

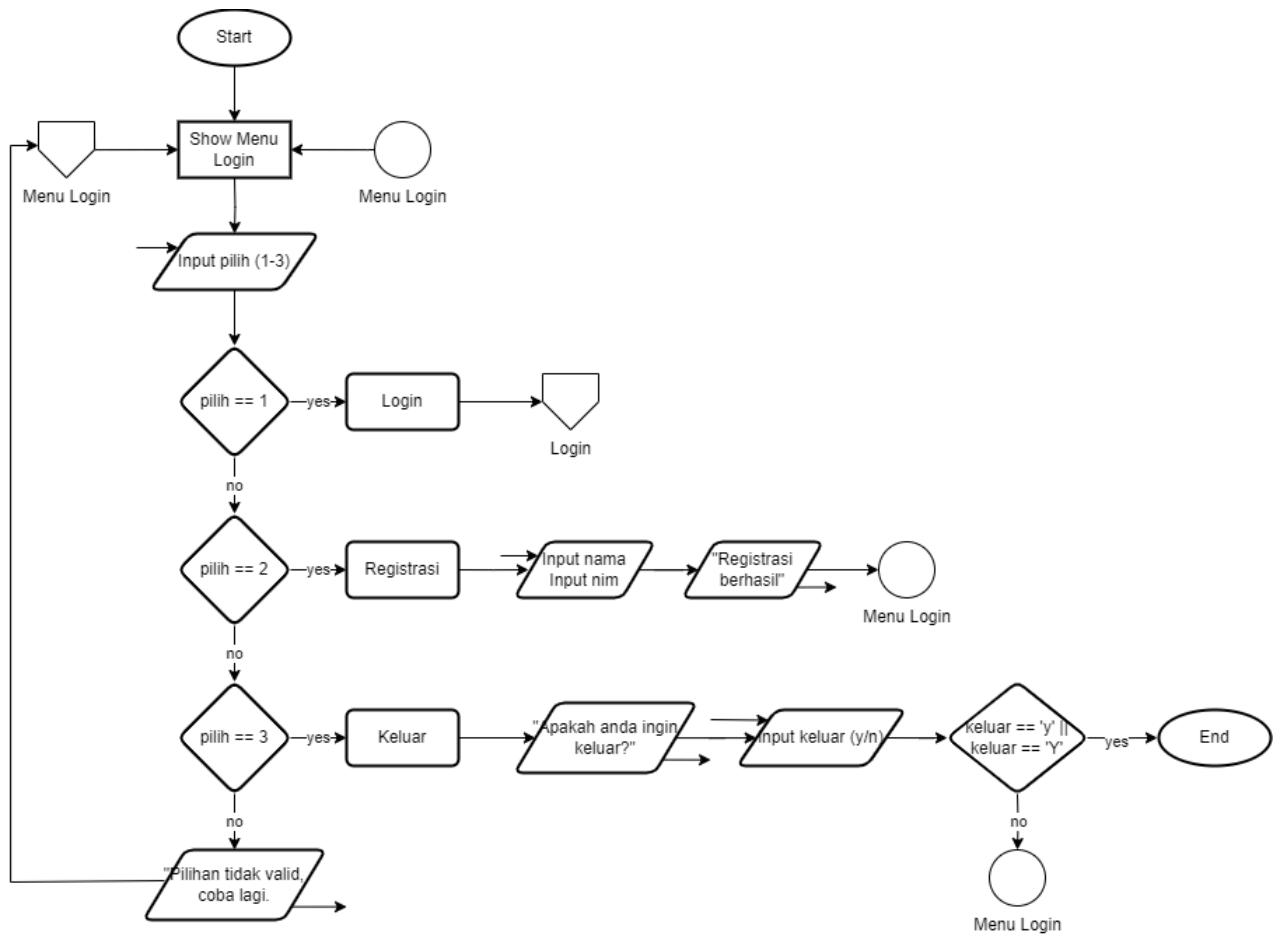
LAPORAN PRAKTIKUM
POSTTEST (5)
ALGORITMA PEMROGRAMAN LANJUT



