

**LAPORAN PRAKTIKUM**  
**POSTTEST 5**  
**ALGORITMA PEMROGRAMAN LANJUT**

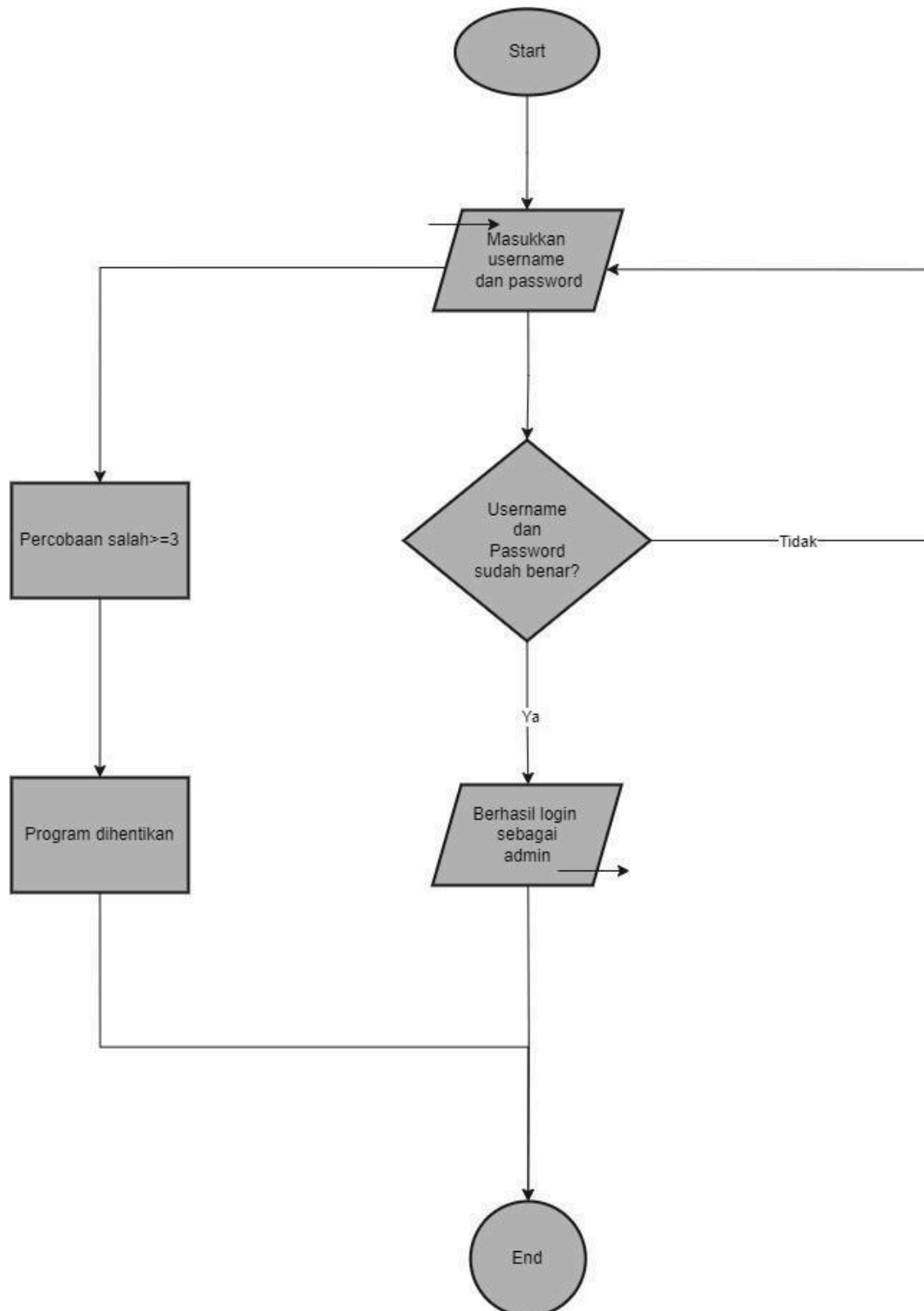


**Disusun oleh:**  
**Rusdiansyah(2409106013)**  
**Kelas (A1 '24)**

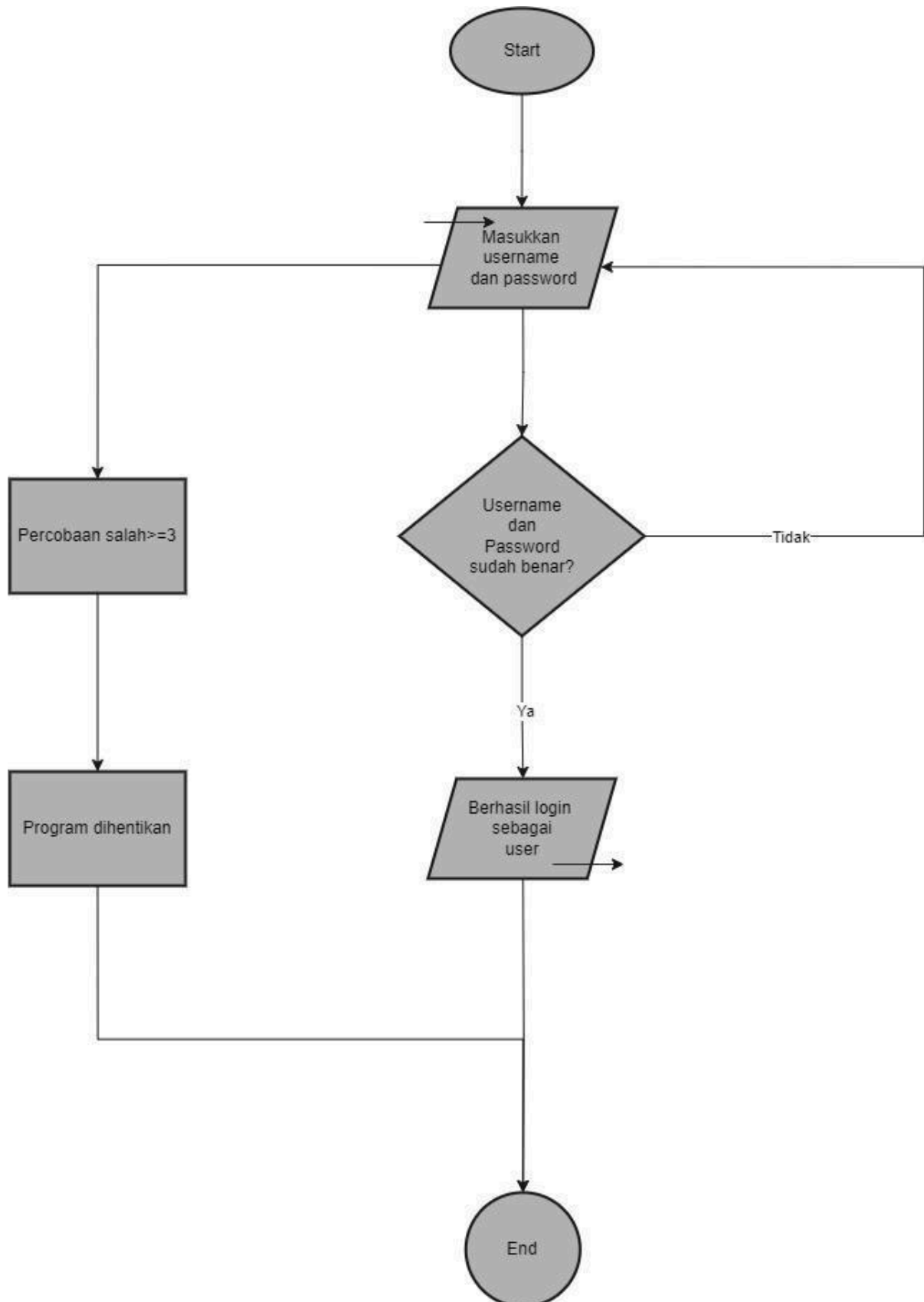
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**

