

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

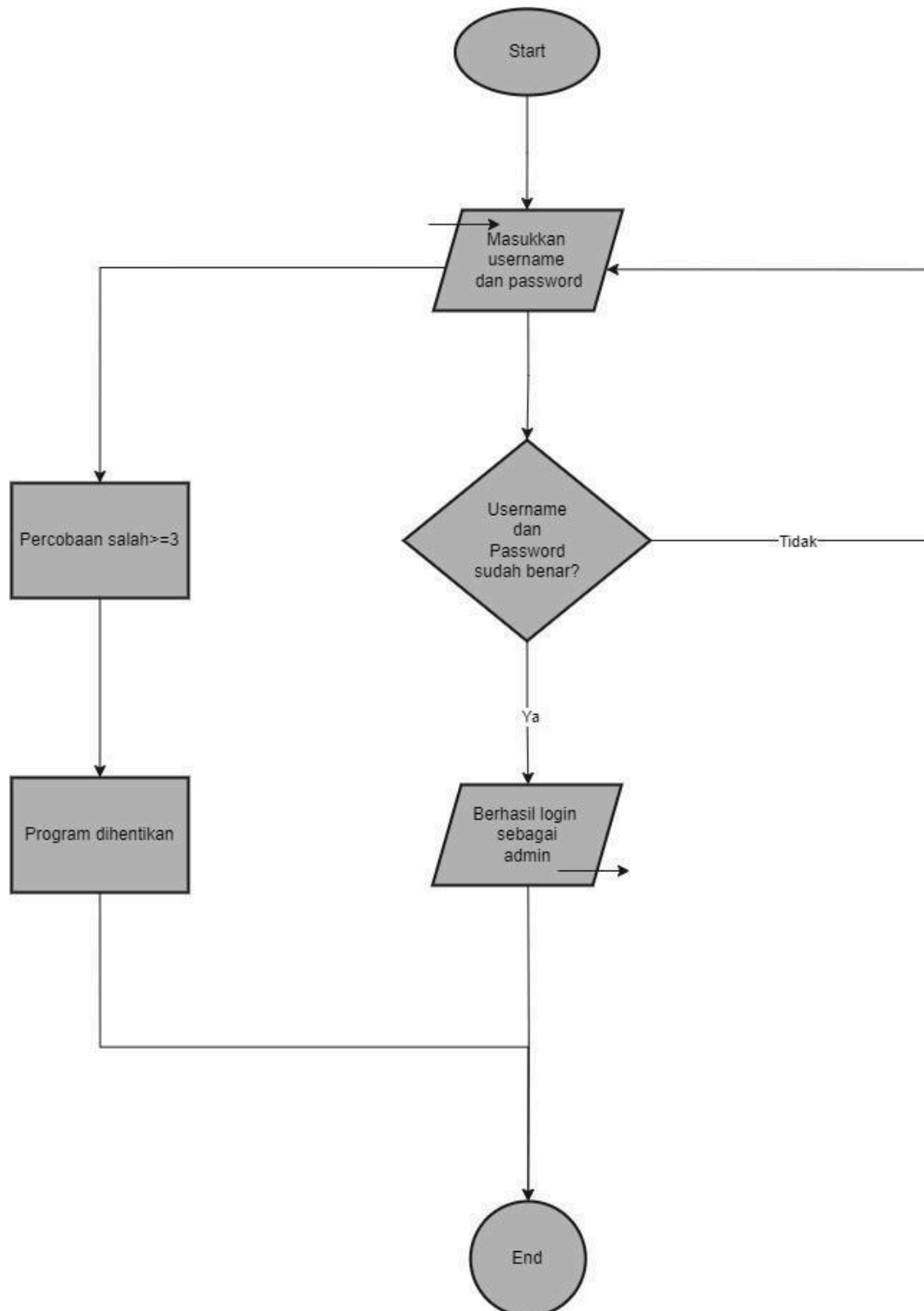


Disusun oleh:
Rusdiansyah(2409106013)
Kelas (A1 '24)

