

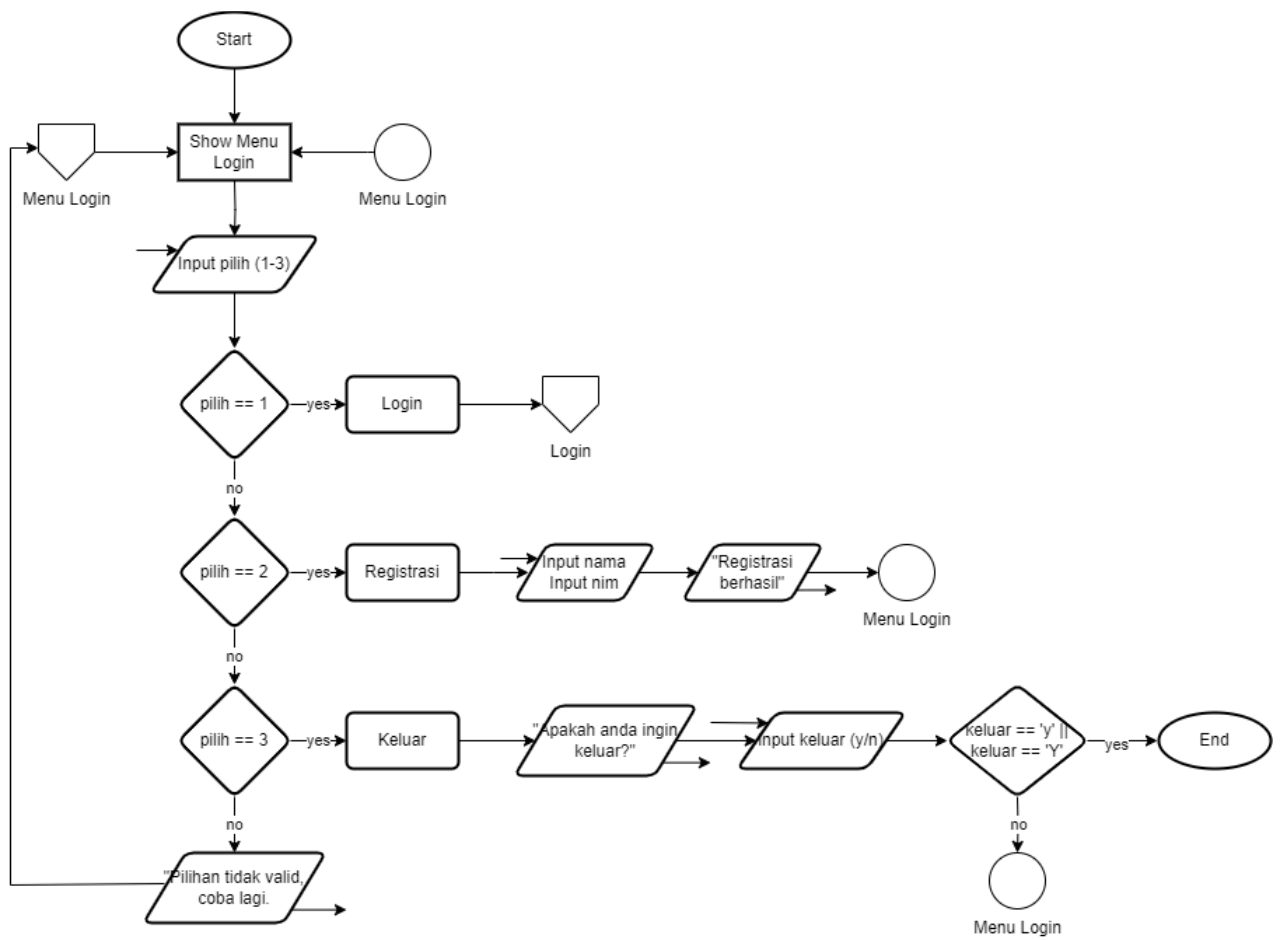
**LAPORAN PRAKTIKUM**  
**POSTTEST (6)**  
**ALGORITMA PEMROGRAMAN LANJUT**