**2025**

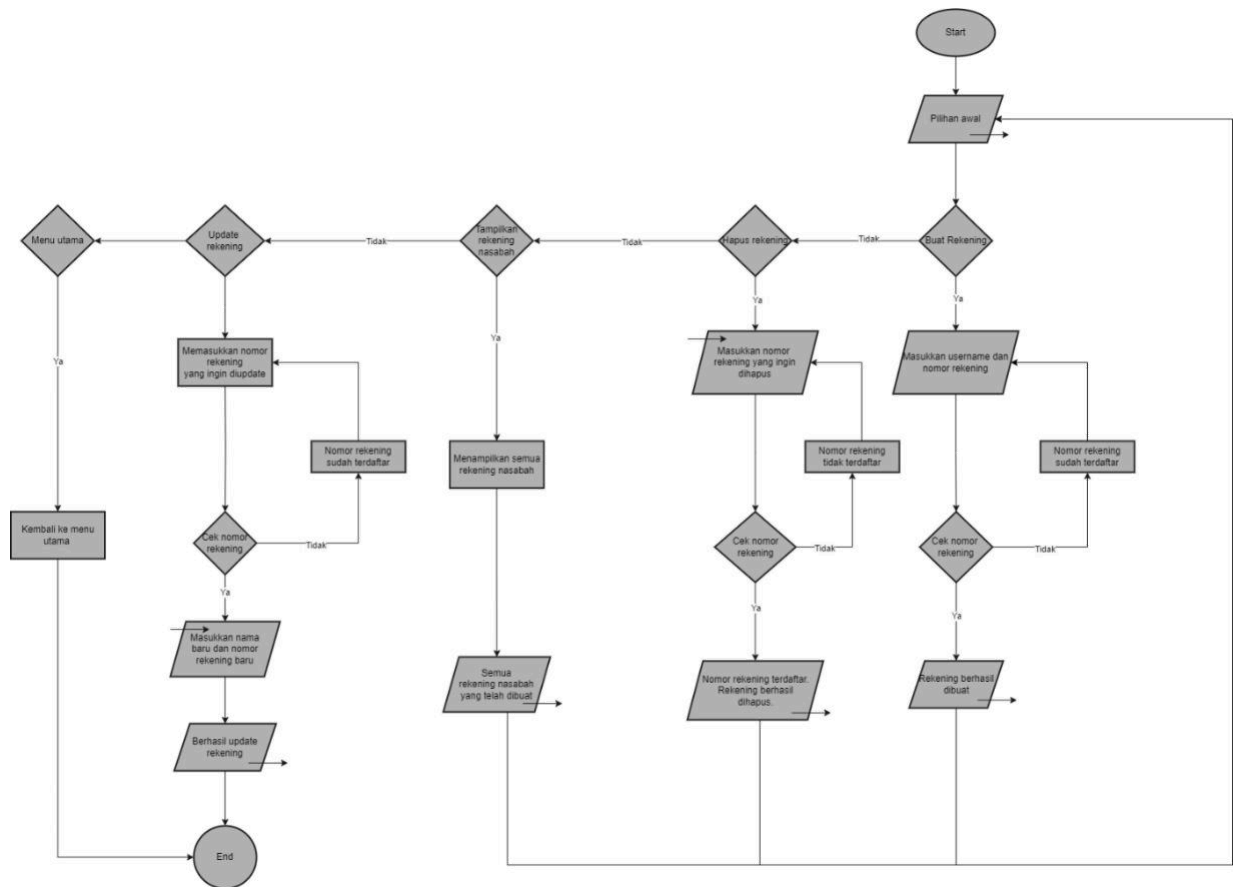
## 1.Flowchart



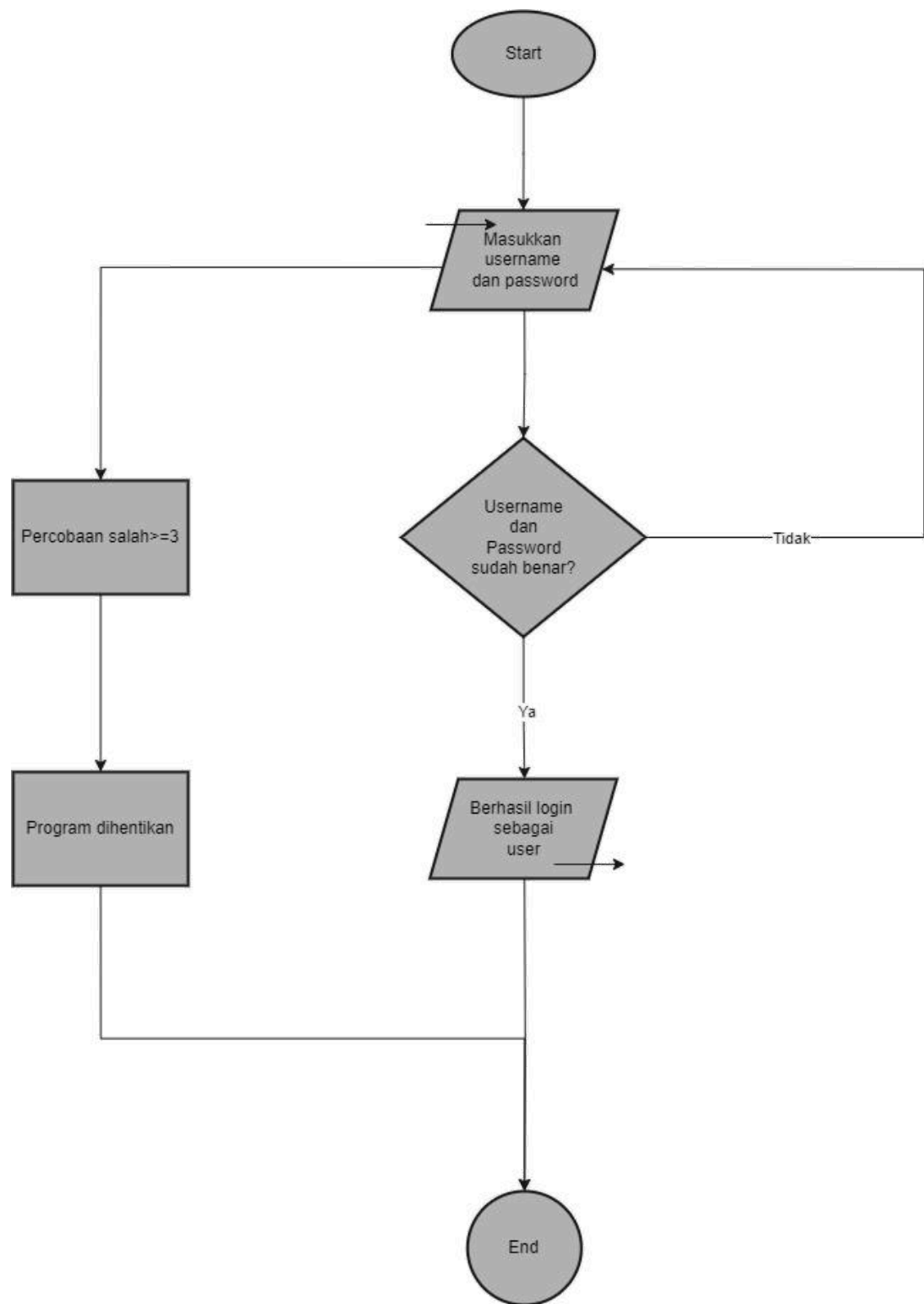
Gambar 1.1 Menu utama



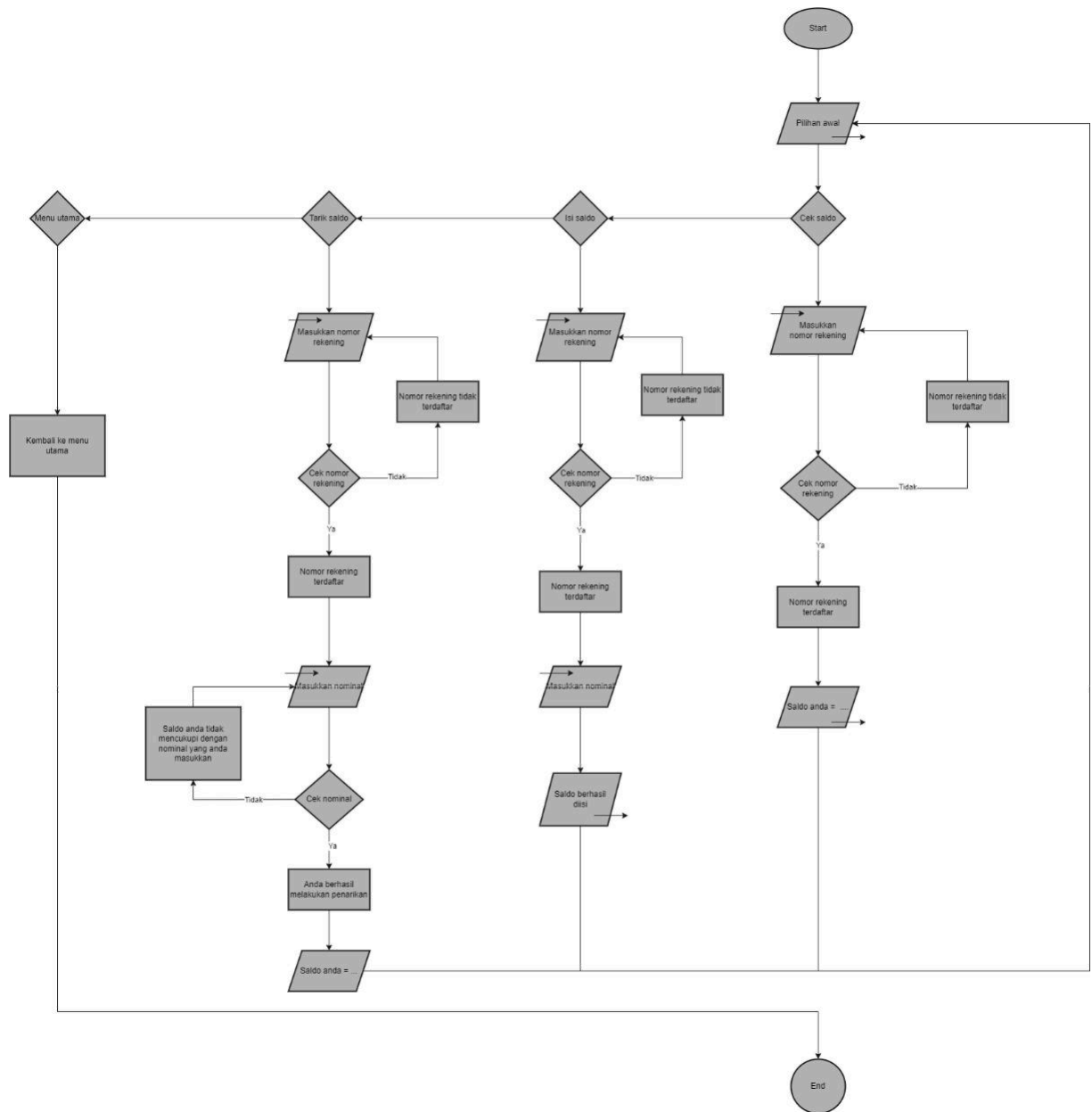
Gambar 1.2 Login sebagai admin



Gambar 1.3 Menu admin



Gambar 1.4 Login sebagai user



Gambar 1.5 Menu user



## 2. Analisis Program

Program ini merupakan simulasi sistem perbankan sederhana yang memungkinkan user atau pengguna untuk melakukan operasi dasar seperti pembuatan rekening, update rekening, penghapusan rekening, pengecekan saldo, pengisian saldo, dan penarikan saldo. Terdapat dua jenis pengguna yakni admin dan user. Yang dimana, admin bertugas mengelola rekening nasabah seperti membuat, menghapus, update, dan menampilkan data rekening. Sedangkan user dapat mengakses informasi saldo, mengisi saldo, dan menarik saldo.

Pointer adalah sebuah variabel khusus dalam bahasa pemrograman C++ yang digunakan untuk menyimpan alamat memori dari suatu data atau variabel lain. Alamat memori ini merujuk ke lokasi di mana data tersebut disimpan di dalam memori komputer. Dengan menggunakan pointer, program dapat mengakses, memanipulasi, atau bahkan mengelola data secara langsung di tingkat memori, memberikan fleksibilitas dan efisiensi yang lebih besar dalam pengelolaan resources.

Pointer dideklarasikan dengan menambahkan tanda \* (simbol asterisk) sebelum nama variabel, dan alamat memori suatu variabel dapat diperoleh menggunakan operator & (address-of). Untuk mengakses nilai yang disimpan di alamat memori yang ditunjuk oleh pointer, digunakan operator \* (dereference).

Pada posttest kali ini, praktikan diminta untuk menambahkan fungsi dengan parameter address-of dan fungsi dengan parameter dereference pada program posttest sebelumnya.

### 3.Source Code

#### 1) Deklarasi fungsi dengan parameter

```
//Deklarasi fungsi dengan parameter
void buatRekening(Nasabah* data, int* size);
void hapusRekening(Nasabah* data, int* size);
void tampilkanRekening(const Nasabah* data, int size);
void updateRekening(Nasabah* data, int size);
void cekSaldo(const Nasabah* data, int size);
void isiSaldo(Nasabah* data, int size);
void tarikSaldo(Nasabah* data, int size);
void menuAdmin(Nasabah* data, int* size, int* errors);
void menuUser(Nasabah* data, int size, int* errors);
```

Gambar 3.1 Deklarasi fungsi dengan parameter

#### 2) Create

```
//Fungsi menggunakan parameter dereference