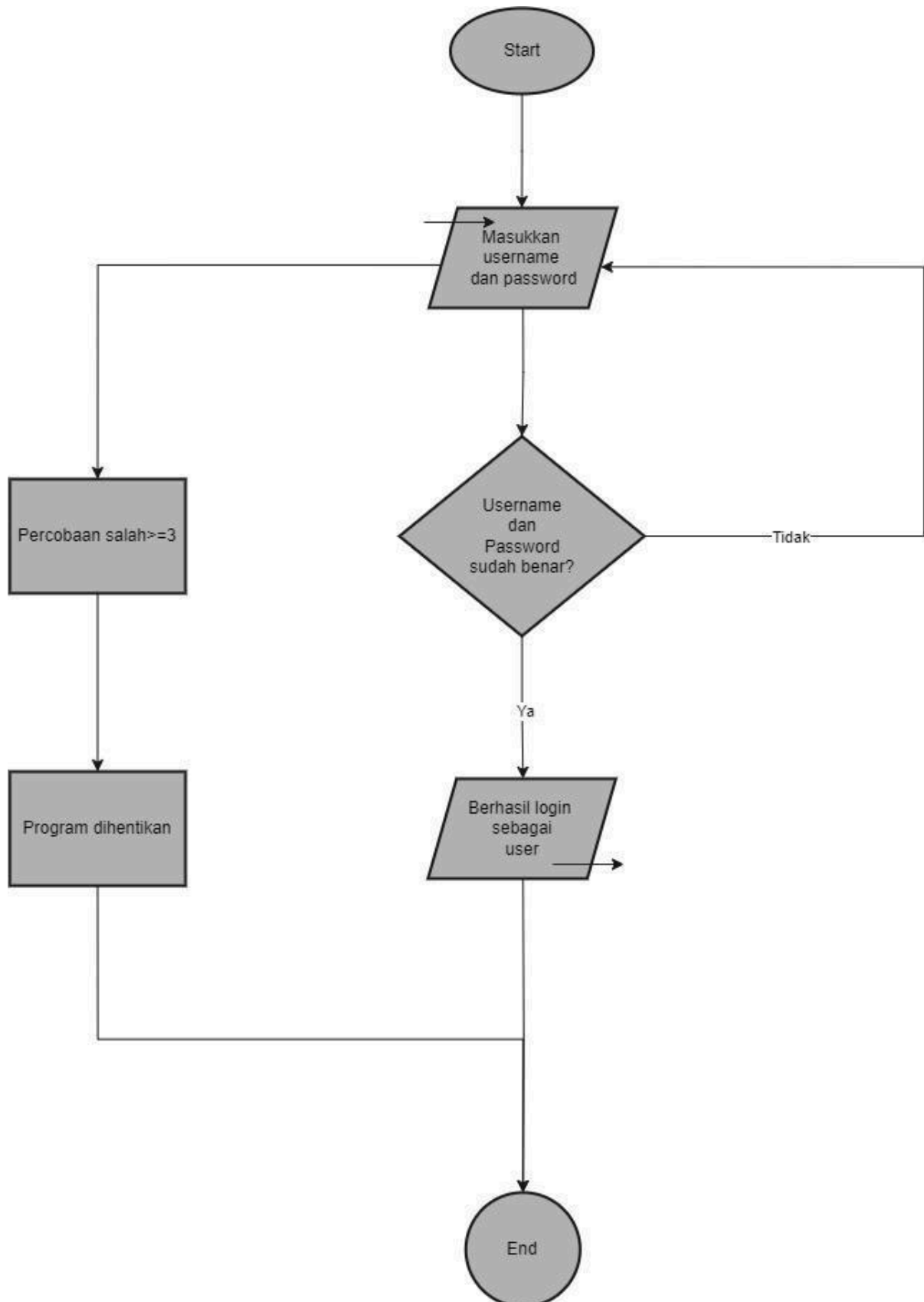
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

