

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



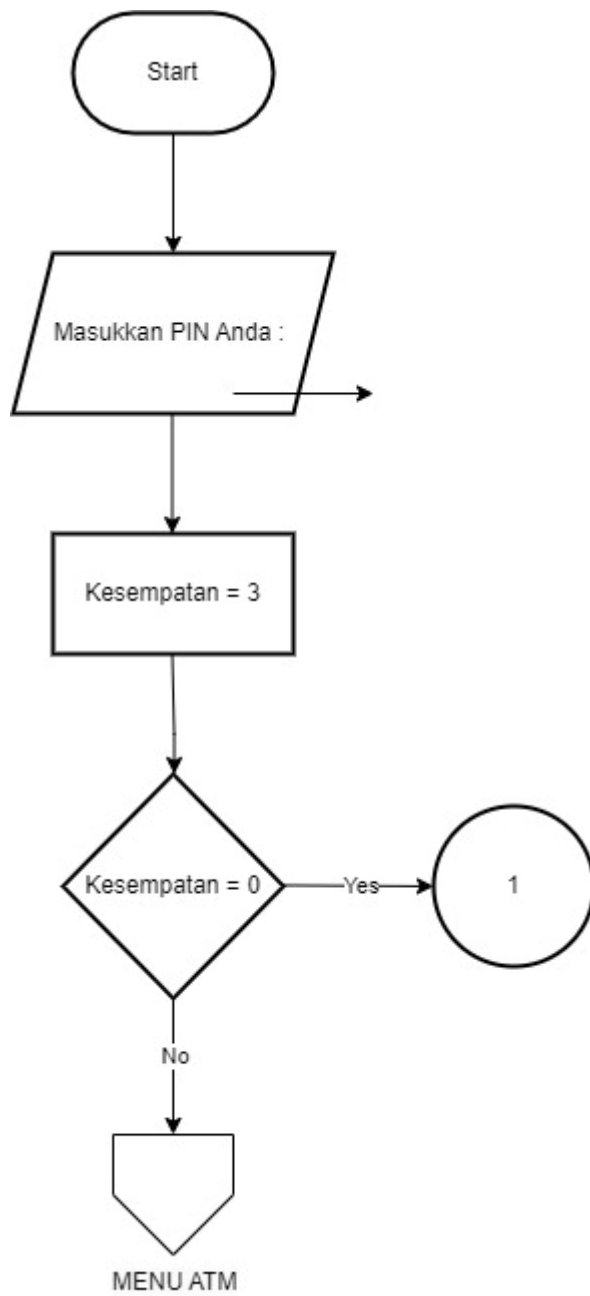
**Disusun oleh:**

**Anindtya Puji Astari (2409106063)**

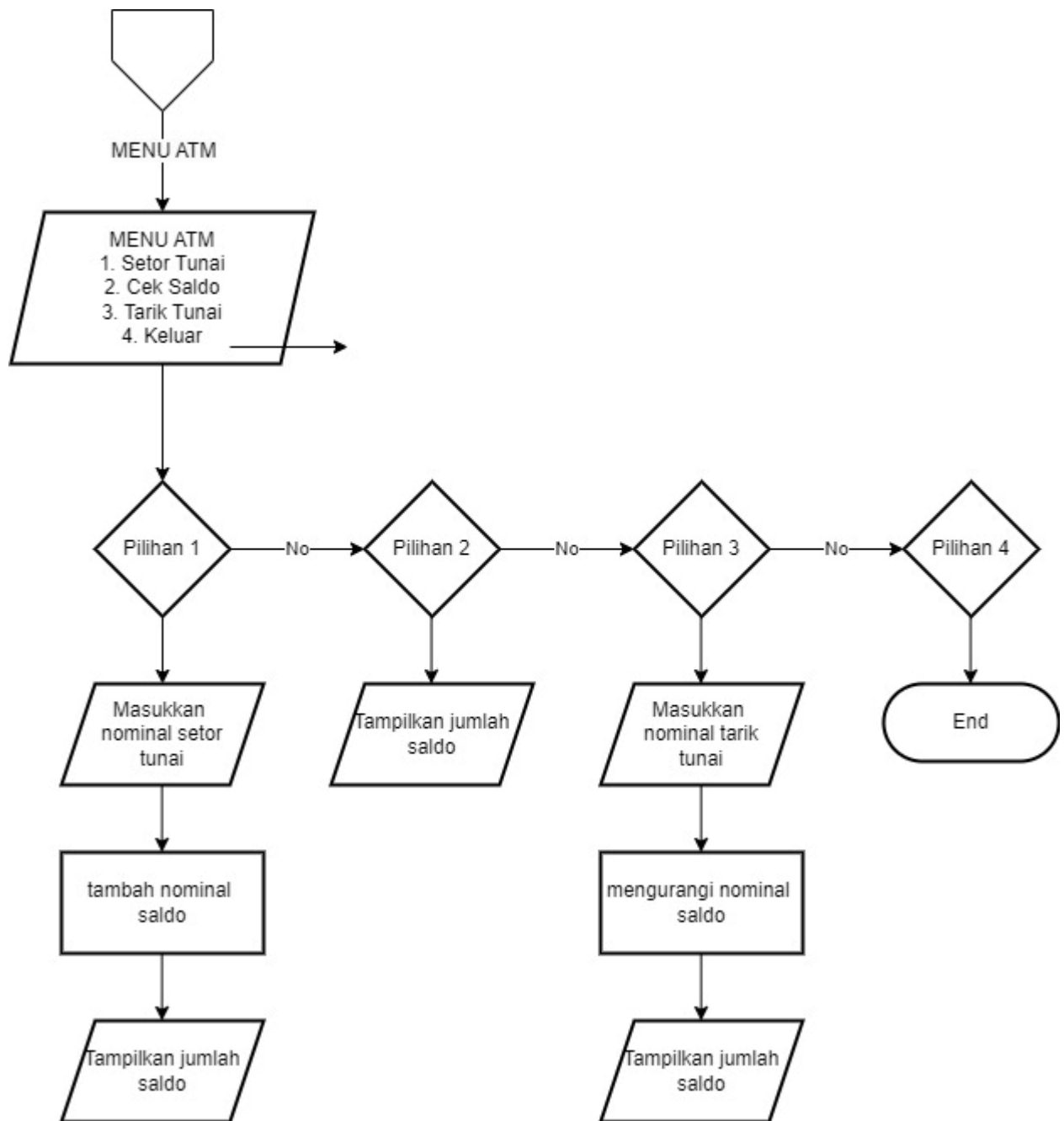
**Kelas (B1'24)**

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

## 1. Flowchart



Gambar 1.1 Flowchart Program Login ATM



Gambar 1.2 Flowchart Menu Utama ATM

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini merupakan program ATM sederhana yang dibuat untuk simulasi fungsi dasar ATM. Terdapat sistem login dengan 4 digit angka sebagai PIN dengan maksimal 3 kali percobaan. Setelah berhasil login user akan mendapat 4 menu, yaitu Setor Tunai, Cek Saldo, Tarik Tunai, dan menu Keluar.

### **2.2 Penjelasan Alur & Algoritma**

#### **1. Deklarasi dan Inisialisasi Variabel**

Program dimulai dengan mendeklarasikan variabel-variabel

- PIN(1037) : PIN yang digunakan untuk autentikasi.
- Saldo (1000000000) : Saldo awal pengguna.
- inputPIN : Menyimpan input PIN pengguna.
- Kesempatan(3) : Menyimpan jumlah percobaan login yang tersisa.

#### **2. Verifikasi PIN**

Pengguna memiliki 3 kesempatan untuk memasukkan PIN yang benar, program menggunakan perulangan for untuk login. Jika PIN sesuai, pengguna bisa mengakses menu utama. Jika salah 3 kali berturut-turut, akun akan terblokir dan program berhenti.

