

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

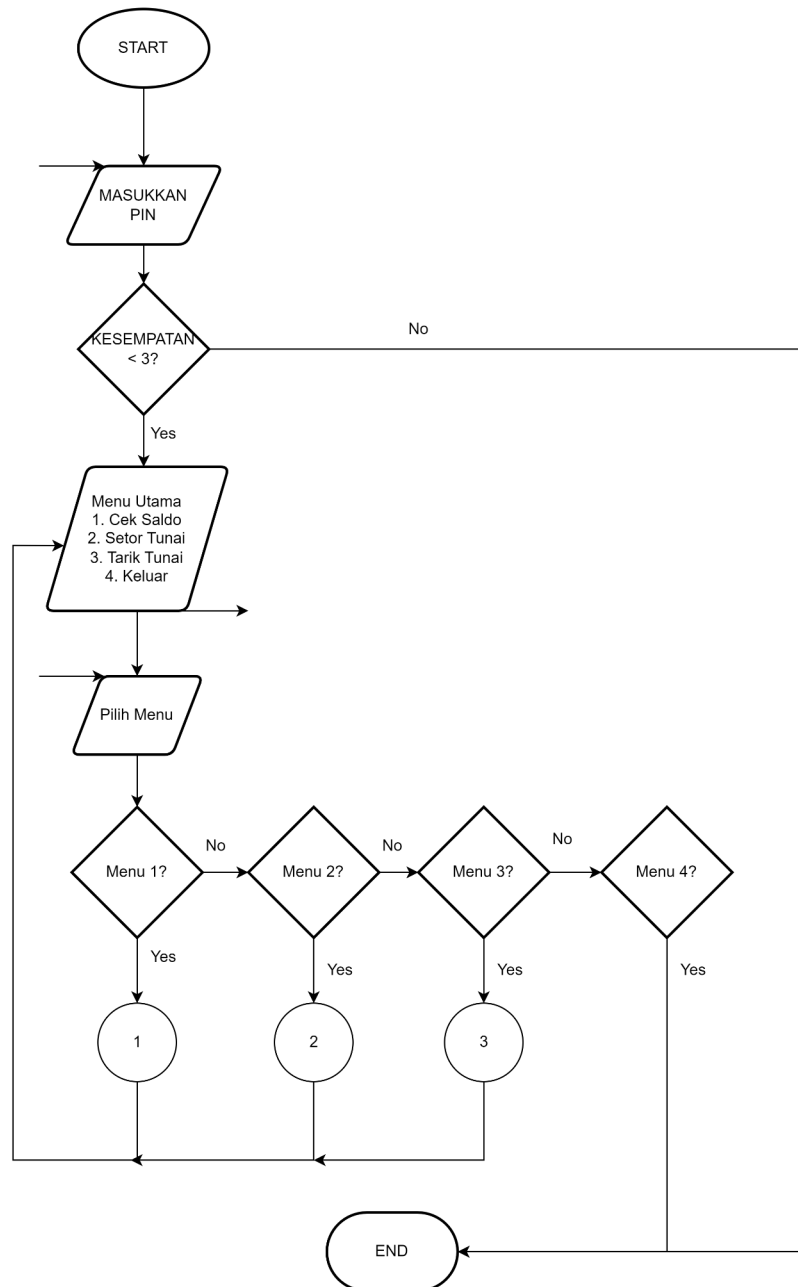


**Disusun oleh:**  
**Muhammad Riva Fachrodhiya (2409106053)**  
**Kelas (B1 '24)**

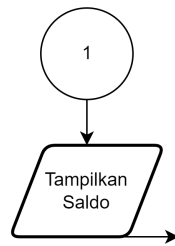
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart

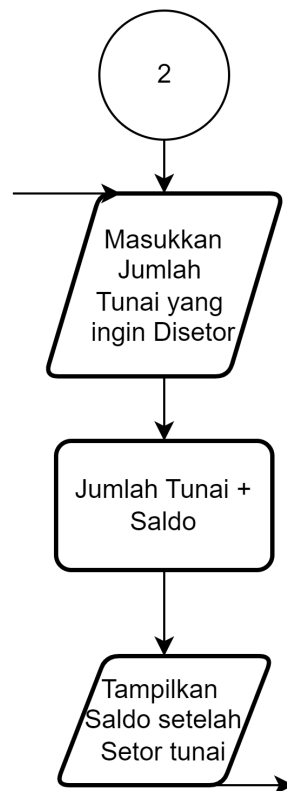
Berikut adalah flowchart menu utama.



