

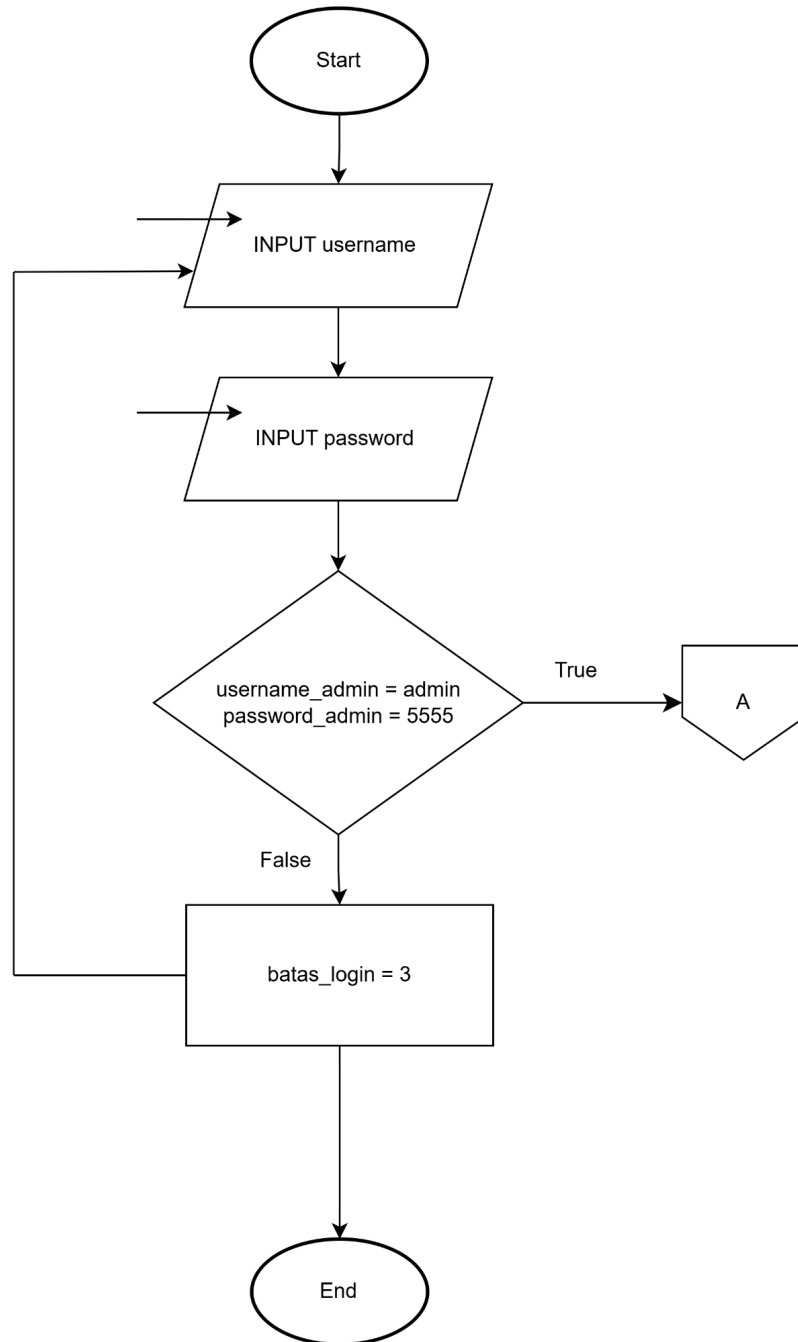
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