#### **3. Menu Utama ATM**

Setelah login berhasil, pengguna masuk ke menu utama dengan 4 pilihan opsi:

1. Setor Tunai : Pengguna memasukkan nominal uang yang ingin disetor.
2. Cek Saldo : Menampilkan saldo pengguna
3. Tarik Tunai : Pengguna memasukkan nominal uang yang ingin ditarik.
4. Keluar : Mengakhiri program.

#### **4. Validasi Input dan Proses Transaksi**

- Program memastikan setiap input valid, seperti memastikan nominal lebih dari 0 dan saldo mencukupi untuk penarikan.
- Dari pilihan menu, program memproses transaksi dan menampilkan output kepada pengguna, misalnya Setor tunai atau Tarik tunai dan outputnya akan ditampilkan.
- Program terus berjalan hingga pengguna memilih Keluar.

#### **5. Keluar dari Program**

- Program menampilkan output "Terima kasih telah menggunakan layanan ATM" dan berhenti.

### **Penjelasan Detail Blok Kode:**

Loop for : Berfungsi untuk membatasi pengguna maksimal 3 kali percobaan login sebelum akun terblokir. Jika PIN benar, pengguna masuk ke menu ATM. Jika gagal 3 kali, akun terblokir dan program berhenti.

Loop while : Berfungsi untuk memastikan menu ATM tetap berjalan sampai pengguna memilih menu Keluar (4). Program akan meminta pengguna memilih menu setidaknya satu kali.

Switch Case : Berfungsi untuk membaca input pengguna dan menjalankan transaksi sesuai pilihan menu, serta memastikan pengguna tidak memasukkan angka yang tidak valid.