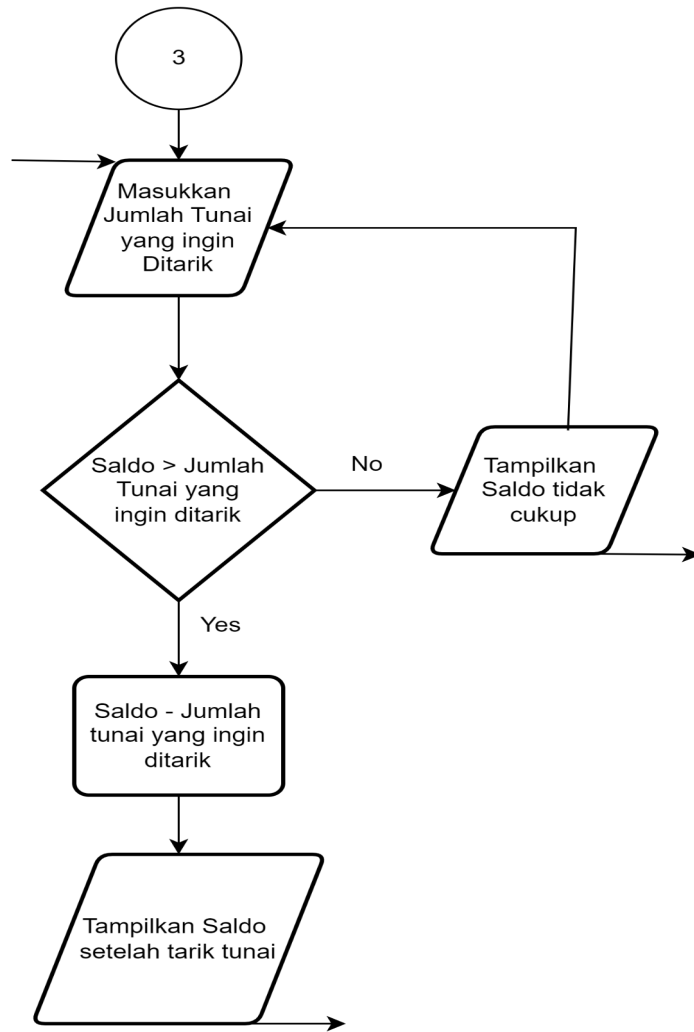
Gambar 1.1 Flowchart Menu Utama



Gambar 1.2 Flowchart Menu Cek Saldo



Gambar 1.3 Flowchart Menu Setor Tunai



Gambar 1.4 Flowchart Menu Tarik Saldo

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Judul dari program ini ialah membuat program ATM Sederhana. Tujuan dibuatnya program ini adalah mempermudah user untuk cek saldo, setor tunai, dan tarik tunai.

## 2.2 Penjelasan Alur & Algoritma

2.2.1 Pertama-tama, user harus login ke program dengan cara memasukkan pin. User diberi kesempatan untuk memasukkan pin sebanyak 3 kali. Jika user salah sebanyak 3 kali percobaan, maka program akan diberhentikan. Namun, jika user tidak salah sebanyak 3 kali percobaan, maka user akan masuk ke menu utama.

2.2.2 Kedua, user akan ditampilkan menu utama. Dalam menu utama, terdapat beberapa sub menu, yaitu cek saldo, setor tunai, tarik tunai, dan keluar dari program. User dapat memilih sub menu mana yang ingin diakses. Anggap saja kita akan mulai dari cek saldo.

2.2.3 Dalam menu cek saldo, user akan langsung dapat tampilan saldo dari user tersebut.

2.2.4 Dalam menu setor tunai, user akan diminta memasukkan jumlah uang tunai yang ingin disetor. Setelah uang tunai disetor, sistem akan mendeteksi ada sejumlah uang yang masuk dan akan langsung menambahkan ke dalam saldo user. Terakhir, user akan dapat melihat jumlah saldo setelah proses setor tunai tersebut.