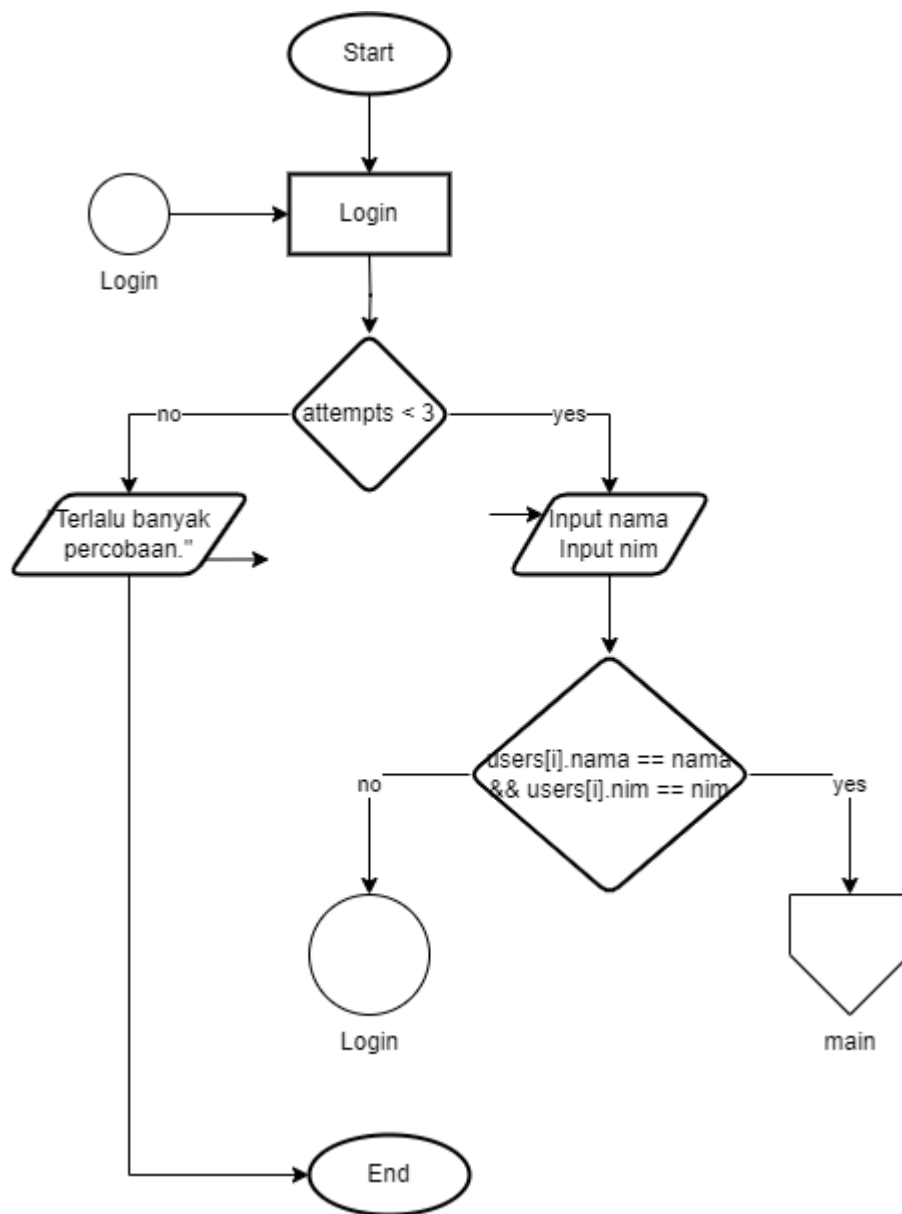
**Disusun oleh:**  
**Azira Faradina (2409106016)**  
**Kelas (A1 '24)**