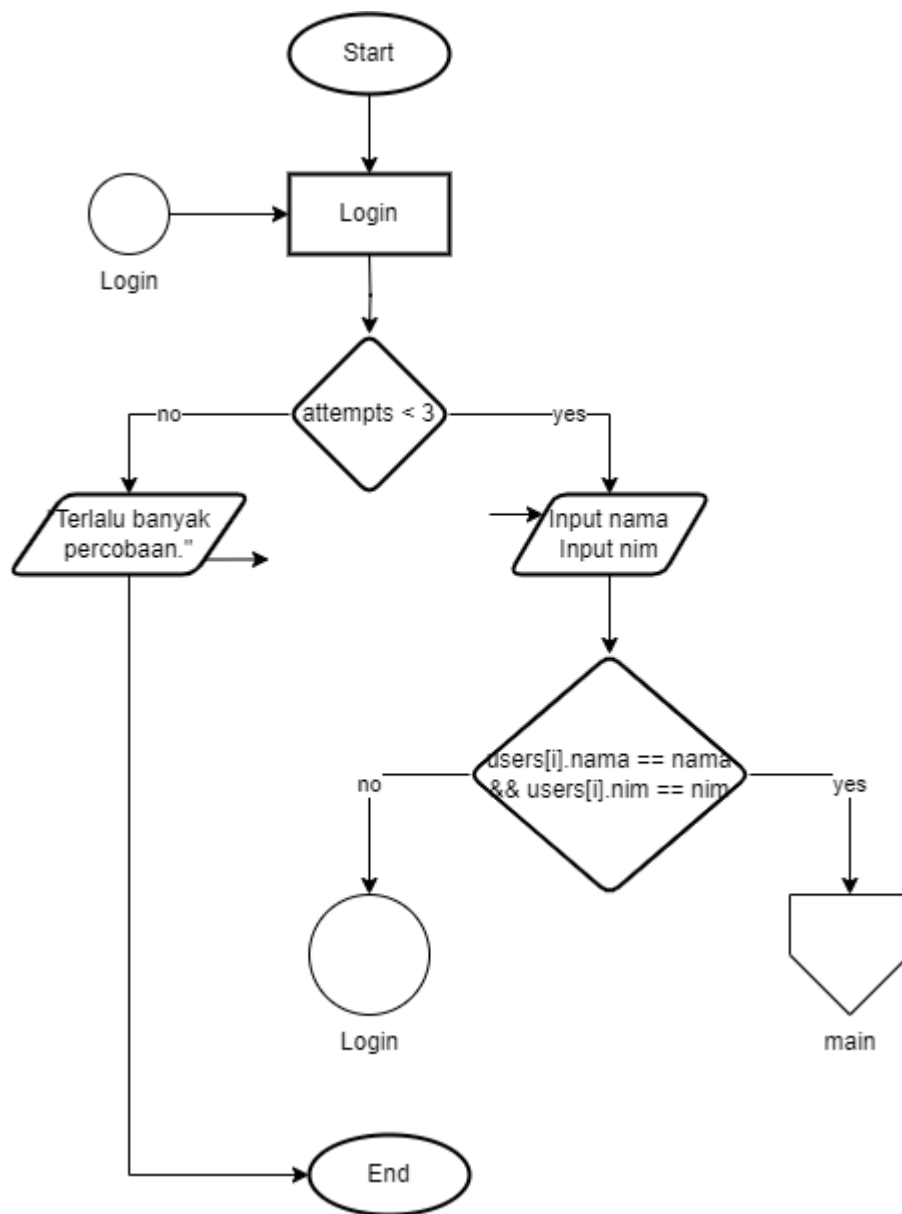
Disusun oleh:
Azira Faradina (2409106016)
Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

