

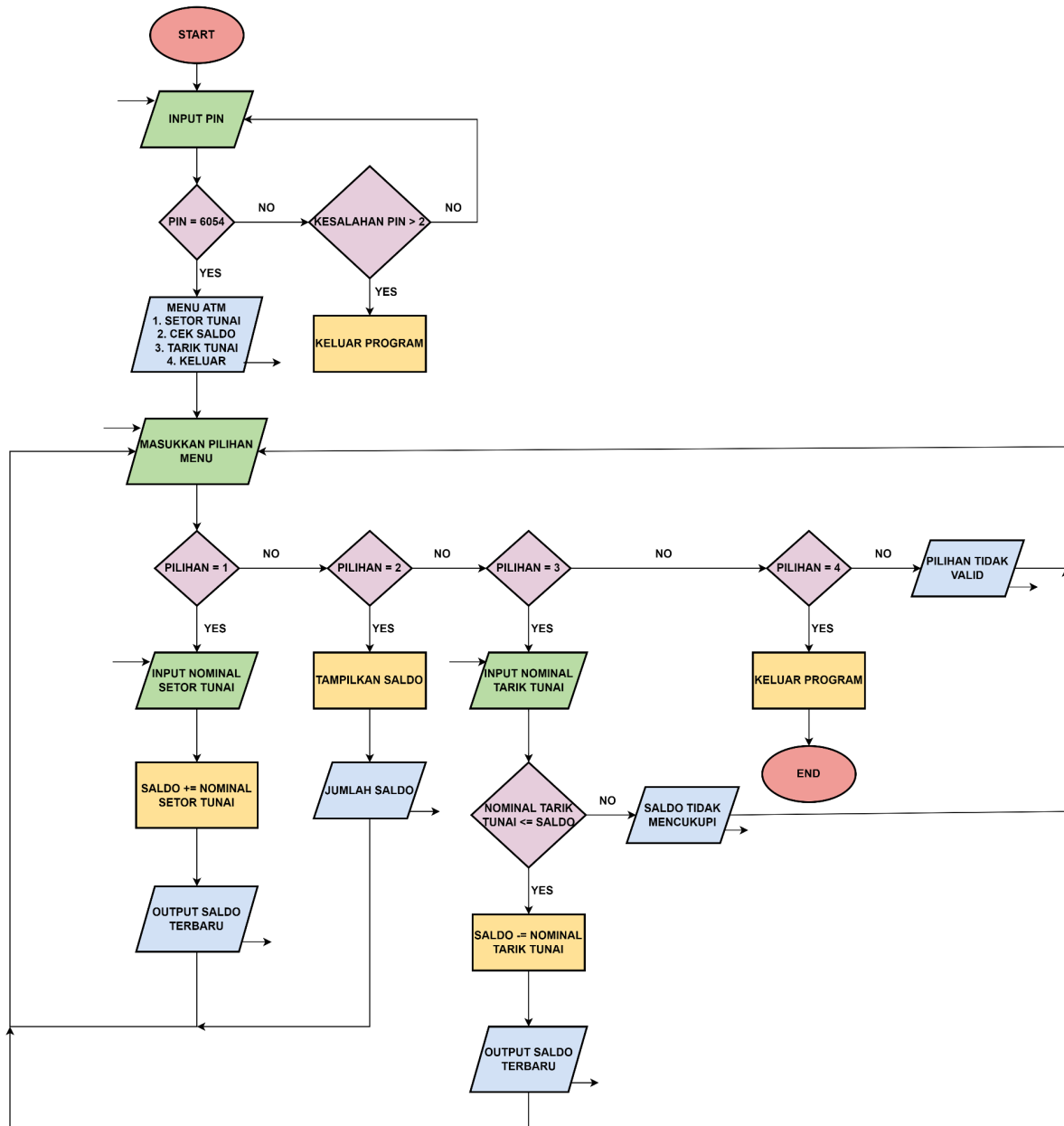
**LAPORAN PRAKTIKUM**  
**POSTTEST 1**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Alya Mayasha - 2409106054**  
**Kelas B1 2024**

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

## 1. Flowchart



Gambar 1.1 Flowchart Program ATM Sederhana

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini merupakan simulasi mesin ATM sederhana dengan bahasa pemrograman C++. Pada program ini pengguna diberikan tiga kesempatan untuk memasukkan PIN dengan benar, PIN adalah empat digit NIM terakhir yaitu 6054. Jika berhasil, pengguna dapat mengakses menu ATM yang menyediakan tiga opsi: Setor Tunai, Cek Saldo, dan Tarik Tunai. Saldo awal pengguna adalah Rp. 2.000.000,- dan setiap transaksi akan mempengaruhi saldo tersebut. Program akan terus berjalan hingga pengguna memilih opsi keluar dari menu. Jika pengguna gagal memasukkan PIN sebanyak tiga kali, akses akan diblokir dan otomatis keluar dari program.

