

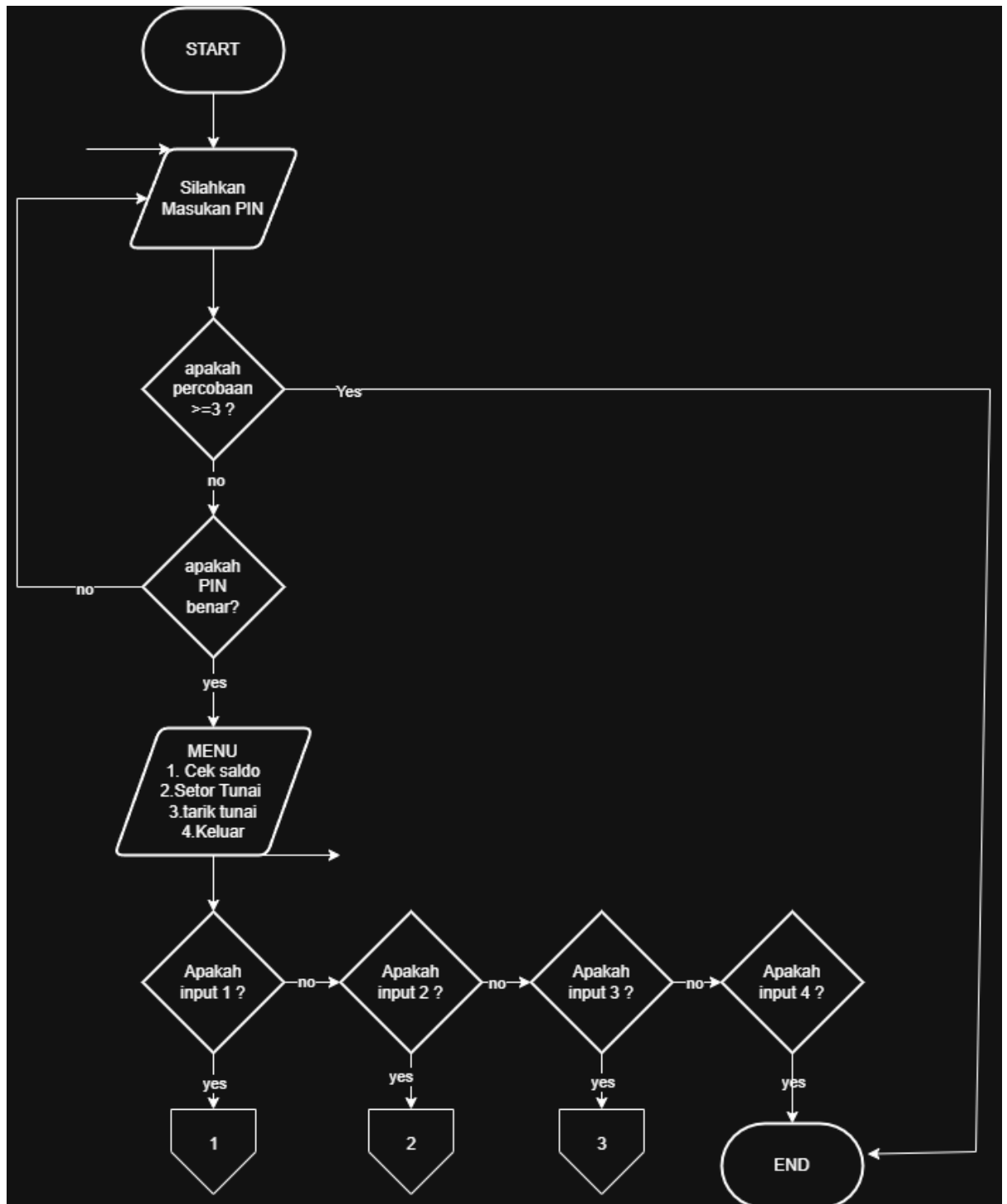
**LAPORAN PRAKTIKUM**  
**POSTTEST 1**  
**ALGORITMA PEMROGRAMAN LANJUT**



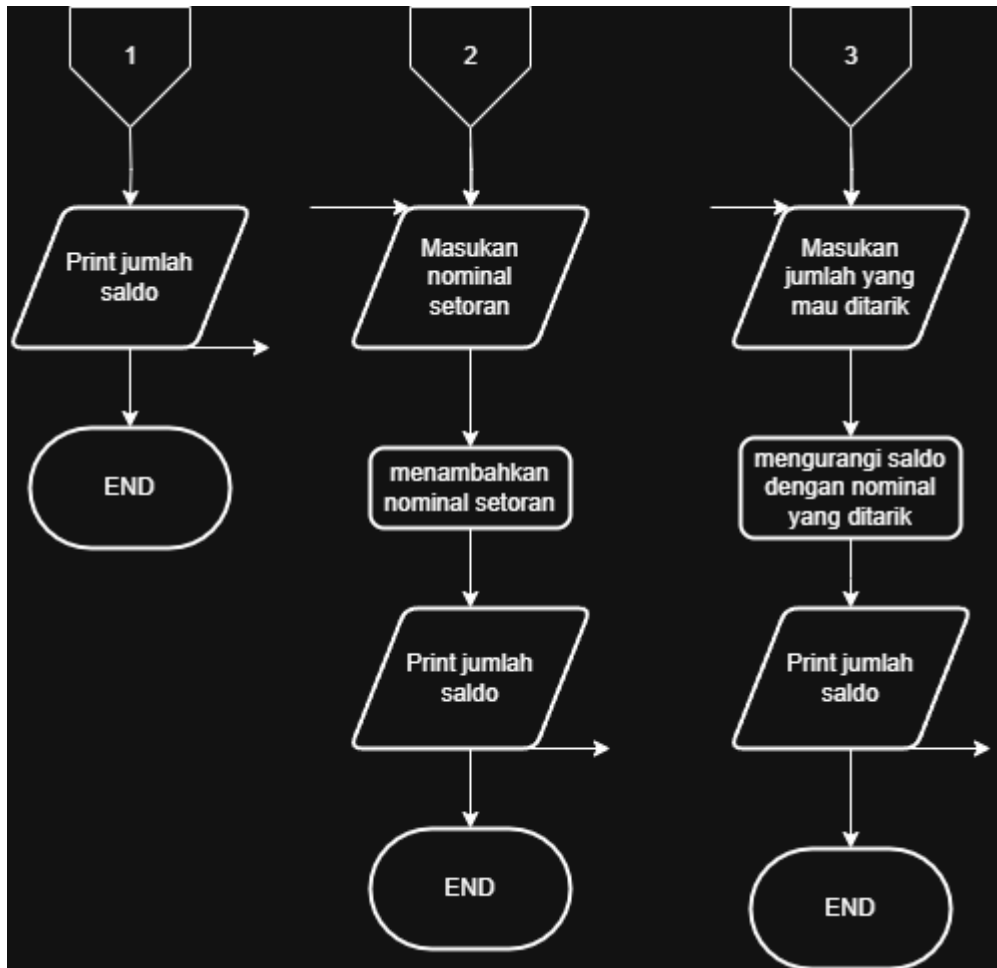
**Disusun oleh:**  
**Muhammad Dzaki Rifa'I (2409106056)**  
**Kelas (B1'24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart



Gambar 1.1 Flowchart program utama



Gambar 1.2 Flowchart menu

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini dibuat untuk mensimulasikan program mesin ATM dengan sistem login menggunakan PIN sebanyak 4 digit dan maksimal 3 kali percobaan, setelah berhasil login maka akan muncul 4 menu yaitu lihat saldo, tarik tunai, setor tunai dan keluar.

### 2.2 Penjelasan Alur & Algoritma

#### 1. Deklarasi dan Inisialisasi Variabel:

- PIN, pinInput: Untuk menyimpan PIN yang benar dan masukan pengguna.

#### Alur Kerja Program

#### 1. Inisialisasi:

- Program memulai dengan mendeklarasikan variabel-variabel utama yang digunakan, seperti PIN, pinInput, saldo, dan lainnya.
- Variabel akses digunakan untuk menentukan apakah pengguna berhasil login atau tidak.

#### 2. Verifikasi PIN:

- Pengguna diberikan 3 kesempatan untuk memasukkan PIN melalui perulangan for.
- Jika PIN yang dimasukkan cocok dengan PIN yang sudah ditentukan (6056), pengguna diberikan akses ke menu utama.
- Jika pengguna gagal memasukkan PIN dengan benar dalam 3 percobaan, program akan menampilkan pesan "Akses ditolak" dan berhenti.

#### 3. Menu Utama:

- Setelah login berhasil, program masuk ke menu utama menggunakan perulangan while.
- Menu ini menyediakan 4 opsi:
  - **1. Cek saldo:** Menampilkan saldo pengguna.
  - **2. Setor tunai:** Meminta nominal uang untuk disetor dan menambah saldo pengguna.
  - **3. Tarik tunai:** Meminta nominal uang untuk ditarik dan mengurangi saldo jika cukup.
  - **4. Keluar:** Mengakhiri program.