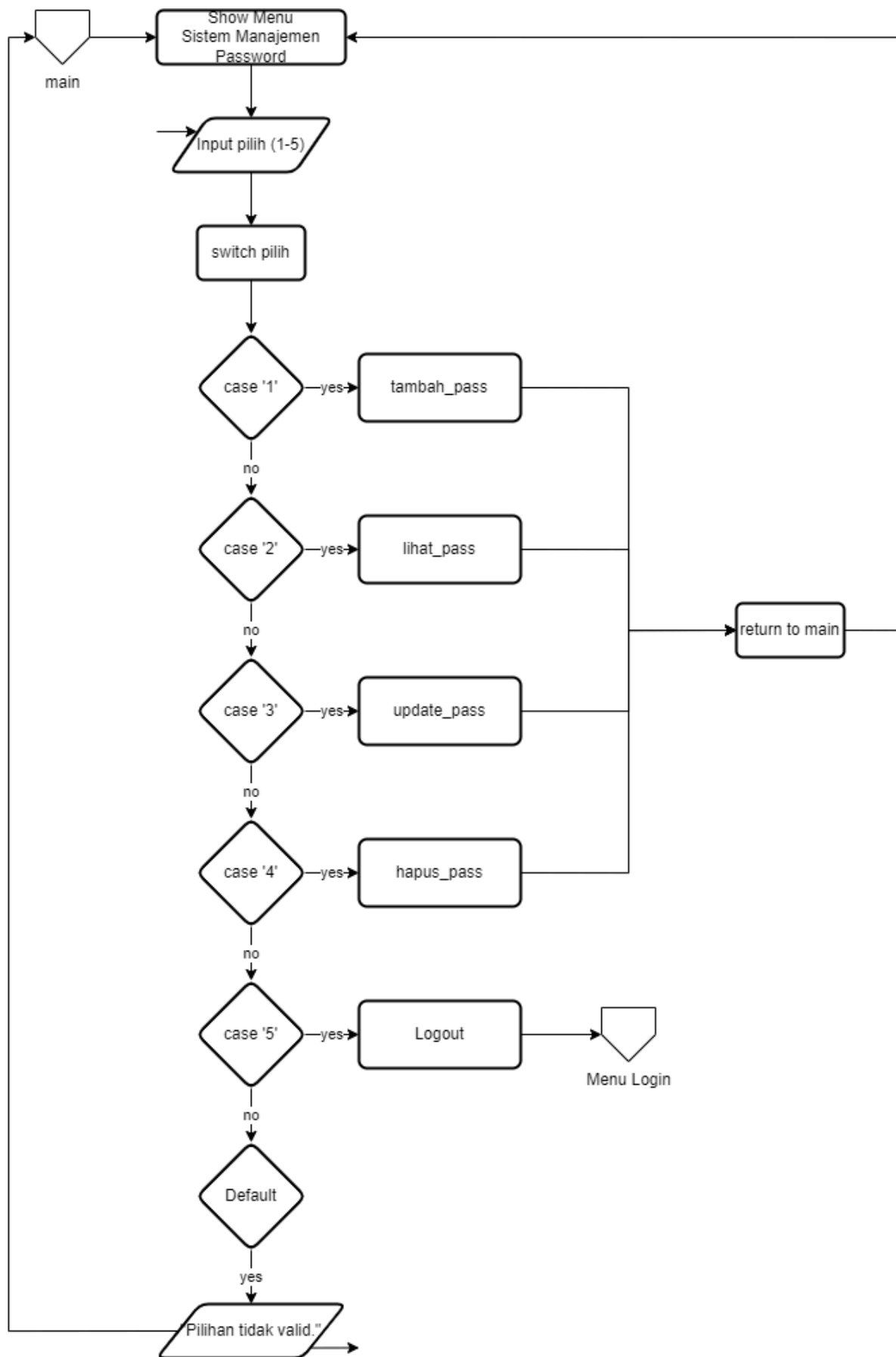
1. Flowchart



Gambar 1.1 Flowchart Menu Login



Gambar 1.2 Flowchart Fitur Login



Gambar 1.3 Flowchart Menu Utama

2. Analisis Program

Program Manajemen Password Sederhana ini adalah program yang memungkinkan pengguna untuk menyimpan, melihat, memperbarui, ataupun menghapus password yang digunakan untuk berbagai akun website atau aplikasi. Adapun manfaat dari program ini adalah pengguna tidak perlu kesulitan untuk menghafal banyak username dan password akun yang telah dibuat sebelumnya. Pada Post Test kali ini, praktikan diminta untuk menerapkan prinsip pointer dan menggunakan fungsi dengan parameter address-of dan dereference.

3. Source Code

A. Menu Login

Fitur ini menampilkan menu dengan tiga opsi yaitu Login, Registrasi user baru, dan Keluar dari program.

```
void menu_login() {
    while (true) {
        cout << "\n===== " << endl;
        cout << "|      Menu Login      |" << endl;
        cout << "===== " << endl;
        cout << "|    1. Login          |" << endl;
        cout << "|    2. Registrasi     |" << endl;
        cout << "|    3. Keluar         |" << endl;
        cout << "===== " << endl;
        cout << "Pilih opsi (1-3): ";
    }
}
```