void buatRekening(Nasabah* data, int* size) {
    if (*size >= MAX_NASABAH) {
        cout << "Maaf jumlah nasabah sudah penuh. Tidak dapat menambahkan rekening baru." << endl;
    } else {
        string nama, nomorRekening;
        cout << "Masukkan nama nasabah : ";
        cin >> nama;
        cout << "Masukkan nomor rekening : ";
        cin >> nomorRekening;

        bool nomorSudahDigunakan = false;
        for (int i = 0; i < *size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                nomorSudahDigunakan = true;
                break;
            }
        }

        if (nomorSudahDigunakan) {
            cout << "Nomor rekening sudah digunakan. Tidak dapat membuat rekening baru." << endl;
        } else {
            data[*size].nama = nama;
            data[*size].nomorRekening = nomorRekening;
            data[*size].saldo = 0;
            (*size)++;
            cout << "Rekening berhasil dibuat." << endl;
        }
    }
}
```

}

Gambar 3.2 Membuat rekening

### 3) Read

```
//Fungsi menggunakan parameter pointer
void tampilkanRekening(const Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        cout <<
        "<=====>" <<
        endl;
        cout << "          |                      DAFTAR NASABAH
|          " << endl;
        cout <<
        "<=====>" <<
        endl;
        for (int i = 0; i < size; i++) {
            cout << "          | " << i + 1 << ". Nama : " << data[i].nama
                << " | Nomor Rekening : " << data[i].nomorRekening
                << " | Saldo : Rp." << data[i].saldo << endl;
        }
        cout <<
        "<=====>"<<endl;
    }
}
```

Gambar 3.3 Menampilkan daftar rekening

```

//Fungsi menggunakan parameter pointer
void cekSaldo(const Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        string nomorRekening;
        cout << "Masukkan nomor rekening : ";
        cin >> nomorRekening;

        bool ditemukan = false;
        for (int i = 0; i < size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                cout << "Nama : " << data[i].nama;
                cout << "Nomor Rekening : " << data[i].nomorRekening << endl;
                cout << "Saldo : Rp. " << data[i].saldo << endl;
                ditemukan = true;
                break;
            }
        }

        if (!ditemukan) {
            cout << "Nomor rekening tidak ditemukan." << endl;
        }
    }
}

