

Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

TUGAS PRA PERTEMUAN 6 PEMROGRAMAN BERORIENTASI OBJEK

Tugas Anda adalah melengkapi kekurangan pada sourcecode tersebut agar aplikasi simulasi ini bisa berjalan dengan benar dan mengupload sourcecode yang telah dilengkapi ke google classroom. Sertakan juga hasil screenshot simulasi pada menu 3 – 7 yang telah dilengkapi. Sebagai simulasi buatlah minimal 2 objek yang masing-masing memiliki 2 akun bank. Terkait pengisian kode bank dibebaskan selama memiliki Panjang 3 digit. Dan Pastikan untuk no.rekening WAJIB 9 digit (sudah termasuk kode bank)!!. Untuk nama nasabah, umur, nama bank, dan jumlah uang saat pembukaan akun bank juga dibebaskan tetapi tolong saat menampilkan jumlah saldo formatnya dibatasi sampai 2 angka dibelakang koma! Misalnya Rp. 30000,00, karena saat ini keluaran pada aplikasi masih seperti ini (menu 7) :

Penyelesaian

- BankAccount.java

```
import java.util.ArrayList; // import the ArrayList class
import java.util.Random;
class Person {
    // variables
    private String nama;
    private int umur;
    // array list akun bank (aggregation)
    private ArrayList<BankAccount> akunBank = new ArrayList<>();

    //constructor
    public Person (String nama, int umur) {
        this.nama = nama;
        this.umur = umur;
        System.out.println("Objek Person " + nama + " Tercipta!!!");
    }

    public String getNama() {
        return nama;
    }

    public int getUmur() {
        return umur;
    }
}
```

```

// membuka akun bank baru untuk objek person
public void bukaAkunBank (String namaBank,String kodeBank,double setoranAwal)
{
    akunBank.add(new BankAccount(namaBank, kodeBank, setoranAwal,
this.nama));
    System.out.println(this.nama+ " ( Berhasil!!, Membuka Akun Bank " +
namaBank + " Saldo awal : " + setoranAwal);
}

/**
 * mengembalikan akunBank dari objek person berdasarkan parameter noRekening.
 * @param noRekening (String)
 * @return Akun Bank (BankAccount)
 */
public BankAccount getAkunBank(String noRekening){
    int idxAkun = 0;
    for (BankAccount akun:this.akunBank) {
        if (akun.getNoRek().equals(noRekening)) {
            return this.akunBank.get(idxAkun);
        }
        idxAkun++;
    }
    return null;
}

/**
 * Menampilkan daftar akun bank berupa no rekening dan nama bank dan saldo
dari objek person.
 * ===== Daftar Rekening Bank [NAMA Objek Person] =====
 * 1. [BRI] [12312313] [Rp. xxxxx]
 * 2. [BCA] [*****] [Rp. xxxxx]
 * 3. ...
 */
public void tampilListRekeningBank() {
    System.out.println("===== Daftar Rekening Bank " + this.nama+ "
=====");
    // TO DO
    //=====

=====

    // TAMBAHKAN KODE DISINI !!!
    int index = 1;
    for(BankAccount akun: akunBank){
        System.out.println(index + ". [" + akun.getBank() + "] [" +
akun.getNoRek() + "] [Rp. "+String.format("%.2f",akun.getSaldo()) +"]");
        index++;
    }
}

```

```

    }
    //=====
===
    System.out.println("===== ");
}

/**
 * menghitung dan mengembalikan semua total saldo dari semua akunbank yg
dimiliki objek person
 * @return rekapSaldo
 */
public double rekapSaldoRekening() {
    double rekapSaldo = 0;
    // TO DO
    //=====
=====
    // TAMBAHKAN KODE DISINI
    //=====
=====
    for(BankAccount akun : akunBank){
        rekapSaldo += akun.getSaldo(); //akan menghitung jumlah saldo
keseluruhan
    }
    System.out.println("Total Saldo Semua Rekening "+this.nama+ " : Rp." +
String.format("%.2f",rekapSaldo));
    return rekapSaldo;
}
}

/**
 * Class Bank
 * @author nanoy
 */
class Bank {
    public String namaBank;
    public String kodeBank;

    /**
     * mengenerate dan mengembalikan nomor rekening bank .
     * panjang no.rekening HARUS 9 digit : (kode bank 3 digit + 6 digit).
     *
     * @return nomorRekening dengan panjang 9 digit (String)
     */
    public String generateNoRek() {
        // TODO
    }
}

