

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT

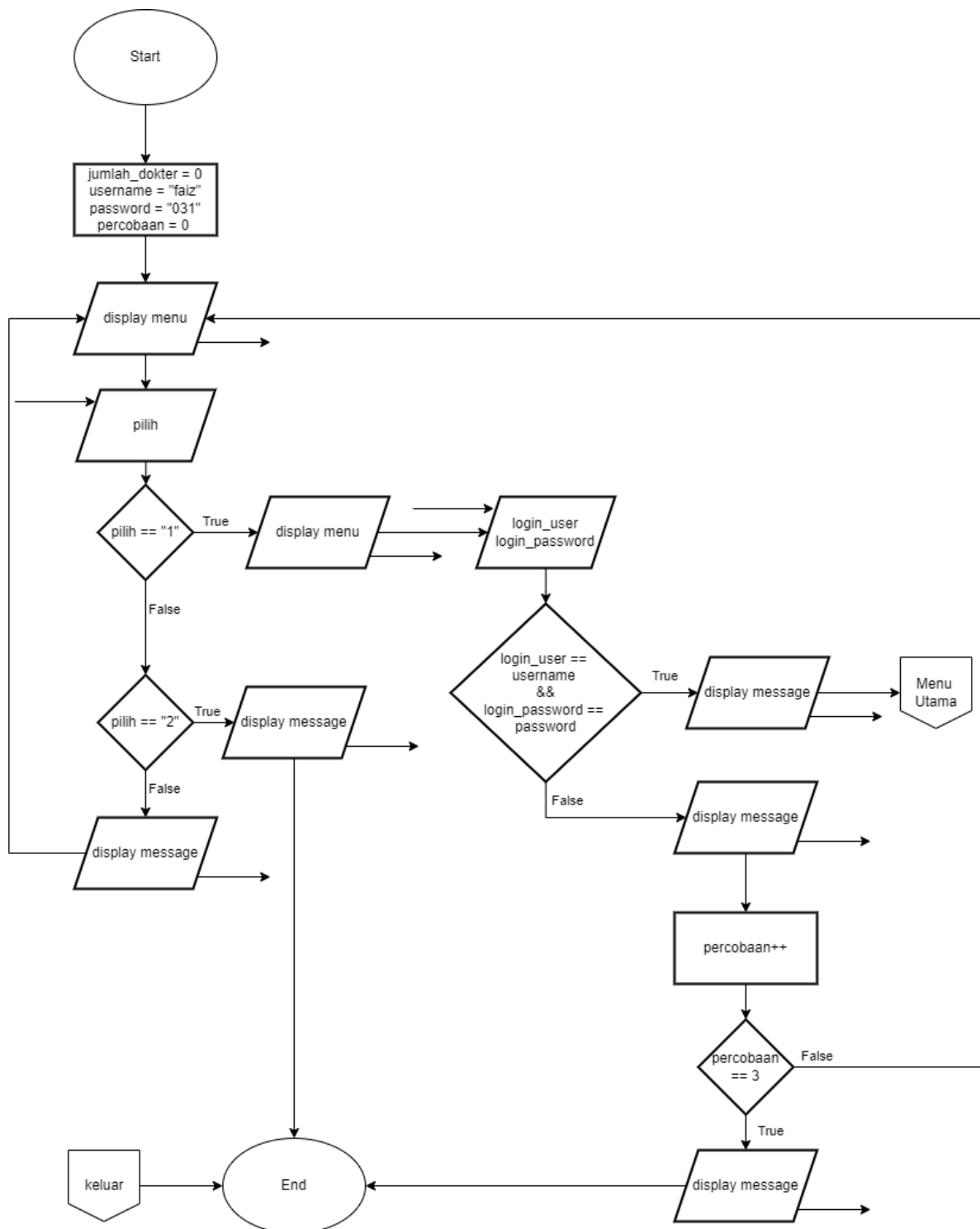


Disusun oleh:
Muhammad Faiz Lazuardi (2409106031)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