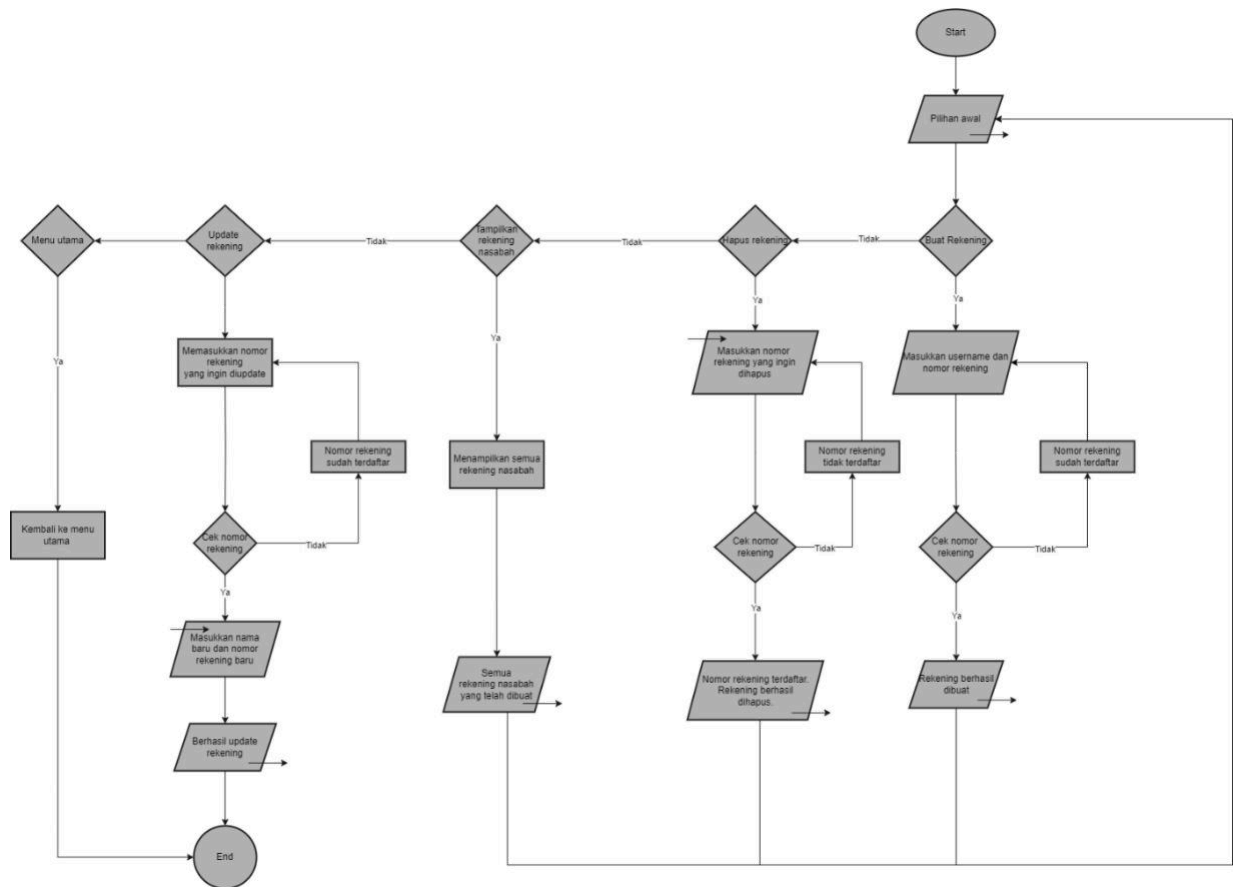
1.Flowchart



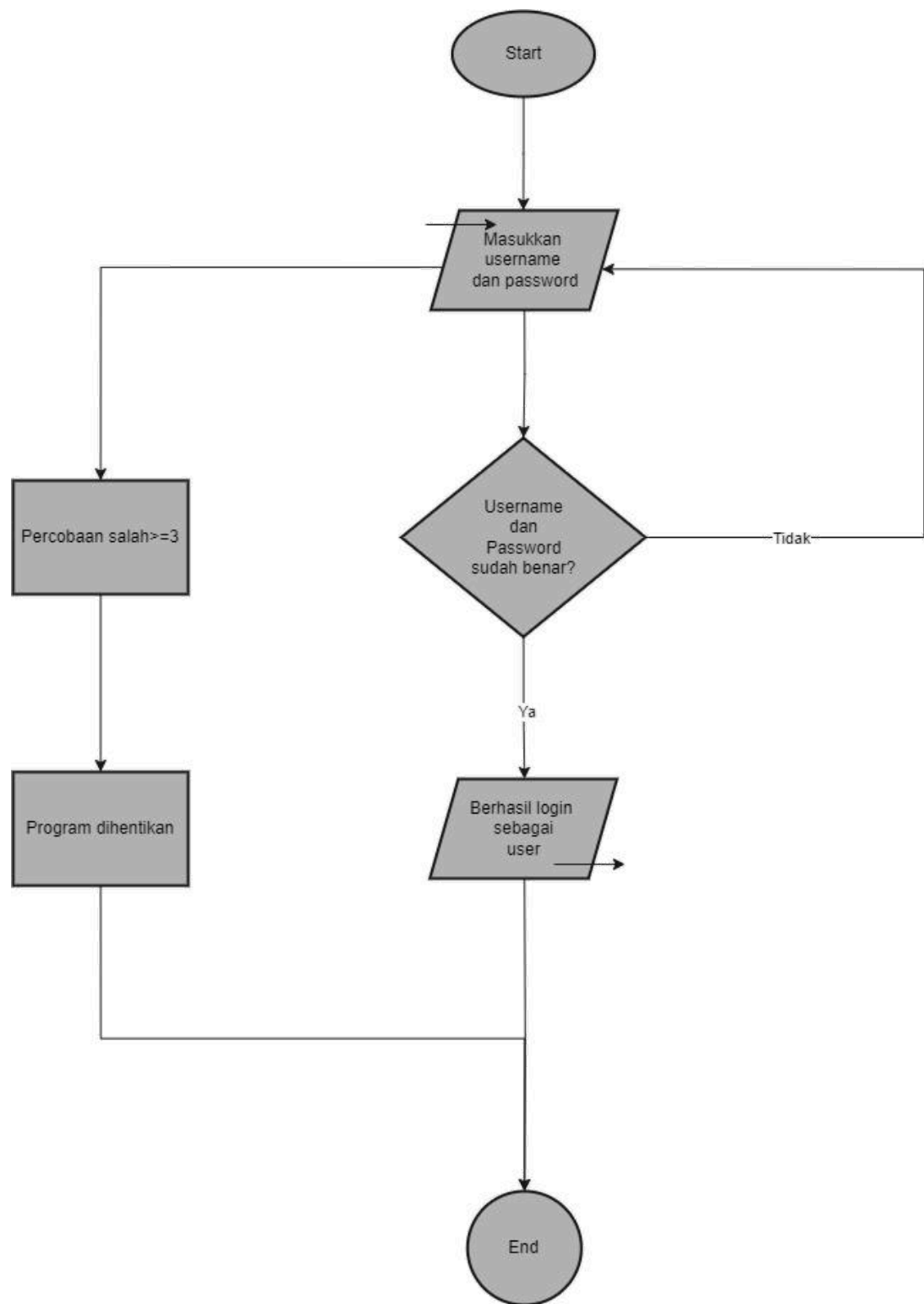
Gambar 1.1 Menu utama



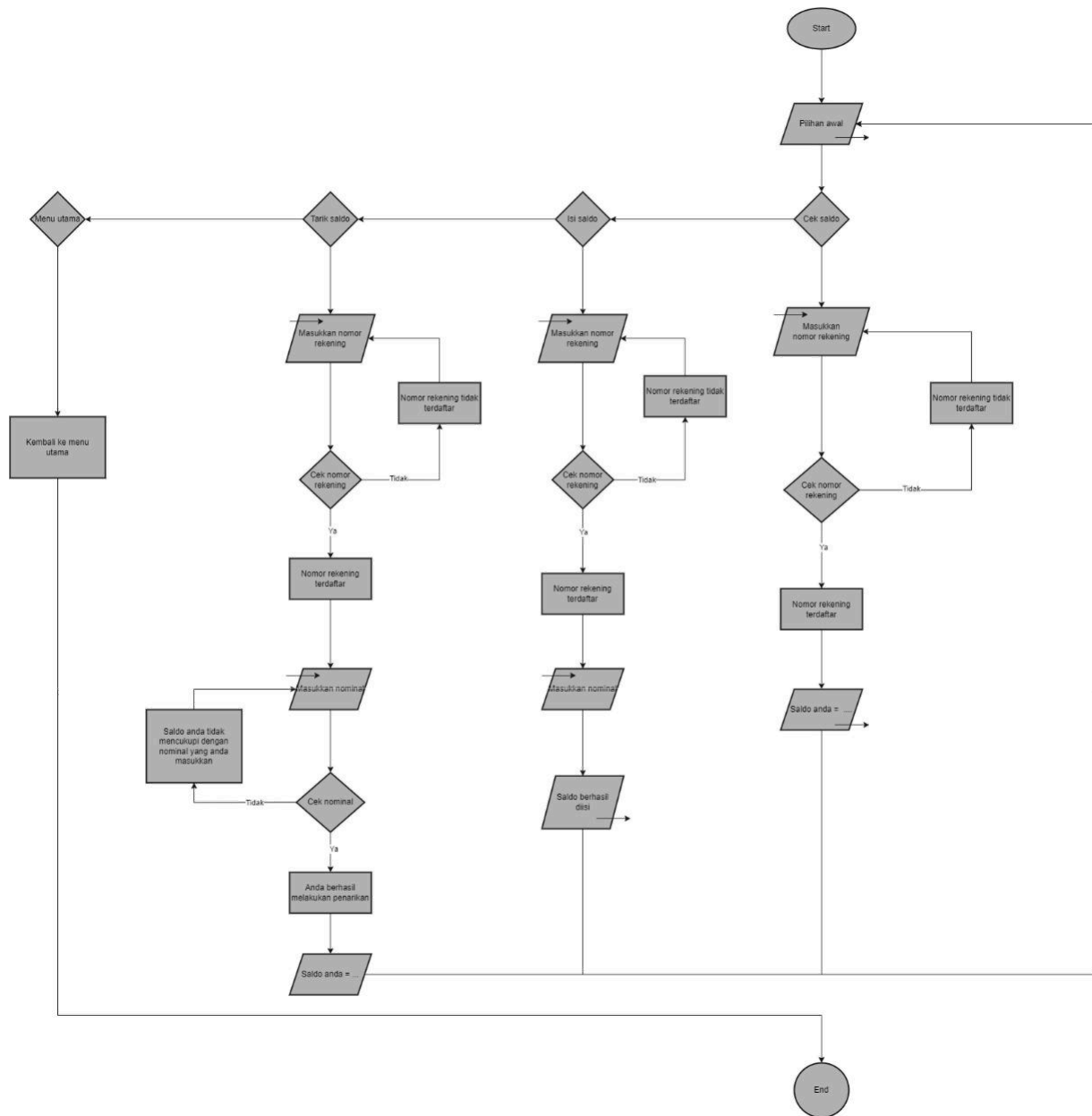
Gambar 1.2 Login sebagai admin



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.

Sorting, dalam bahasa Indonesia disebut pengurutan, adalah proses mengatur elemen-elemen dalam sebuah himpunan data ke dalam urutan tertentu, seperti dari terkecil ke terbesar (ascending) atau sebaliknya (descending), atau berdasarkan urutan alfabetik. Proses ini mengubah data yang acak menjadi teratur sesuai dengan kriteria tertentu.

Pada posttest kali ini, praktikan diminta untuk menambahkan metode sorting pada program posttest sebelumnya.