### 2.2 Penjelasan Alur & Algoritma

#### 1. Inisialisasi Variabel

Di awal program, beberapa variabel dideklarasikan:

- PIN\_BENAR : Menyimpan PIN yang benar (6054).
- BATAS\_GAGAL : Menentukan batas maksimal percobaan memasukkan PIN yaitu tiga kali.
- saldo : Menyimpan saldo awal pengguna yaitu Rp. 2.000.000,-.
- aksesditerima : Menandakan apakah pengguna berhasil memasukkan PIN dengan benar atau tidak.

#### 2. Memasukkan PIN dan Validasi

- Pengguna diberikan 3 kesempatan untuk memasukkan PIN.
- Jika PIN benar, program menampilkan pesan sambutan dan mengatur aksesditerima menjadi true, lalu keluar dari perulangan.
- Jika PIN salah, program menampilkan jumlah percobaan yang telah dilakukan.
- Jika pengguna gagal memasukkan PIN dalam 3 kali percobaan, akses diblokir, dan program berakhir.

#### 3. Menampilkan Menu ATM

Jika pengguna berhasil memasukkan PIN, program akan masuk ke menu utama ATM yang menggunakan perulangan *while (true)* agar menu terus ditampilkan sampai pengguna memilih untuk keluar. Selanjutnya program akan menampilkan daftar opsi

transaksi dan pengguna diminta memasukkan pilihan dalam bentuk angka (1-4). Opsi tersebut berisi :

1. Setor Tunai
2. Cek Saldo
3. Tarik Tunai
4. Keluar

#### 4. Pemrosesan Pilihan Menu

##### A. Setor Tunai

- Pengguna memasukkan jumlah uang yang ingin disetorkan.
- Saldo bertambah sesuai dengan jumlah setoran.
- Program menampilkan saldo terbaru.

##### B. Cek Saldo

- Program akan menampilkan saldo pengguna.

##### C. Tarik Tunai

- Pengguna memasukkan jumlah uang yang ingin ditarik.
- Jika saldo cukup, saldo akan dikurangi dan transaksi berhasil.
- Jika saldo tidak mencukupi, program menampilkan pesan error.

##### D. Keluar

- Jika pengguna memilih opsi ini, program akan keluar dari perulangan dan menampilkan pesan perpisahan.
- *return 0;* digunakan untuk mengakhiri eksekusi program.

##### E. Pilihan Tidak Valid

Jika pengguna memasukkan angka di luar 1-4, program akan menampilkan pesan error dan kembali ke menu utama.

#### 5. Akhir Program

- Jika pengguna memilih keluar dari menu, program akan berhenti.
- Jika pengguna salah memasukkan PIN 3 kali berturut-turut, akses diblokir dan program berhenti.

### 3. Source Code

#### A. Fitur Validasi PIN

Fitur ini memastikan pengguna memasukkan PIN yang benar dalam maksimal 3 kali percobaan.

**Source Code:**

```
for (int percobaan = 1; percobaan <= BATAS_GAGAL; percobaan++) {
    cout << "Masukkan PIN Anda: ";
    cin >> pin;

    if (pin == PIN_BENAR) {
        cout << "PIN benar. Selamat datang!\n";
        aksesditerima = true;
        break;
    } else {
        cout << "PIN salah! Percobaan ke-" << percobaan << " dari " <<
        BATAS_GAGAL << ".\n";
    }
}
```

#### B. Fitur Setor Tunai

Fitur ini memungkinkan pengguna menambahkan saldo ke rekening mereka.

**Source Code:**

```
case 1: {
    int setor;
    cout << "Masukkan jumlah setor tunai: ";
    cin >> setor;
    saldo += setor;
    cout << "Setor tunai berhasil. Saldo Anda saat ini: Rp" << saldo << endl;
    break;
}
```

#### C. Fitur Cek Saldo

Fitur ini digunakan untuk menampilkan saldo terbaru pengguna.

**Source Code:**

```
case 2:
    cout << "Saldo Anda saat ini: Rp" << saldo << endl;
    break;
```

#### D. Fitur Tarik Tunai

Fitur ini memungkinkan pengguna menarik uang dengan memastikan saldo mencukupi.

