

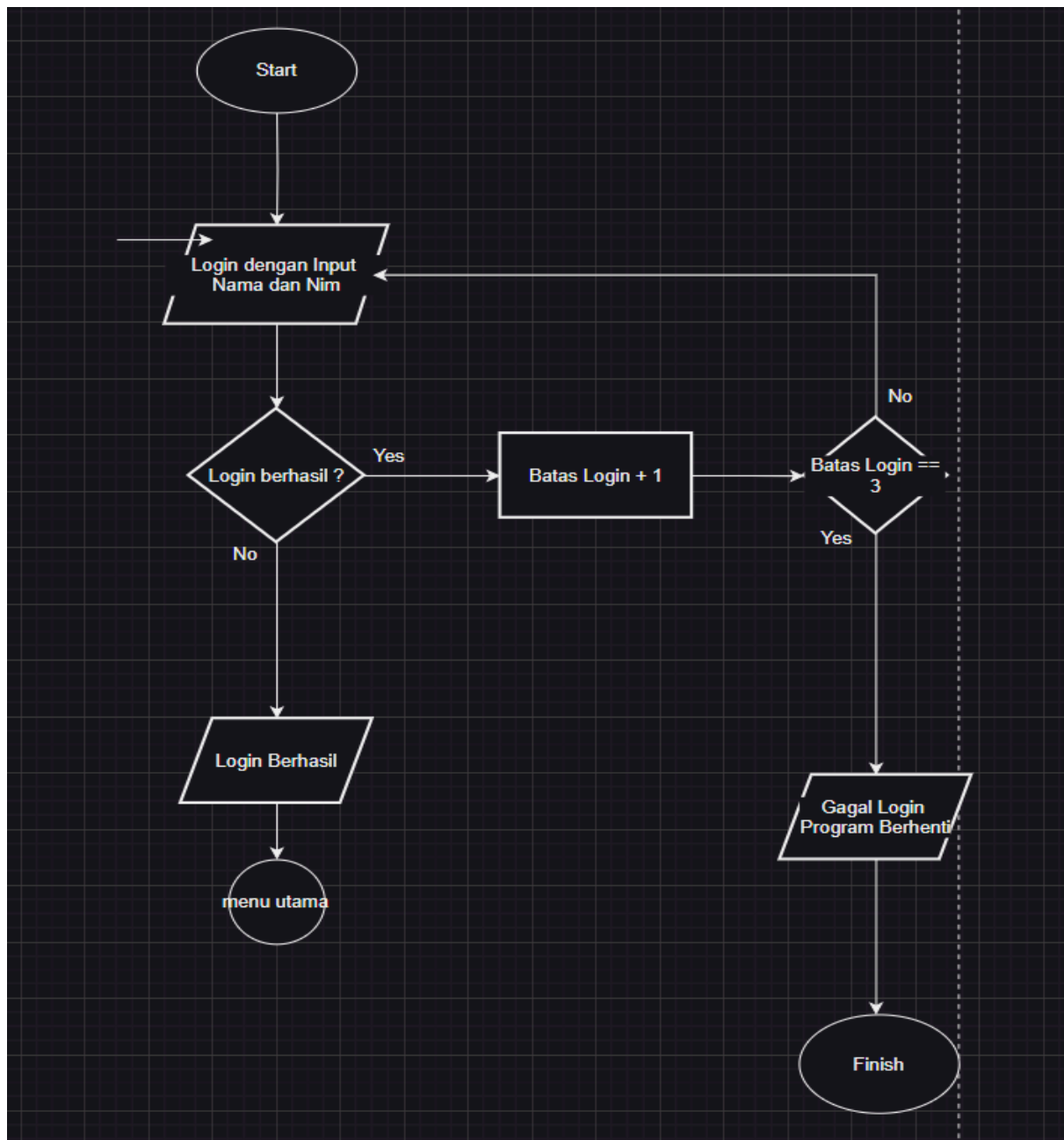
**LAPORAN PRAKTIKUM**  
**POSTTEST (2)**  
**ALGORITMA PEMROGRAMAN LANJUT**



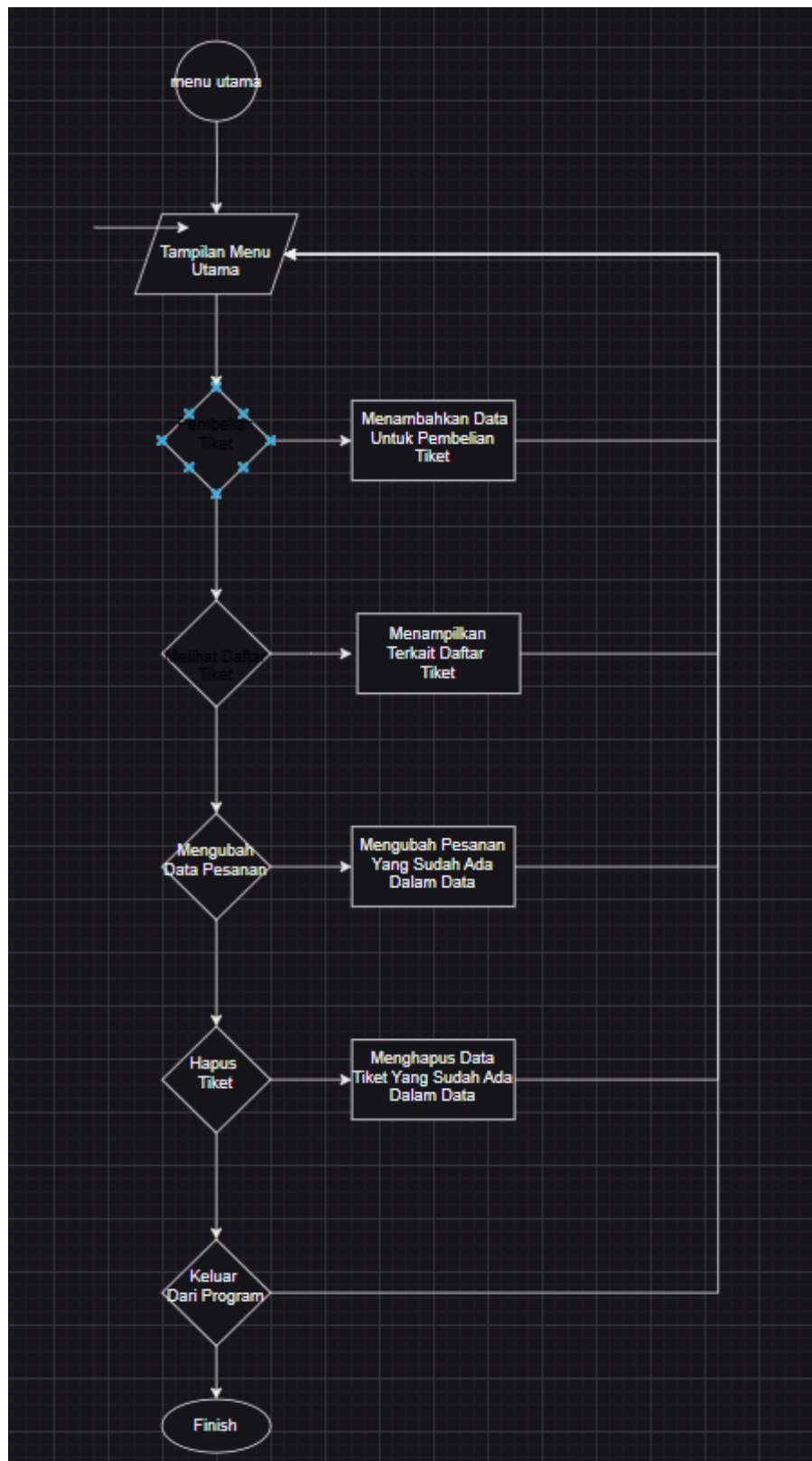
**Disusun oleh:**  
**ANDI FACHRY ALAM TENGKO (2409106006)**  
**Kelas (A1'24)**

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

## 1. Flowchart



Gambar 1.1 FC-Main



Gambar 1.2 FC-Menu-Utama

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Pada program kali ini, kita akan menjelaskan bagaimana program bekerja dari awal hingga akhir, mulai dari proses login, pemesanan tiket, hingga keluarnya program. Program ini merupakan sistem pemesanan tiket pesawat yang memiliki beberapa fitur utama: login, pemesanan tiket, melihat daftar tiket, mengubah data pemesanan, dan menghapus tiket.