#### 4. Validasi Input:

- Program memastikan setiap input valid, seperti memastikan nominal uang yang dimasukkan lebih besar dari 0 dan memastikan saldo mencukupi untuk penarikan.

5. **Proses Transaksi:**

- Berdasarkan pilihan menu, program memproses data (misalnya, menambah atau mengurangi saldo) dan menampilkan output kepada pengguna.
- Program akan terus menampilkan menu hingga pengguna memilih opsi "Keluar".

6. **Keluar dari Program:**

- Ketika pengguna memilih keluar, program menampilkan pesan perpisahan dan berhenti.

**Penjelasan Detail Blok Kode:**

- **Loop for:** Digunakan untuk membatasi percobaan login hingga 3 kali.
- **if-else di Login:** Memeriksa apakah PIN sesuai dan memutuskan apakah pengguna bisa lanjut atau tidak.
- **Loop while:** Menjaga interaksi pengguna di menu hingga mereka memilih keluar.
- **Kondisi if-else di Menu:**
  - Memastikan pemrosesan berdasarkan input pengguna untuk setiap opsi di menu.
  - Memberikan feedback interaktif dan validasi input transaksi.

### 3. Source Code

#### A. Verifikasi pin

Fitur ini digunakan untuk memverifikasi input PIN yang dimasukan oleh user, jika user salah memasukan PIN sampai 3 kali maka program akan berhenti.

```
//Verifikasi PIN
for (int percobaan = 1; percobaan <= maxCoba; percobaan++) {
    cout << "Masukkan PIN Anda: ";
    cin >> pinInput;

    if (pinInput == PIN) {
        cout << "Login berhasil! Selamat datang!" << endl;
        cin.ignore();
        cin.get();
        system("cls");
        akses = true;
        break;
    } else {
        cout << "PIN salah. Percobaan (" << percobaan << "/" << maxCoba << ")"
<< endl;
        system("cls");
    }
}

if (!akses) {
    cout << "Anda telah gagal login sebanyak " << maxCoba << " kali. Akses
ditolak." << endl;
    cin.ignore();
    cin.get();
    return 0;
}
```

## B. Menu utama

Fitur ini terdiri atas 4 pilihan yaitu cek saldo, setor tunai, tarik tunai dan keluar

```
// Menu utama
int pilihan = 0;
while (akses) {
    system("cls");
    cout << "===== " << endl;
    cout << "        Selamat datang        " << endl;
    cout << "===== " << endl;
    cout << "1 -> Cek saldo" << endl;
    cout << "2 -> Setor tunai" << endl;
    cout << "3 -> Tarik tunai" << endl;
    cout << "4 -> Keluar" << endl;
    cout << "Masukkan pilihan: ";
    cin >> pilihan;

    if (cin.fail()) {
        cin.clear();
        cin.ignore(1000, '\n');
        cout << "Masukkan angka yang valid!" << endl;
        cin.ignore();
        continue;
    }

    if (pilihan == 1) {
        cout << "Saldo Anda adalah Rp " << saldo << "." << endl;
        cin.ignore();
        cin.get();
    } else if (pilihan == 2) {
        cout << "Masukkan jumlah uang yang ingin disetor: Rp ";
        cin >> saldoin;
```

```

        if (saldoin > 0) {
            saldo += saldoin;
            cout << "Setor tunai berhasil. Saldo Anda sekarang adalah Rp " <<
saldo << "." << endl;
        } else {
            cout << "Nominal setor tunai harus lebih besar dari 0." << endl;
        }
        cin.ignore();
        cin.get();
    } else if (pilihan == 3) {
        cout << "Masukkan nominal yang ingin Anda tarik: Rp ";
        cin >> saldout;

        if (saldout > saldo) {
            cout << "Saldo Anda tidak cukup untuk melakukan penarikan." <<
endl;
        } else if (saldout > 0) {
            saldo -= saldout;
            cout << "Tarik tunai berhasil. Saldo Anda sekarang adalah Rp " <<
saldo << "." << endl;
        } else {
            cout << "Nominal penarikan harus lebih besar dari 0." << endl;
        }
        cin.ignore();
        cin.get();
    } else if (pilihan == 4) {
        cout << "Keluar. Terima kasih telah menggunakan layanan kami." <<
endl;

        akses = false;
    } else {
        cout << "Pilihan tidak valid! Harap masukkan angka antara 1-4." <<
endl;

        cin.ignore();
        cin.get();
    }
}

