



# Laporan Asistensi

## Tugas Pertama

Ubah Project Studi Kasus dengan menerapkan Konsep MVC.

## Source Code

Class Bank.java

```
package Controller;

import Entity.Transaksi;
import Entity.Transfer;
import Entity.User;
import Entity.Rekening;
import Model.Authentication;

import java.util.ArrayList;
import java.util.Scanner;

public class Bank {

    private static ArrayList<User> userTerdaftar = new
    ArrayList<User>();

    public static ArrayList<User> getUserTerdaftar() {
        return userTerdaftar;
    }

    public static boolean tambahUser(User akunBaru, Rekening
    rekening){
        if(userTerdaftar.size() != 0){
            for (User pengguna: userTerdaftar){
                if
                ((pengguna.getNik().equals(akunBaru.getNik())) ||
                (pengguna.getNoTelpon().equals(akunBaru.getNoTelpon()))){
                    return false;
                }
            }
        }
        akunBaru.buatRekening(rekening);
        userTerdaftar.add(akunBaru);
        return true;
    }
}
```



# Laporan Asistensi

```
        public static void setorTunai(User pengguna, Transaksi
transaksi){
            pengguna.getRekening().tambahTransaksi(transaksi);

pengguna.getRekening().tambahSaldo(transaksi.getNilaiNominal()
);
        }

        public static boolean tarikTunai(User pengguna, Transaksi
transaksi){
            boolean status =
pengguna.getRekening().ambilSaldo(transaksi.getNilaiNominal())
;
            if (status) {
                pengguna.getRekening().tambahTransaksi(transaksi);
            }
            return status;
        }

        public static boolean transfer(Transaksi transaksi){
            Transfer transfer = (Transfer) transaksi;
            User userAsal = transfer.getUserAsal();
            User userTujuan = transfer.getUserTujuan();

            for (User akunTujuan : userTerdaftar) {
                if (akunTujuan.getIdUser() ==
userTujuan.getIdUser()) {

userTujuan.getRekening().tambahTransaksi(transfer);

userTujuan.getRekening().tambahSaldo(transfer.getNilaiNominal(
));

userAsal.getRekening().tambahTransaksi(transfer);

userAsal.getRekening().ambilSaldo(transfer.getNilaiNominal());
                return true;
            }
        }
        return false;
    }
}
```



# Laporan Asistensi

```
public static boolean verifPin() {
    Scanner input = new Scanner(System.in);
    int percobaan = 3;

    do {
        percobaan -= 1;
        System.out.print("Masukan PIN: ");
        String pin = input.nextLine();

        boolean status = Authentication.verifPin(pin);

        if (!status) {
            if (percobaan == 0) {
                System.out.println("Anda Salah memasukan
PIN sebanyak 3x");
                return false;
            } else {
                System.out.println("PIN Salah !");
            }
        } else {
            return true;
        }
    } while (percobaan != 0);
    return false;
}
```

Class Authentication.java

```
package Model;

import Entity.SetorTunai;
import Entity.User;
import Utils.RandomNumberString;
import Entity.Rekening;
import Controller.Bank;
import View.HomePage;
import View.MainPage;

public class Authentication {

    static User userLogged = null;

    public static void initialUser(){
```



# Laporan Asistensi

```
User pengguna = new User("Test 1", "123456", "0811",
"user1", "123");
Bank.tambahUser(pengguna, new Rekening("123456",
"123"));
Bank.setorTunai(pengguna, new SetorTunai(1000000));
User pengguna2 = new User("Test 2", "654321", "08122",
"user2", "321");
Bank.tambahUser(pengguna2, new Rekening("654321",
"321"));
}

public static User getUserLogged() {
    return userLogged;
}

public static void register(String namaLengkap, String
nik, String noTelpon, String username, String kodeAkses, String
pin){
    User userBaru = new User(namaLengkap, nik, noTelpon,
username, kodeAkses);

    String noRekening = RandomNumberString.getNumeric(6);
    Rekening rekening = new Rekening(noRekening, pin);
    Bank.tambahUser(userBaru, rekening);
}

public static boolean login(String username, String
kodeAkses){
    if (Bank.getUserTerdaftar().size() != 0) {
        for (User pengguna: Bank.getUserTerdaftar()){
            if ((pengguna.getUsername().equals(username))
&& (pengguna.getKodeAkses().equals(kodeAkses))) {
                userLogged = pengguna;
                return true;
            }
        }
    }
    return false;
}

public static boolean verifPin(String pin){
    return userLogged.getRekening().getPin().equals(pin);
}
```



# Laporan Asistensi

```
public static User cariRekening(String rekening){
    for (User pengguna : Bank.getUserTerdaftar()) {
        if
        (pengguna.getRekening().getNoRekening().equals(rekening)) {
            return pengguna;
        }
    }
    return null;
}

public static void logOut(char pilihan){
    if (pilihan == 'y' || pilihan == 'Y'){
        new HomePage();
    }
    else if (pilihan == 'n' || pilihan == 'N'){
        new MainPage();
    }
}

public static void exit(char pilihan){
    if (pilihan == 'Y' || pilihan == 'y'){
        System.out.println("-----Keluar Aplikasi-----");
        userLogged = null;
    }
    else if (pilihan == 'N' || pilihan == 'n'){
        new HomePage();
    }
}
}
```