3.Source Code

1)

a.Deklarasi fungsi Sorting

```
// Deklarasi fungsi sorting
void bubbleSortNama(Nasabah arr[], int n); // Sorting nama ascending (Bubble Sort)
void selectionSortSaldo(Nasabah arr[], int n); // Sorting saldo descending (Selection Sort)
void insertionSortNomorRekening(Nasabah arr[], int n); // Sorting nomor rekening ascending (Insertion Sort)

void buatRekening(Nasabah* data, int* size);
void hapusRekening(Nasabah* data, int* size);
void tampilkanRekening(Nasabah* data, int size, int sortMethod = 0);
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 Sorting

b. 3 Metode sorting (Bubble Sort, Selection Sort, Insertion Sort)

```
// Implementasi Bubble Sort untuk sorting nama (ascending)
void bubbleSortNama(Nasabah arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j].nama > arr[j+1].nama) {
                swap(arr[j], arr[j+1]);
            }
        }
    }
}

// Implementasi Selection Sort untuk sorting saldo (descending)
void selectionSortSaldo(Nasabah arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        int max_idx = i;
        for (int j = i+1; j < n; j++) {
            if (arr[j].saldo > arr[max_idx].saldo) {
                max_idx = j;
            }
        }
        swap(arr[max_idx], arr[i]);
    }
}

// Implementasi Insertion Sort untuk sorting nomor rekening (ascending)
void insertionSortNomorRekening(Nasabah arr[], int n) {
```

```

    for (int i = 1; i < n; i++) {
        Nasabah key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j].nomorRekening > key.nomorRekening) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

```

Gambar 3.2 Metode sorting (Bubble Sort, Selection Sort, Insertion Sort)

2) Create

```

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 3.3 Membuat rekening

3) Read

```
void tampilkanRekening(Nasabah* data, int size, int sortMethod) {
    if (size == 0) {
        cout << "Belum ada rekening yang terdaftar." << endl;
    } else {
        // Membuat salinan data untuk diurutkan
        Nasabah sortedData[MAX_NASABAH];
        for (int i = 0; i < size; i++) {
            sortedData[i] = data[i];
        }

        // Metode sorting
        switch (sortMethod) {
            case 1: // Bubble Sort by Nama (A-Z)
                bubbleSortNama(sortedData, size);
                break;
            case 2: // Selection Sort by Saldo (Descending)
                selectionSortSaldo(sortedData, size);
                break;
            case 3: // Insertion Sort by Nomor Rekening (Ascending)
                insertionSortNomorRekening(sortedData, size);
                break;
            default: // Tidak diurutkan (sortMethod = 0)
                break;
        }

        cout <<
        "<=====>" <<
        endl;

        cout << "          |                      DAFTAR NASABAH"
        |
        " << endl;
        cout <<
        "<=====>" <<
        endl;

        for (int i = 0; i < size; i++) {
            cout << "          | " << i + 1 << ". Nama : " << sortedData[i].nama
            << " | Nomor Rekening : " << sortedData[i].nomorRekening
            << " | Saldo : Rp." << sortedData[i].saldo << endl;
        }
        cout <<
        "<=====>" << endl;
    }
}
```

Gambar 3.4 Menampilkan daftar rekening

```

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 3.5 Cek saldo dalam rekening

4) Update

```
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 3.6 Mengupdate rekening

```

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 3.7 Isi saldo