2.2.5 Dalam menu tarik tunai, user akan diminta memasukkan jumlah uang yang ingin ditarik. Setelah user memasukkan jumlah uang yang ingin ditarik, sistem akan melakukan pengecekan apakah saldo cukup untuk melakukan proses tarik tunai. Jika saldo tidak cukup, maka user akan dapat tampilan berupa “Saldo tidak cukup”, namun jika saldo cukup, maka saldo akan langsung berkurang sesuai dengan jumlah uang yang ingin ditarik oleh user.

2.2.6 Terakhir, menu keluar. Dalam menu ini, user akan langsung keluar dari program.

## 3. Source Code

### A. Fitur Clear Screen

Fitur ini digunakan untuk membersihkan terminal.

#### Source Code:

```
void clearScreen() {  
    system("cls");  
}
```

## B. Fitur Cek Saldo

Fitur ini digunakan untuk user agar dapat cek saldo.

```
void cekSaldo() {  
    cout << fixed << setprecision(0);  
    cout << "Saldo Anda saat ini: Rp " << saldo << endl;  
}
```

## C. Fitur Setor Tunai

Fitur ini digunakan untuk user agar dapat setor tunai.

```
void setorTunai() {  
    double jumlah;  
    cout << "Masukkan jumlah setor tunai: Rp ";  
    cin >> jumlah;  
  
    if (cin.fail()) {  
        cin.clear();  
        cin.ignore(1000, '\n');  
        cout << "Input tidak valid! Harus angka.\n";  
        return;  
    }  
  
    if (jumlah > 0) {  
        saldo += jumlah;  
        cout << fixed << setprecision(0);  
        cout << "Setor tunai berhasil! Saldo Anda sekarang: Rp " << saldo <<  
endl;  
    } else {  
        cout << "Jumlah tidak valid!" << endl;  
    }  
}
```

## D. Fitur Tarik Tunai.

Fitur ini digunakan agar user dapat menarik tunai dari saldo.

```
void tarikTunai() {  
    double jumlah;  
    cout << "Masukkan jumlah tarik tunai: Rp ";  
    cin >> jumlah;  
  
    if (cin.fail()) {
```

```

        cin.clear();
        cin.ignore(1000, '\n');
        cout << "Input tidak valid! Harus angka.\n";
        return;
    }

    if (jumlah > 0 && jumlah <= saldo) {
        saldo -= jumlah;
        cout << fixed << setprecision(0);
        cout << "Tarik tunai berhasil! Saldo Anda sekarang: Rp " << saldo <<
endl;
    } else {
        cout << "Jumlah tidak valid atau saldo tidak mencukupi!" << endl;
    }
}

```

### E. Fitur Login.

Fitur ini digunakan agar user dapat login ke program dengan password yang telah ditentukan.

```

int main() {
    string pin;
    int kesempatan = 0;
    bool isAuthenticated = false;

    while (kesempatan < kesempatanMaks) {
        clearScreen();
        cout << "=====";
        cout << "\nMasukkan PIN";
        cout << "\n=====";
        cout << "\nPIN : ";
        cin >> pin;

        if (pin.length() != 4) {
            cout << "PIN harus 4 digit!" << endl;
        } else if (pin == pinATM) {
            isAuthenticated = true;
            break;
        } else {
            cout << "PIN salah! Percobaan ke-" << (kesempatan + 1) << endl;
        }

        delay(1);
        kesempatan++;
    }

    if (!isAuthenticated) {

```

```

        cout << "Anda telah 3 kali salah memasukkan PIN. ATM diblokir!" <<
endl;
        return 0;
    }

```

## F. Fitur Delay.

Fitur ini digunakan untuk delay program agar user dapat membaca output yang berada di terminal terlebih dahulu.

```

void delay(int detik) {
    this_thread::sleep_for(chrono::seconds(detik));
}

```

## G. Fitur Utama.

Fitur ini mencakup seluruh fitur-fitur diatas.

```

int pilihan;
do {
    clearScreen();
    cout << "===== ";
    cout << "\n\t\t\t\t\t Menu ATM\n";
    cout << "===== ";
    cout << "\n1. Cek Saldo >>\n2. Setor Tunai >>\n3. Tarik Tunai >>\n4.
Keluar >>\n";
    cout << "===== ";
    cout << "\nPilih menu: ";

    cin >> pilihan;

    if (cin.fail()) {
        cin.clear();
        cin.ignore(1000, '\n');
        cout << "Input tidak valid! Harap masukkan angka 1-4.\n";
        delay(2);
        continue;
    }

    clearScreen();

    switch (pilihan) {
        case 1:
            cekSaldo();
            break;