#### **1. Proses Login**

Pada bagian pertama program, pengguna diminta untuk melakukan login. Program akan meminta input nama dan NIM pengguna. Sistem hanya akan menerima login jika nama dan NIM yang dimasukkan sesuai dengan data yang sudah ditentukan sebelumnya, yaitu:

- Nama: "Andi Fachry Alam Tengko"
- NIM: "006"

Proses login dilakukan dengan percobaan sebanyak 3 kali. Jika login gagal setelah 3 kali percobaan, program akan berhenti. Berikut adalah langkah-langkahnya:

- Input Nama dan NIM: Program meminta pengguna memasukkan nama dan NIM.
- Validasi: Setelah itu, program akan memeriksa apakah nama dan NIM yang dimasukkan sesuai dengan yang benar.
- Percobaan Login: Jika salah, program akan meminta pengguna untuk mencoba lagi. Jika berhasil login, program akan lanjut ke menu utama.

#### **2. Menu Utama Sistem Pemesanan Tiket**

Setelah login berhasil, program akan menampilkan menu dengan pilihan-pilihan berikut:

1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar

Program akan menerima input pilihan menu dari pengguna dan akan mengeksekusi fungsi terkait berdasarkan pilihan tersebut.

### 3. Membeli Tiket

Jika pengguna memilih menu Membeli Tiket (pilihan 1), berikut adalah langkah-langkahnya:

- Cek Kapasitas: Program pertama-tama akan memeriksa apakah tiket masih tersedia. Jika sudah penuh (jumlah tiket mencapai kapasitas maksimum yaitu MAX\_TIKET), program akan memberi tahu pengguna bahwa tiket sudah penuh.
- Input Data Pemesan: Jika tiket masih tersedia, pengguna diminta untuk memasukkan:
  - Nama Pemesan
  - Tujuan Penerbangan
  - Tanggal Penerbangan (dalam format DD-MM-YYYY)
- Proses Penyimpanan Tiket: Data yang dimasukkan akan disimpan dalam array tiketPesawat dan jumlah tiket (jumlahTiket) akan bertambah 1.

### 4. Melihat Daftar Tiket

Jika pengguna memilih menu Melihat Daftar Tiket (pilihan 2), berikut adalah langkah-langkahnya:

- Program akan memeriksa apakah ada tiket yang sudah dipesan. Jika belum ada tiket yang dipesan ( $\text{jumlahTiket} == 0$ ), program akan menampilkan pesan bahwa belum ada tiket yang dipesan.
- Jika ada tiket yang dipesan, program akan menampilkan daftar tiket yang sudah dipesan, termasuk informasi nama pemesan, tujuan, dan tanggal penerbangan untuk setiap tiket yang terdaftar.

### 5. Mengubah Data Pemesanan

Jika pengguna memilih menu Mengubah Data Pemesanan (pilihan 3), berikut adalah langkah-langkahnya:

- Program akan meminta nomor tiket yang ingin diubah.

- Validasi Nomor Tiket: Program memeriksa apakah nomor tiket yang dimasukkan valid (apakah berada dalam rentang tiket yang sudah dipesan).
- Input Perubahan: Jika tiket valid, program akan menampilkan data tiket yang ada dan meminta pengguna untuk memasukkan data baru (nama, tujuan, dan tanggal penerbangan).
- Penyimpanan Perubahan: Setelah data baru dimasukkan, tiket akan diperbarui sesuai dengan input baru dari pengguna.