```

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 Menarik saldo

5) Delete

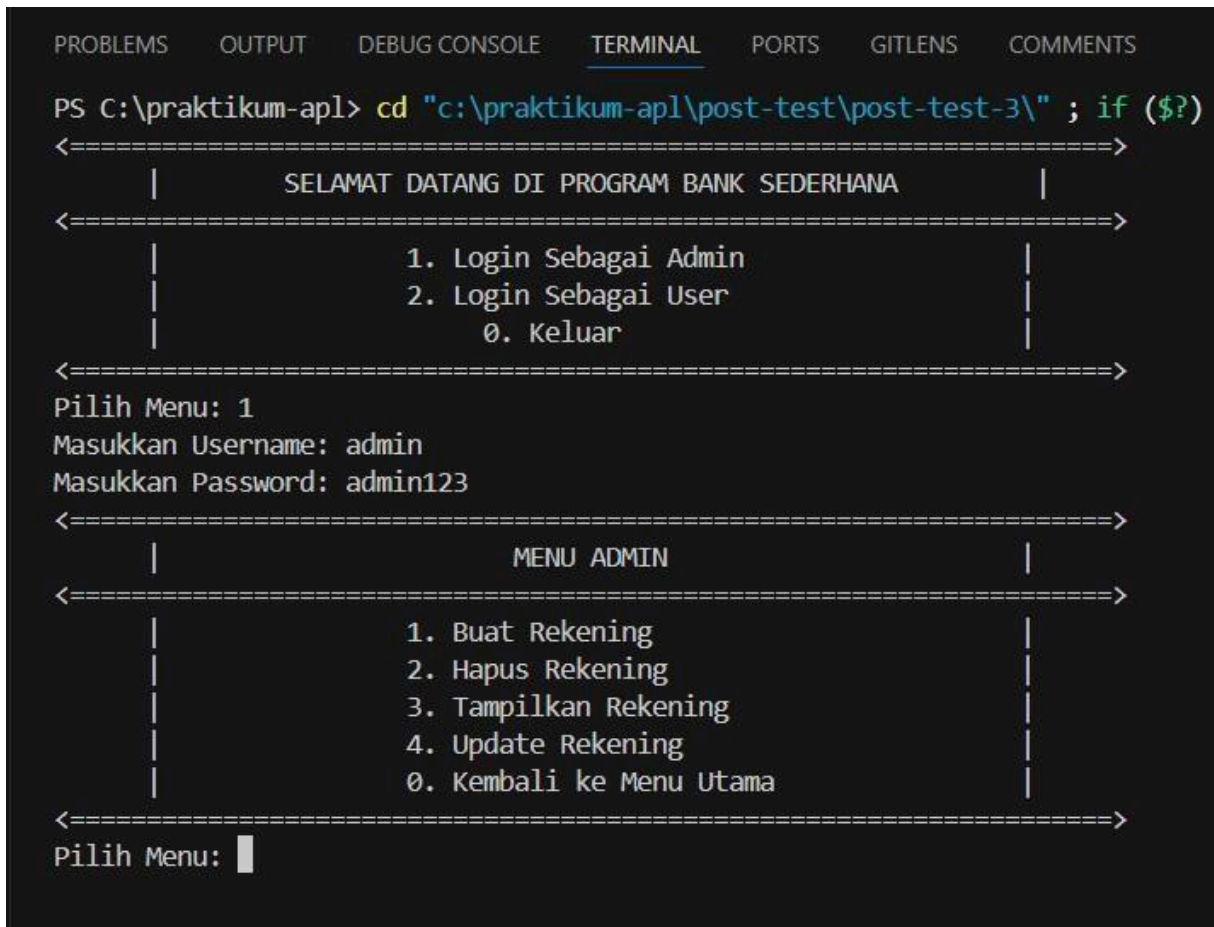
```
void hapusRekening(Nasabah* data, int* size) {
    if (*size == 0) {
        cout << "Belum ada rekening yang terdaftar" << endl;
    } else {
        string nomorRekening;
        cout << "Masukkan nomor rekening yang ingin dihapus: " ;
        cin >> nomorRekening;

        bool ditemukan = false;
        for (int i = 0; i < *size; i++) {
            if (data[i].nomorRekening == nomorRekening) {
                for (int j = i; j < *size - 1; j++) {
                    data[j] = data[j + 1];
                }
                (*size)--;
                ditemukan = true;
                cout << "Rekening berhasil dihapus." << endl;
                break;
            }
        }

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

Gambar 3.9 Menghapus rekening

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 ADMIN           |
<=====>
|           1. Buat Rekening      |
|           2. Hapus Rekening    |
|           3. Tampilkan Rekening|
|           4. Update Rekening   |
|           0. Kembali ke Menu Utama|
<=====>
Pilih Menu: 3
Pilih yang akan disorting (Pilih 0: Tidak sorting, Pilih 1: Nama A-Z, Pilih 2: Saldo, Pilih 3: No.Rek): 1
<=====>
|           DAFTAR NASABAH       |
<=====>
| 1. Nama : Ahnap | Nomor Rekening : 1234 | Saldo : Rp.50000 |
| 2. Nama : Melchi | Nomor Rekening : 4321 | Saldo : Rp.1000000 |
| 3. Nama : Rusdy | Nomor Rekening : 3456 | Saldo : Rp.5000 |
| 4. Nama : Zifa | Nomor Rekening : 2345 | Saldo : Rp.19000000 |
<=====>
```

Gambar 4.9 Sorting nama (A-Z) menggunakan Bubble Sort

```
<=====>
|           MENU ADMIN           |
<=====>
|           1. Buat Rekening      |
|           2. Hapus Rekening    |
|           3. Tampilkan Rekening|
|           4. Update Rekening   |
|           0. Kembali ke Menu Utama|
<=====>
Pilih Menu: 3
Pilih yang akan disorting (Pilih 0: Tidak sorting, Pilih 1: Nama A-Z, Pilih 2: Saldo, Pilih 3: No.Rek): 2
<=====>
|           DAFTAR NASABAH       |
<=====>
| 1. Nama : Zifa | Nomor Rekening : 2345 | Saldo : Rp.19000000 |
| 2. Nama : Melchi | Nomor Rekening : 4321 | Saldo : Rp.1000000 |
| 3. Nama : Ahnap | Nomor Rekening : 1234 | Saldo : Rp.50000 |
| 4. Nama : Rusdy | Nomor Rekening : 3456 | Saldo : Rp.5000 |
<=====>
```

Gambar 4.10 Sorting saldo (Descending) menggunakan Selection Sort

```
<=====>
|           MENU ADMIN           |
<=====>
|           1. Buat Rekening      |
|           2. Hapus Rekening    |
|           3. Tampilkan Rekening|
|           4. Update Rekening   |
|           0. Kembali ke Menu Utama|
<=====>
Pilih Menu: 3
Pilih yang akan disorting (Pilih 0: Tidak sorting, Pilih 1: Nama A-Z, Pilih 2: Saldo, Pilih 3: No.Rek): 3
<=====>
|           DAFTAR NASABAH       |
<=====>
| 1. Nama : Ahnap | Nomor Rekening : 1234 | Saldo : Rp.50000
| 2. Nama : Zifa  | Nomor Rekening : 2345 | Saldo : Rp.19000000
| 3. Nama : Rusdy | Nomor Rekening : 3456 | Saldo : Rp.5000
| 4. Nama : Melchi| Nomor Rekening : 4321 | Saldo : Rp.1000000
<=====>
```