```

Gambar 3.4 Cek saldo dalam rekening

#### 4) Update

```
//Fungsi menggunakan parameter pointer
void updateRekening(Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        string nomorRekening;
        cout << "Masukkan nomor rekening yang ingin diupdate : ";
        cin >> nomorRekening;

        bool ditemukan = false;
        for (int i = 0; i < size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                cout << "Masukkan nama baru : ";
                cin >> data[i].nama;
                cout << "Masukkan nomor rekening baru : ";
                cin >> data[i].nomorRekening;
                cout << "Rekening berhasil diupdate." << endl;
                ditemukan = true;
                break;
            }
        }

        if (!ditemukan) {
            cout << "Nomor rekening tidak ditemukan." << endl;
        }
    }
}
```

Gambar 3.5 Mengupdate rekening

```

//Fungsi menggunakan parameter pointer
void isiSaldo(Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        string nomorRekening;
        int jumlah;
        cout << "Masukkan nomor rekening : ";
        cin >> nomorRekening;
        cout << "Masukkan jumlah saldo yang ingin diisi : Rp.";
        cin >> jumlah;

        bool ditemukan = false;
        for (int i = 0; i < size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                data[i].saldo += jumlah;
                cout << "Saldo berhasil diisi! Saldo Anda sekarang : Rp." <<
data[i].saldo << endl;
                ditemukan = true;
                break;
            }
        }

        if (!ditemukan) {
            cout << "Nomor rekening tidak ditemukan." << endl;
        }
    }
}
}

```

Gambar 3.6 Isi saldo

```

//Fungsi menggunakan parameter pointer
void tarikSaldo(Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        string nomorRekening;
        int jumlah;
        cout << "Masukkan nomor rekening : ";
        cin >> nomorRekening;
        cout << "Masukkan jumlah saldo yang ingin diambil : Rp.";
        cin >> jumlah;

        bool ditemukan = false;
        for (int i = 0; i < size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                if (data[i].saldo >= jumlah) {
                    data[i].saldo -= jumlah;
                    cout << "Saldo berhasil diambil! Saldo Anda sekarang : Rp."
<< data[i].saldo << endl;
                } else {
                    cout << "Saldo tidak cukup." << endl;
                }
                ditemukan = true;
                break;
            }
        }

        if (!ditemukan) {
            cout << "Nomor rekening tidak ditemukan." << endl;
        }
    }
}

```

Gambar 3.7 Menarik saldo



## 5) Delete

```
//Fungsi menggunakan parameter pointer
void tarikSaldo(Nasabah* data, int size) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        string nomorRekening;
        int jumlah;
        cout << "Masukkan nomor rekening : ";
        cin >> nomorRekening;
        cout << "Masukkan jumlah saldo yang ingin diambil : Rp.";
        cin >> jumlah;

        bool ditemukan = false;
        for (int i = 0; i < size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                if (data[i].saldo >= jumlah) {
                    data[i].saldo -= jumlah;
                    cout << "Saldo berhasil diambil! Saldo Anda sekarang : Rp."
<< data[i].saldo << endl;
                } else {
                    cout << "Saldo tidak cukup." << endl;
                }
                ditemukan = true;
                break;
            }
        }

        if (!ditemukan) {
            cout << "Nomor rekening tidak ditemukan." << endl;
        }
    }
}
```

Gambar 3.8 Menghapus rekening

## 6) Fungsi parameter address-of dan dereference

```
//Fungsi menggunakan parameter address-of dan dereference
void menuAdmin(Nasabah* data, int* size, int* errors) {
    while (true) {
        cout <<
        "<=====>" <<
        endl;
        cout << "          |          MENU ADMIN
|          " << endl;
        cout <<
        "<=====>" <<
        endl;
        cout << "          |          1. Buat Rekening
|          " << endl;
        cout << "          |          2. Hapus Rekening
|          " << endl;
        cout << "          |          3. Tampilkan Rekening
|          " << endl;
        cout << "          |          4. Update Rekening
|          " << endl;
        cout << "          |          0. Kembali ke Menu Utama
|          " << endl;
        cout <<
        "<=====>" <<
        endl;

        int pilihanAdmin;
        cout << "Pilih Menu: ";
        cin >> pilihanAdmin;

        switch (pilihanAdmin) {
            case 1:
                buatRekening(data, size);
                break;
            case 2:
                hapusRekening(data, size);
                break;
            case 3:
                tampilkanRekening(data, *size);
                break;
            case 4:
                updateRekening(data, *size);
                break;
            case 0:
                return;
            default:
                cout << "Pilihan tidak valid." << endl;
                (*errors)++;
                if (*errors >= 3) {
```

```

        cout << "Anda gagal sebanyak 3 kali, program dihentikan." <<
endl;
        exit(0);
    }
}
}
}

//Fungsi menggunakan parameter address-of dan dereference
void menuUser(Nasabah* data, int size, int* errors) {
    while (true) {
        cout <<
"<=====>" <<
endl;
        cout << "          |          MENU USER          |
" << endl;
        cout <<
"<=====>" <<
endl;
        cout << "          |          1. Cek Saldo          |
" << endl;
        cout << "          |          2. isi Saldo          |
" << endl;
        cout << "          |          3. Tarik Saldo          |
" << endl;
        cout << "          |          0. Kembali Ke Menu Utama    |
" << endl;
        cout <<
"<=====>" <<
endl;

        int pilihanUser;
        cout << "Pilih Menu User :";
        cin >> pilihanUser;

        switch (pilihanUser) {
            case 1:
                cekSaldo(data, size);
                break;
            case 2:
                isiSaldo(data, size);
                break;
            case 3:
                tarikSaldo(data, size);
                break;
            case 0:
                return;
            default:
                cout << "Pilihan tidak valid." << endl;
                (*errors)++;
                if (*errors >= 3) {