## 6. Menghapus Tiket

Jika pengguna memilih menu Menghapus Tiket (pilihan 4), berikut adalah langkah-langkahnya:

- Program akan meminta nomor tiket yang ingin dihapus.
- Validasi Nomor Tiket: Program memeriksa apakah nomor tiket yang dimasukkan valid.
- Proses Penghapusan: Jika tiket valid, program akan menggeser data tiket setelah tiket yang dihapus untuk mengisi tempat yang kosong, dan mengurangi jumlah tiket yang dipesan sebanyak 1.

## 7. Keluar dari Program

Jika pengguna memilih menu Keluar (pilihan 5), program akan menampilkan pesan terima kasih dan kemudian berhenti.

## 8. Alur Program secara Keseluruhan

- Program dimulai dengan percakapan login.
- Setelah berhasil login, program menampilkan menu utama yang dapat diakses berulang kali hingga pengguna memilih untuk keluar (pilihan 5).
- Pada setiap pilihan menu, program akan melakukan operasi terkait (membeli tiket, melihat tiket, mengubah data tiket, menghapus tiket) sesuai dengan input pengguna.
- Program akan terus berjalan dalam sebuah loop hingga pilihan 5 dipilih (keluar).
- Pada akhir program, jika login gagal setelah 3 percobaan atau setelah keluar, program berhenti.

## Penjelasan Blok Kode atau Fungsi:

### 1. Deklarasi dan Struktur Tiket:

- `const int MAX_TIKET = 10;`: Mendefinisikan kapasitas maksimum tiket.
- `struct Tiket`: Menyimpan data tiket (nama, tujuan, tanggal).
- `Tiket tiketPesawat[MAX_TIKET];`: Array untuk menyimpan data tiket.
- `int jumlahTiket = 0;`: Menyimpan jumlah tiket yang sudah dipesan.

### 2. Login:

- Pengguna memasukkan nama dan NIM.
- Program membandingkan input dengan data yang benar, memberikan percobaan login hingga 3 kali.

### 3. Menu Utama:

- Program menampilkan menu pilihan yang berisi 5 opsi dan menunggu input pengguna.
- Setiap pilihan menu akan menjalankan fungsi yang sesuai (membeli tiket, melihat daftar tiket, mengubah tiket, menghapus tiket).

### 4. Membeli Tiket:

- Input nama, tujuan, dan tanggal penerbangan untuk tiket baru.
- Mengecek kapasitas tiket, menambah jumlah tiket yang sudah dipesan.

### 5. Melihat Daftar Tiket:

- Menampilkan daftar tiket yang sudah dipesan.

### 6. Mengubah dan Menghapus Tiket:

- Mengubah data tiket atau menghapus tiket berdasarkan nomor tiket yang diberikan.

### 7. Keluar:

- Program berhenti ketika memilih opsi 5.

## 2.2 Penjelasan Alur & Algoritma

Program dimulai dengan proses login, di mana pengguna harus memasukkan nama dan NIM yang telah ditentukan. Jika input yang dimasukkan salah, pengguna diberikan kesempatan hingga tiga kali untuk mencoba kembali. Jika gagal dalam tiga percobaan, program akan berhenti. Jika berhasil, pengguna akan diarahkan ke menu utama.

Setelah berhasil login, pengguna dapat memilih berbagai opsi dalam menu utama, seperti membeli tiket, melihat daftar tiket yang sudah dipesan, mengubah data tiket, menghapus tiket, atau keluar dari program.

Dalam proses pembelian tiket, sistem akan memeriksa apakah masih ada slot tiket yang tersedia. Jika sudah penuh, pembelian tidak dapat dilakukan. Jika masih tersedia, pengguna akan diminta untuk memasukkan nama pemesan, tujuan penerbangan, dan tanggal keberangkatan dalam format DD-MM-YYYY. Setelah itu, data tiket akan disimpan, dan jumlah tiket yang sudah dipesan akan bertambah.

