404012243", nama:"Fuller ", tlp:"571-7062", email:"convallis@vestibulumanteipsum.org", saldo:5000));
    b.addRek(new Rekening(norek:"162705250112", nama:"Malcolm", tlp:"623-0234", email:"porttitor.tellus.non@curabitur.ca", saldo:44000));
    b.addRek(new Rekening(norek:"169712042416", nama:"Geoffrey", tlp:"1-683-416-8323", email:"ut.pellentesque@luctusutpellentesque.com", saldo:50000));
    b.addRek(new Rekening(norek:"161007278862", nama:"Rudyard", tlp:"650-5379", email:"Proin.eget@velitegestaslacinia.ca", saldo:123000));
    b.addRek(new Rekening(norek:"164603294259", nama:"Troy", tlp:"897-7608", email:"pede.Suspendisse.dui@a.ca", saldo:100000));
    b.addRek(new Rekening(norek:"163204213437", nama:"Alec ", tlp:"792-4447", email:"non@mus.com", saldo:34000));
    b.addRek(new Rekening(norek:"161807297229", nama:"Walter", tlp:"863-8209 ", email:"Pellentesque.ut.ipsun@neque.ca", saldo:334544));
    b.addRek(new Rekening(norek:"169503136823", nama:"Simon", tlp:"592-6919 ", email:"tellus.justo.sit@commodoauator.net", saldo:23444));
    b.addRek(new Rekening(norek:"168507083528", nama:"Kamal", tlp:"1-115-339-7678", email:"dictum@nec.edu", saldo:567770));
```

Pertama instansiasi dulu kelas Bank dengan isi parameter jumlah max dari list rekening yang akan diinput dan dimasukkan ke variable b.

Panggil fungsi addRek dengan isi parameter instance dari kelas Rekening. Jadi kita memasukkan parameter sambil menginstansiasi kelas rekening.

Menampilkan semua rekening

```
b.showAllRek();
```

```
633ff2b\redhat.java\jdt_ws\uts_ca39da64\bin' 'bank.Bank' a\uts> & 'C:\Progr  
===== Inforasi Rekening =====  
No Rekening : 160309273084  
Nama : Wallace  
Telepon : 1-458-264-3263  
Email : ligula.Nullam@tacitisociosqu.edu  
Saldo : 10000.0  
  
===== Inforasi Rekening =====  
No Rekening : 161006170573  
Nama : Darius  
Telepon : 1-357-843-0547  
Email : nec@lectusjusto.org  
Saldo : 7000.0  
  
===== Inforasi Rekening =====  
No Rekening : 162404012243  
Nama : Fuller  
Telepon : 571-7062  
Email : convallis@vestibulumanteipsum.org  
Saldo : 5000.0  
  
===== Inforasi Rekening =====  
No Rekening : 162705250112  
Nama : Malcolm  
Telepon : 623-0234  
Email : porttitor.tellus.non@curabitur.ca  
Saldo : 44000.0  
  
===== Inforasi Rekening =====  
No Rekening : 169712042416  
Nama : Geoffrey  
Telepon : 1-683-416-8323  
Email : ut.pellentesque@luctusutpellentesque.com  
Saldo : 50000.0
```

Mengurutkan rekening dari saldo terbesar

```
// Urutkan saldo rekening dari yang terbesar  
b.sortRekByDesc();
```



```
633ff2b\redhat.java\jdt_ws\uts_ca39da64\bin' 'bank.Bank' a\uts>
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 168507083528
```

```
Nama : Kamal
```

```
Telepon : 1-115-339-7678
```

```
Email : dictum@nec.edu
```

```
Saldo : 567770.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 161807297229
```

```
Nama : Walter
```

```
Telepon : 863-8209
```

```
Email : Pellentesque.ut.ipsum@neque.ca
```

```
Saldo : 334544.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 161007278862
```

```
Nama : Rudyard
```

```
Telepon : 650-5379
```

```
Email : Proin.eget@velitegestaslacinia.ca
```

```
Saldo : 123000.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 164603294259
```

```
Nama : Troy
```

```
Telepon : 897-7608
```

```
Email : pede.Suspendisse.dui@a.ca
```

```
Saldo : 100000.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 169712042416
```

```
Nama : Geoffrey
```

```
Telepon : 1-683-416-8323
```

Menampilkan rekening yang memiliki saldo dibawah 50k

```
// Tampilkan Data Rekening yang Saldo nya kurang dari 50.000
```

```
b.showRekUnder50();
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 162705250112
```

```
Nama : Malcolm
```

```
Telepon : 623-0234
```

```
Email : porttitor.tellus.non@Curabitur.ca
```

```
Saldo : 44000.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 163204213437
```

