**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 10 – ATM”

****

Oleh:

Nama : Daffa Yassar Ahmad

NPM : 4523210032

Kelas : A

Dosen:

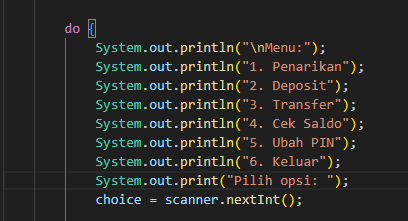
Adi Wahyu Pribadi, S.Si., M.Kom

**S1-Teknik Informatika**

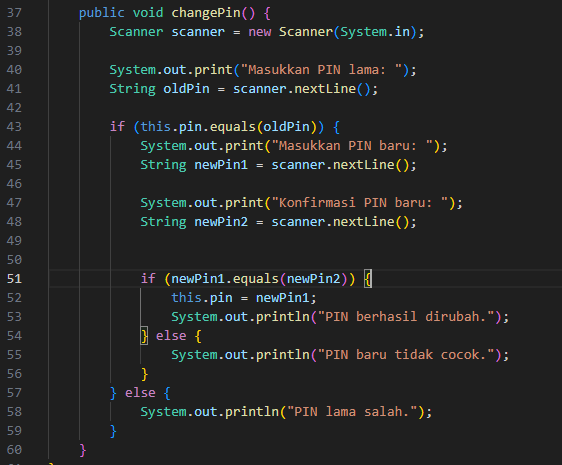
**Fakultas Teknik Universitas Pancasila**

**2023/2024**

1. Tambahkan fitur Ubah PIN
   1. Tambahkan opsi “Ubah PIN” pada menu utama



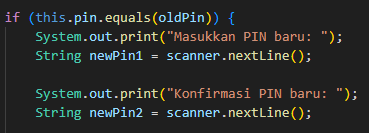
* 1. Buat mdeo dalam kelas Account untuk mengubah PIN, yaitu: changePin()



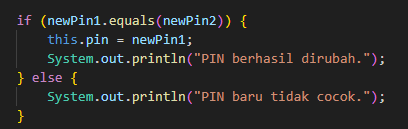
* **Scanner scanner = new Scanner(System.in)** : Berguna untuk membuat objek Scanner untuk mengambil input dari pengguna melalui konsol.
  1. Dalam method tersebut lakukan hal berikut:
     1. Verifikasi PIN lama



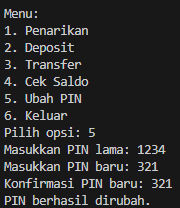
* **System.out.print**: Menampilkan pesan kepada pengguna untuk memasukkan PIN lama.
* **String oldPin = scanner.nextLine();**: Membaca input dari pengguna dan menyimpannya ke variabel oldPin
  + 1. Minta nasabah memasukkan PIN baru dua kali



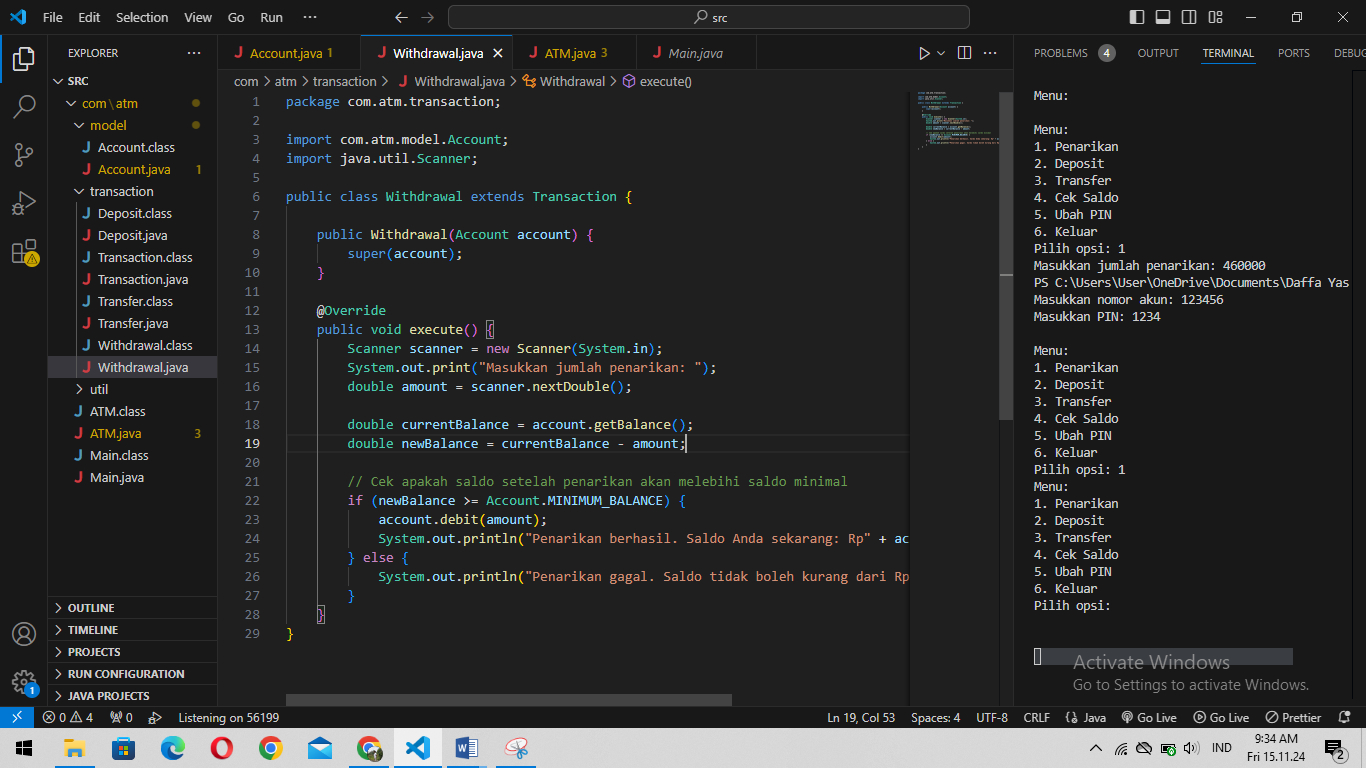
* **System.out.print("Masukkan PIN baru: ");**: Meminta pengguna untuk memasukkan PIN baru.
* **String newPin1 = scanner.nextLine();**: Menyimpan PIN baru yang dimasukkan ke variabel newPin1.
* **System.out.print("Konfirmasi PIN baru: ");**: Meminta pengguna untuk mengonfirmasi PIN baru.
* **String newPin2 = scanner.nextLine();**: Menyimpan konfirmasi PIN baru yang dimasukkan ke variabel newPin2.
* **if (this.pin.equals(oldPin))**: Mengecek apakah PIN yang dimasukkan pengguna (oldPin) cocok dengan PIN yang tersimpan dalam objek (this.pin). Jika cocok, maka lanjut ke langkah berikutnya.
  + 1. Validasi bahwa kedua input PIN baru cocok