B. Login

Fitur ini memastikan hanya pengguna dengan nama dan NIM yang sudah tersimpan di array of struct yang dapat mengakses program. Pointer pada program digunakan untuk mengakses array of struct bertipe “user” yang memungkinkan fungsi login untuk membaca dan membandingkan data pengguna secara langsung dari alamat memori. Jika nama dan NIM tidak cocok dengan elemen struct yang ditunjuk oleh pointer, program memberi kesempatan sebanyak 3 kali sebelum berhenti.

```
int login(user* users, int jumlah_user) {
    string nama, nim;
    int attempts = 0;
    while (attempts < 3) {
```

```

        cout << "\n=== Login ===" << endl;
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, nim);

        for (int i = 0; i < jumlah_user; ++i) {
            if (users[i].nama == nama && users[i].nim == nim) {
                cout << "Login berhasil! Selamat datang, " << nama << "!" <<
endl;
                return i;
            }
        }
        cout << "Nama atau NIM salah. Coba lagi." << endl;
        attempts++;
    }
    cout << "Terlalu banyak percobaan. Program berhenti." << endl;
    exit(0);
}

```

C. Registrasi

Fitur ini memungkinkan untuk menambah pengguna baru dengan pointer “user* pUser = &users[jumlah_user]” yang menunjuk langsung ke elemen struct baru. Data diinisialisasi melalui operator “->” yang memungkinkan manipulasi langsung di memori tanpa menyalin nilai. Jika berhasil, jumlah pengguna akan ditambah.

```

void regis() {
    cout << "\n=== Registrasi ===" << endl;
    cout << "Masukkan Nama: ";
    getline(cin, users[jumlah_user].nama);
    cout << "Masukkan NIM: ";
    getline(cin, users[jumlah_user].nim);

    user* pUser = &users[jumlah_user]; // menyimpan pointer ke user yang baru
    pUser->jumlah_pass = 0; //inisialisasi jumlah password menjadi 0

    jumlah_user++;
    cout << "Registrasi berhasil! Silakan login kembali." << endl;
}

```

D. Menu Utama

Fitur ini menampilkan menu pilihan untuk menambahkan password, lihat password, update password, hapus password, serta logout dari akun.

```
void main_menu(user &u) {
    while (true) {
        cout << "\n===== " << endl;
        cout << "|   Sistem Manajemen Password   |" << endl;
        cout << "===== " << endl;
        cout << "|       1. Tambah Password       |" << endl;
        cout << "|       2. Lihat Password        |" << endl;
        cout << "|       3. Update Password       |" << endl;
        cout << "|       4. Hapus Password        |" << endl;
        cout << "|       5. Logout                |" << endl;
        cout << "===== " << endl;
        cout << "Pilih opsi: ";
```

E. Fitur-Fitur Utama

Source Code:

1) Fitur Tambah Password

Fitur ini meminta pengguna memasukkan data terkait password yang ingin disimpan dengan mengakses secara langsung elemen array melalui pointer “data_password* p = &u.daftar_pass[u.jumlah_pass]”. Input pengguna lalu dimasukkan ke dalam struct menggunakan operator “->” dan kekuatan password akan dievaluasi melalui pointer. Waktu penyimpanan data juga dicatat menggunakan “ctime()” dan disimpan langsung lewat pointer, memungkinkan program memanipulasi data secara efisien tanpa penyalinan nilai.

```
void tambah_pass(user &u) {
    if (u.jumlah_pass >= 100) {
        cout << "Penyimpanan penuh!" << endl;
        return;
    }
    data_password* p = &u.daftar_pass[u.jumlah_pass]; // akses data dengan
    dereference

    cout << "\n=== Tambah Password ===" << endl;
    cout << "Website/Platform: "; getline(cin, p->website);
```

```

cout << "Username: "; getline(cin, p->username);
cout << "Password: "; getline(cin, p->password);
cout << "Kategori (sosmed, bank, dll): "; getline(cin, p->kategori);
cout << "Email yang terhubung: "; getline(cin, p->email);

// mengecek kekuatan password
p->strength = cek_strength(&p->password);
time_t now = time(0);
p->waktu = ctime(&now);

u.jumlah_pass++;
cout << "Password berhasil ditambahkan!" << endl;
}