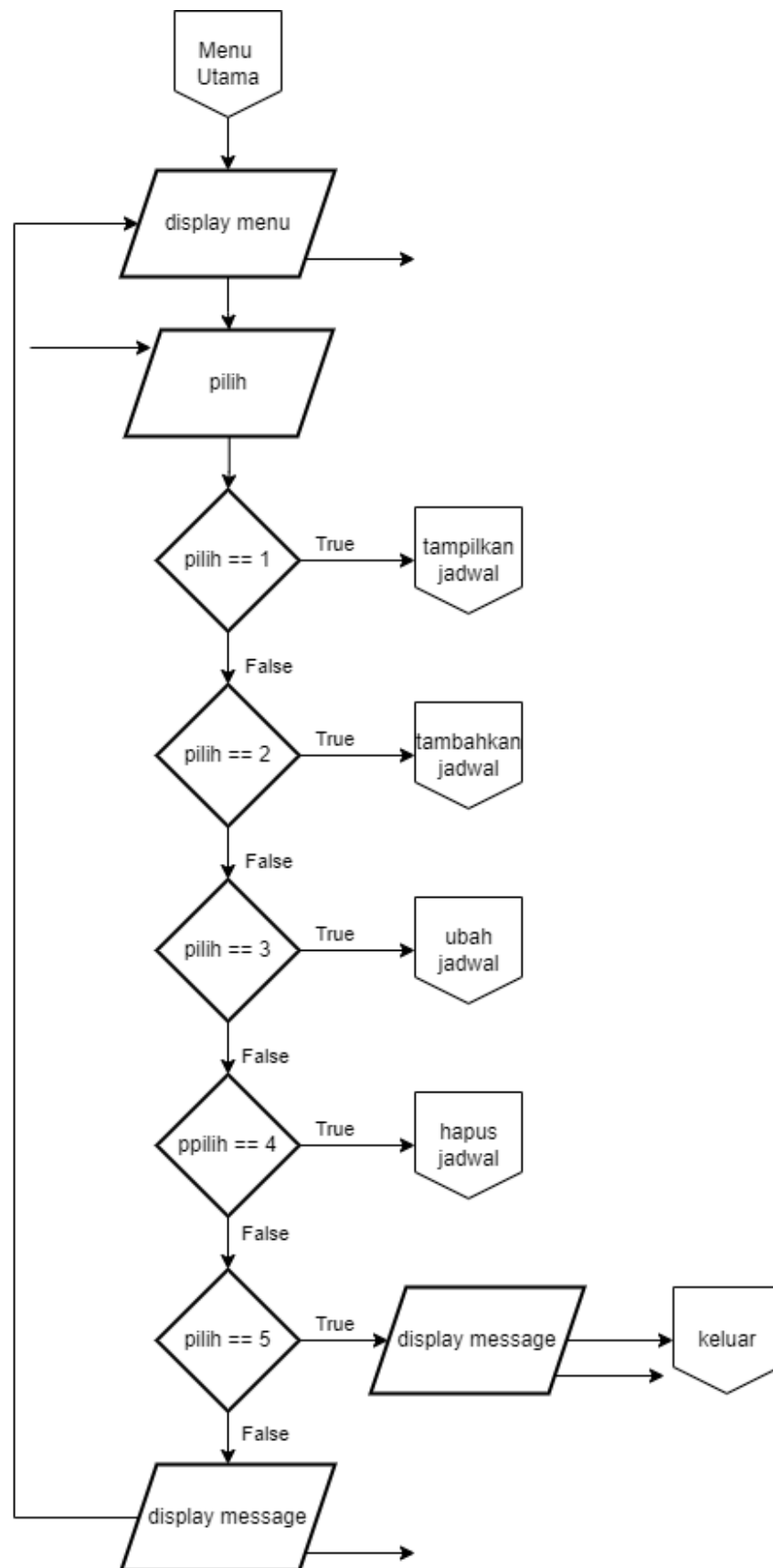
1. Flowchart

1.1 Menu Login



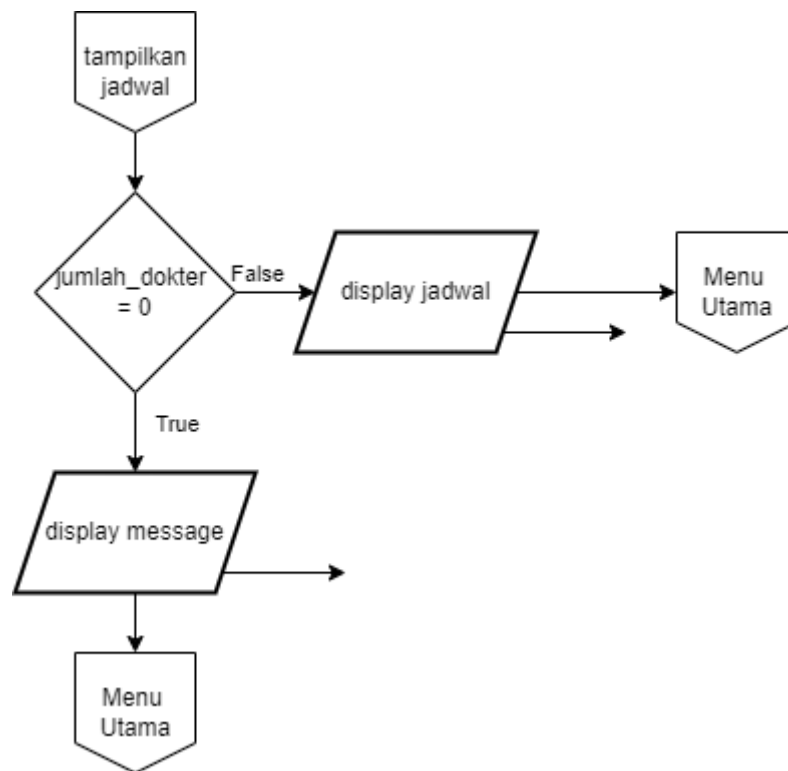
