**UTS**

**Nama: Rizqi Bagus Andrean**

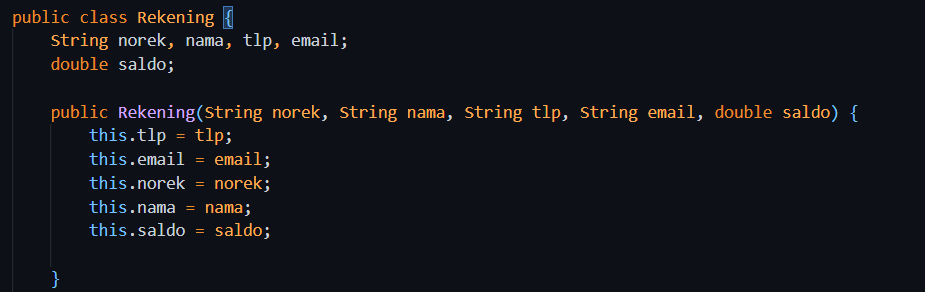
**Kelas: TI-1D**

**Absen: 25**

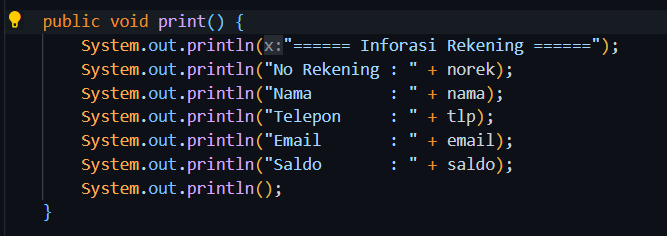
1. **Diagram Class Rekening**

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

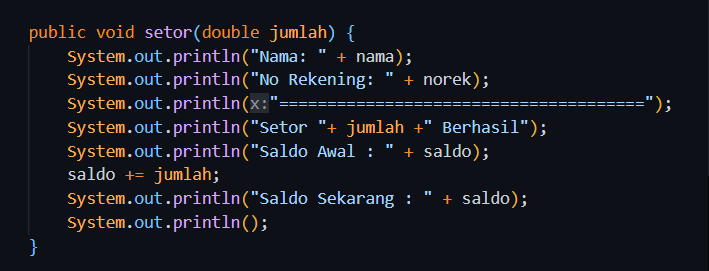
**Penjelasan Kode Program**

****

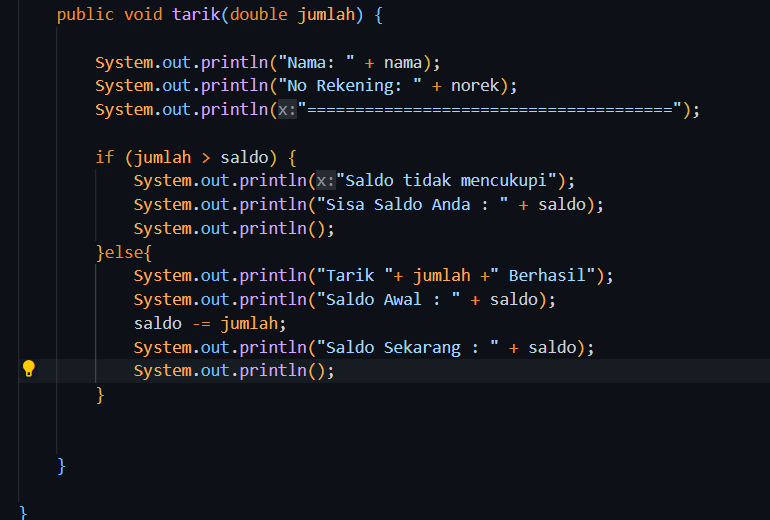
Pada class Rekening terdapat atrribut untuk menyimpan norek, nama, tlp, email, dan saldo. Terdapat juga konstruktor untuk memasukkan data awal saat instansiasi.



Fungsi print digunakan untuk menampilkan data pemilik rekening beserta saldonya.