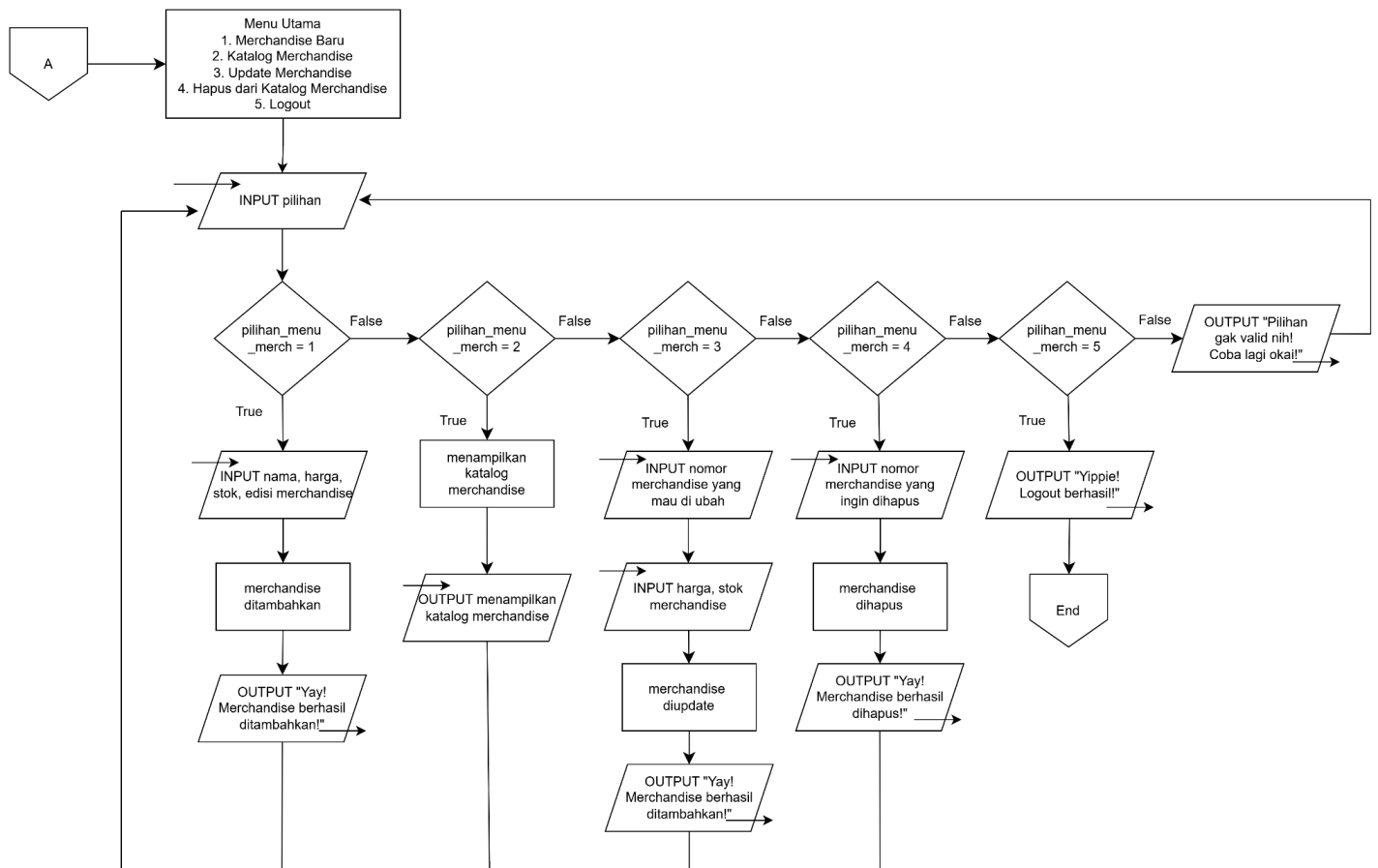
Disusun oleh:
AULIA NATASYA
2409106084
KELAS B2'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart

2. Analisis Program

Program ini memungkinkan admin untuk create, read, update, dan delete data merchandise. Lalu dengan adanya fitur login yang memiliki batas percobaan sebanyak tiga kali kesempatan, program memastikan bahwa hanya admin yang berwenang dapat mengakses sistem, sehingga keamanan data terjaga. Keuntungan utama dari program ini adalah untuk meningkatkan keteraturan dalam pengelolaan data merchandise. Dimana admin dapat dengan mudah memperbarui informasi stok dan harga, sehingga mengurangi kemungkinan kesalahan input. Selain itu, tampilan data yang rapi dapat membantu admin mudah untuk membaca dan mengelola katalog. Dengan adanya fitur penghapusan data, merchandise yang sudah tidak tersedia dapat langsung dihapus dari program, dan memastikan katalog selalu terupdate.