```

2) Fitur Lihat Password

Fitur ini menampilkan daftar password yang tersimpan dalam bentuk tabel yang rapi menggunakan setw() dari <iomanip>. Prinsip pointer diterapkan saat data password dikirim sebagai alamat (&u.daftar-pass[i]) ke fungsi “show_data_pass” yang kemudian mengakses isi struct menggunakan operator panah “->”.

```

// fungsi untuk menampilkan header data
void show_pass_header() {
    cout << left << setw(5) << "No"
        << setw(20) << "Website"
        << setw(15) << "Username"
        << setw(20) << "Password"
        << setw(15) << "Kategori"
        << setw(30) << "Email"
        << setw(10) << "Strength"
        << setw(25) << "Waktu" << endl;
    cout << string(140, '-') << endl;
}

// fungsi untuk menampilkan data password menggunakan prinsip pointer dan
// dengan parameter dereference
void show_pass_data(int indeks, const data_password* p) {
    cout << left << setw(5) << indeks
        << setw(20) << p->website
        << setw(15) << p->username
        << setw(20) << p->password
        << setw(15) << p->kategori
        << setw(30) << p->email
        << setw(10) << p->strength
        << setw(25) << p->waktu;
}

```



```
// fungsi tampilkan data menggunakan prinsip pointer dan menggunakan
parameter address-of
void lihat_pass(const user &u) {
    if (u.jumlah_pass == 0) {
        cout << "Belum ada password tersimpan :(" << endl;
        return;
    }
    show_pass_header();
    for (int i = 0; i < u.jumlah_pass; ++i) {
        show_pass_data(i + 1, &u.daftar_pass[i]);
    }
}
```

3) Fitur Update Password

Fitur ini menggantikan data lama dengan data baru yang di input pengguna. Prinsip pointer digunakan untuk menunjuk langsung ke elemen “daftar_pass” yang dipilih, melalui “data_password* p = &u.daftar_pass[idx - 1]”. Dengan pointer “p”, data lama diakses dan diperbarui secara langsung di memori menggunakan operator panah “->” tanpa perlu menyalin nilai antar variabel.

```
void update_pass(user &u) {
    lihat_pass(u);
    if (u.jumlah_pass == 0)
        return;
    int idx;
    cout << "Pilih nomor password untuk diupdate: ";
    cin >> idx; cin.ignore();

    if (idx < 1 || idx > u.jumlah_pass) {
        cout << "Nomor tidak valid." << endl;
        return;
    }

    data_password* p = &u.daftar_pass[idx - 1]; // akses data dengan
dereference
    cout << "Website/Platform baru: "; getline(cin, p->website);
    cout << "Username baru: "; getline(cin, p->username);
    cout << "Password baru: "; getline(cin, p->password);
    cout << "Kategori baru: "; getline(cin, p->kategori);
    cout << "Email baru: "; getline(cin, p->email);

    p->strength = cek_strength(&p->password);
    time_t now = time(0);
```

```

    p->waktu = ctime(&now);
    cout << "Password berhasil diperbarui!" << endl;
}

```

4) Fitur Hapus Password

Fitur ini menghapus data dengan menggeser elemen array sehingga tidak ada celah kosong. Prinsip pointer digunakan untuk mengakses dan memindahkan data antar elemen struct secara langsung. Pointer “p1” dan “p2” menggunakan dereference (*p1 = *p2) yang memungkinkan manipulasi data langsung di memoir tanpa perlu variabel tambahan.

```

void hapus_pass(user &u) {
    lihat_pass(u);
    if (u.jumlah_pass == 0) return;
    int idx;
    cout << "Pilih nomor password untuk dihapus: ";
    cin >> idx; cin.ignore();

    if (idx < 1 || idx > u.jumlah_pass) {
        cout << "Nomor tidak valid." << endl;
        return;
    }
    // penerapan pointer untuk memindahkan data
    for (int i = idx - 1; i < u.jumlah_pass - 1; ++i) {
        // akses data dengan dereference
        data_password* p1 = &u.daftar_pass[i]; // elemen yang akan dipindah
        data_password* p2 = &u.daftar_pass[i + 1]; // elemen yang akan
        menggantikan
        *p1 = *p2; // copy data dari p2 ke p1
    }
    u.jumlah_pass--;

    cout << "Password berhasil dihapus!" << endl;
}

```

4. Uji Coba dan Hasil Output

```
=====
|      Menu Login      |
=====
|      1. Login        |
|      2. Registrasi   |
|      3. Keluar        |
=====
Pilih opsi (1-3): 2

=== Registrasi ===
Masukkan Nama: Nuno
Masukkan NIM: 2401036017
Registrasi berhasil! Silakan login kembali.
```

Gambar 4.1 Output Program

```
=====
|      Menu Login      |
=====
|      1. Login        |
|      2. Registrasi   |
|      3. Keluar        |
=====
Pilih opsi (1-3): 1

=== Login ===
Masukkan Nama: Nuno
Masukkan NIM: 2401036017
Login berhasil! Selamat datang, Nuno!
```

Gambar 4.2 Output Program

```

=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====

Pilih opsi: 1

=== Tambah Password ===
Website/Platform: tiktok
Username: nunoganteng
Password: aziracantik
Kategori (sosmed, bank, dll): sosmed uhuy
Email yang terhubung: nunoazira@forever.com
Password berhasil ditambahkan!