```

```
        cout << "Anda gagal sebanyak 3 kali, program dihentikan." <<  
endl;  
        exit(0);  
    }  
}  
}
```

Gambar 3.9 fungsi parameter address-of dan dereference

#### 4. Uji Coba dan Hasil Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

PS C:\praktikum-apl> cd "c:\praktikum-apl\post-test\post-test-3\" ; if ($?)
<=====>
|          SELAMAT DATANG DI PROGRAM BANK SEDERHANA          |
<=====>
|          1. Login Sebagai Admin          |
|          2. Login Sebagai User          |
|          0. Keluar                        |
<=====>
Pilih Menu: 1
Masukkan Username: admin
Masukkan Password: admin123
<=====>
|          MENU ADMIN          |
<=====>
|          1. Buat Rekening          |
|          2. Hapus Rekening          |
|          3. Tampilkan Rekening          |
|          4. Update Rekening          |
|          0. Kembali ke Menu Utama          |
<=====>
Pilih Menu: █
```

Gambar 4.1 Login sebagai admin

```
<=====>
|          MENU ADMIN          |
<=====>
|          1. Buat Rekening      |
|          2. Hapus Rekening    |
|          3. Tampilkan Rekening |
|          4. Update Rekening   |
|          0. Kembali ke Menu Utama |
<=====>
Pilih Menu: 1
Masukkan nama nasabah : Ahnap
Masukkan nomor rekening : 1234
Rekening berhasil dibuat.
<=====>
|          MENU ADMIN          |
<=====>
|          1. Buat Rekening      |
|          2. Hapus Rekening    |
|          3. Tampilkan Rekening |
|          4. Update Rekening   |
|          0. Kembali ke Menu Utama |
<=====>
Pilih Menu: 3
<=====>
|          DAFTAR NASABAH          |
<=====>
| 1. Nama : Ahnap | Nomor Rekening : 1234 | Saldo : Rp.0 |
<=====>
```

Gambar 4.2 Membuat rekening dan menampilkan rekening nasabah.

```

<=====
|                                     |
|                                     | MENU ADMIN                             |
|                                     |                                     |
<=====
|                                     |                                     |
|                                     | 1. Buat Rekening                       |
|                                     | 2. Hapus Rekening                     |
|                                     | 3. Tampilkan Rekening                 |
|                                     | 4. Update Rekening                   |
|                                     | 0. Kembali ke Menu Utama             |
|                                     |                                     |
<=====
Pilih Menu: 4
Masukkan nomor rekening yg ingin anda update : 1234
Masukkan nama baru : Melchi
Masukkan nomor rekening baru : 4321
Rekening berhasil diupdate.
<=====
|                                     |
|                                     | MENU ADMIN                             |
|                                     |                                     |
<=====
|                                     |                                     |
|                                     | 1. Buat Rekening                       |
|                                     | 2. Hapus Rekening                     |
|                                     | 3. Tampilkan Rekening                 |
|                                     | 4. Update Rekening                   |
|                                     | 0. Kembali ke Menu Utama             |
|                                     |                                     |
<=====
Pilih Menu: 3
<=====
|                                     |
|                                     | DAFTAR NASABAH                         |
|                                     |                                     |
<=====
| 1. Nama : Melchi | Nomor Rekening : 4321 | Saldo : Rp.0 |
|                                     |                                     |
<=====
|                                     |
|                                     | MENU ADMIN                             |
|                                     |                                     |
<=====
|                                     |                                     |
|                                     | 1. Buat Rekening                       |
|                                     | 2. Hapus Rekening                     |
|                                     | 3. Tampilkan Rekening                 |
|                                     | 4. Update Rekening                   |
|                                     | 0. Kembali ke Menu Utama             |
|                                     |                                     |
<=====
Pilih Menu: █

```

Gambar 4.3 Update rekening dan menampilkan daftar rekening

```
<=====>
Pilih Menu: 0
<=====>
|          SELAMAT DATANG DI PROGRAM BANK SEDERHANA          |
<=====>
|          1. Login Sebagai Admin          |
|          2. Login Sebagai User          |
|          0. Keluar          |
<=====>
Pilih Menu: 2
Masukkan username : user
Masukkan password : user123
<=====>
|          MENU USER          |
<=====>
|          1. Cek Saldo          |
|          2. isi Saldo          |
|          3. Tarik Saldo          |
|          0. Kembali Ke Menu Utama          |
<=====>
Pilih Menu User : |
```

Gambar 4.4 Kembali ke menu utama, login sebagai user

```
<=====>
|          MENU USER          |
<=====>
|          1. Cek Saldo          |
|          2. isi Saldo          |
|          3. Tarik Saldo          |
|          0. Kembali Ke Menu Utama          |
<=====>
Pilih Menu User :2
Masukkan nomor rekening : 4321
Masukkan jumlah saldo yg ingin diisi : Rp.55000
Saldo berhasil diisi! Saldo anda sekarang : Rp.55000
<=====>
|          MENU USER          |
<=====>
|          1. Cek Saldo          |
|          2. isi Saldo          |
|          3. Tarik Saldo          |
|          0. Kembali Ke Menu Utama          |
<=====>
```

Gambar 4.5 Isi saldo



```
<=====>
|                                     |
|                                     | MENU USER                             |
|                                     |                                     |
<=====>
|                                     |                                     |
|                                     | 1. Cek Saldo                         |
|                                     | 2. isi Saldo                        |
|                                     | 3. Tarik Saldo                      |
|                                     | 0. Kembali Ke Menu Utama           |
|                                     |                                     |
<=====>
Pilih Menu User :1
Masukkan nomor rekening : 3
Nomor rekening tidak ditemukan.
```

Gambar 4.6 Tidak menemukan nomor rekening yang tidak terdaftar.