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

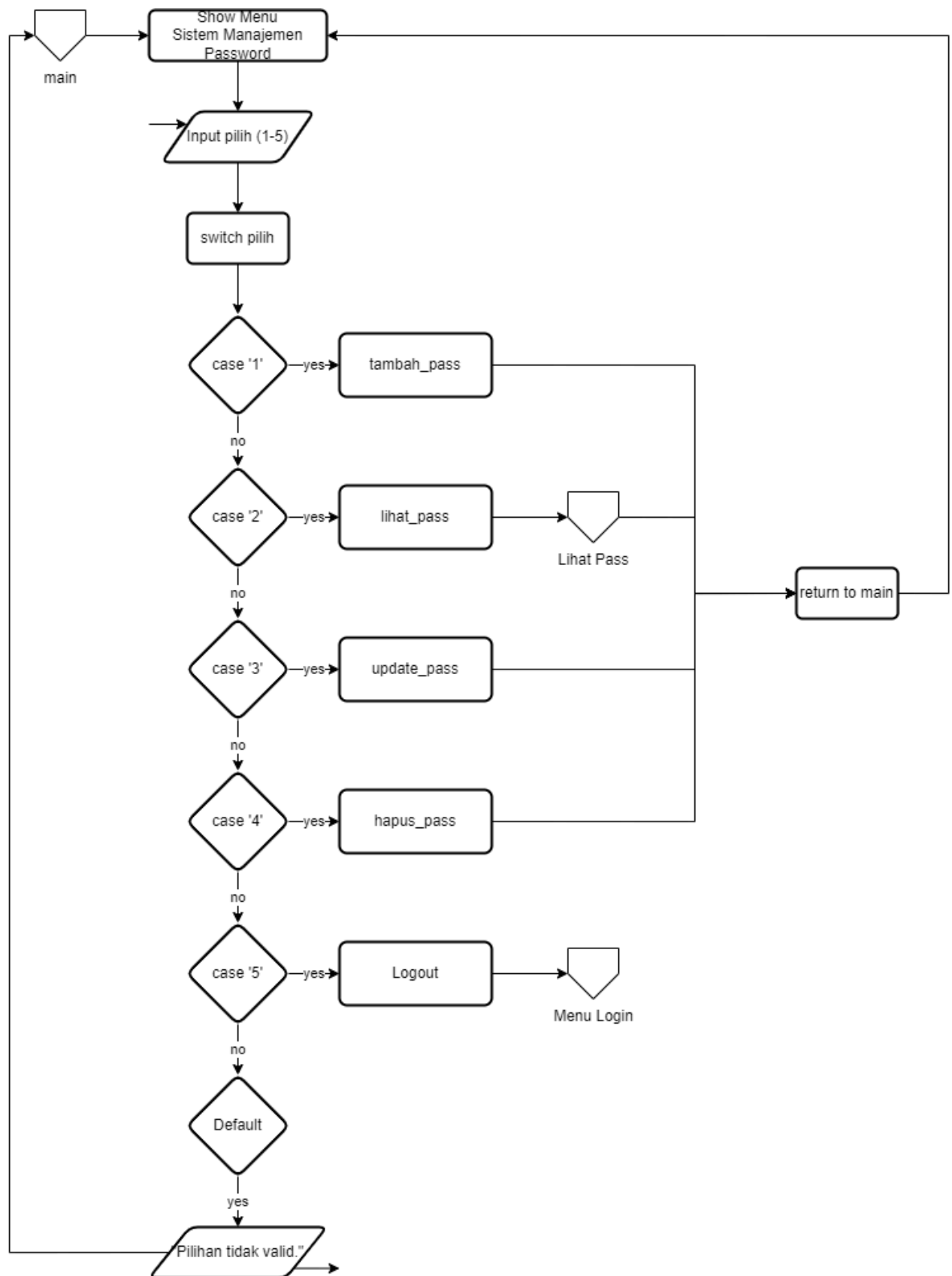
## 1. Flowchart



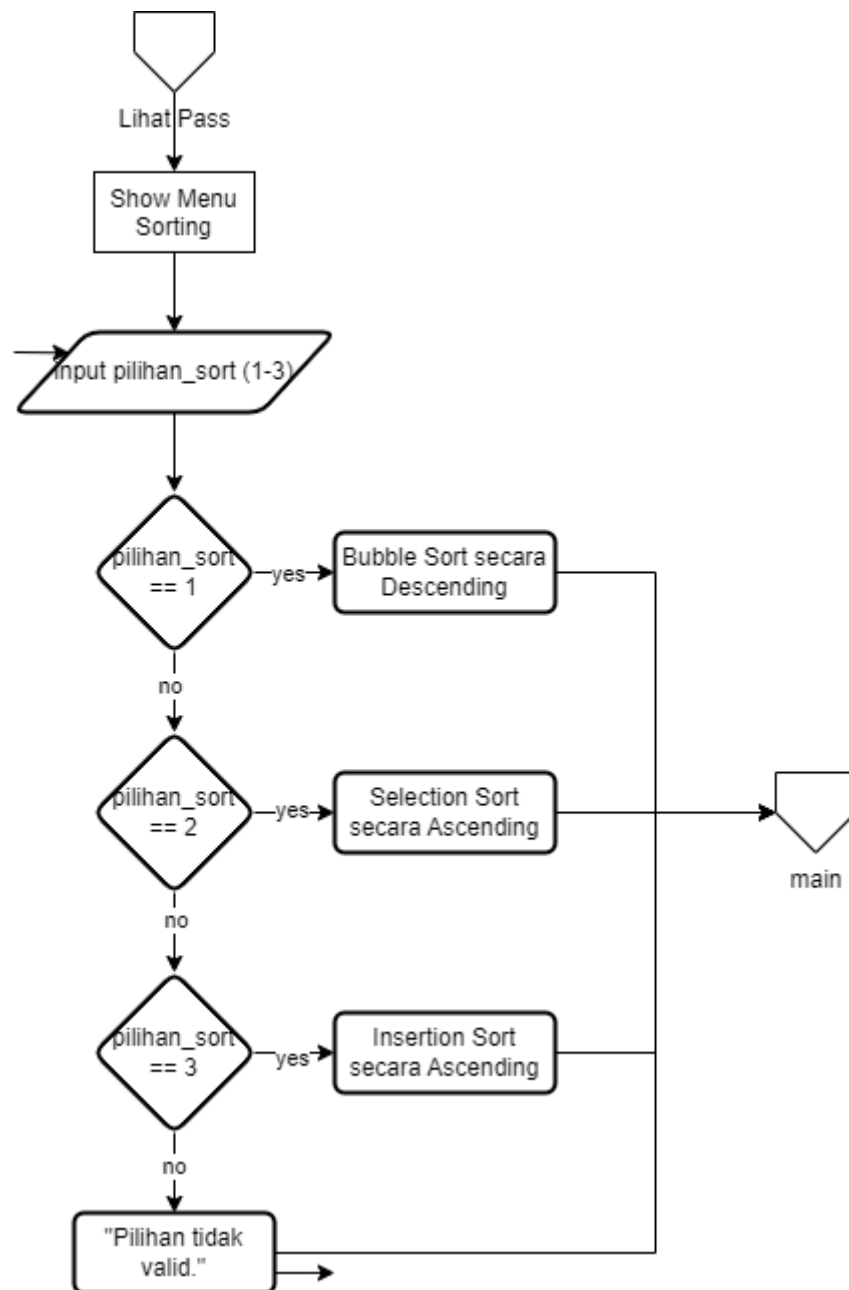
Gambar 1.1 Flowchart Menu Login



Gambar 1.2 Flowchart Fitur Login



Gambar 1.3 Flowchart Menu Utama



Gambar 1.4 Flowchart Fitur Lihat Password

## 2. Analisis Program

Program Manajemen Password Sederhana ini adalah program yang memungkinkan pengguna untuk menyimpan, melihat, memperbaiki, 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 sorting.

### 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.

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

    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);

    p->password_length = p->password.length();
    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 tampilkan 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) << "Panjang"
        << setw(25) << "Waktu" << endl;
    cout << string(150, '-') << endl;
}

// fungsi tampilkan data
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->password_length
        << setw(25) << p->waktu;
}

// fungsi lihat password
int lihat_pass(const user &u, int mapping[]) {
    if (u.jumlah_pass == 0) {
        cout << "Belum ada password tersimpan :(" << endl;
        return 0;
    }
}
```