3. Source Code

3.1 Fitur Login Admin

Fitur ini memungkinkan admin untuk melakukan login dengan username dan password yang sudah ditentukan. Pengguna memiliki batas percobaan sebanyak tiga kali.

```
string username_admin, password_admin;
int batas_login = 3;

while (batas_login > 0) {
    cout << "===== " << endl;
    cout << "        Selamat Datang di Halaman Login!        " << endl;
    cout << "===== " << endl;
    cout << "Username: ";
    cin >> username_admin;
    cout << "Password: ";
    cin >> password_admin;

    if (username_admin == "aulianatasya" && password_admin == "084") {
        cout << "Yippie! Login berhasil!" << endl;
        cout << "-----" << endl;
        break;
    } else {
        batas_login--;
        cout << "Login gagal! Sisa percobaan " << batas_login << endl;
        cout << "-----" << endl;
    }
}

if (batas_login == 0) {
    cout << "Oopsie! Kesempatan login habis" << endl;
    cout << "-----" << endl;
    return 0;
}
```

3.2 Create Merchandise Baru

Pada bagian create merchandise baru digunakan untuk menambahkan merchandise ke dalam katalog. Admin dapat memasukan nama, harga, stok dan edisi merchandise yang baru, merchandise baru dapat ditambahkan selama kapasitas maksimum belum tercapai. Apabila kapasitas penuh, akan menampilkan peringatan bahwa merchandise tidak bisa ditambahkan lagi.

```

if (pilihan_menu_merch == 1) {
    if (panjang_array < max_merch) {
        cout << "===== " <<
endl;
        cout << "                Merchandise Baru                " <<
endl;
        cout << "===== " <<
endl;
        cout << "Masukan nama merchandise: ";
        getline(cin, katalog[panjang_array].nama_merch);
        cout << "Masukan harga merchandise: ";
        cin >> katalog[panjang_array].rincian.harga_merch;
        cout << "Masukan stok merchandise: ";
        cin >> katalog[panjang_array].rincian.stok_merch;
        cout << "Masukan edisi merchandise: ";
        cin >> katalog[panjang_array].edisi_merch;
        panjang_array++;
        cout << "Yay! Merchandise berhasil ditambahkan!" << endl;
        cout << "-----" << endl;
    } else {
        cout << "Oh naww! Kapasitas penuh! Tidak bisa menambah
merchandise lagi!" << endl;
        cout <<
"-----" << endl;
    }
}

```

3.3 Read Katalog Merchandise

Pada bagian ini digunakan untuk menampilkan daftar merchandise yang tersedia. Apabila katalog masih kosong, sistem akan memberitahu bahwa belum ada merchandise. Jika ada, sistem menampilkan data yang rapi sesuai kolom nomor urut, nama, harga, stok dan edisi merchandise.

```

else if (pilihan_menu_merch == 2){
    if (panjang_array == 0){
        cout << "Belum ada merchandise nih!" << endl;
    } else {
        cout <<
"=====
=" << endl;
        cout << "                                Katalog Merchandise
" << endl;
    }
}

```

```

        cout <<
"=====
=" << endl;
        cout << left << setw(5) << "No"
        << setw(45) << "Nama Merchandise"
        << setw(15) << "Harga"
        << setw(10) << "Stok"
        << setw(10) << "Edisi" << endl;
        cout <<
"-----
-" << endl;

        for (int i = 0; i < panjang_array; i++){
            cout << setw(5) << i + 1;
            cout << setw(45) << katalog[i].nama_merch;
            cout << setw(15) << katalog[i].rincian.harga_merch;
            cout << setw(10) << katalog[i].rincian.stok_merch;
            cout << setw(10) << katalog[i].edisi_merch << endl << endl;
        }
    }
}

```

3.4 Update Merchandise

Pada bagian ini memungkinkan admin untuk mengubah harga dan stok merchandise. Jika katalog kosong, akan menampilkan pesan bahwa tidak ada merchandise yang bisa diperbarui. Jika ada, pengguna memilih merchandise berdasarkan nomor dan memasukan data baru.

```

else if (pilihan_menu_merch == 3) {
    if (panjang_array == 0){
        cout << "Belum ada merchandise yang diubah nih!"
<< endl;
        cout <<
"-----" << endl;
    } else {
        cout <<
"=====
" << endl;
        cout << "Update Merchandise
" << endl;
        cout <<