```

```

        //=====
        // TAMBAHKAN KODE DISINI !!!
        //=====
        Random random = new Random();
        int randomNumber = random.nextInt(900000) + 100000; // generate angka
random 6-digit
        String noRek = String.format("%03d", Integer.parseInt(kodeBank)) +
String.format("%06d", randomNumber); // concatenate bank code dan random number
        return noRek; // ubah returnnya
    }
}

/**
 * interface transaksi keuangan
 */
interface ITransaksi {
    public void tarikSaldo(double jumlahUang);
    public void setor(double jumlahUang);
    public void transfer(double jumlahUang, Person p, String noRek);
}

/**
 * BankAccount Class
 */
public class BankAccount implements ITransaksi {
    // composition
    private Bank bank;
    private String noRek;
    private double saldo;
    // constructor
    public BankAccount(String namaBank, String kodeBank, double
depositAwal,String namaNasabah) {
        this.bank = new Bank();
        this.bank.kodeBank = kodeBank;
        this.bank.namaBank = namaBank;
        this.noRek = this.bank.generateNoRek();
        this.saldo = depositAwal;
    }

    /**
     * method tarikSaldo, jumlah uang pada akun bank bersangkutan akan berkurang
     * @param jumlahUang
     */
    @Override

```

```

public void tarikSaldo(double jumlahUang) {
    // TODO
    //=====
    // TAMBAHKAN KODE DISINI !!!
    if (saldo >= jumlahUang) {
        saldo -= jumlahUang; //saat saldo ditarik, maka saldo akan berkurang
sebanyak jumlahUang yang ditarik
        System.out.println("Penarikan berhasil. Sisa saldo : Rp." +
String.format("%.2f",saldo));
    }else{
        System.out.println("Saldo tidak mecukupi untuk melakukan penarikan
sebesar Rp." + String.format("%.2f",jumlahUang));
    }
    //=====
}

/**
 * method setor, jumlah uang pada akun bank bersangkutan akan bertambah
 * @param jumlahUang
 */
@Override
public void setor(double jumlahUang) {
    // TO DO
    //=====
    // TAMBAHKAN KODE DISINI !!
    //=====
    saldo += jumlahUang; //saat menyetorkan sejumlah uang ke rekening, maka
saldo akan bertambah sebanyak jumlahUang yang disetorkan
    System.out.println("Berhasil setor ke rekening sejumlah : Rp."
+ String.format("%.2f",jumlahUang));
}

/**
 * Method transfer, saldo akunbank pengirim akan berkurang dan saldo akunbank
penerima akan bertambah
 * sesuai jumlah uang yang ditransfer pada parameter.
 * @param jumlahUang
 * @param penerima
 * @param noRek
 */
@Override
public void transfer(double jumlahUang, Person penerima, String noRek) {
    // TO DO
    // =====
    // TAMBAHKAN KODE DISINI !!!

```

```

// =====
if (saldo >= jumlahUang) {
    saldo -= jumlahUang;
    penerima.getAkunBank(noRek).setor(jumlahUang);
    System.out.println("Transfer berhasil ke "+noRek+" a.n.
"+penerima.getNama()+
        " sebesar Rp. "+ String.format("%.2f",jumlahUang));
} else {
    System.out.println("Saldo tidak mencukupi untuk melakukan transfer
sebesar Rp." +String.format("%.2f",jumlahUang));
}

}

public String getBank() {
    return bank.namaBank;
}

public String getNoRek() {
    return noRek;
}

public double getSaldo() {
    return saldo;
}
}

```

- BankAccountTest.java

```

import java.util.Scanner; // untuk menampung input dari user
import java.util.ArrayList; // import the ArrayList class

public class BankAccountTest {

    static void tampilMenu(ArrayList<Person> p) {
        boolean quit = false;
        do {
            System.out.println("==== Pilih Menu ===");
            System.out.println("1.Buat Objek Person!");
            System.out.println("2.Buat Akun Bank pada Objek Person!");
            System.out.println("3.Lihat daftar rekening dari objek Person!");
            System.out.println("4.Setor Uang ke Rekening dari objek Person");
            System.out.println("5.Tarik Uang dari Rekening objek Person");