```
Nama : Alec
```

```
Telepon : 792-4447
```

```
Email : non@mus.com
```

```
Saldo : 34000.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 169503136823
```

```
Nama : Simon
```

```
Telepon : 592-6919
```

```
Email : tellus.justo.sit@commodoactor.net
```

```
Saldo : 23444.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 160309273084
```

```
Nama : Wallace
```

```
Telepon : 1-458-264-3263
```

```
Email : ligula.Nullam@tacetisociosqu.edu
```

```
Saldo : 10000.0
```

```
===== Inforasi Rekening =====
```

```
No Rekening : 163006470573
```

Mencari rekening berdasarkan nama

```
// Cari Rekening berdasarkan nama
b.findRekByName("Malcolm");
b.findRekByName("Kamal");
// Nama Joni Tidak Ada
b.findRekByName("Joni");
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\uts>
633ff2b\redhat.java\jdt_ws\uts_ca39da64\bin' 'bank.Bank'
===== Inforasi Rekening =====
No Rekening : 162705250112
Nama       : Malcolm
Telepon    : 623-0234
Email      : porttitor.tellus.non@Curabitur.ca
Saldo      : 44000.0

===== Inforasi Rekening =====
No Rekening : 168507083528
Nama       : Kamal
Telepon    : 1-115-339-7678
Email      : dictum@nec.edu
Saldo      : 567770.0

Rekening dengan nama Joni tidak ditemukan
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\uts>
```

Tarik Uang dengan Jumlah tertentu

```
b.addSaldo("161007278862", 250000);
// Rekening Salah
b.addSaldo("111111111", 250000);

633ff2b\redhat.java\jdt_ws\uts_ca39da64\bin' 'bank.Bank' a\uts>
Nama: Rudyard
No Rekening: 161007278862
=====
Setor 250000.0 Berhasil
Saldo Awal : 123000.0
Saldo Sekarang : 373000.0

Rekening tidak ditemukan
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\uts>
```

Setor uang jumlah tertentu

```
// Setor Uang Rekening Jmlah Tertentu
b.tarikSaldo("161807297229", 100000);
// Saldo Tidak Cukup
b.tarikSaldo("161807297229", 4000000);
```

```
// Rekening Salah  
b.tarikSaldo("111111111", 250000);
```

```
Rekening tidak ditemukan  
● PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\uts> & 'C:\Program Files\Java\jdk-633ff2b\redhat.java\jdt_ws\uts_ca39da64\bin' 'bank.Bank'  
Nama: Walter  
No Rekening: 161807297229  
=====
```

Tarik 100000.0 Berhasil
Saldo Awal : 334544.0
Saldo Sekarang : 234544.0

Nama: Walter
No Rekening: 161807297229
=====

Saldo tidak mencukupi
Sisa Saldo Anda : 234544.0

Rekening tidak ditemukan
○ PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\uts> █

Full Source Code

Class Rekening

```
package bank;  
  
public class Rekening {  
    String norek, nama, tlp, email;  
    double saldo;  
  
    public Rekening(String norek, String nama, String tlp, String email,  
double saldo) {  
        this.tlp = tlp;  
        this.email = email;  
        this.norek = norek;  
        this.nama = nama;  
        this.saldo = saldo;  
    }  
  
    public void print() {  
        System.out.println("===== Inforasi Rekening =====");  
    }  
}
```

```

        System.out.println("No Rekening : " + norek);
        System.out.println("Nama      : " + nama);
        System.out.println("Telepon   : " + tlp);
        System.out.println("Email     : " + email);
        System.out.println("Saldo     : " + saldo);
        System.out.println();
    }

    public void setor(double jumlah) {
        System.out.println("Nama: " + nama);
        System.out.println("No Rekening: " + norek);
        System.out.println("=====");
        System.out.println("Setor "+ jumlah +" Berhasil");
        System.out.println("Saldo Awal : " + saldo);
        saldo += jumlah;
        System.out.println("Saldo Sekarang : " + saldo);
        System.out.println();
    }

    public void tarik(double jumlah) {

        System.out.println("Nama: " + nama);
        System.out.println("No Rekening: " + norek);
        System.out.println("=====");

        if (jumlah > saldo) {
            System.out.println("Saldo tidak mencukupi");
            System.out.println("Sisa Saldo Anda : " + saldo);
            System.out.println();
        }else{
            System.out.println("Tarik "+ jumlah +" Berhasil");
            System.out.println("Saldo Awal : " + saldo);
            saldo -= jumlah;
            System.out.println("Saldo Sekarang : " + saldo);
            System.out.println();
        }

