LAPORAN PRAKTIKUM

POSTTEST 1

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

DEVON FALEN PASAE

2409106055

Kelas B1‘24

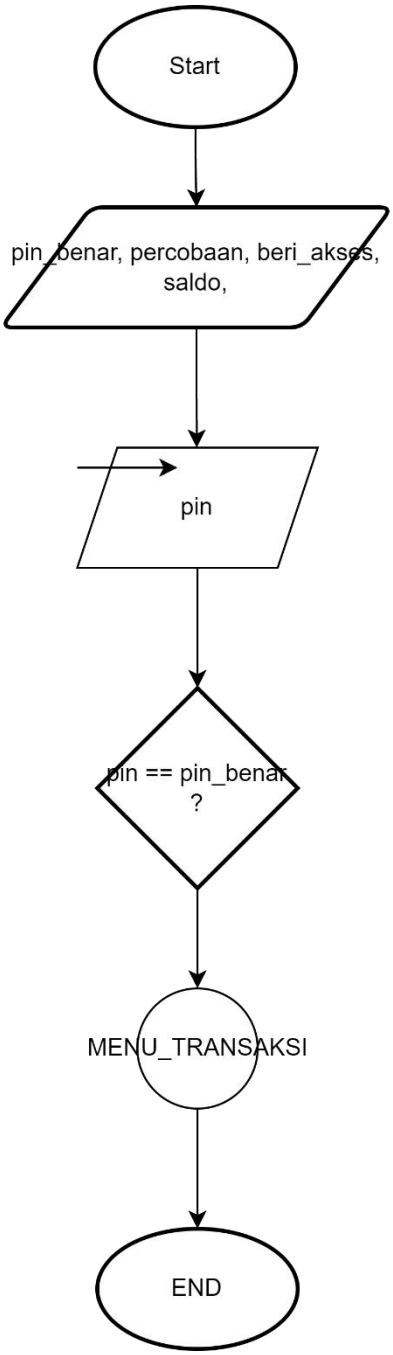
PROGRAM STUDI INFORMATIKA

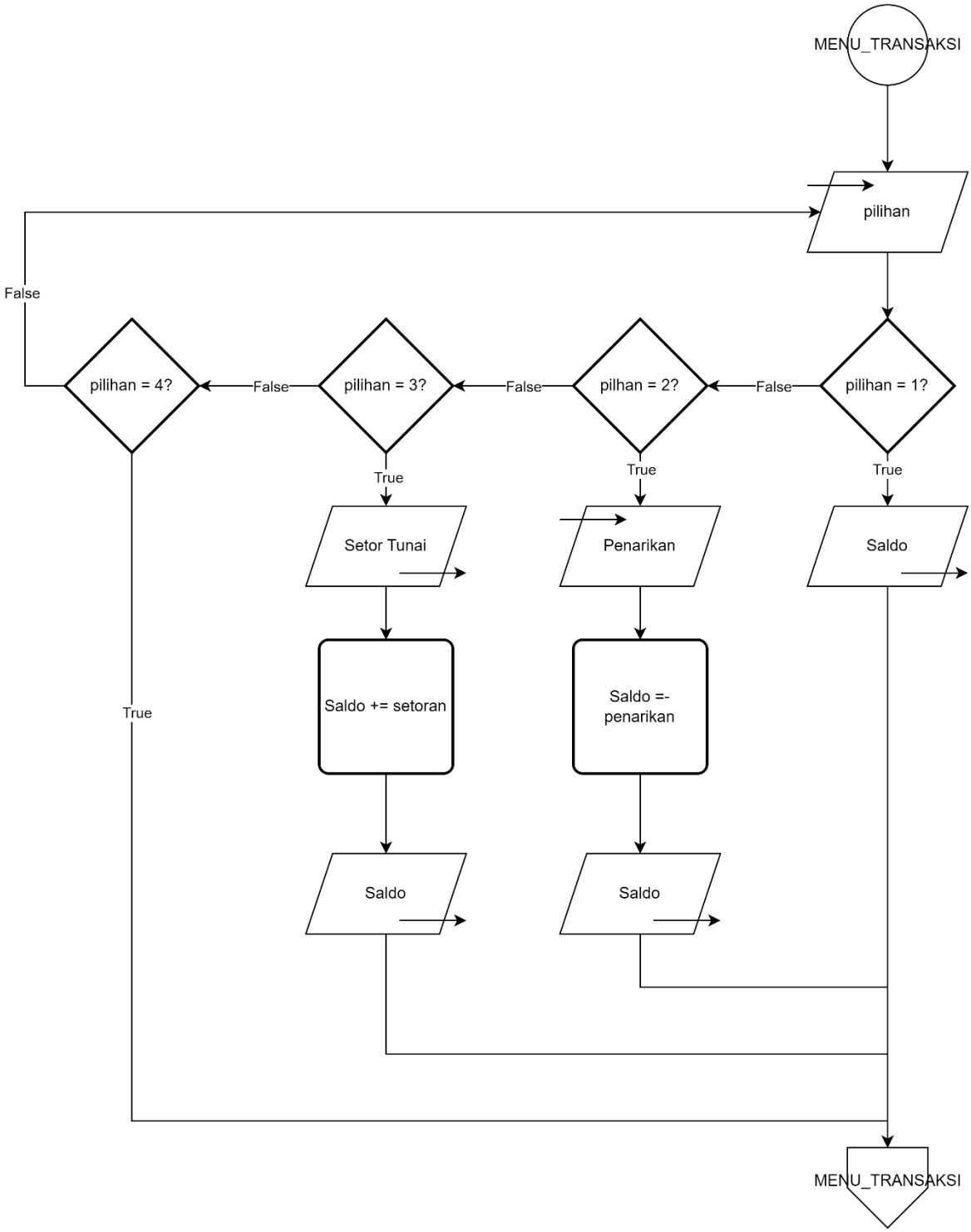
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