```

```

System.out.println("6.Transfer Uang!");
System.out.println("7.Tampilkan daftar Objek Person!");
System.out.println("=====");

Scanner in = new Scanner(System.in);
System.out.println("Pilih Menu no : ");
String menu = in.nextLine();
in.close();

switch(menu) {
    // buat objek person baru
    case "1" -> {
        try {
            String nama;
            int umur;
            do {
                System.out.println("Nama objek Person : (tidak boleh
blank) ");

                nama = in.nextLine();
                nama = nama.replaceAll("\\s", "");
            }while(nama.equals(""));
            System.out.println("Umur : ");
            umur = in.nextInt();
            p.add(new Person(nama,umur));
        }catch(Exception e) {
            System.out.println("Terjadi kesalahan input!");
        }
        break;
    }
    // buka akun bank dari objek person yang ada
    case "2" -> {
        try {
            // list objek person yang ada
            for (int i=0;i<p.size();i++) {
                System.out.println(String.format("%d. [nama:
%s]",i+1,p.get(i).getNama()));
            }
            System.out.println("Pilih [No.Urut] objek person yang
akan dibuat akun bank ? contoh: 1");
            int i = Integer.parseInt(in.nextLine());
            System.out.println("Ketik Nama Bank : (contoh: BRI / BCA
/ BSI / BNI / ...)");

            String namaBank = in.nextLine();
            System.out.println("Ketik Kode Bank 3 digit angka:
(contoh: 002 / 003 / 005 / 006 / ...)");

```

```

        String kdBank = in.nextLine();
        System.out.println("Masukkan jumlah setoran awal Rp. :
(tanpa pemisah ribuan,contoh: 100000)");
        Double setorAwal = in.nextDouble();
        p.get(i-1).bukaAkunBank(namaBank, kdBank, setorAwal);
    }catch (Exception e) {
        System.out.println("Terjadi kesalahan input atau belum
ada objek Person yang dibuat!");
    }
    break;
}
// lihat daftar rekening dari objek person yang ada
case "3" -> {
    try {
        // list objek person yang ada
        for (int i=0;i<p.size();i++) {
            System.out.println(String.format("%d. [nama:
%s]",i+1,p.get(i).getNama()));
        }
        System.out.println("Pilih [No.Urut] objek person yg ingin
dilihat daftar rekeningnya ? contoh: 1");
        int i = Integer.parseInt(in.nextLine());
        //tampilkan list rekening bank dari objek person terpilih
        p.get(i-1).tampilListRekeningBank();
    }catch (Exception e) {
        System.out.println("Terjadi kesalahan input atau belum
ada objek Person yang dibuat!");
    }
    break;
}
// setor uang ke salah satu rekening dari objek person
case "4" -> {
    try {
        // list objek person yang ada
        for (int i=0;i<p.size();i++) {
            System.out.println(String.format("%d. [nama:
%s]",i+1,p.get(i).getNama()));
        }
        System.out.println("Pilih [No.Urut] objek person yg ingin
setor ? contoh: 1 / 2");
        int i = Integer.parseInt(in.nextLine());
        // tampilkan daftar rekening dari objek terpilih
        p.get(i-1).tampilListRekeningBank();
        System.out.println("Pilih Nomor Rekening objek person yg
ingin disetor (9 digit)? contoh: 002393873");
    }
}