```

```

"===== " << endl;
        cout << "Masukan nomor merchandise yang ingin di
ubah: ";
        cin >> index;

        if (index > 0 && index <= panjang_array) {
            cout << "Masukan harga baru: ";
            cin >> katalog[index - 1].rincian.harga_merch;
            cout << "Masukan stok baru: ";
            cin >> katalog[index - 1].rincian.stok_merch;
            cout << "Yay! Merchandise berhasil
diperbaharui!" << endl;
        } else {
            cout << "Oopsie! Nomor merchandise ga valid
nih!" << endl;
        }
        cout <<
"-----" << endl;
    }
}

```

3.5 Delete Merchandise

Pada bagian ini, memungkinkan admin menghapus merchandise berdasarkan nomor. Apabila katalog kosong, akan menampilkan sebuah pesan. Jika nomor valid, merchandise dihapus dan data digeser.

```

        else if (pilihan_menu_merch == 4) {
            if (panjang_array == 0){
                cout << "Belum ada merchandise yang dihapus nih!" << endl;
                cout << "-----" <<
endl;
            } else {
                cout << "===== " <<
endl;
                cout << "                Hapus Merchandise                " <<
endl;
                cout << "===== " <<
endl;
            }
        }
    }
}

```

```

        cout << "Masukan nomor merchandise yang ingin dihapus: ";
        cin >> index;

        if (index > 0 && index <= panjang_array){
            for (int i = index - 1; i < panjang_array - 1; i++){
                katalog[i] = katalog[i + 1];
            }
            panjang_array--;
            cout << "Yay! Merchandise berhasil dihapus!" << endl;
        } else {
            cout << "Oopsie! Nomor merchandise ga valid nih!" << endl;
        }
        cout << "-----" <<
endl;
    }
}

```

3.6 Struct dan Nested Struct

Rincian merupakan struct yang isinya informasi mengenai harga dan stok merchandise. Merchandise adalah struct yang memiliki anggota Rincian rincian, yang berarti Merchandise menyimpan sebuah struct Rincian di dalamnya. Karena Rincian berada di dalam Merchandise, maka disebut dengan nested struct.

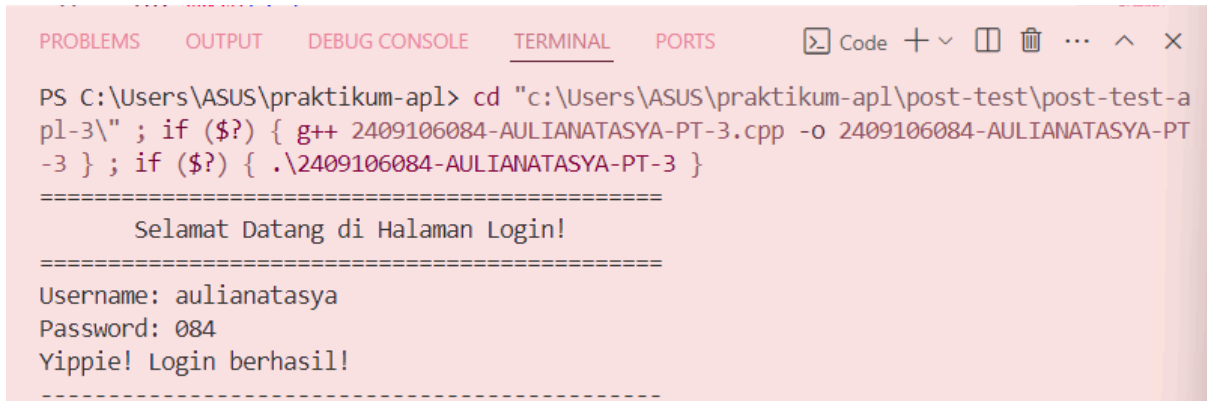
```

struct Rincian {
    int harga_merch;
    int stok_merch;
};

struct Merchandise {
    string nama_merch;
    Rincian rincian;
    int edisi_merch;
};