```
<=====>
Pilih Menu User :3
Masukkan nomor rekening :4321
Masukkan jumlah saldo yg ingin anda tarik : Rp.56000
Saldo tidak cukup.
<=====>
|                                     |
|                                     | MENU USER                             |
|                                     |                                     |
<=====>
|                                     |                                     |
|                                     | 1. Cek Saldo                         |
|                                     | 2. isi Saldo                        |
|                                     | 3. Tarik Saldo                      |
|                                     | 0. Kembali Ke Menu Utama           |
|                                     |                                     |
<=====>
```

Gambar 4.7 Tidak dapat ditarik, karena nominal melebihi saldo yang ada direkening.

```
<=====>
Pilih Menu User :3
Masukkan nomor rekening :4321
Masukkan jumlah saldo yg ingin anda tarik : Rp.50000
Saldo berhasil ditarik! Saldo anda sekarang : Rp.5000
<=====>
|                                     |
|                                     | MENU USER                             |
|                                     |                                     |
<=====>
|                                     |                                     |
|                                     | 1. Cek Saldo                         |
|                                     | 2. isi Saldo                        |
|                                     | 3. Tarik Saldo                      |
|                                     | 0. Kembali Ke Menu Utama           |
|                                     |                                     |
<=====>
Pilih Menu User :█
```

Gambar 4.8 Tarik saldo berhasil.