```

Gambar 4.3 Output Program

```

=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====

Pilih opsi: 2

```

No	Website	Username	Password	Kategori	Email	Strength	Waktu
1	tiktok	nunoganteng	aziracantik	sosmed uhuy	nunoazira@forever.com	Kuat	Mon Apr 21 17:47:19 2025

Gambar 4.4 Output Program

```

=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====

Pilih opsi: 4

```

No	Website	Username	Password	Kategori	Email	Strength	Waktu
1	tiktok	nunoganteng	aziracantik	sosmed uhuy	nunoazira@forever.com	Kuat	Mon Apr 21 17:47:19 2025

```

Pilih nomor password untuk dihapus: 1
Password berhasil dihapus!

```

Gambar 4.5 Output Program

```

=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====
Pilih opsi: 2
Belum ada password tersimpan :(

=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====
Pilih opsi: 5
Logout berhasil!

```

Gambar 4.6 Output Program

```

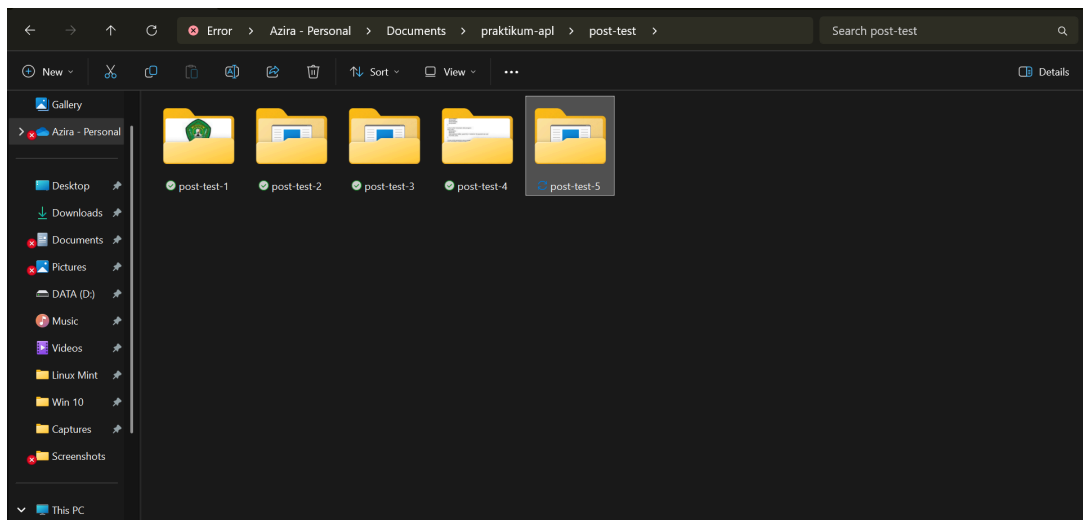
=====
|   Menu Login                 |
=====
|   1. Login                   |
|   2. Registrasi              |
|   3. Keluar                  |
=====
Pilih opsi (1-3): 3
Apakah Anda ingin keluar? (y/n): y
Terima kasih telah menggunakan program ini!

