**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 10 – Studi Kasus ATM*”*

****

Oleh:

Nama : Avryzel Alifian Hakim

NPM : 4523210121

Kelas : A

Dosen:

Adi Wahyu Pribadi , S.Si., M.Kom

**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2023/2024**

1. **Soure Code**

* Account.java

package *com.atm.model*;

import *java.util.Scanner*;

*public* *class* Account {

*private* String accountNumber;

*private* String pin;

*private* double balance;

*public* Account(String accountNumber, String pin, double balance) {

        this.accountNumber = accountNumber;

        this.pin = pin;

        this.balance = balance;

    }

*// Getter dan Setter*

*public* String getAccountNumber() {

        return accountNumber;

    }

*public* String getPin() {

        return pin;

    }

*public* double getBalance() {

        return balance;

    }

*public* void credit(double amount) {

        this.balance += amount;

    }

*public* void debit(double amount) {

        this.balance -= amount;

    }

*//\* Method untuk mengubah PIN*

*public* void changePin() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan PIN lama: ");

        String oldPin = scanner.nextLine();

        if (this.pin.equals(oldPin)) {

            System.out.print("Masukkan PIN baru: ");

            String newPin = scanner.nextLine();

            System.out.print("Konfirmasi PIN baru: ");

            String confirmPin = scanner.nextLine();

            if (newPin.equals(confirmPin)) {

                this.pin = newPin;

                System.out.println("PIN berhasil diubah.");

            } else {

                System.out.println("PIN baru tidak cocok. Gagal mengubah PIN.");

            }

        } else {

            System.out.println("PIN lama salah. Gagal mengubah PIN.");

        }

    }

}

* Transaction.java

package *com.atm.transaction*;

import *com.atm.model.Account*;;

*public* *abstract* *class* Transaction {

*protected* Account account;

*public* Transaction(Account account) {

        this.account = account;

    }

*public* *abstract* void execute();

}

* Withdrawal.java

package *com.atm.transaction*;

import *com.atm.model.Account*;

import *java.util.Scanner*;

*public* *class* Withdrawal *extends* Transaction {

*private* *static* *final* double MINIMUM\_BALANCE = 50000;

*public* Withdrawal(Account account) {

        super(account);

    }

    @Override

*public* void execute() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Masukkan jumlah penarikan: ");

        double amount = scanner.nextDouble();

        if (amount <= account.getBalance() - MINIMUM\_BALANCE) {

            account.debit(amount);

            System.out.println("Penarikan berhasil. Saldo Anda sekarang: " + account.getBalance());

        } else {

            System.out.println("Saldo tidak mencukupi untuk penarikan dengan saldo minimal Rp50,000.");

        }

    }

}

* Deposit.java

package *com.atm.transaction*;

import *com.atm.model.Account*;

import *java.util.Scanner*;

*public* *class* Deposit *extends* Transaction {

*public* Deposit(Account account) {

        super(account);

    }

    @Override

*public* void execute() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Masukkan jumlah deposit: ");

        double amount = scanner.nextDouble();

        account.credit(amount);

        System.out.println("Deposit berhasil. Saldo Anda sekarang: " + account.getBalance());

    }

}

* Transfer.java

package *com.atm.transaction*;

import *com.atm.model.Account*;

import *java.util.Scanner*;

*public* *class* Transfer *extends* Transaction {

*private* Account targetAccount;

*public* Transfer(Account account, Account targetAccount) {

        super(account);

        this.targetAccount = targetAccount;

    }

    @Override

*public* void execute() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan jumlah transfer: ");

        double amount = scanner.nextDouble();

        if (amount <= account.getBalance()) {

            account.debit(amount);

            targetAccount.credit(amount);

            System.out.println("Transfer berhasil. Saldo Anda sekarang: " + account.getBalance());

        } else {

            System.out.println("Saldo tidak mencukupi.");

        }

    }

}

* ATM.java

package *com.atm*;

import *com.atm.model.Account*;

import *com.atm.transaction.\**;

import *java.util.HashMap*;

import *java.util.Map*;

import *java.util.Scanner*;

*public* *class* ATM {

*private* Map<String, Account> accounts;

*public* ATM() {

        accounts = new HashMap<>();

        initializeAccounts();

    }

*private* void initializeAccounts() {

*// Menambahkan akun contoh*

        accounts.put("123456", new Account("123456", "1234", 500000));

        accounts.put("654321", new Account("654321", "4321", 1000000));

    }

*public* void start() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan nomor akun: ");

        String accountNumber = scanner.nextLine();

        System.out.print("Masukkan PIN: ");

        String pin = scanner.nextLine();

        Account account = authenticateUser(accountNumber, pin);

        if (account != null) {

            showMenu(account);

        } else {

            System.out.println("Autentikasi gagal.");

        }

    }

*private* Account authenticateUser(String accountNumber, String pin) {

        Account account = accounts.get(accountNumber);

        if (account != null && account.getPin().equals(pin)) {

            return account;

        }

        return null;

    }

*private* void showMenu(Account account) {

        Scanner scanner = new Scanner(System.in);

        int choice = 0;

        do {

            System.out.println("\nMenu:");

            System.out.println("1. Penarikan");

            System.out.println("2. Deposit");

            System.out.println("3. Transfer");

            System.out.println("4. Cek Saldo");

            System.out.println("5. Ubah PIN");

            System.out.println("6. Keluar");

            System.out.print("Pilih opsi: ");

            choice = scanner.nextInt();

            switch (choice) {

                case 1:

                    Transaction withdrawal = new Withdrawal(account);

                    withdrawal.execute();

                    break;

                case 2:

                    Transaction deposit = new Deposit(account);

                    deposit.execute();

                    break;

                case 3:

                    scanner.nextLine(); *// Membersihkan buffer*

                    System.out.print("Masukkan nomor akun tujuan: ");

                    String targetAccountNumber = scanner.nextLine();

                    Account targetAccount = accounts.get(targetAccountNumber);

                    if (targetAccount != null) {

                        Transaction transfer = new Transfer(account, targetAccount);

                        transfer.execute();

                    } else {

                        System.out.println("Akun tujuan tidak ditemukan.");

                    }

                    break;

                case 4:

                    System.out.println("Saldo Anda: " + account.getBalance());

                    break;

                case 5:

                    account.changePin();

                    break;

                case 6:

                    System.out.println("Terima kasih telah menggunakan ATM kami.");

                    break;

                default:

                    System.out.println("Opsi tidak valid.");

            }

        } while (choice != 6);

    }

}

* Main.java

package *com.atm*;

*public* *class* Main {

*public* *static* void main(String[] args) {

        ATM atm = new ATM();

        atm.start();

    }

}

1. **Fitur Ubah Pin**

* Hasil Running

**A computer screen with white text

Description automatically generated**

1. **Validasi Saldo Minimal pada saat penarikan**

* Hasil Running

**A screen shot of a computer

Description automatically generated**