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



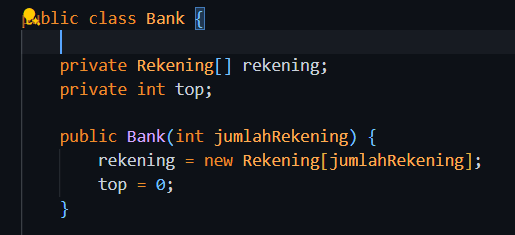
Fungsi Tarik digunakan untuk menarik saldo, saat fungsi ini dijalankan akan menampilkan nama dan norek nasabah lalu melakukan pengecekan apakah jumlah yang ditarik lebih besar dari saldo. Apabila iya maka penarikan gagal dan akan menampilkan “Saldo tidak mencukupi”.

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

1. **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**

****

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

Attribute top digunakan untuk menampung index arry 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 memangil 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 nomer rekening. Fungsi ini melakukan looping sesuai jumlah maksimal rekening yang disimpan, dan setiap loopnya akan mencocokkan apakah norek yang diinputkan ada yang sama dengan rekeing 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 pirnt dari instance rekening.

    public void addRek(Rekening rek) {

        rekening[top] = rek;

        top++;

    }

Fungsi addRek digunakan untuk menambah rekening dengan parameter rek yang bertipedata Rekening. Didalam fungsi ini dia akan memasukkan instance dari rekening ke array rekening dengan index yang kosong, lalu menambah atrribut 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 memanggit 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 memanggit 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");

        }

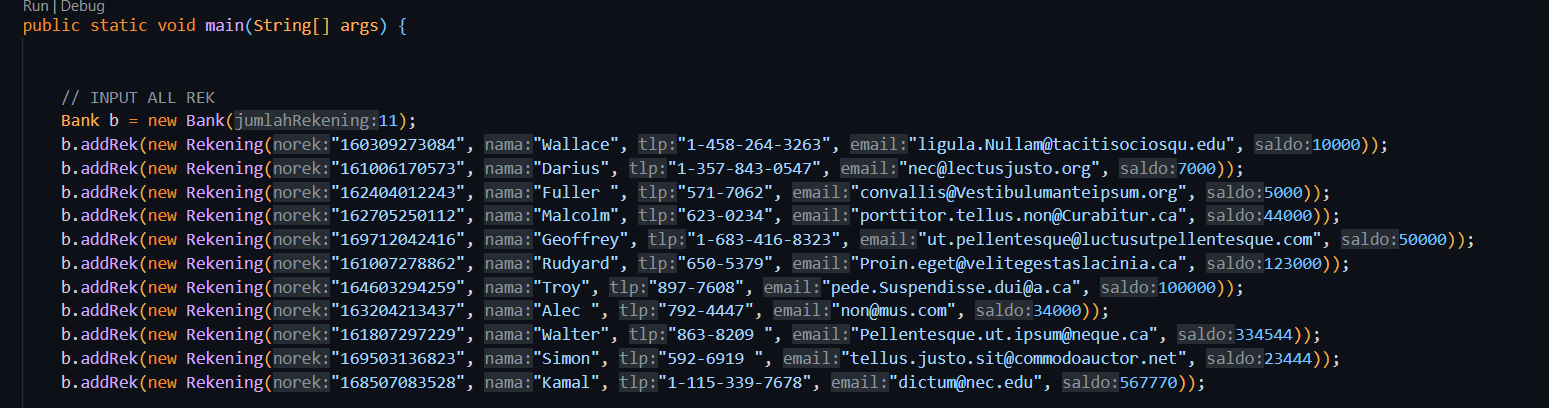
    }

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 memanggit 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**



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**

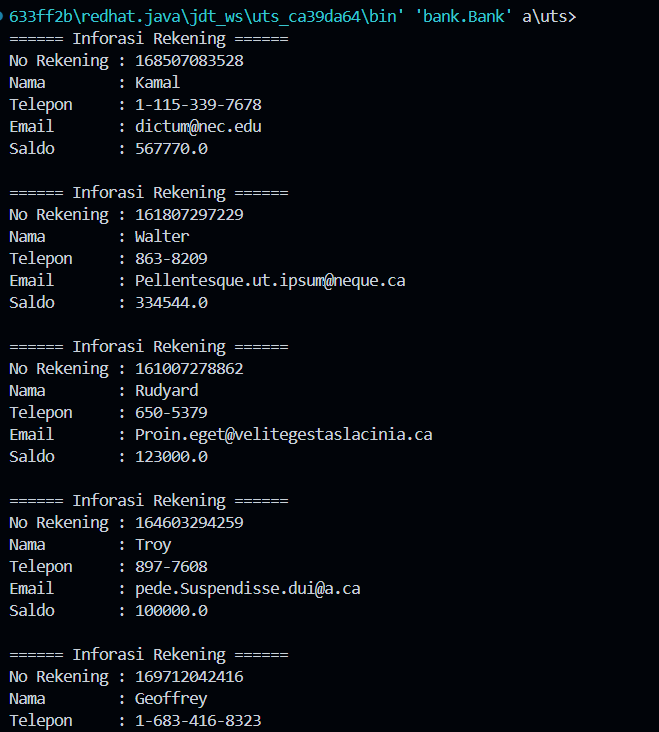
        b.showAllRek();