```

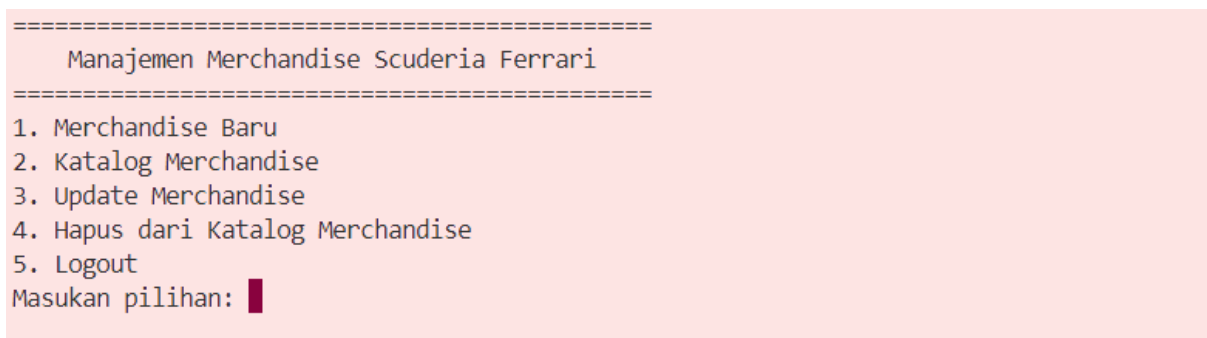

4. Uji Coba dan Hasil Output



```
PS C:\Users\ASUS\praktikum-apl> cd "c:\Users\ASUS\praktikum-apl\post-test\post-test-a
pl-3\" ; if ($?) { g++ 2409106084-AULIANATASYA-PT-3.cpp -o 2409106084-AULIANATASYA-PT
-3 } ; if ($?) { .\2409106084-AULIANATASYA-PT-3 }

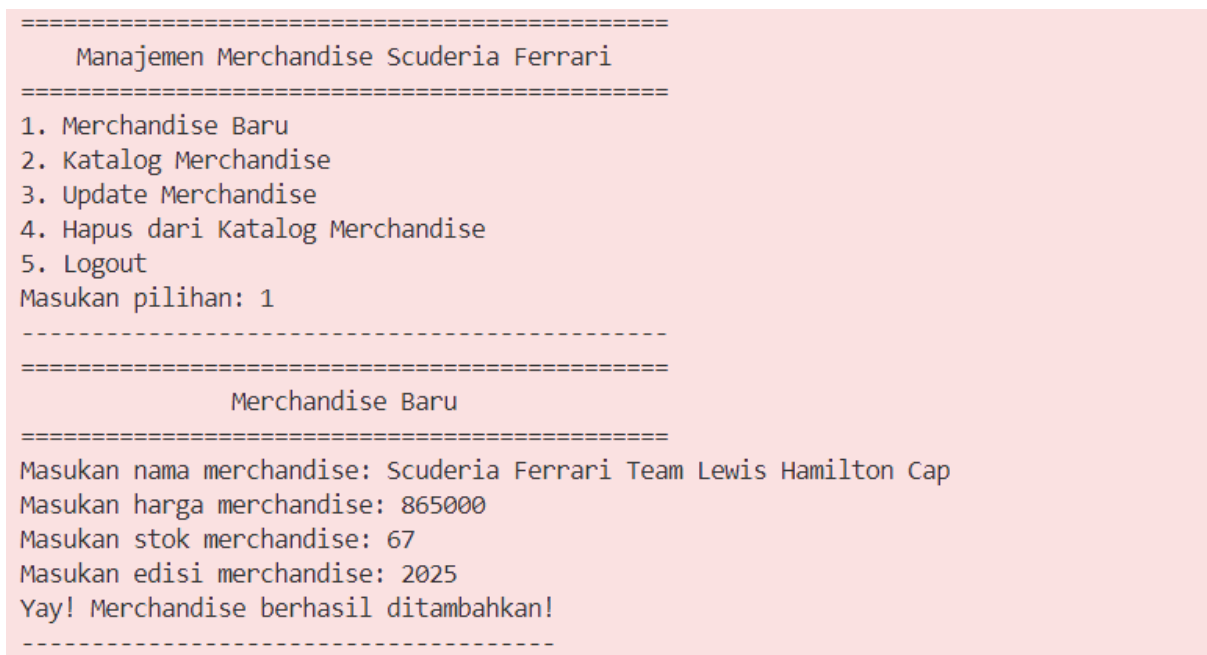
=====
                Selamat Datang di Halaman Login!
=====
Username: aulianatasya
Password: 084
Yippee! Login berhasil!
=====
```