```



```

        String noRek = in.nextLine();
        System.out.println("Masukkan jumlah uang (Rp.) yang ingin
disetor ? contoh: 10000");
        Double jumlahUang = in.nextDouble();
        // Setorkan ke akunbank dari objek person terkait
        BankAccount ba = p.get(i-1).getAkunBank(noRek);
        //setor dilakukan
        ba.setor(jumlahUang);
        //tampilkan saldo akhir dari rek. yang disetor
        System.out.println(String.format("Saldo akhir objek %s
no.rek %s Bank %s adalah %f",
                                p.get(i-1).getNama(),
                                noRek,
                                p.get(i-1).getAkunBank(noRek).getBank(),
                                p.get(i-1).getAkunBank(noRek).getSaldo()
                                ));
    }catch (Exception e) {
        System.out.println("Terjadi kesalahan Input atau No.
Rekening Tidak Terdaftar!!");
    }
    break;
}
// tarik uang dari salah satu rekening objek person
case "5" -> {
    try {
        // list objek person yang ada
        for (int i=0;i<p.size();i++) {
            System.out.println(String.format("%d. [nama:
%s]",i+1,p.get(i).getNama()));
        }
        System.out.println("Pilih [No.Urut] objek person yg ingin
melakukan penarikan uang ? contoh: 1 / 2");
        int i = Integer.parseInt(in.nextLine());
        // tampilkan daftar rekening dari objek terpilih
        p.get(i-1).tampilListRekeningBank();
        System.out.println("Pilih Nomor Rekening objek person yg
ingin ditarik (9 digit)? contoh: 002393873");
        String noRek = in.nextLine();
        System.out.println("Masukkan jumlah uang (Rp.) yang ingin
ditarik (Pastikan saldo tersedia)? contoh: 10000");
        Double jumlahUang = in.nextDouble();
        // Setorkan ke akunbank dari objek person terkait
        BankAccount ba = p.get(i-1).getAkunBank(noRek);
        ba.tarikSaldo(jumlahUang);

```

```

        System.out.println(String.format("Saldo akhir objek %s
no.rek %s Bank %s adalah %f",
                                p.get(i-1).getNama(),
                                noRek,
                                p.get(i-1).getAkunBank(noRek).getBank(),
                                p.get(i-1).getAkunBank(noRek).getSaldo()
                                ));
    }catch (Exception e) {
        System.out.println("Terjadi kesalahan Input atau No.
Rekening Tidak Teraftar!!");
    }
    break;
}
case "6" -> {
    try {
        // list objek person yang ada
        for (int i=0;i<p.size();i++) {
            System.out.println(String.format("%d. [nama:
%s]",i+1,p.get(i).getNama()));
        }
        System.out.println("Pilih [No.Urut] objek person yg ingin
melakukan transfer/kirim uang ? contoh: 1 / 2");
        int i = Integer.parseInt(in.nextLine());
        // tampilkan daftar rekening dari objek terpilih
        p.get(i-1).tampilListRekeningBank();
        System.out.println("Pilih Nomor Rekening Pengirim (9
digit)? contoh: 002393873");
        String noRek = in.nextLine();
        System.out.println("Masukkan jumlah uang (Rp.) yang ingin
dikirim (Pastikan saldo tersedia)? contoh: 10000");
        Double jumlahUang = Double.parseDouble(in.nextLine());
        // Dapatkan akun bank yang digunakan untuk
mengirim/transfer
        BankAccount ba = p.get(i-
1).getAkunBank(noRek);
        // Pilih penerima Transfer dari list objek person yang
ada
        for (int x=0;x<p.size();x++) {
            System.out.println(String.format("%d. [nama:
%s]",x+1,p.get(x).getNama()));
        }
        System.out.println("Pilih [No.Urut] objek
person penerima transfer uang ? contoh: 1 / 2");
        int x = Integer.parseInt(in.nextLine());

```

```

        // tampilkan daftar rekening Penerima transfer (objek
Person)
        p.get(x-1).tampilListRekeningBank();
        System.out.println("Pilih Nomor Rekening Penerima (9
digit)? contoh: 002393873");
        String noRekPenerima = in.nextLine();
        // transfer
        ba.transfer(jumlahUang, p.get(x-1), noRekPenerima);
        // informasi saldo pengirim dan penerima
        System.out.println(String.format(" (Pengirim) Saldo akhir
objek %s no.rek %s Bank %s adalah %f",
                p.get(i-1).getNama(),
                noRek,
                p.get(i-1).getAkunBank(noRek).getBank(),
                p.get(i-1).getAkunBank(noRek).getSaldo()
                ));
        System.out.println(String.format(" (Penerima) Saldo akhir
objek %s no.rek %s Bank %s adalah %f",
                p.get(x-1).getNama(),
                noRekPenerima,
                p.get(x-
1).getAkunBank(noRekPenerima).getBank(),
                p.get(x-
1).getAkunBank(noRekPenerima).getSaldo()
                ));
        }catch(Exception e) {
            System.out.println("Terjadi kesalahan Input atau No.
Rekening Tidak Teraftar!!");
        }
        break;
    }
    // tampilkan daftar objek Person yang ada
    case "7" -> {
        for (int i=0;i<p.size();i++) {
            System.out.println(String.format("%d. [nama: %s] [umur:
%d] [total saldo semua bank : Rp. %.2f]",
                    i+1,
                    p.get(i).getNama(),
                    p.get(i).getUmur(),
                    p.get(i).rekapSaldoRekening()
                    ));
        }
        break;
    }
    // program dihentikan/selesai