```



```

        case 2:
            setorTunai();
            break;
        case 3:
            tarikTunai();
            break;
        case 4:
            cout << "Terima kasih telah menggunakan ATM ini!" << endl;
            break;
        default:
            cout << "Pilihan tidak valid! Harap masukkan angka 1-4.\n";
    }

    if (pilihan != 4) {
        cout << "\nTekan Enter untuk kembali ke menu... \n(Jika tekan
sekali tidak bisa, maka tekan 2 kali)";
        cin.ignore();
        cin.get();
    }

    } while (pilihan != 4);

    return 0;
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Jadi, skenario yang akan saya gunakan ada 3, yaitu:

1. Skenario 1, user akan salah dalam memasukkan pin sebanyak 3 kali.
2. Skenario 2, user akan menarik uang, namun saldo user tidak cukup.
3. Skenario 3, user akan menggunakan atm secara normal.

### 4.2 Hasil Output

```

=====
Masukkan PIN
=====
PIN : 1111
PIN salah! Percobaan ke-3
Anda telah 3 kali salah memasukkan PIN. ATM diblokir!

```

Gambar 4.1 Skenario 1

```
Masukkan jumlah tarik tunai: Rp 20000
Jumlah tidak valid atau saldo tidak mencukupi!

Tekan Enter untuk kembali ke menu...
(Jika tekan sekali tidak bisa, maka tekan 2 kali)
```

Gambar 4.1 Skenario 2

```
=====
                        Menu ATM
=====
1. Cek Saldo >>
2. Setor Tunai >>
3. Tarik Tunai >>
4. Keluar >>
=====
Pilih menu: 
```

Gambar 4.1 Skenario 3

```
Saldo Anda saat ini: Rp 20000

Tekan Enter untuk kembali ke menu...
(Jika tekan sekali tidak bisa, maka tekan 2 kali)
```

Gambar 4.1 Skenario 3

```
Masukkan jumlah setor tunai: Rp 10000
Setor tunai berhasil! Saldo Anda sekarang: Rp 30000

Tekan Enter untuk kembali ke menu...
(Jika tekan sekali tidak bisa, maka tekan 2 kali)
```

Gambar 4.1 Skenario 3

```
Masukkan jumlah tarik tunai: Rp 15000
Tarik tunai berhasil! Saldo Anda sekarang: Rp 15000

Tekan Enter untuk kembali ke menu...
(Jika tekan sekali tidak bisa, maka tekan 2 kali)
```

Gambar 4.1 Skenario 3

## 5. Langkah - Langkah GIT

### 5.1 Git Init

Git init dilakukan untuk menginisiasikan repository git dengan cara menuliskan “git init” di directory kita. Disini saya banyak melakukan kesalahan di git init, seharusnya directory yang dituju adalah C:\praktikum-apl\post-test\post-test-1

```
PS C:\APL.SMT2\praktikum-apl\post-test\post-test-1> git init
Initialized empty Git repository in C:/APL.SMT2/praktikum-apl/post-test/post-test-1/.git/
```

Gambar 5.1 Git Init

## 5.2 Git Add

Git add berfungsi untuk melakukan perubahan sebelum di commit menggunakan git commit.

```
PS C:\APL.SMT2\praktikum-apl\post-test\post-test-1> git add .
PS C:\APL.SMT2\praktikum-apl\post-test\post-test-1> |
```

Gambar 5.2 Git Add

## 5.3 Git Commit

Git commit berfungsi untuk melakukan perubahan pada file yang kita tinjau di git add dan dapat memberi komentar untuk perubahan apa yang kita lakukan.

```
PS C:\praktikum-apl> git commit -m "Upload filee"
On branch main
nothing to commit, working tree clean
```

Gambar 5.3 Git Commit

## 5.4 Git Remote

Git remote berfungsi menyambungkan file dari git bash ke github.

```
PS C:\praktikum-apl> git remote add origin https://github.com/MuhammadRivaFachrodhiya/praktikum-apl
error: remote origin already exists.
```

Gambar 5.4 Git Remote

## 5.5 Git Push

Git push adalah langkah terakhir yang akan mengirimkan langkah langkah sebelumnya ke github.

```
PS C:\praktikum-apl\post-test\post-test-1> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 679.94 KiB | 2.50 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MuhammadRivaFachrodhiya/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.5 Git Push