Gambar 4.1 Hasil output berhasil login



```
=====
                Manajemen Merchandise Scuderia Ferrari
=====
1. Merchandise Baru
2. Katalog Merchandise
3. Update Merchandise
4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: █
```

Gambar 4.2 Hasil output setelah berhasil login



```
=====
                Manajemen Merchandise Scuderia Ferrari
=====
1. Merchandise Baru
2. Katalog Merchandise
3. Update Merchandise
4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: 1
-----
=====
                Merchandise Baru
=====
Masukan nama merchandise: Scuderia Ferrari Team Lewis Hamilton Cap
Masukan harga merchandise: 865000
Masukan stok merchandise: 67
Masukan edisi merchandise: 2025
Yay! Merchandise berhasil ditambahkan!
-----
```

Gambar 4.3 Hasil output pilihan 1 yaitu merchandise baru

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... Code + - [ ] [X] ... ^ X

3. Update Merchandise
4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: 2
-----
=====
                        Katalog Merchandise
=====
=====
No      Nama Merchandise                        Harga      Stok      Edisi
-----
1       Scuderia Ferrari Team Hooded Sweat      2160000    24        2024
2       Scuderia Ferrari Drivers Oversize T-Shirt  1655000    33        2024
3       Scuderia Ferrari Team Charles Leclerc Cap   730000     35        2024
4       Scuderia Ferrari Race Jacquard Polo         140000     87        2025
5       Scuderia Ferrari Team Softshell Jacket     3270000    65        2025
6       Scuderia Ferrari Team Carlos Sainz Cap      645000     23        2024
7       Scuderia Ferrari Team Lewis Hamilton Cap    865000     67        2025
```

Gambar 4.4 Hasil output pilihan 2 yaitu menampilkan katalog merchandise

```

4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: 3
-----
=====
                        Update Merchandise
=====
Masukan nomor merchandise yang ingin di ubah: 6
Masukan harga baru: 599000
Masukan stok baru: 3
Yay! Merchandise berhasil diperbaharui!
-----

```

Gambar 4.5 Hasil output pilihan 3 update merchandise

```

4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: 4
-----
=====
                        Hapus Merchandise
=====
Masukan nomor merchandise yang ingin dihapus: 6
Yay! Merchandise berhasil dihapus!
-----

```

Gambar 4.6 Hasil output pilihan 4 yaitu hapus merchandise

```

=====
Manajemen Merchandise Scuderia Ferrari
=====
1. Merchandise Baru
2. Katalog Merchandise
3. Update Merchandise
4. Hapus dari Katalog Merchandise
5. Logout
Masukan pilihan: 5
-----
Yippie! Logout berhasil!
-----
PS C:\Users\ASUS\praktikum-apl\post-test\post-test-apl-3>

```

Gambar 4.7 Hasil output pilihan 5 yaitu logout atau keluar dari sistem

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^ X
PS C:\Users\ASUS\praktikum-apl> cd "c:\Users\ASUS\praktikum-apl\post-test\post-t
est-apl-3\" ; if ($?) { g++ 2409106084-AULIANATASYA-PT-3.cpp -o 2409106084-AULIA
NATASYA-PT-3 } ; if ($?) { .\2409106084-AULIANATASYA-PT-3 }
=====
Selamat Datang di Halaman Login!
=====
Username: wosasie
Password: 0805
Login gagal! Sisa percobaan 2
-----
=====
Selamat Datang di Halaman Login!
=====
Username: saciw
Password: 239
Login gagal! Sisa percobaan 1
-----
=====
Selamat Datang di Halaman Login!
=====
Username: sasie
Password: 0055
Login gagal! Sisa percobaan 0
-----
Oopsie! Kesempatan login habis
-----
PS C:\Users\ASUS\praktikum-apl\post-test\post-test-apl-3> █
```

Gambar 4.8 Hasil output login gagal ketika admin memasukkan username dan password yang salah sampai tiga kali kesempatan.