****

**Mengurutkan rekening dari saldo terbesar**

        // Urutkan saldo rekening dari yang terbesar

        b.sortRekByDesc();

****

**Menampilkan rekening yang memiliki saldo dibawah 50k**

 // Tampilkan Data Rekening yang Saldo nya kurang dari 50.000

        b.showRekUnder50();

****

**Mencari rekening berdasarkan nama**

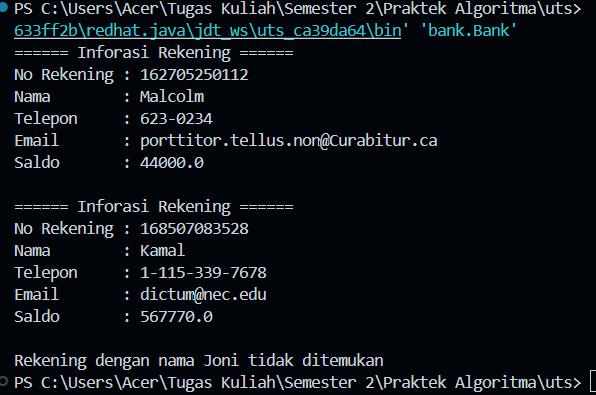
        // Cari Rekening berdasarkan nama

        b.findRekByName("Malcolm");

        b.findRekByName("Kamal");

        // Nama Joni Tidak Ada

        b.findRekByName("Joni");

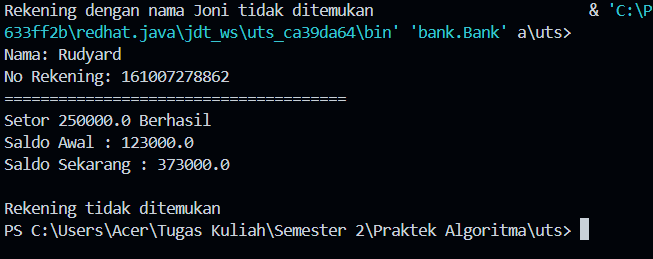
****

**Tarik Uang dengan Jumlah tertentu**

        b.addSaldo("161007278862", 250000);

        // Rekening Salah

        b.addSaldo("111111111", 250000);

****

**Setor uang jumlah tertentu**

        // Setor Uang Rekening Jmlah Tertentu

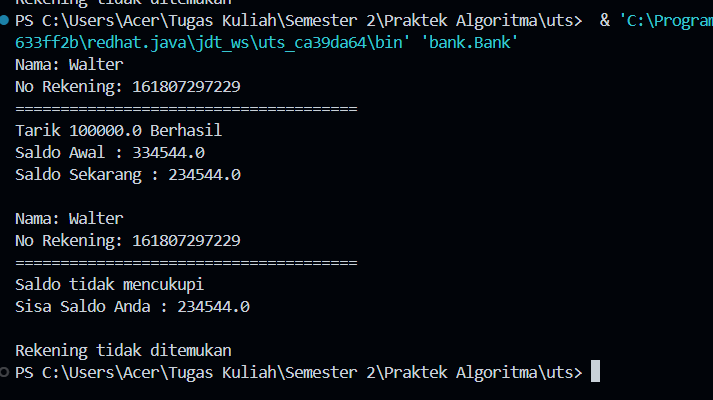
        b.tarikSaldo("161807297229", 100000);

        // Saldo Tidak Cukup

        b.tarikSaldo("161807297229", 4000000);

        // Rekening Salah

        b.tarikSaldo("111111111", 250000);

****