## □ Menu Sorting

Fitur ini menampilkan menu untuk memilih kriteria dan metode pengurutan yang diinginkan.

```

int pilihan_sort;
cout << "\n===== " << endl;
cout << "|          Menu Sorting          |" << endl;
cout << "===== " << endl;
cout << "|  1. Sort Website (Z - A)  |" << endl;
cout << "|  2. Sort Panjang Password |" << endl;
cout << "|  3. Sort Username (A - Z) |" << endl;
cout << "===== " << endl;
cout << "Pilih opsi (1-3): ";
cin >> pilihan_sort;
cin.ignore();

```

### □ Simpan Data

Kode ini menyalin data asli ke array “temp” untuk diurutkan, sementara array “mapping” akan menyimpan indeks asli dari elemen tersebut.

```

data_password temp[100];
for (int i = 0; i < u.jumlah_pass; i++) {
    temp[i] = u.daftar_pass[i];
    mapping[i] = i; // mapping awal
}

if (pilihan_sort == 1) {
    bubble_sort(temp, mapping, u.jumlah_pass);
} else if (pilihan_sort == 2) {
    selection_sort(temp, mapping, u.jumlah_pass);
} else if (pilihan_sort == 3) {
    insertion_sort(temp, mapping, u.jumlah_pass);
} else {
    cout << "Pilihan tidak valid." << endl;
    return 0;
}

```

### □ Metode Sorting

Ada 3 metode sorting yang diterapkan pada program ini.

- Bubble Sort secara Descending (Website). Prinsipnya yaitu membandingkan pasangan elemen pada “arr[j].website” berurutan dan menukarnya jika tidak sesuai dengan urutan.