Gambar 4.11 Sorting nomor rekening (Ascending) menggunakan Insertion Sort

```
<=====>
|           MENU ADMIN           |
<=====>
|           1. Buat Rekening      |
|           2. Hapus Rekening    |
|           3. Tampilkan Rekening|
|           4. Update Rekening   |
|           0. Kembali ke Menu Utama|
<=====>
Pilih Menu: 3
Pilih yang akan disorting (Pilih 0: Tidak sorting, Pilih 1: Nama A-Z, Pilih 2: Saldo, Pilih 3: No.Rek): 0
<=====>
|           DAFTAR NASABAH       |
<=====>
| 1. Nama : Ahnap | Nomor Rekening : 1234 | Saldo : Rp.50000
| 2. Nama : Melchi| Nomor Rekening : 4321 | Saldo : Rp.1000000
| 3. Nama : Zifa  | Nomor Rekening : 2345 | Saldo : Rp.19000000
| 4. Nama : Rusdy | Nomor Rekening : 3456 | Saldo : Rp.5000
<=====>
```

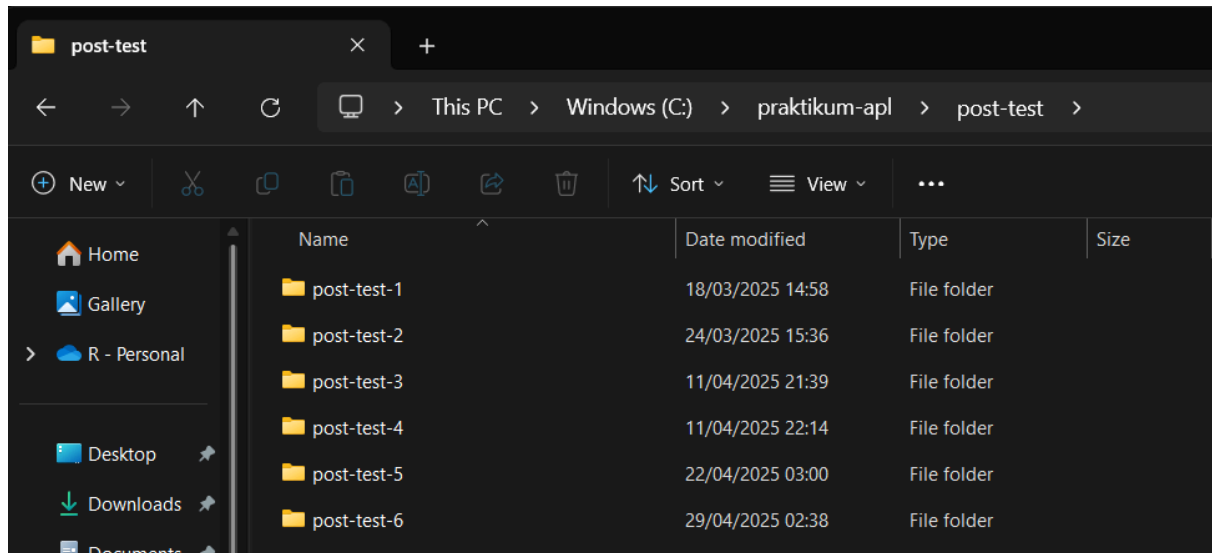
Gambar 4.12 Tidak melakukan sorting


```
<=====>
|                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 4.13 Keluar dari menu user dan program (berhenti dari program)

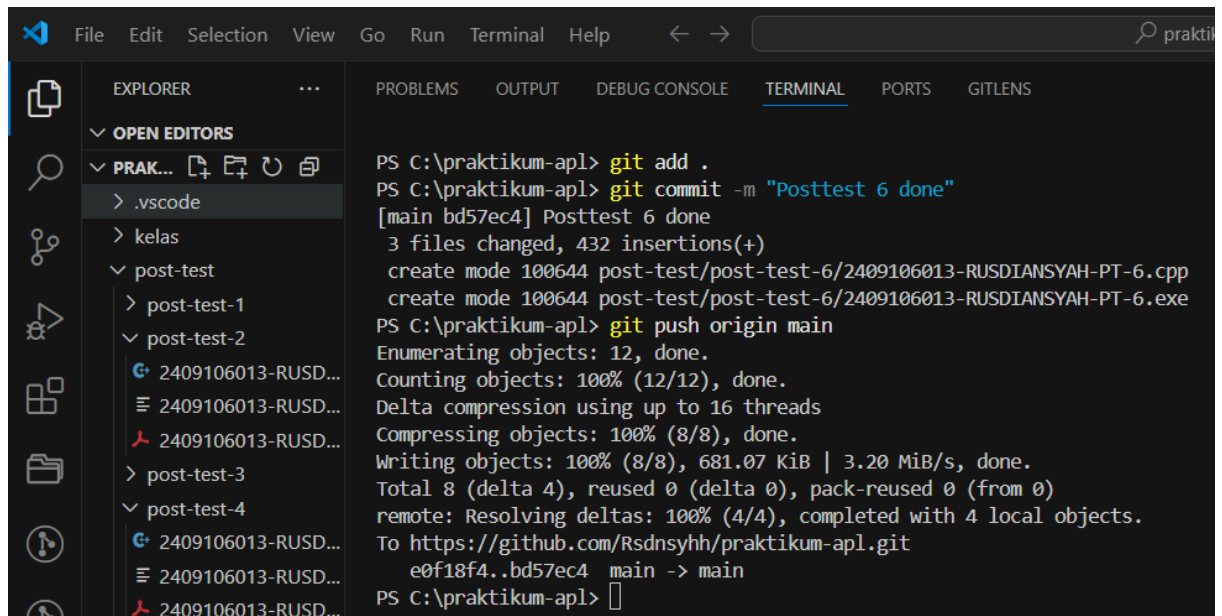
5.Langkah-Langkah Git pada VSCode.

- 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-6.



Gambar 5.1 Membuat folder post-test-6.

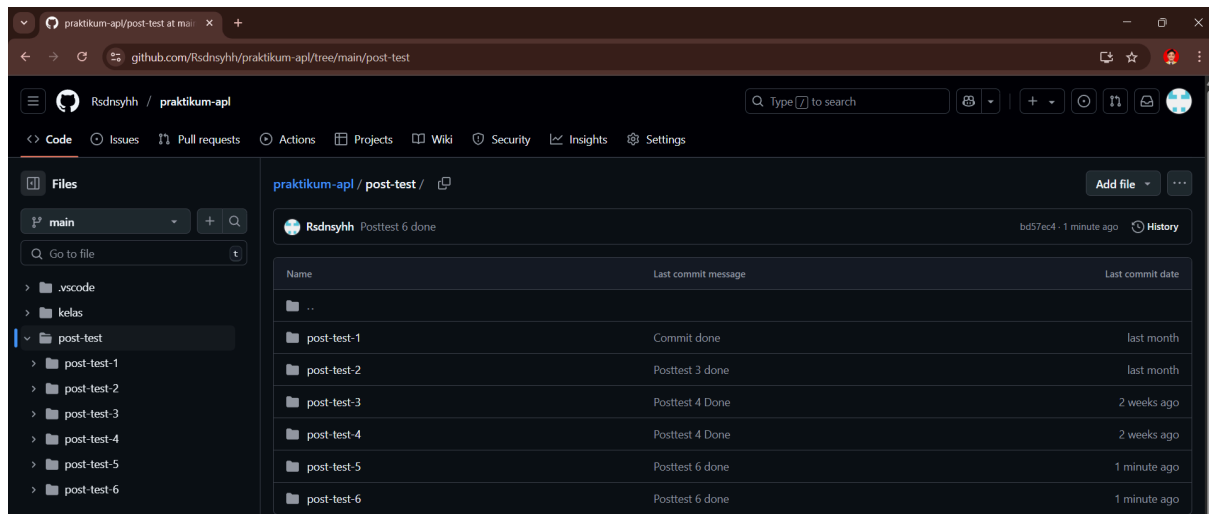