```

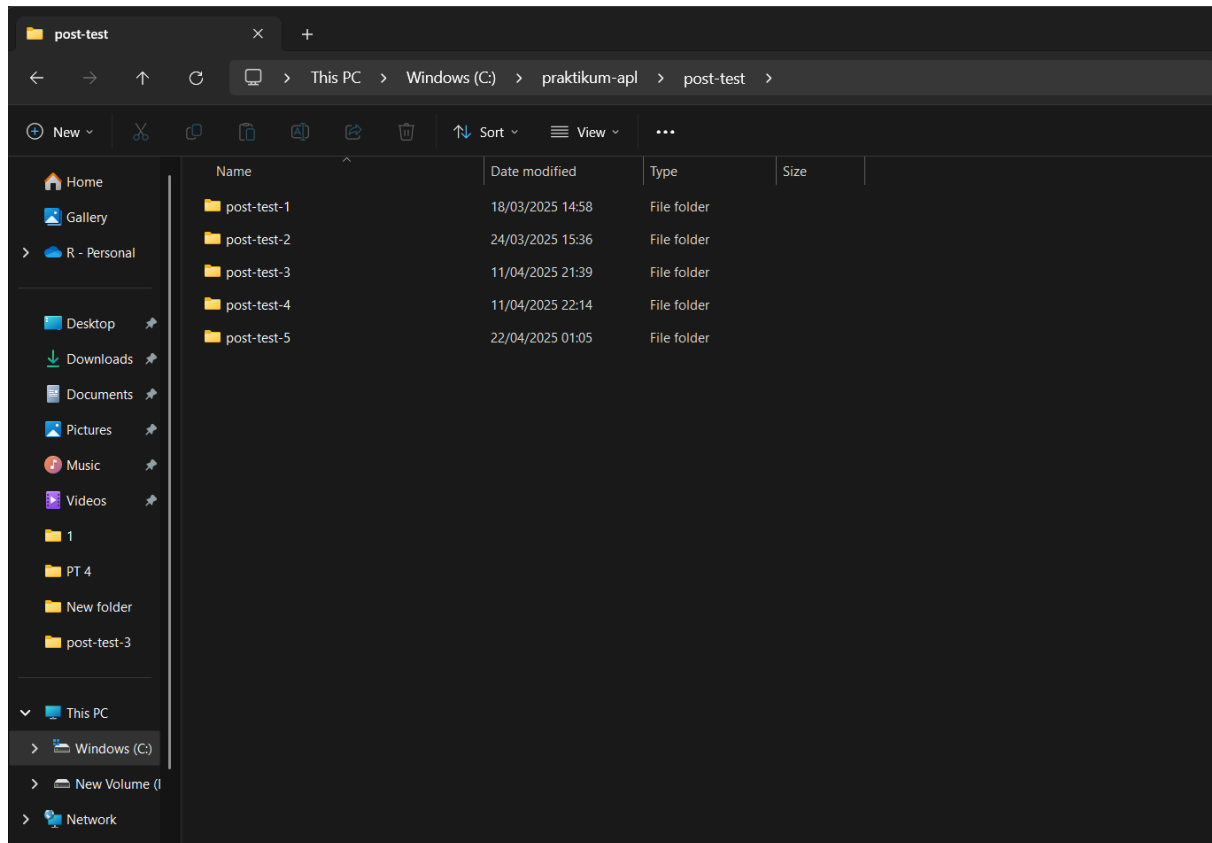
<=====>
|                                |
|                                | MENU USER                                |
|                                |                                |
<=====>
|                                |                                |
|                                | 1. Cek Saldo                    |
|                                | 2. isi Saldo                   |
|                                | 3. Tarik Saldo                 |
|                                | 0. Kembali Ke Menu Utama      |
|                                |                                |
<=====>
Pilih Menu User :0
<=====>
|                                |                                |
|                                | SELAMAT DATANG DI PROGRAM BANK Sederhana |
|                                |                                |
<=====>
|                                |                                |
|                                | 1. Login Sebagai Admin        |
|                                | 2. Login Sebagai User        |
|                                | 0. Keluar                    |
|                                |                                |
<=====>
Pilih Menu: 0
Terimakasih telah menggunakan program ini.

```

Gambar 4.9 Keluar dari menu user dan program (berhenti dari program)

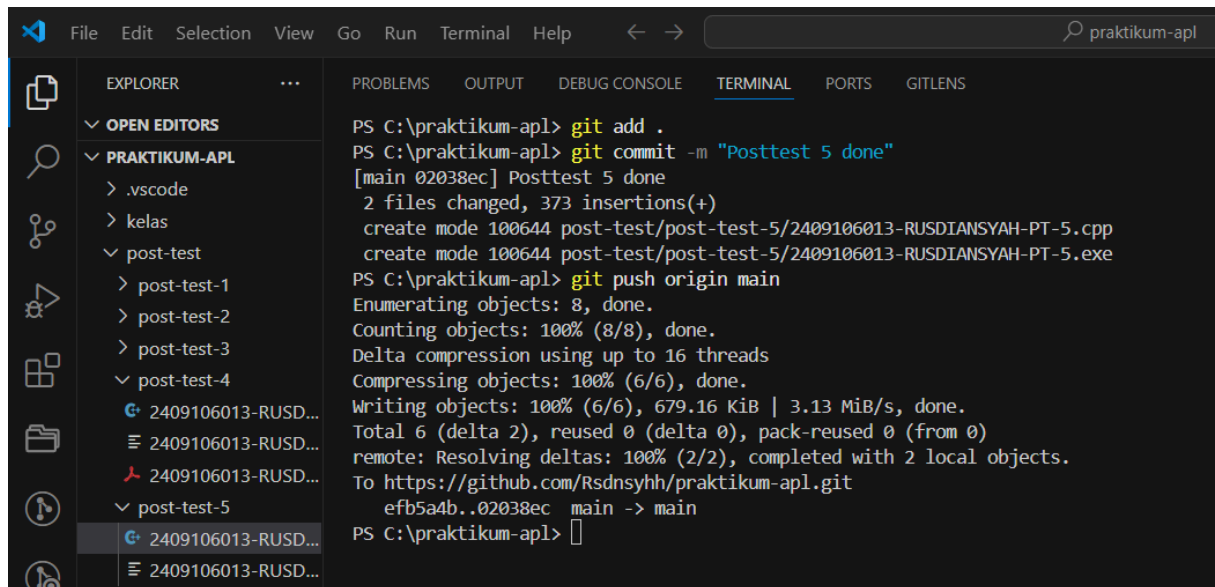
## 5.Langkah-Langkah Git pada VSCode.

- a) Membuat folder praktikum-apl,kemudian didalam folder praktikum-apl buat folder kelas dan post-test,dan yang terakhir didalam file post-test buat lagi folder post-test-5.