Jika pengguna ingin melihat daftar tiket yang telah dipesan, program akan menampilkan semua tiket yang tersedia. Jika belum ada tiket yang dipesan, sistem akan memberi tahu pengguna bahwa daftar tiket masih kosong.

Untuk mengubah data pemesanan tiket, pengguna harus memasukkan nomor tiket yang ingin diperbarui. Program akan memeriksa apakah nomor tiket tersebut valid. Jika valid, program akan menampilkan data tiket yang ada dan meminta pengguna untuk memasukkan data baru, seperti nama, tujuan, dan tanggal keberangkatan. Setelah perubahan dikonfirmasi, tiket akan diperbarui sesuai dengan input yang diberikan.

Jika pengguna ingin menghapus tiket, mereka harus memasukkan nomor tiket yang ingin dihapus. Program akan memeriksa apakah nomor tiket tersebut valid. Jika valid, tiket akan dihapus, dan data tiket lainnya akan digeser untuk mengisi slot yang kosong. Jumlah tiket yang tersedia akan berkurang.

Program akan terus berjalan hingga pengguna memilih opsi keluar. Jika pengguna memilih keluar, program akan menampilkan pesan terima kasih sebelum berhenti.

#### Source Code:

```
int main() {  
    // Proses Login  
    string namaInput, nimInput;  
    string namaBenar = "Andi Fachry Alam Tengko";  
    string nimBenar = "006";  
    int attempts = 0;  
    bool loginBerhasil = false;  
  
    // Percobaan Login sebanyak 3 kali  
    while (attempts < 3) {  
        cout << "Masukkan Nama: ";  
        getline(cin, namaInput);  
  
        cout << "Masukkan NIM: ";  
        cin >> nimInput;
```



```

    cin.ignore();

    if (namaInput == namaBenar && nimInput == nimBenar) {
        loginBerhasil = true;
        break;
    } else {
        cout << "Nama atau NIM salah! Coba lagi.\n";
        attempts++;
    }
}

if (!loginBerhasil) {
    cout << "Login gagal setelah 3 kali percobaan. Program berhenti.\n";
    return 0;
}

int pilihan;
do {
    cout << "\nMenu Sistem Pemesanan Tiket Pesawat\n";
    cout << "1. Membeli Tiket\n";
    cout << "2. Melihat Daftar Tiket\n";
    cout << "3. Mengubah Data Pemesanan\n";
    cout << "4. Menghapus Tiket\n";
    cout << "5. Keluar\n";
    cout << "Pilih menu: ";
    cin >> pilihan;
    cin.ignore();

    switch (pilihan) {
        case 1: {
            if (jumlahTiket >= 100) {
                cout << "Tiket sudah penuh!\n";
                break;
            }

            cout << "Masukkan Nama Pemesan: ";
            getline(cin, namaPemesan[jumlahTiket]);

            cout << "Masukkan Tujuan: ";
            getline(cin, tujuanPenerbangan[jumlahTiket]);

            cout << "Masukkan Tanggal Penerbangan (DD-MM-YYYY): ";
            getline(cin, tanggalPenerbangan[jumlahTiket]);

            jumlahTiket++;
            cout << "Tiket berhasil dipesan!\n";
            break;
        }
    }
}