```

// Bubble Sort untuk sort website (Descending Z-A)
void bubble_sort(data_password arr[], int mapping[], int n) {

```

```

    for (int i = 0; i < n-1; ++i) {
        for (int j = 0; j < n-i-1; ++j) {
            if (arr[j].website < arr[j+1].website) {
                swap(arr[j], arr[j+1]);
                swap(mapping[j], mapping[j+1]);
            }
        }
    }
}

```

- Selection Sort secara Ascending (int Panjang Password). Metode ini mencari elemen minimum dari bagian yang belum diurutkan dan menempatkannya di posisi yang benar.

```

// Selection Sort untuk sort panjang password (Ascending 1-9)
void selection_sort(data_password arr[], int mapping[], int n) {
    for (int i = 0; i < n-1; ++i) {
        int min_idx = i;
        for (int j = i+1; j < n; ++j) {
            if (arr[j].password_length < arr[min_idx].password_length) {
                min_idx = j;
            }
        }
        swap(arr[i], arr[min_idx]);
        swap(mapping[i], mapping[min_idx]);
    }
}

```

- Insertion Sort secara Ascending (Username). Memasukkan elemen pada string “arr[j].username” ke dalam posisi yang sesuai dalam subset data yang sudah terurut.

```

// Insertion Sort untuk sort username (Ascending A-Z)
void insertion_sort(data_password arr[], int mapping[], int n) {
    for (int i = 1; i < n; i++) {
        data_password key = arr[i];
        int key_idx = mapping[i];
        int j = i - 1;
        while (j >= 0 && arr[j].username > key.username) {
            arr[j + 1] = arr[j];
            mapping[j + 1] = mapping[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

```

```

        mapping[j + 1] = key_idx;
    }
}

```

### 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.

```

// fungsi update password
void update_pass(user &u) {
    int mapping[100];
    int n = lihat_pass(u, mapping);
    if (n == 0) return;

    int idx;
    cout << "\nPilih nomor password untuk diupdate: ";
    cin >> idx;
    cin.ignore();

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

    data_password* p = &u.daftar_pass[mapping[idx - 1]];
    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->password_length = p->password.length();
    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) {
    int mapping[100];
    int n = lihat_pass(u, mapping);
    if (n == 0) return;

    int idx;
    cout << "\nPilih nomor password untuk dihapus: ";
    cin >> idx;
    cin.ignore();

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

    int asli = mapping[idx - 1];
    for (int i = asli; i < u.jumlah_pass - 1; ++i) {
        u.daftar_pass[i] = u.daftar_pass[i + 1];
    }
    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): 1

=== Login ===
Masukkan Nama: Azira
Masukkan NIM: 2409106016
Login berhasil! Selamat datang, Azira!
```

Gambar 4.1 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: ig
Username: ziraaaxo
Password: azira
Kategori (sosmed, bank, dll): sosmed
Email yang terhubung: azira@gmail.com
Password berhasil ditambahkan!

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

```

Gambar 4.2 Output Program

```
=== Tambah Password ===  
Website/Platform: twitter  
Username: azirabfr  
Password: aziracantik  
Kategori (sosmed, bank, dll): sosmed  
Email yang terhubung: azira@gmail.com  
Password berhasil ditambahkan!
```

```
=====
```

Sistem Manajemen Password
---------------------------

```
=====
```

1. Tambah Password
2. Lihat Password
3. Update Password
4. Hapus Password
5. Logout

```
=====
```

Pilih opsi: 1

```
=== Tambah Password ===  
Website/Platform: youtube  
Username: mapoft  
Password: cantikmakasih  
Kategori (sosmed, bank, dll): ent  
Email yang terhubung: azira@gmail.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

=====
|   Menu Sorting                |
=====
|   1. Sort Website (Z - A)    |
|   2. Sort Panjang Password   |
|   3. Sort Username (A - Z)   |
=====
Pilih opsi (1-3): 1