```

Gambar 4.7 Output Program

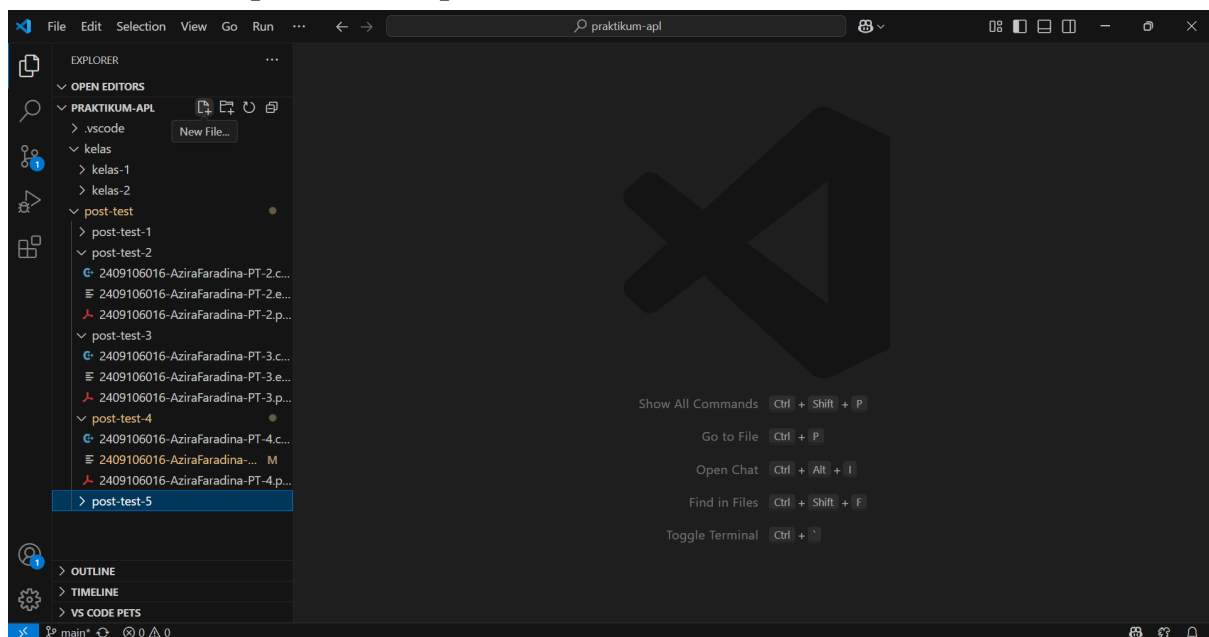
5. Langkah-Langkah Git pada VSCode

5.1 Buat Folder Baru di Explorer

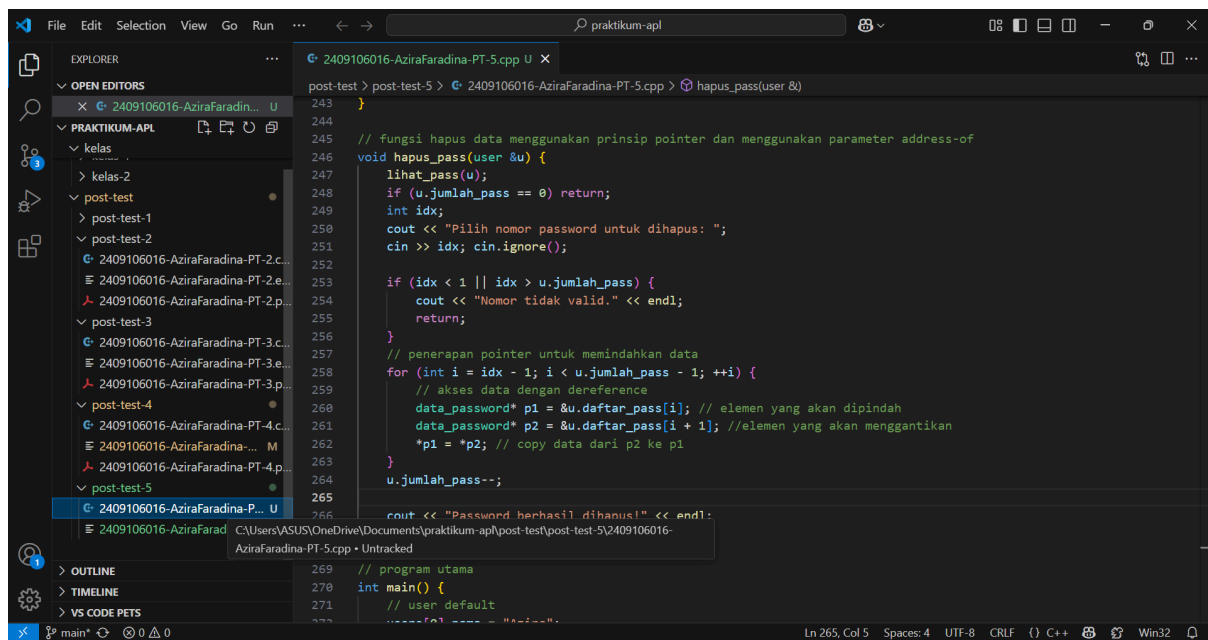


Buat folder baru “post-test-5” di dalam folder “post-test”.