```

```

        default -> {
            System.out.println("Inputan Tidak Terdaftar, Program
dihentikan!");
            quit = true;
        }
    }
}
while(!quit);
}
/**
 * Entry Point
 * @param args
 */
public static void main(String[] args) {
    ArrayList<Person> p = new ArrayList<Person>();
    tampilMenu(p);
}
}

```

- Simulasi

- Membuat 2 objek Person

```

==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
1
Nama objek Person : (tidak boleh blank)
Gazali
Umur :
21
Objek Person Gazali Tercipta!!!

==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
1
Nama objek Person : (tidak boleh blank)
La Ode
Umur :
20
Objek Person LaOde Tercipta!!!

```

- Masing-masing objek Person memiliki 2 akun bank

Misal Objek Gazali memiliki akun Bank BRI (setoran awal RP 1jt) dan Bank BCA (setoran awal Rp 5jt)

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
2
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yang akan dibuat akun bank ? contoh: 1
1
Ketik Nama Bank : (contoh: BRI / BCA / BSI / BNI /...)
BRI
Ketik Kode Bank 3 digit angka: (contoh: 002 / 003 / 005 / 006 /...)
002
Masukkan jumlah setoran awal Rp. : (tanpa pemisah ribuan,contoh: 100000)
1000000
Gazali ( Berhasil!!, Membuka Akun Bank BRI Saldo awal : 1000000.0
```

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
2
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yang akan dibuat akun bank ? contoh: 1
1
Ketik Nama Bank : (contoh: BRI / BCA / BSI / BNI /...)
BCA
Ketik Kode Bank 3 digit angka: (contoh: 002 / 003 / 005 / 006 /...)
003
Masukkan jumlah setoran awal Rp. : (tanpa pemisah ribuan,contoh: 100000)
5000000
Gazali ( Berhasil!!, Membuka Akun Bank BCA Saldo awal : 5000000.0
```

Kemudian misal objek LaOde mempunyai akun Bank BSI (setoran awal Rp 500rb) dan Bank BNI (setoran awal 800rb)

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
2
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yang akan dibuat akun bank ? contoh: 1
2
Ketik Nama Bank : (contoh: BRI / BCA / BSI / BNI /...)
BSI
Ketik Kode Bank 3 digit angka: (contoh: 002 / 003 / 005 / 006 /...)
005
Masukkan jumlah setoran awal Rp. : (tanpa pemisah ribuan,contoh: 100000)
500000
LaOde ( Berhasil!!, Membuka Akun Bank BSI Saldo awal : 500000.0
```

```

==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
2
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yang akan dibuat akun bank ? contoh: 1
2
Ketik Nama Bank : (contoh: BRI / BCA / BSI / BNI /...)
BNI
Ketik Kode Bank 3 digit angka: (contoh: 002 / 003 / 005 / 006 /...)
006
Masukkan jumlah setoran awal Rp. : (tanpa pemisah ribuan,contoh: 100000)
800000
LaOde ( Berhasil!!, Membuka Akun Bank BNI Saldo awal : 800000.0

```

- Melakukan pengecekan daftar rekening objek Person (Menu 3)
Objek Person Gazali:

```

==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
3
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yg ingin dilihat daftar rekeningnya ? contoh: 1
1
===== Daftar Rekening Bank Gazali =====
1. [BRI] [002556462] [Rp. 1000000.00]
2. [BCA] [003538892] [Rp. 5000000.00]
=====

```

Objek Person LaOde