```

```

        case 2: {
            if (jumlahTiket == 0) {
                cout << "Belum ada tiket yang dipesan.\n";
                break;
            }

            cout << "Daftar Tiket yang sudah dipesan:\n";
            for (int i = 0; i < jumlahTiket; i++) {
                cout << "Tiket " << (i + 1) << " - Nama: " <<
namaPemesan[i]
                << ", Tujuan: " << tujuanPenerbangan[i]
                << ", Tanggal: " << tanggalPenerbangan[i] << endl;
            }
            break;
        }
        case 3: {
            int nomor;
            cout << "Masukkan nomor tiket yang ingin diubah (1-" <<
jumlahTiket << "): ";
            cin >> nomor;
            cin.ignore();

            if (nomor < 1 || nomor > jumlahTiket) {
                cout << "Tiket tidak ditemukan.\n";
                break;
            }

            cout << "Data tiket yang akan diubah:\n";
            cout << "Nama: " << namaPemesan[nomor - 1] << endl;
            cout << "Tujuan: " << tujuanPenerbangan[nomor - 1] << endl;
            cout << "Tanggal: " << tanggalPenerbangan[nomor - 1] <<
endl;

            cout << "Masukkan Nama Baru: ";
            getline(cin, namaPemesan[nomor - 1]);

            cout << "Masukkan Tujuan Baru: ";
            getline(cin, tujuanPenerbangan[nomor - 1]);

            cout << "Masukkan Tanggal Baru: ";
            getline(cin, tanggalPenerbangan[nomor - 1]);

            cout << "Tiket berhasil diperbarui!\n";
            break;
        }
        case 4: {
            int nomor;
            cout << "Masukkan nomor tiket yang ingin dihapus (1-" <<

```

```

jumlahTiket << "): ";
    cin >> nomor;
    cin.ignore();

    if (nomor < 1 || nomor > jumlahTiket) {
        cout << "Tiket tidak ditemukan.\n";
        break;
    }

    for (int i = nomor - 1; i < jumlahTiket - 1; i++) {
        namaPemesan[i] = namaPemesan[i + 1];
        tujuanPenerbangan[i] = tujuanPenerbangan[i + 1];
        tanggalPenerbangan[i] = tanggalPenerbangan[i + 1];
    }

    jumlahTiket--;
    cout << "Tiket berhasil dihapus!\n";
    break;
}
case 5: {
    cout << "Terima kasih, program selesai.\n";
    break;
}
default: {
    cout << "Pilihan tidak valid, coba lagi.\n";
}
}
} while (pilihan != 5);

return 0;
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

#### 1. Skenario 1

Pada skenario ini user memasukkan input sesuai dengan aturan program, program berjalan sesuai dengan senormalnya.

#### 2. Skenario 2

Pada skenario ini user salah menginput password ataupun username sebanyak tiga kali dan program akan berhenti otomatis.

### 4.2 Hasil Output

```
Masukkan Nama: Andi Fachry Alam Tengko
Masukkan NIM: 006

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 1
Masukkan Nama Pemesan: Fachry
Masukkan Tujuan: Bontang
Masukkan Tanggal Penerbangan (DD-MM-YYYY): 22232006
Tiket berhasil dipesan!

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 5
Terima kasih, program selesai.
```

Gambar 4.1 Percobaan-1

```
Masukkan Nama: a
Masukkan NIM: a
Nama atau NIM salah! Coba lagi.
Masukkan Nama: a
Masukkan NIM: a
Nama atau NIM salah! Coba lagi.
Masukkan Nama: a
Masukkan NIM: a
Nama atau NIM salah! Coba lagi.
Login gagal setelah 3 kali percobaan. Program berhenti.
```

Gambar 4.2 Percobaan-2

## 5. Git

### 1. Git add

Digunakan untuk mengubah status file ke staging area, penggunaan `.` di akhir untuk mengubah semua file yang hijau.

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git add .
```

Gambar 5.1 Add

### 2. Git Commit

Digunakan untuk menaikkan file ke github dan siap untuk di push.

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git commit -m "Finish Post Test 2"
[main 9a8132c] Finish Post Test 2
3 files changed, 149 insertions(+)
create mode 100644 post-test/post-test-apl-2/2409106006-AndiFachryAlamTengko-PT-2.cpp
create mode 100644 post-test/post-test-apl-2/2409106006-AndiFachryAlamTengko-PT-2.exe
create mode 100644 post-test/post-test-apl-2/2409106006-AndiFachryAlamTengko-PT-2.pdf
```

Gambar 5.2 Commit

### 3. Git Push

Digunakan agar file yang sudah di commit bisa dilihat oleh publik.

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.07 MiB | 717.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AndiFachryAlamTengko/praktikum-apl.git
   59c9ec7..9a8132c  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Push