```



```
}  
}
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

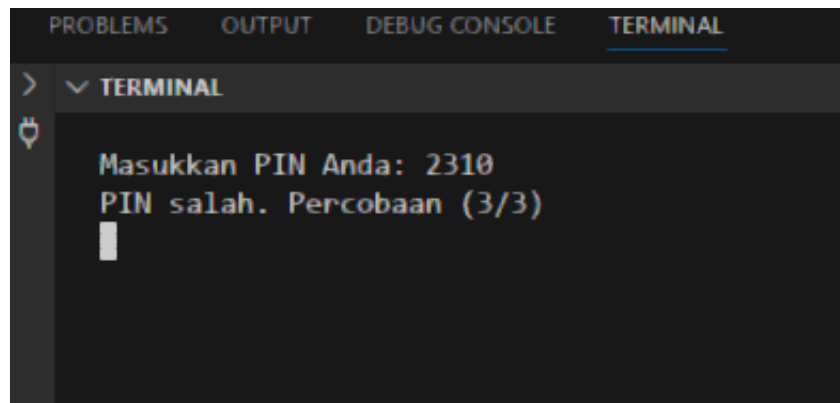
#### 1. Skenario 1

Uji coba verifikasi input PIN, jika user memasukan PIN selain 6056 maka akan disalahkan dan percobaan akan berkurang sampai 3 kali.

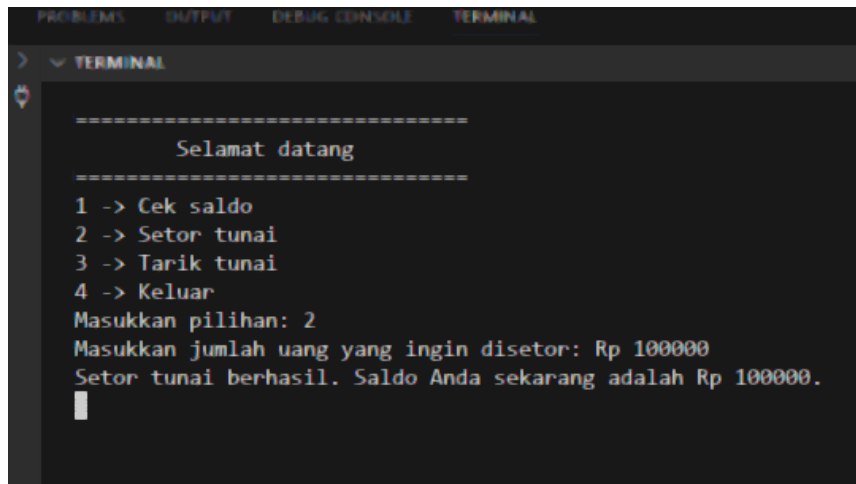
#### 2. Skenario 2

Uji coba penambahan dan melihat saldo apakah saldo sudah bertambah dengan benar.

### 4.2 Hasil Output



Gambar 4.1 Hasil Skenario 1



Gambar 4.2 Hasil Skenario 2

## 5. Git

### 5.1 Git Init

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL (main)
$ git init
Initialized empty Git repository in C:/Users/ASUS/Praktikum-APL/.git/
```

Git init berfungsi untuk membuat repository git pada folder tertentu

### 5.2 Git Add

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git add Posttest1.cpp
```

Git add berfungsi untuk Memasukkan file ke staging area agar siap untuk commit.

### 5.3 Git Commit

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git commit -m "login done"
[main b75faf2] login done
1 file changed, 29 insertions(+)
create mode 100644 Post-test/Post-test-APL-1/Posttest1.cpp
```

Git commit berfungsi menyimpan perubahan di repository lokal.

Pesan di -m "..." menjelaskan perubahan yang dilakukan.

## 5.4 Git remote add origin

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL (main)
$ git remote add origin https://github.com/dzak2323/Praktikum-APL.git
```

Menghubungkan repository lokal ke repository GitHub.

## 5.5 Git push origin main

```
ASUS@Dzakzak MINGW64 ~/Praktikum-APL (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 20 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (23/23), 1.01 MiB | 448.00 KiB/s, done.
Total 23 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/dzak2323/Praktikum-APL.git
 * [new branch]      main -> main
```

Git push origin main Mengunggah commit yang ada di branch main ke repository GitHub.

Jika branch belum ada di GitHub, jalankan `git push -u origin main` agar branch main menjadi default untuk push berikutnya.