

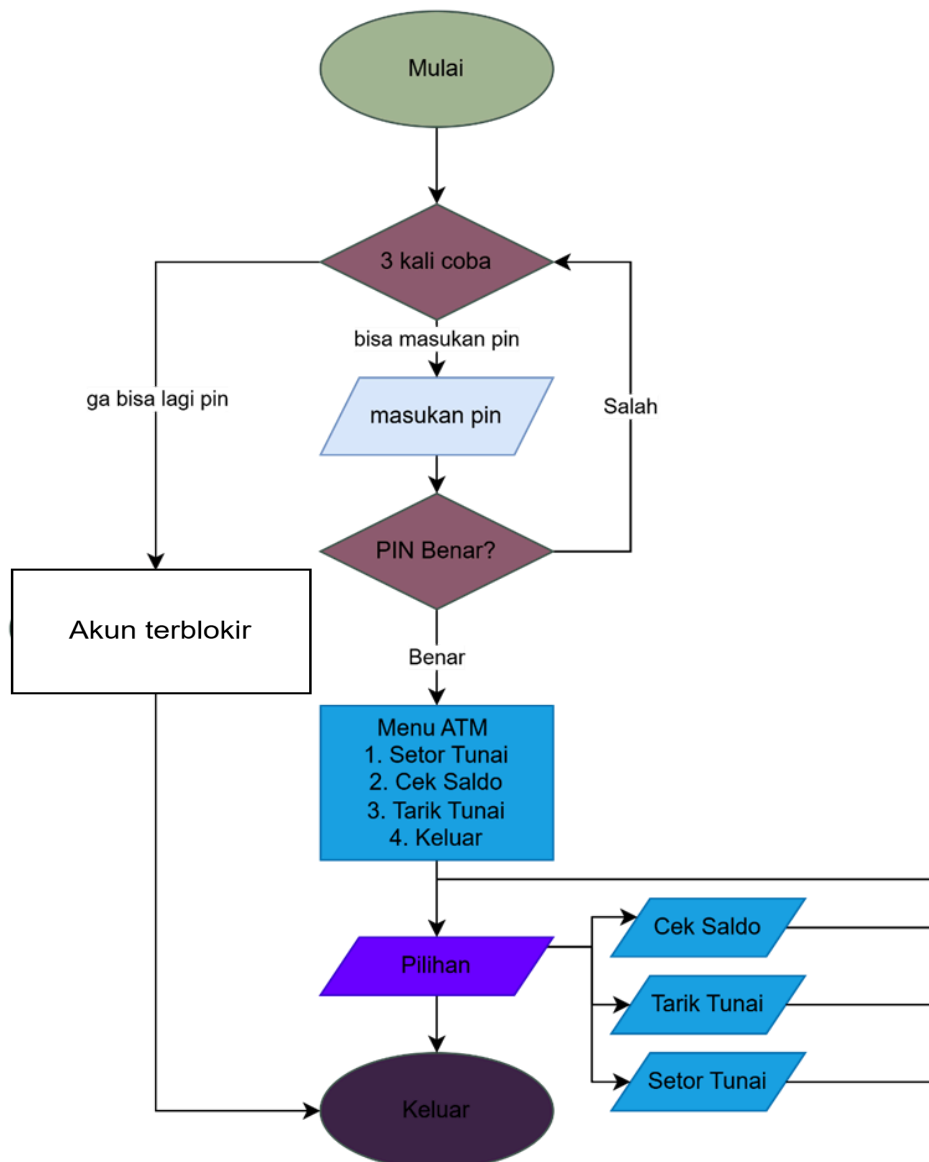
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Fachlevi Muhammad (2409106059)
B24 (B1)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

2.1 Deskripsi Singkat Program

Tujuan utama dari program ini adalah untuk mensimulasikan sistem ATM sederhana yang memungkinkan pengguna untuk:

1. Login dengan PIN 4 digit (PIN: 2025)
2. Melakukan transaksi seperti setor tunai, cek saldo, dan tarik tunai
3. Menghindari akses yang salah dengan batasan 3 kali percobaan PIN
4. Memastikan keamanan saldo dengan validasi jumlah saldo sebelum tarik tunai
5. Menggunakan sistem menu yang interaktif hingga pengguna memilih dengan mudah