5.2 Buka Folder praktikum-apl di VSCode

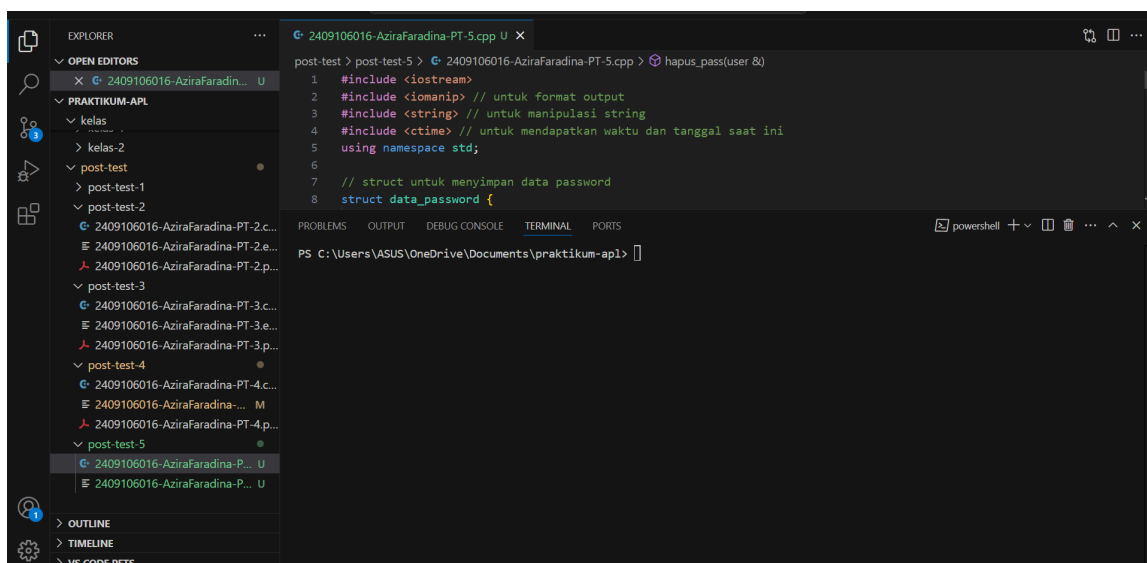


5.3 Buat File Baru di folder post-test-5



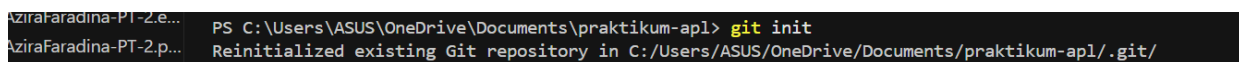
Buat file .cpp baru dengan nama file “2409106016-AziraFaradina-PT-5”.

5.4 Buka Terminal



Path sudah ada di paling awal dari folder.

5.5 Git init



Praktikan melakukan git init untuk menginisiasi repository git.

5.6 Git Add

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git add .
```

Lalu, praktikan melakukan git add untuk menambahkan file yang ingin di commit (git add .).

5.7 Git Commit

```
radina-PT-3.c... PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git commit -m "tambah file cpp dan exe"
radina-PT-3.e... [main 52266ab] tambah file cpp dan exe
radina-PT-3.p... 3 files changed, 279 insertions(+)
                  create mode 100644 post-test/post-test-5/2409106016-AziraFaradina-PT-5.cpp
                  create mode 100644 post-test/post-test-5/2409106016-AziraFaradina-PT-5.exe
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl>
```

Praktikan melakukan git commit untuk membuat checkpoint, yaitu upload file cpp & exe.

5.8 Git Push

```
iraFaradina-PT-4.c... PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git push -u origin main
iraFaradina-PT-4.e... Enumerating objects: 12, done.
iraFaradina-PT-4.p... Counting objects: 100% (12/12), done.
                        Delta compression using up to 12 threads
                        Compressing objects: 100% (8/8), done.
iraFaradina-PT-5.c... Writing objects: 100% (8/8), 74.70 KiB | 4.98 MiB/s, done.
iraFaradina-PT-5.e... 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/ziraaaxo/praktikum-apl.git
                        386130c..52266ab main -> main
                        branch 'main' set up to track 'origin/main'.
```

Terakhir, praktikan melakukan git push untuk meng-upload semua hal yang ada pada repository lokal ke Github.