1. Flowchart


Start
pin_benar, percobaan, beri_akses, saldo,
pin
pin == pin_benar ?
MENU_TRANSAKSI
END


MENU_TRANSAKSI
pilihan
pilihan = 1?
pilhan = 2?
pilihan = 3?
pilihan = 4?
False
False
False
False
True
True
True
True
Saldo
Penarikan
Setor Tunai
Saldo =- penarikan
Saldo += setoran
Saldo
Saldo
MENU_TRANSAKSI

2. Analisis Program

2.1 Deskripsi Singkat Program

Program di atas adalah simulasi sederhana dari sistem ATM berbasis C++. Pengguna harus memasukkan PIN yang benar untuk mendapatkan akses ke transaksi perbankan. Setelah berhasil masuk, mereka dapat memilih dari beberapa opsi: **cek saldo**, **tarik tunai**, **setor tunai**, atau **keluar**. Program terus berjalan hingga pengguna memilih untuk keluar.

2.2 Penjelasan Alur & Algoritma

Program dimulai dengan mengimpor library iostream untuk operasi input dan output. Variabel pin_benar didefinisikan sebagai konstanta dengan nilai **6055**, yang merupakan PIN yang benar. Variabel percobaan diinisialisasi sebagai **3**, yang menunjukkan jumlah maksimal percobaan login. Variabel beri_akses bertipe boolean digunakan untuk menentukan apakah pengguna berhasil masuk ke dalam sistem, dengan nilai awal false. Selain itu, variabel saldo diinisialisasi dengan nilai **0** untuk menyimpan saldo pengguna, dan pilihan digunakan untuk menyimpan opsi yang dipilih dalam menu transaksi.

Di dalam fungsi main(), program meminta pengguna untuk memasukkan PIN. Jika PIN yang dimasukkan sesuai dengan pin_benar, program akan memberikan akses ke menu transaksi dan menampilkan pesan autentikasi berhasil. Jika PIN yang dimasukkan salah, program menampilkan jumlah percobaan yang tersisa dan memberikan kesempatan hingga **3 kali**. Jika pengguna gagal memasukkan PIN yang benar setelah **3 percobaan**, program akan menampilkan pesan bahwa akun terkunci dan berhenti.

Jika login berhasil, program akan menampilkan **menu utama**, yang memungkinkan pengguna untuk memilih antara:

1. **Cek Saldo** → Menampilkan saldo saat ini.
2. **Tarik Tunai** → Meminta input jumlah uang yang akan ditarik. Jika saldo mencukupi, jumlah saldo akan dikurangi dan program menampilkan saldo tersisa. Jika saldo tidak mencukupi, program menampilkan pesan kesalahan.
3. **Setor Tunai** → Meminta input jumlah uang yang akan disetorkan. Program menambahkan jumlah tersebut ke saldo dan menampilkan saldo terbaru.
4. **Keluar** → Menghentikan program dan menampilkan pesan terima kasih.

Selama pengguna tidak memilih opsi **Keluar (4)**, program akan terus menampilkan menu utama setelah setiap transaksi selesai, memungkinkan pengguna untuk melakukan transaksi lainnya. Jika pengguna memasukkan opsi yang tidak valid, program akan menampilkan pesan kesalahan dan meminta input ulang.

Setiap kali pengguna memasukkan input, program memastikan bahwa input tersebut valid (misalnya, jumlah penarikan tidak melebihi saldo) dengan menggunakan pernyataan kondisional. Setelah transaksi selesai, hasilnya ditampilkan, dan pengguna dapat memilih untuk kembali ke menu utama atau keluar.