5. Langkah-Langkah Git

5.1 Git add

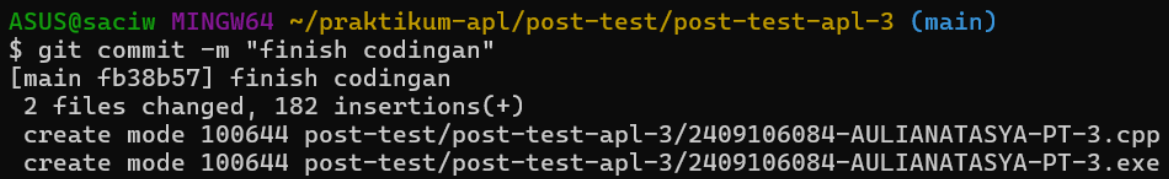
Menambahkan file yang telah diubah atau dibuat ke dalam staging area, sehingga siap untuk dikomit.

```
ASUS@saciw MINGW64 ~/praktikum-apl/post-test/post-test-apl-3 (main)
$ git add .
```

Gambar 5.2 Git add

5.2 Git commit

Menyimpan perubahan dalam repository dengan pesan deskriptif.

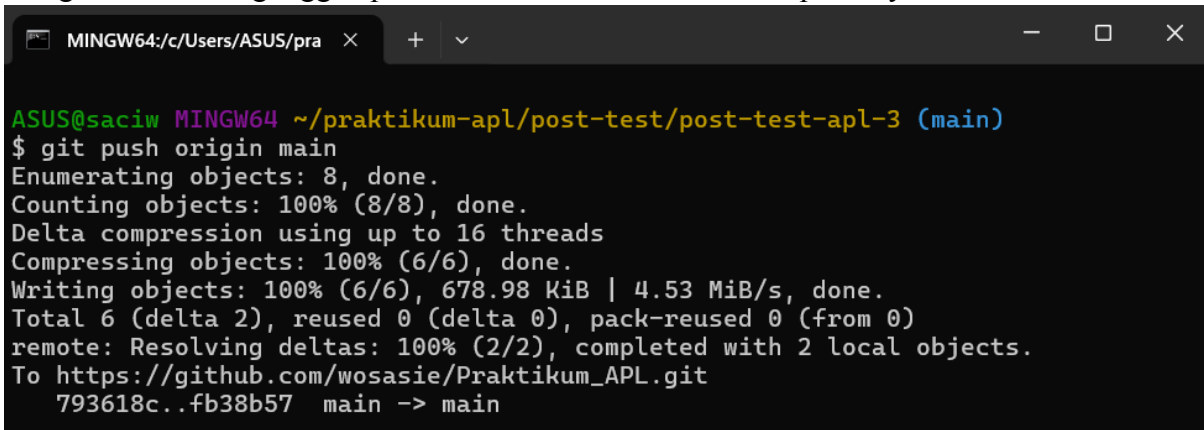


```
ASUS@saciw MINGW64 ~/praktikum-apl/post-test/post-test-apl-3 (main)
$ git commit -m "finish codingan"
[main fb38b57] finish codingan
2 files changed, 182 insertions(+)
create mode 100644 post-test/post-test-apl-3/2409106084-AULIANATASYA-PT-3.cpp
create mode 100644 post-test/post-test-apl-3/2409106084-AULIANATASYA-PT-3.exe
```

Gambar 5.2 Git commit

5.3 Git push origin main

Mengirim atau mengunggah perubahan dari branch main ke repository.



```
ASUS@saciw MINGW64 ~/praktikum-apl/post-test/post-test-apl-3 (main)
$ 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), 678.98 KiB | 4.53 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/wosasie/Praktikum_APL.git
793618c..fb38b57  main -> main
```

Gambar 5.3 Git push origin main