- b) Buka VS Code kemudian buka terminal. Git init merupakan perintah untuk menginisialisasi repositori Git baru di direktori proyek user. Git add merupakan perintah digunakan untuk menambahkan file ke staging area (area persiapan sebelum perubahan tersebut di commit). Commit adalah perintah yang digunakan 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 file tree containing folders like '.vscode', 'kelas', 'post-test', and 'post-test-1' through 'post-test-4'. The main area is the Terminal, which shows the output of several Git commands executed in a PowerShell prompt at 'PS C:\praktikum-apl>'. The commands and their outputs are: 1. 'git add .' followed by 'git commit -m "Posttest 6 done"', resulting in a commit on the 'main' branch with hash 'bd57ec4'. 2. 'git push origin main', which shows progress for enumerating, counting, compressing, and writing objects, and finally pushing to the remote repository at 'https://github.com/Rsdnsyhh/praktikum-apl.git'. The terminal output is as follows:

```
PS C:\praktikum-apl> git add .
PS C:\praktikum-apl> git commit -m "Posttest 6 done"
[main bd57ec4] Posttest 6 done
3 files changed, 432 insertions(+)
create mode 100644 post-test/post-test-6/2409106013-RUSDIANSYAH-PT-6.cpp
create mode 100644 post-test/post-test-6/2409106013-RUSDIANSYAH-PT-6.exe
PS C:\praktikum-apl> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 681.07 KiB | 3.20 MiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/Rsdnsyhh/praktikum-apl.git
   e0f18f4..bd57ec4  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