Program berakhir ketika pengguna memilih **Keluar**.

3. Source Code

- Autentikasi

  Fitur ini dibuat sebagai bentuk autentikasi program dengan cara pengguna input password yang benar dan diberi kesempatan 3 kali. Jika pengguna gagal menginput password benar, maka program akan otomatis berakhir.

```
int main() {
    const int pin_benar = 6055;
    const int max_percobaan = 3;
    bool beri_akses = false;
    int saldo = 0;
    int pilihan;

    for(int percobaan = 1; percobaan <= max_percobaan; percobaan++) {
        int input_pengguna;
        cout << "Masukkan PIN Anda: ";
        cin >> input_pengguna;

        if(input_pengguna == pin_benar) {
            beri_akses = true;
            cout << "\n== Autentikasi berhasil! ==\n";
            break;
        } else {
            cout << "PIN salah! Sisa percobaan: " << (max_percobaan - percobaan) << "\n\n";
        }
    }

    if(!beri_akses) {
        cout << "\n== Autentikasi gagal. Akun terkunci. ==\n";
        return 0;
    }
```

- Menu Utama

```
    do {
        cout << "\n=== MENU TRANSAKSI ==="
            << "\n1. Cek Saldo"
            << "\n2. Tarik Tunai"
            << "\n3. Setor Tunai"
            << "\n4. Keluar"
            << "\nPilih opsi (1-4): ";
        cin >> pilihan;
```

```
        switch(pilihan) {
            case 1:
                cout << "\nSaldo Anda: Rp" << saldo << endl;
                break;

            case 2: {
                int jumlah;
                cout << "\nMasukkan jumlah penarikan: Rp";
                cin >> jumlah;

                if(jumlah > saldo) {
                    cout << "Saldo tidak cukup!\n";
                } else {
                    saldo -= jumlah;
                    cout << "Penarikan berhasil. Saldo tersisa: Rp" << saldo << endl;
                }
                break;
            }

            case 3: {
                int jumlah;
                cout << "\nMasukkan jumlah setoran: Rp";
                cin >> jumlah;

                saldo += jumlah;
                cout << "Setoran berhasil. Saldo saat ini: Rp" << saldo << endl;
                break;
            }

            case 4:
                cout << "\nTerima kasih telah menggunakan layanan kami.\n";
                break;

            default:
                cout << "\nPilihan tidak valid! Silakan coba lagi.\n";
        }

    } while(pilihan != 4);

    return 0;
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

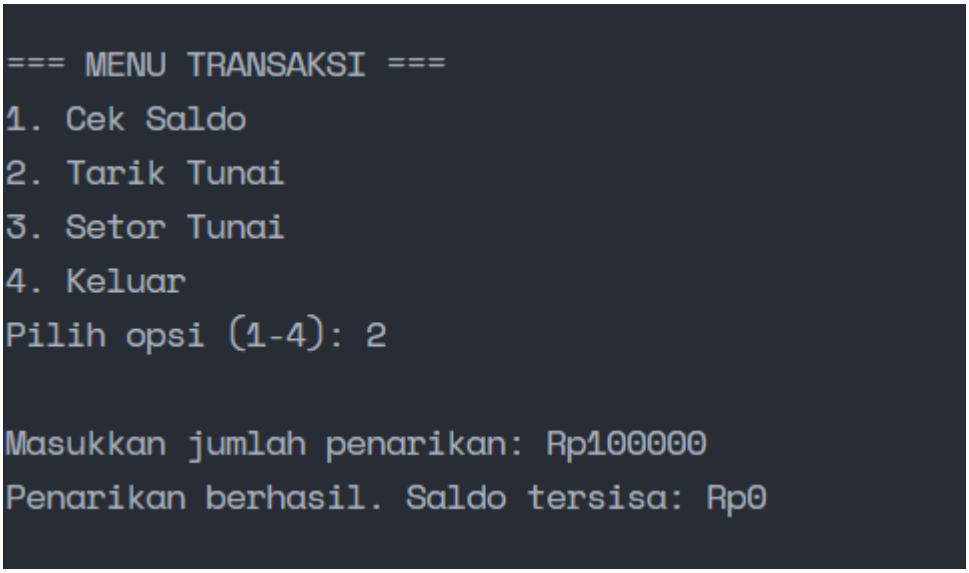
   Jika pengguna memasukkan PIN yang salah selama 3 kali berturut-turut, maka program akan terblokir dan berakhir secara otomatis.

2. Skenario 2

   Saat pengguna mencoba menu “Tarik Tunai’ dengan saldo yang mencukupi atau tidak, dan apakah saldo berkurang dengan perhitungan yang benar.

4.2 Hasil Output

Gambar 4.1 Skenario 1



Gambar 4.2 Skenario 2