#### Source Code:

```
case 3: {
    int tarik;
    cout << "Masukkan jumlah tarik tunai: ";
    cin >> tarik;
    if (tarik > saldo) {
        cout << "Mohon maaf saldo tidak mencukupi!\n";
    } else {
        saldo -= tarik;
        cout << "Tarik tunai berhasil. Saldo Anda saat ini: Rp" << saldo <<
endl;
    }
    break;
}
```

#### E. Fitur Keluar

Fitur ini mengakhiri sesi pengguna di ATM.

#### Source Code:

```
case 4:
    cout << "Terima kasih telah menggunakan ATM ini. Have a great day!\n";
    return 0;
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

1. Skenario 1  
Ketika pengguna memasukkan PIN yang salah sebanyak tiga kali.
2. Skenario 2  
Ketika pengguna memilih menu setor tunai dan memasukkan nominal 500000.
3. Skenario 3  
Ketika pengguna memilih menu tarik tunai dengan nominal yang melebihi saldo awal.
4. Skenario 4  
Ketika pengguna memilih pilihan menu selain 1-4.

### 4.2 Hasil Output

```
=== SELAMAT DATANG DI ATM ===  
Masukkan PIN Anda: 2112  
PIN salah! Percobaan ke-1 dari 3.  
Masukkan PIN Anda: 1103  
PIN salah! Percobaan ke-2 dari 3.  
Masukkan PIN Anda: 2005  
PIN salah! Percobaan ke-3 dari 3.  
Anda telah salah memasukkan PIN sebanyak 3 kali. Akses diblokir.  
PS C:\Users\HP-GK\Desktop\PRAKTIKUM APL>
```

*Gambar 4.1 Output Skenario 1*

```
=== MENU ATM ===  
1. Setor Tunai  
2. Cek Saldo  
3. Tarik Tunai  
4. Keluar  
Pilih menu (1-4): 1  
Masukkan jumlah setor tunai: 500000  
Setor tunai berhasil. Saldo Anda saat ini: Rp2500000
```

*Gambar 4.2 Output Skenario 2*

```

=== MENU ATM ===
1. Setor Tunai
2. Cek Saldo
3. Tarik Tunai
4. Keluar
Pilih menu (1-4): 3
Masukkan jumlah tarik tunai: 5000000
Mohon maaf saldo tidak mencukupi!

```

*Gambar 4.3 Output Skenario 3*

```

Pilih menu (1-4): 7
Pilihan tidak valid, silakan coba lagi.

=== MENU ATM ===
1. Setor Tunai
2. Cek Saldo
3. Tarik Tunai
4. Keluar
Pilih menu (1-4): 

```

*Gambar 4.4 Output Skenario 4*

## 5. Penjelasan GIT

### 5.1 Git Init

Git Init berfungsi untuk menginisialisasi repository Git baru di dalam folder proyek. Ini akan membuat direktori .git untuk melacak perubahan.

```

HP-GK@DESKTOP-N8AD402 MINGW64 ~/Desktop/praktikum-apl (master)
$ git init
Initialized empty Git repository in C:/Users/HP-GK/Desktop/praktikum-apl/.git/

```

*Gambar 5.1 Git Init*

### 5.2 Git Add

Git Add berfungsi untuk menambahkan file tertentu ke staging area agar siap untuk dikomit.

```

HP-GK@DESKTOP-N8AD402 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (master)
$ git add 2409106054-AlyaMayasha-PT-1.cpp

```

*Gambar 5.2 Git Add*

### 5.3 Git Commit

Git Commit berfungsi untuk menyimpan perubahan yang ada di staging area ke dalam repository disertai dengan pesan deskriptif.



```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (master)
$ git commit -m "commit pertama"
[master (root-commit) 57efd0e] commit pertama
2 files changed, 73 insertions(+)
create mode 100644 post-test/post-test-1/2409106054-AlyaMayasha-PT-1.cpp
create mode 100644 post-test/post-test-1/2409106054-AlyaMayasha-PT-1.exe
```

*Gambar 5.3 Git Commit*

## 5.4 Git Remote Add Origin

Git Remote Add Origin berfungsi untuk menghubungkan repository lokal dengan repository remote (misalnya, di GitHub atau GitLab) menggunakan URL yang diberikan.

```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/desktop/praktikum-apl (master)
$ git remote add origin https://github.com/AlyaMayasha/praktikum-apl.git
```

*Gambar 5.4 Git Remote Add Origin*

## 5.5 Git Push Origin Main

Git Push Origin Main berfungsi untuk mengirim (mengunggah) perubahan dari branch main di repository lokal ke repository remote yang sudah dikonfigurasi sebelumnya.

```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/desktop/praktikum-apl (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 671.68 KiB | 4.51 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AlyaMayasha/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

*Gambar 5.5 Git Push Origin Main*