2.2 Penjelasan Alur & Algoritma

Tahap 1: Login dengan PIN

1. user diminta memasukkan PIN
2. Jika PIN benar, lanjut ke menu ATM
3. Jika salah, diberikan kesempatan maksimal 3 kali
4. Jika 3 kali salah, program berhenti dengan pesan "Akun Terblokir"

```
int main() {
    int pin = 2025;
    int inputPin, attempts = 0;
    long long saldo = 1000000;
    int pilihan;

    while (attempts < 3) {
        cout << "Masukkan PIN: ";
        cin >> inputPin;

        if (inputPin == pin) {
            cout << "login berhasil!\n";
            break;
        } else {
            cout << "PIN salah Coba lagi\n";
            attempts++;
        }
    }

    if (attempts == 3) {
        cout << "Terlalu banyak percobaan. Akun diblokir\n";
        return 0;
    }
}
```

Tahap 2: Menampilkan Menu ATM

1. Setelah login berhasil, program akan masuk ke menu utama
2. Pengguna bisa memilih Setor Tunai, Cek Saldo, Tarik Tunai, atau Keluar
3. Menu terus berulang (looping) sampai pengguna memilih keluar

```
do {
    cout << "\nMENU ATM\n";
    cout << "1.Setor Tunai\n";
    cout << "2.Cek saldo\n";
    cout << "3.Tarik Tunai\n";
    cout << "4.Keluar\n";
    cout << "Pilih menu: ";
    cin >> pilihan;
}
```

Tahap 3: Fitur Fitur dalam Menu ATM

Setor Tunai

1. Pengguna memasukkan jumlah uang yang ingin disetor
2. Saldo bertambah sesuai jumlah setor

```
case 1: {
    long long setor;
    cout << "Masukkan jumlah setor tunai: ";
    cin >> setor;
    saldo += setor;
    cout << "Setor tunai berhasil! Saldo sekarang: Rp " << saldo <<
    "\n";
    break;
```

Cek Saldo

1. Menampilkan saldo saat ini

```
case 2:
    cout << "Saldo Anda saat ini: rp " << saldo << "\n";
    break;
```

Tarik Tunai

1. Pengguna memasukkan jumlah uang yang ingin ditarik
2. Jika saldo mencukupi, saldo dikurangi
3. Jika saldo kurang, muncul pesan error

```
case 3: {
    long long tarik;
    cout << "Masukkan jumlah tarik tunai: ";
    cin >> tarik;

    if (tarik > saldo) {
        cout << "Saldo tidak mencukupi\n";
    } else {
        saldo -= tarik;
        cout << "Tarik tunai berhasil Saldo sekarang: rp " << saldo
        << "\n";
    }
    break;
```

Keluar dari ATM

1. Program berhenti dengan menampilkan pesan "Terima kasih telah menggunakan ATM ini"

```
case 4:
    cout << "Terima kasih telah menggunakan ATM ini :]\n";
    break;
```

3. Source Code

Login dengan PIN (Maksimal 3 Percobaan)

```
int main() {
    int pin = 2025;
    int inputPin, attempts = 0;
    long long saldo = 1000000;
    int pilihan;

    while (attempts < 3) {
        cout << "Masukkan PIN: ";
        cin >> inputPin;

        if (inputPin == pin) {
            cout << "login berhasil!\n";
            break;
        } else {
            cout << "PIN salah Coba lagi\n";
            attempts++;
        }
    }

    if (attempts == 3) {
        cout << "Terlalu banyak percobaan. Akun diblokir\n";
        return 0;
    }
}
```

Menu ATM (Looping Hingga Pilih Keluar)

```
do {  
    cout << "\nMENU ATM\n";  
    cout << "1.Setor Tunai\n";  
    cout << "2.Cek saldo\n";  
    cout << "3.Tarik Tunai\n";  
    cout << "4.Keluar\n";  
    cout << "Pilih menu: ";  
    cin >> pilihan;
```

Setor Tunai fungsi Menambah saldo sesuai nominal yang dimasukkan pengguna

```
case 1: {  
    long long setor;  
    cout << "Masukkan jumlah setor tunai: ";  
    cin >> setor;  
    saldo += setor;  
    cout << "Setor tunai berhasil! Saldo sekarang: Rp " << saldo <<  
    "\n";  
    break;  
}
```

Mengurangi saldo jika saldo mencukupi, atau menampilkan pesan error jika saldo kurang

```
case 3: {  
    long long tarik;  
    cout << "Masukkan jumlah tarik tunai: ";  
    cin >> tarik;  
  
    if (tarik > saldo) {  
        cout << "Saldo tidak mencukupi\n";  
    } else {  
        saldo -= tarik;  
        cout << "Tarik tunai berhasil Saldo sekarang: rp " << saldo  
        << "\n";  
    }  
    break;
```