Gambar 5.1 Membuat folder post-test-5.

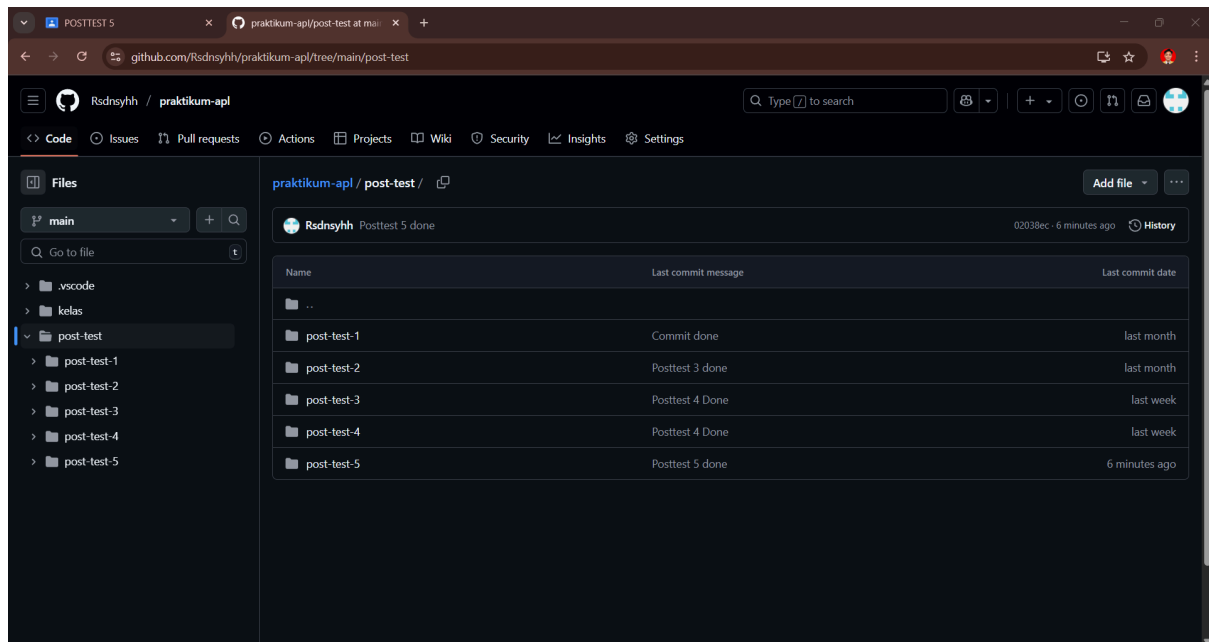
- b) Buka VS Code kemudian buka terminal. Git init merupakan perintah untuk menginisialisasi repositori Git baru di direktori proyek user. Git add merupakan perintah digunakan untuk menambahkan file ke staging area (area persiapan sebelum perubahan tersebut di commit). Commit adalah perintah yang digunakan untuk menyimpan perubahan yang telah di staging ke dalam riwayat repositori Git. Git push adalah perintah yang digunakan untuk mengirim perubahan yang telah di commit di repositori lokal ke repositori remote.

The image is a screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows the Explorer view with a project named 'PRAKTIKUM-APL' containing folders like '.vscode', 'kelas', 'post-test', and several files. The main area is the Terminal, which shows the execution of several Git commands. The commands and their outputs are as follows:

```
PS C:\praktikum-apl> git add .
PS C:\praktikum-apl> git commit -m "Posttest 5 done"
[main 02038ec] Posttest 5 done
2 files changed, 373 insertions(+)
create mode 100644 post-test/post-test-5/2409106013-RUSDIANSYAH-PT-5.cpp
create mode 100644 post-test/post-test-5/2409106013-RUSDIANSYAH-PT-5.exe
PS C:\praktikum-apl> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 679.16 KiB | 3.13 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Rsdnsyhh/praktikum-apl.git
 efb5a4b..02038ec  main -> main
PS C:\praktikum-apl>
```

Gambar 5.2 Git init, Git add, commit, dan Git push

c) Lakukan refresh atau reload pada tab browser atau chrome Github yang digunakan kemudian selesai.



Gambar 5.3 Refresh tab Github