## **3. Source Code**

### **A. Verifikasi pin**

Fitur ini digunakan untuk memverifikasi input PIN yang dimasukan oleh user. Jika PIN benar, pengguna masuk ke menu ATM, dan jika user salah memasukan PIN sampai 3 kali berturut-turut maka program akan berhenti.

```

int PIN = 6063;
int saldo = 1000000000;

int menuATM();

int main()
{
    int inputPIN;
    int kesempatan = 3;

    for (int i = 0; i < 3; i++)
    {
        cout << "Masukkan PIN Anda : ";
        cin >> inputPIN;

        if (inputPIN == PIN)
        {
            cout << "\nLogin Berhasil \nSelamat Datang di ATM"
<< endl;

            int pilihan;
            do
            {
                pilihan = menuATM();
            }
            while (pilihan != 4);
            cout << "Terima Kasih telah menggunakan layanan ATM"
<< endl;

            return 0;
        }
        else
        {
            kesempatan--;
            cout << "\nPIN SALAH! \nKesempatan Tersisa : " <<
kesempatan << "\n";
        }
    }
}

```

```
}  
    cout << "\nAkun Anda terblokir karena salah memasukkan PIN 3  
kali" << endl;  
    return 0;  
}
```

## B. Menu Utama ATM

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

```
int menuATM()
{
    int pilihan;
    int nominal;

    cout << "\nMENU ATM" << endl;
    cout << "1. Setor Tunai" << endl;
    cout << "2. Cek Saldo" << endl;
    cout << "3. Tarik Tunai" << endl;
    cout << "4. Keluar" << endl;
    cout << "Pilih Menu : ";
    cin >> pilihan;

    switch (pilihan)
    {
        case 1:
            cout << "Masukkan nominal setor tunai : ";
            cin >> nominal;
            if (nominal > 0)
            {
                saldo += nominal;
                cout << "\nSetor tunai berhasil \nSaldo
Anda sekarang : " << saldo << endl;
            }
            else
            {
                cout << "Nominal harus lebih dari 0!" <<
endl;
            }
            break;

        case 2:
```



```

        cout << "\nSaldo Anda saat ini : " << saldo
<< endl;

        break;

    case 3:
        cout << "Masukkan nominal tarik tunai : ";
        cin >> nominal;
        if (nominal > saldo)
        {
            cout << "Saldo tidak mencukupi" << endl;
        }
        else if (nominal > 0)
        {
            saldo -= nominal;
            cout << "\nTarik tunai berhasil \nSaldo
Anda sekarang : " << saldo << endl;
        }
        else
        {
            cout << "Nominal harus lebih dari 0!" <<
endl;
        }
        break;

    case 4:
        cout << "Keluar dari Menu ATM..." << endl;
        break;

    default:
        cout << "Pilihan tidak valid! \nSilahkan pilih
menu yang tersedia" << endl;
    }
    return pilihan;
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

#### 1. Skenario 1

Uji coba Setor Tunai dan melihat apakah nominalnya bertambah dengan benar.

#### 2. Skenario 2

Uji coba Tarik Tunai dan melihat apakah nominal berkurang dengan benar.

### 4.2 Hasil Output

```
MENU ATM
1. Setor Tunai
2. Cek Saldo
3. Tarik Tunai
4. Keluar
Pilih Menu : 1
Masukkan nominal setor tunai : 100000

Setor tunai berhasil
Saldo Anda sekarang : 1000100000
```

Gambar 4.1 Hasil Skenario 1

```
MENU ATM
1. Setor Tunai
2. Cek Saldo
3. Tarik Tunai
4. Keluar
Pilih Menu : 3
Masukkan nominal tarik tunai : 55000000

Tarik tunai berhasil
Saldo Anda sekarang : 945100000
```

Gambar 4.2 Hasil Skenario 2

## 5. Git

### 5.1 Git Init

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git init
Reinitialized existing Git repository in C:/Users/Lenovo/Desktop/Praktikum-APL/Post-test/Post-test-APL-1/.git/
```

Git init berfungsi membuat repository git pada folder tertentu.

### 5.2 Git Add

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git add .
```

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

### 5.3 Git Commit

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git commit -m "commit1"
[main (root-commit) f7d8381] commit1
3 files changed, 96 insertions(+)
create mode 100644 2409106063-AnindtyaPujiAstari-PT-1.cpp
create mode 100644 2409106063-AnindtyaPujiAstari-PT-1.exe
create mode 100644 2409106063-AnindtyaPujiAstari-PT-1.pdf
```

Git commit berfungsi untuk menyimpan perubahan di repository lokal.

Pesan di -m "... " menjelaskan perubahan yang dilakukan, sebagai tanda.

### 5.4 Git remote add origin

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git remote add origin https://github.com/Yyak135/Praktikum-APL.git
```

Menghubungkan repository lokal ke repository GitHub.

### 5.5 Git push origin main

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL/Post-test/Post-test-APL-1 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.06 MiB | 465.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Yyak135/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.