```

No	Website	Username	Password	Kategori	Email	Panjang	Waktu
1	youtube	mapoft	cantikmakasih	ent	azira@gmail.com	13	Tue Apr 29 20:31:55 2025
2	twitter	azirabfr	aziracantik	sosmed	azira@gmail.com	11	Tue Apr 29 20:31:38 2025
3	ig	ziraaaxo	azira	sosmed	azira@gmail.com	5	Tue Apr 29 20:31:17 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: 3

=====
|   Menu Sorting                |
=====
|   1. Sort Website (Z - A)    |
|   2. Sort Panjang Password   |
|   3. Sort Username (A - Z)   |
=====
Pilih opsi (1-3): 2

```

No	Website	Username	Password	Kategori	Email	Panjang	Waktu
1	ig	ziraaaxo	azira	sosmed	azira@gmail.com	5	Tue Apr 29 20:31:17 2025
2	twitter	azirabfr	aziracantik	sosmed	azira@gmail.com	11	Tue Apr 29 20:31:38 2025
3	youtube	mapoft	cantikmakasih	ent	azira@gmail.com	13	Tue Apr 29 20:31:55 2025

```

Pilih nomor password untuk diupdate: 1
Website/Platform baru: ig
Username baru: ziraaaxo
Password baru: iyakamujugacantik
Kategori baru: sosmed
Email baru: azira@gmail.com
Password berhasil diperbarui!

```

Gambar 4.5 Output Program

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

=====
|   Menu Sorting               |
=====
|   1. Sort Website (Z - A)    |
|   2. Sort Panjang Password  |
|   3. Sort Username (A - Z)  |
=====
Pilih opsi (1-3): 3
No  Website      Username      Password      Kategori      Email              Panjang  Waktu
-----
1   twitter      azirabfr     aziracantik   sosmed        azira@gmail.com    11       Tue Apr 29 20:31:38 2025
2   youtube      mapoft       cantikmakasih ent          azira@gmail.com    13       Tue Apr 29 20:31:55 2025
3   ig           ziraaxo      iyakamujugacantik sosmed        azira@gmail.com    17       Tue Apr 29 20:32:29 2025

Pilih nomor password untuk dihapus: 2
Password berhasil dihapus!
```

Gambar 4.6 Output Program

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

=====
|   Menu Sorting               |
=====
|   1. Sort Website (Z - A)    |
|   2. Sort Panjang Password  |
|   3. Sort Username (A - Z)  |
=====
Pilih opsi (1-3): 1
No  Website      Username      Password      Kategori      Email              Panjang  Waktu
-----
1   twitter      azirabfr     aziracantik   sosmed        azira@gmail.com    11       Tue Apr 29 20:31:38 2025
2   ig           ziraaxo      iyakamujugacantik sosmed        azira@gmail.com    17       Tue Apr 29 20:32:29 2025
```

Gambar 4.7 Output Program

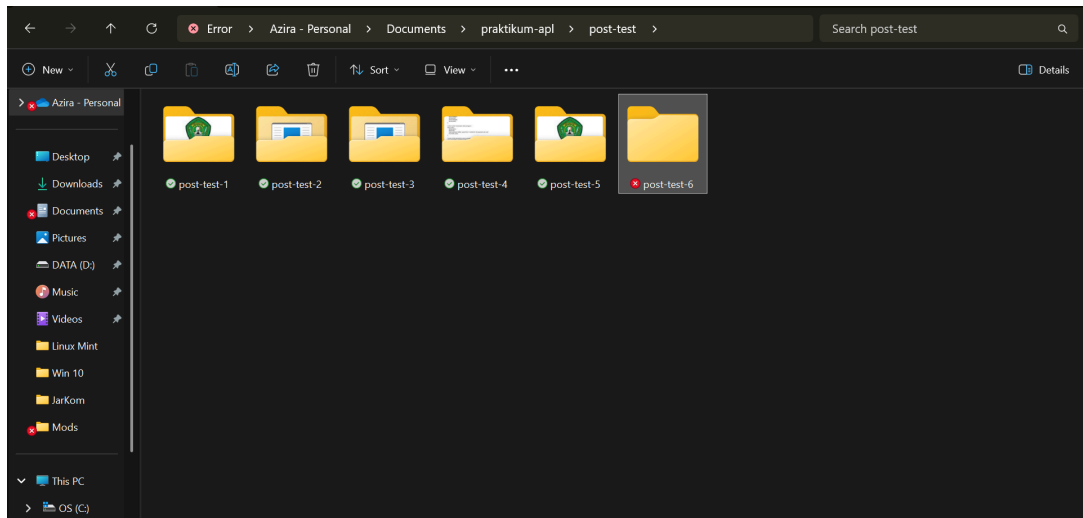
```
=====
|   Sistem Manajemen Password   |
=====
|   1. Tambah Password         |
|   2. Lihat Password          |
|   3. Update Password         |
|   4. Hapus Password          |
|   5. Logout                  |
=====
Pilih opsi: 5
Logout berhasil!