Menghentikan program jika pengguna memilih "Keluar"

```
case 4:  
    cout << "Terima kasih telah menggunakan ATM ini :]\n";  
    break;
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1: Login Berhasil dan Melakukan Transaksi

Input:

1. Memasukkan PIN yang benar (2025)
2. Memilih Setor Tunai dan memasukkan 500000
3. Memilih Cek Saldo untuk memastikan saldo bertambah
4. Memilih Tarik Tunai dengan jumlah 300000
5. Memilih Keluar

Output:

1. Program menerima PIN yang benar dan masuk ke menu
2. Saldo bertambah setelah setor tunai
3. Saldo berkurang setelah tarik tunai
4. Program keluar setelah pengguna memilih "Keluar"

Skenario 2: Salah Memasukkan PIN 3 Kali (Akun Terblokir)

Input:

1. Memasukkan PIN yang salah (misal: 1234)
2. Memasukkan PIN yang salah lagi
3. Memasukkan PIN yang salah untuk ketiga kalinya

Output:

1. Program menampilkan pesan "PIN salah Coba lagi" setelah setiap percobaan
2. Setelah percobaan ke-3, program menampilkan "Terlalu banyak percobaan. Akun diblokir"
3. Program langsung berhenti

Skenario 3: Tarik Tunai Lebih dari Saldo (Gagal)

Input:

1. Memasukkan PIN yang benar (2025)
2. Memilih Cek Saldo (Misal saldo = 1.000.000)
3. Memilih Tarik Tunai dan memasukkan jumlah 2.000.000

Output:

1. Program menampilkan pesan "Saldo tidak mencukupi"
2. Saldo tetap tidak berubah

Skenario 4: Memilih Menu yang Tidak Ada

Input:

1. Memasukkan PIN yang benar (2025)
2. Memilih menu dengan angka di luar opsi (misalnya 9)

Ekspektasi Output:

1. Program menampilkan pesan "Pilihan tidak valid"
2. Menu ATM muncul kembali untuk meminta input ulang

4.2 Hasil Output

Skenario 1

```
Masukkan PIN: 2025
login berhasil!

MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 
MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 1
Masukkan jumlah setor tunai: 500000
Setor tunai berhasil! Saldo sekarang: Rp 1500000

MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 3
Masukkan jumlah tarik tunai: 300000
Tarik tunai berhasil Saldo sekarang: rp 1200000

MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 4
Terima kasih telah menggunakan ATM ini :]
```


Skenario 2

```
Masukkan PIN: 1234
PIN salah Coba lagi
Masukkan PIN: 1234
PIN salah Coba lagi
Masukkan PIN: 1234
PIN salah Coba lagi
Terlalu banyak percobaan. Akun diblokir
```

Skenario 3

```
MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 3
Masukkan jumlah tarik tunai: 2000000
Saldo tidak mencukupi
```

Skenario 4

```
MENU ATM
1.Setor Tunai
2.Cek saldo
3.Tarik Tunai
4.Keluar
Pilih menu: 5
Pilihan tidak valid
```

5. Git

```
Microsoft Windows [Version 10.0.26100.3323]
C:\Users\ACER\Downloads\Praktikum-APL>git init
Initialized empty Git repository in C:/Users/ACER/Downloads/Praktikum-APL/.git/

C:\Users\ACER\Downloads\Praktikum-APL>git add .

C:\Users\ACER\Downloads\Praktikum-APL>git commit -m "Finish Post Test 1"
[main (root-commit) 4c95f06] Finish Post Test 1
 2 files changed, 73 insertions(+)
 create mode 100644 Post-Test/main.cpp
 create mode 100644 Post-Test/main.exe

C:\Users\ACER\Downloads\Praktikum-APL>git remote add origin https://github.com/Diego444666/Praktikum-APL.git

C:\Users\ACER\Downloads\Praktikum-APL>git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 671.23 KiB | 8.09 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Diego444666/Praktikum-APL.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\ACER\Downloads\Praktikum-APL>
```

git init : Menginisialisasi repository Git di dalam folder proyek. Perintah ini membuat folder

git add : Menambahkan semua file dalam proyek ke staging area, yang berarti file sudah siap untuk dikomit

git commit -m "Finish Post Test 1" : Menyimpan perubahan ke dalam repository Git dengan sebuah pesan deskriptif.

git remote add origin https://github.com/Diego444666/Praktikum-APL.git : Menghubungkan repository lokal dengan repository yang ada di GitHub. URL repository harus sesuai dengan yang dibuat di GitHub.

git push -u origin main : Mengunggah semua file yang telah dikomit ke repository GitHub pada branch