    }

}
}

```

Class Bank

```
package bank;
```

```

public class Bank {

    private Rekening[] rekening;
    private int top;

    public Bank(int jumlahRekening) {
        rekening = new Rekening[jumlahRekening];
        top = 0;
    }

    public void findRekByName(String name){
        for (int i = 0; i < rekening.length; i++) {
            if (rekening[i].nama.equalsIgnoreCase(name)) {
                rekening[i].print();
                return;
            }
        }

        System.out.println("Rekening dengan nama "+ name +" tidak ditemukan");
    }

    int findIndexRekByNorek(String norek){
        for (int i = 0; i < rekening.length; i++) {
            if (rekening[i].norek.equalsIgnoreCase(norek)) {
                return i;
            }
        }
        return -1;
    }

    public void showAllRek(){
        for(Rekening rek : rekening){
            rek.print();
        }
    }

    public void addRek(Rekening rek) {
        rekening[top] = rek;
        top++;
    }

    public void sortRekByDesc(){
        for (int i = 0; i < rekening.length; i++) {
            for (int j = 0; j < rekening.length; j++) {
                if (rekening[i].saldo > rekening[j].saldo) {
                    Rekening temp = rekening[i];
                    rekening[i] = rekening[j];
                    rekening[j] = temp;
                }
            }
        }
    }
}

```

```

        }
    }
}

showAllRek();
}

public void showRekUnder50() {
    for(Rekening rek : rekening){
        if (rek.saldo < 50000) {
            rek.print();
        }
    }
}

public void addSaldo(String norek, double jumlah) {
    int index = findIndexRekByNorek(norek);
    if (index != -1) {
        rekening[index].setor(jumlah);
    } else {
        System.out.println("Rekening tidak ditemukan");
    }
}

public void tarikSaldo(String norek, double jumlah) {
    int index = findIndexRekByNorek(norek);
    if (index != -1) {
        rekening[index].tarik(jumlah);
    } else {
        System.out.println("Rekening tidak ditemukan");
    }
}

public static void main(String[] args) {

    // INPUT ALL REK
    Bank b = new Bank(11);
    b.addRek(new Rekening("160309273084", "Wallace", "1-458-264-3263",
"ligula.Nullam@tacitisociosqu.edu", 10000));
    b.addRek(new Rekening("161006170573", "Darius", "1-357-843-0547",
"nec@lectusjusto.org", 7000));
    b.addRek(new Rekening("162404012243", "Fuller ", "571-7062",
"convallis@Vestibulumanteipsum.org", 5000));
    b.addRek(new Rekening("162705250112", "Malcolm", "623-0234",
"porttitor.tellus.non@Curabitur.ca", 44000));
    b.addRek(new Rekening("169712042416", "Geoffrey", "1-683-416-8323",
"ut.pellentesque@luctusutpellentesque.com", 50000));
}

```

```

        b.addRek(new Rekening("161007278862", "Rudyard", "650-5379",
"Proin.eget@velitegestaslacinia.ca", 123000));
        b.addRek(new Rekening("164603294259", "Troy", "897-7608",
"pede.Suspendisse.dui@a.ca", 100000));
        b.addRek(new Rekening("163204213437", "Alec ", "792-4447",
"non@mus.com", 34000));
        b.addRek(new Rekening("161807297229", "Walter", "863-8209 ",
"Pellentesque.ut.ipsium@neque.ca", 334544));
        b.addRek(new Rekening("169503136823", "Simon", "592-6919 ",
"tellus.justo.sit@commodoauctor.net", 23444));
        b.addRek(new Rekening("168507083528", "Kamal", "1-115-339-7678",
"dictum@nec.edu", 567770));

// // Tampil Semua Data Rekening
b.showAllRek();

// Urutkan saldo rekening dari yang terbesar
b.sortRekByDesc();

// Tampilkan Data Rekening yang Saldo nya kurang dari 50.000
b.showRekUnder50();

// // Cari Rekening berdasarkan nama
b.findRekByName("Malcolm");
b.findRekByName("Kamal");
// // Nama Joni Tidak Ada
b.findRekByName("Joni");

// // Tarik Uang Rekening Jmlah Tertentu
b.addSaldo("161007278862", 250000);
// // Rekening Salah
b.addSaldo("111111111", 250000);

// Setor Uang Rekening Jmlah Tertentu
b.tarikSaldo("161807297229", 100000);
// Saldo Tidak Cukup
b.tarikSaldo("161807297229", 4000000);
// Rekening Salah
b.tarikSaldo("111111111", 250000);

}

}

```