* **if (newPin1.equals(newPin2))**: Mengecek apakah kedua input PIN baru cocok. Jika cocok, maka PIN berhasil diganti, dan this.pin diubah menjadi newPin1.
* **System.out.println("PIN berhasil dirubah.");**: Memberi tahu pengguna bahwa PIN telah berhasil diubah.
* **else**: Jika kedua PIN baru tidak cocok, program akan mencetak pesan bahwa PIN baru tidak cocok.
  + 1. Perbarui PIN jika validasi berhasil



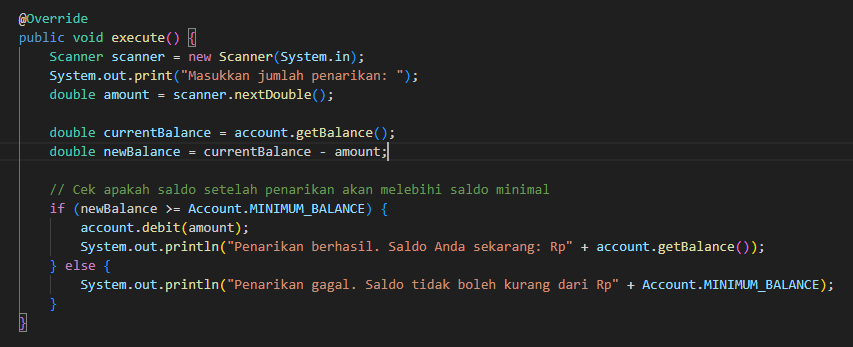
1. Validasi Saldo Minimal pada saat penarikan
   1. Modifikasi fitur penarikan sehingga nasabah harus menyisakan saldo minimal setelah penarikan dilakukan. Misal, saldo minial adalah Rp50,000-



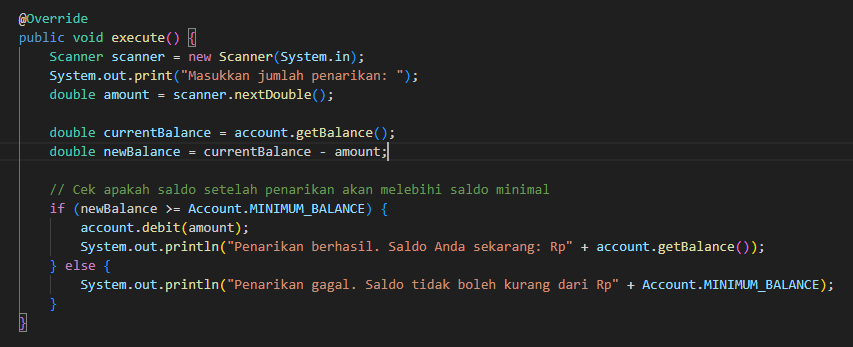
* 1. Langkah-langkah:
     1. Tentukan saldo minimal, tambahkan konstanta MINIMUM\_BALANCE dalam kelas Account



* **MINIMUM\_BALANCE**: Konstanta untuk menentukan saldo minimal (misalnya Rp 50.000).
  + 1. Modifikasi methode execute() dalam kelas Withdrawal untuk memeriksa apakah saldo setelah penarikan tidak kuran dari saldo minimal



* if (newBalance >= Account.MINIMUM\_BALANCE) memeriksa apakah saldo setelah penarikan (newBalance) lebih besar atau sama dengan saldo minimal yang ditetapkan dalam Account.MINIMUM\_BALANCE.
* Jika saldo setelah penarikan mencukupi (tidak kurang dari saldo minimal), method debit(amount) dipanggil untuk mengurangi saldo di akun sebesar jumlah yang ingin ditarik.
* Pesan sukses ditampilkan kepada pengguna yang mencantumkan saldo terkini melalui account.getBalance().
  + 1. Jika saldo tidak mencukupi, tampilkan pesan kesalahan



* Jika saldo baru kurang dari saldo minimal, penarikan tidak dilakukan, dan pesan kesalahan ditampilkan, memberitahukan bahwa saldo tidak boleh kurang dari batas minimal (MINIMUM\_BALANCE).