```

==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
3
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yg ingin dilihat daftar rekeningnya ? contoh: 1
2
===== Daftar Rekening Bank LaOde =====
1. [BSI] [005953044] [Rp. 500000.00]
2. [BNI] [006674117] [Rp. 800000.00]
=====

```

- Setor Uang (Menu 4)

Misal melakukan setor uang sebesar 2jt ke rekening BSI objek LaOde

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
4
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yg ingin setor ? contoh: 1 / 2
2
===== Daftar Rekening Bank LaOde =====
1. [BSI] [005953044] [Rp. 500000.00]
2. [BNI] [006674117] [Rp. 800000.00]
=====
Pilih Nomor Rekening objek person yg ingin disetor (9 digit)? contoh: 002393873
005953044
Masukkan jumlah uang (Rp.) yang ingin disetor ? contoh: 10000
2000000
Berhasil setor ke rekening sejumlah : Rp.2000000.00
Saldo akhir objek LaOde no.rek 005953044 Bank BSI adalah 2500000.00
```

- Tarik Uang (Menu 5)

Misal ingin menarik uang sebesar 200rb dari rekening BCA objek person Gazali

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
5
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yg ingin melakukan penarikan uang ? contoh: 1 / 2
1
===== Daftar Rekening Bank Gazali =====
1. [BRI] [002556462] [Rp. 1000000.00]
2. [BCA] [003538892] [Rp. 5000000.00]
=====
Pilih Nomor Rekening objek person yg ingin ditarik (9 digit)? contoh: 002393873
003538892
Masukkan jumlah uang (Rp.) yang ingin ditarik (Pastikan saldo tersedia)? contoh: 10000
200000
Penarikan berhasil. Sisa saldo : Rp.4800000.00
Saldo akhir objek Gazali no.rek 003538892 Bank BCA adalah 4800000.00
```

- Transfer Uang (Menu 6)

Misal ingin melakukan transfer uang sebesar 150rb dari rekening BRI objek person Gazali ke rekening BNI objek person LaOde

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
6
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person yg ingin melakukan transfer/kirim uang ? contoh: 1 / 2
1
===== Daftar Rekening Bank Gazali =====
1. [BRI] [002556462] [Rp. 100000.00]
2. [BCA] [003538892] [Rp. 480000.00]
=====
Pilih Nomor Rekening Pengirim (9 digit)? contoh: 002393873
002556462
Masukkan jumlah uang (Rp.) yang ingin dikirim (Pastikan saldo tersedia)? contoh: 10000
150000
1. [nama: Gazali]
2. [nama: LaOde]
Pilih [No.Urut] objek person penerima transfer uang ? contoh: 1 / 2
2
===== Daftar Rekening Bank LaOde =====
1. [BSI] [005953044] [Rp. 250000.00]
2. [BNI] [006674117] [Rp. 800000.00]
=====
Pilih Nomor Rekening Penerima (9 digit)? contoh: 002393873
006674117
Berhasil setor ke rekening sejumlah : Rp.150000.00
Transfer berhasil ke 006674117 a.n. LaOde sebesar Rp. 150000.00
(Pengirim) Saldo akhir objek Gazali no.rek 002556462 Bank BRI adalah 850000.00
(Penerima) Saldo akhir objek LaOde no.rek 006674117 Bank BNI adalah 950000.00
```

- Menampilkan daftar objek person (Menu 7)

```
==== Pilih Menu ====
1.Buat Objek Person!
2.Buat Akun Bank pada Objek Person!
3.Lihat daftar rekening dari objek Person!
4.Setor Uang ke Rekening dari objek Person
5.Tarik Uang dari Rekening objek Person
6.Transfer Uang!
7.Tampilkan daftar Objek Person!
=====
Pilih Menu no :
7
Total Saldo Semua Rekening Gazali : Rp.5650000.00
1. [nama: Gazali] [umur: 21] [total saldo semua bank : Rp. 5650000.00]
Total Saldo Semua Rekening LaOde : Rp.3450000.00
2. [nama: LaOde] [umur: 20] [total saldo semua bank : Rp. 3450000.00]
```