=====
|   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 4.8 Output Program

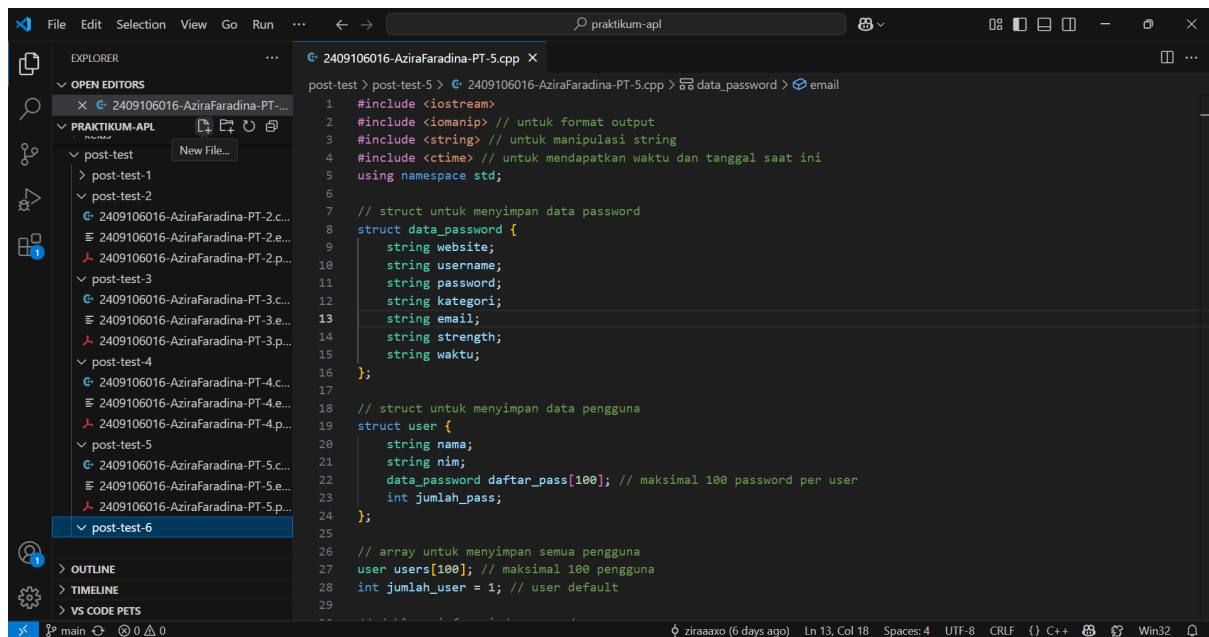
## 5. Langkah-Langkah Git pada VSCode

### 5.1 Buat Folder Baru di Explorer

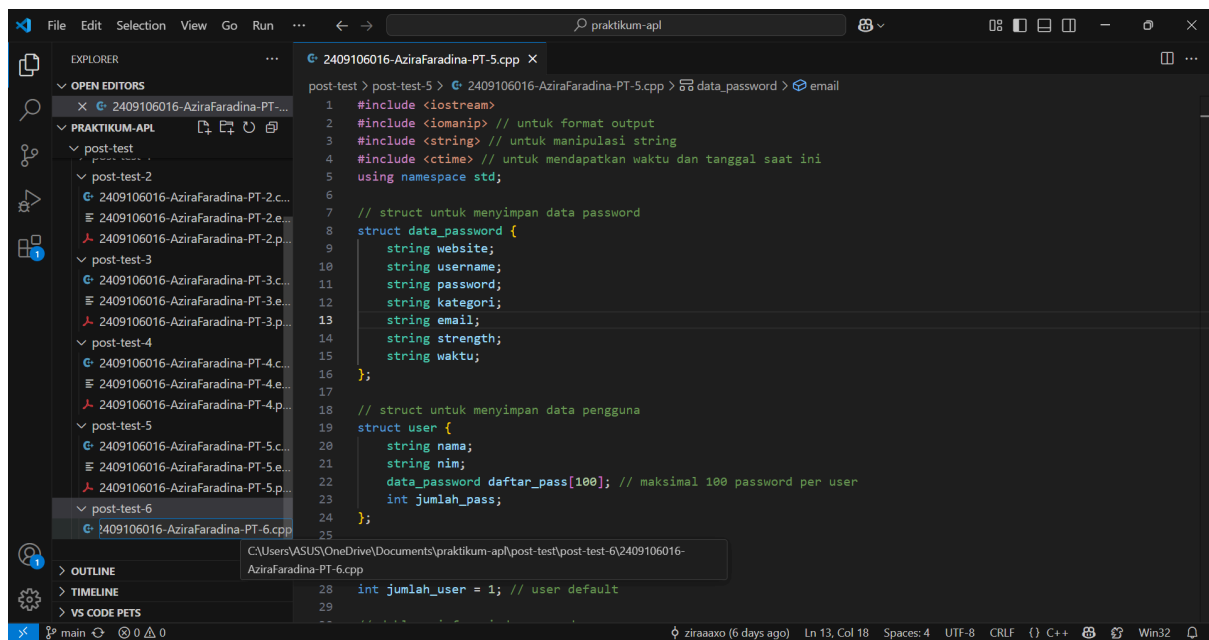


Buat folder baru “post-test-6” di dalam folder “post-test”.

### 5.2 Buka Folder praktikum-apl di VSCode

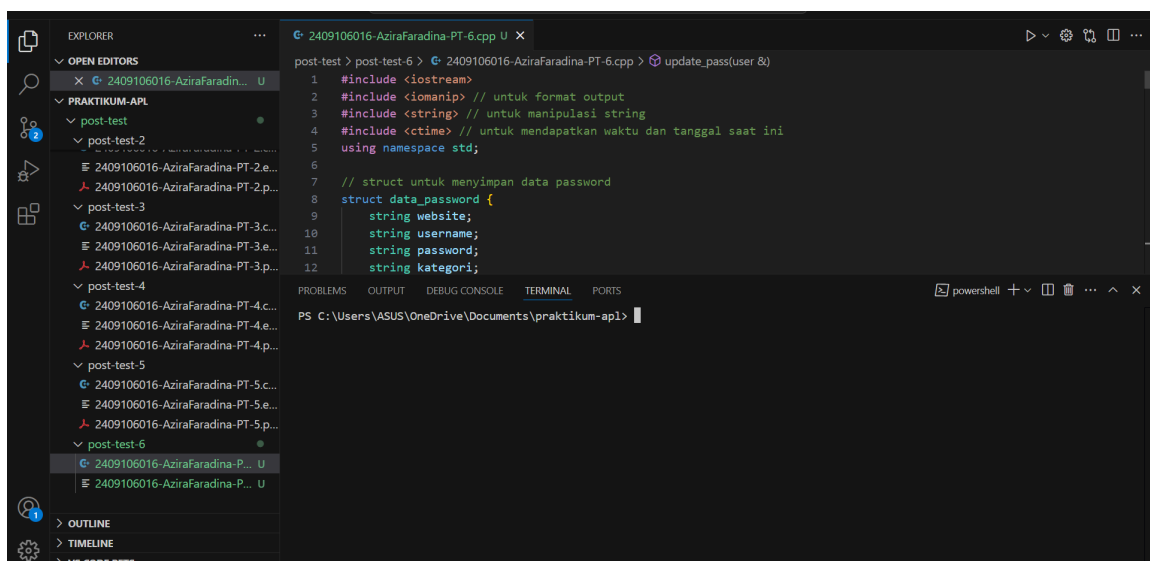


## 5.3 Buat File Baru di folder post-test-5



Buat file .cpp baru dengan nama file “2409106016-AziraFaradina-PT-6”.

## 5.4 Buka Terminal



Path sudah ada di paling awal dari folder.

## 5.5 Git init

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/ASUS/OneDrive/Documents/praktikum-apl/.git/
```

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

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git commit -m "upload file cpp dan exe"
[main c91dda9] upload file cpp dan exe
2 files changed, 338 insertions(+)
create mode 100644 post-test/post-test-6/2409106016-AziraFaradina-PT-6.cpp
create mode 100644 post-test/post-test-6/2409106016-AziraFaradina-PT-6.exe
```

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

## 5.8 Git Push

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 75.97 KiB | 3.62 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/ziraaaxo/praktikum-apl.git
   f4807f4..c91dda9  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apl>
```

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