Gambar 1.1 Menu Login

1.2 Menu Utama



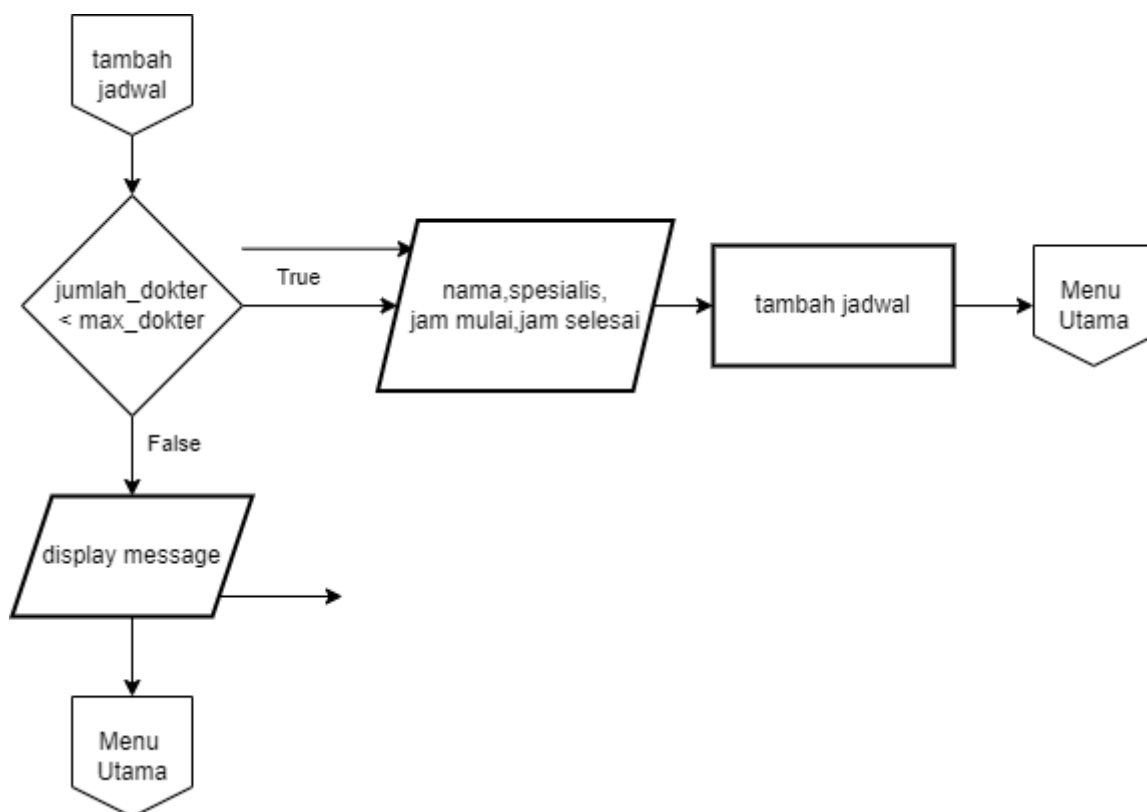
Gambar 1.2 Menu Utama

1.3 Menu Read



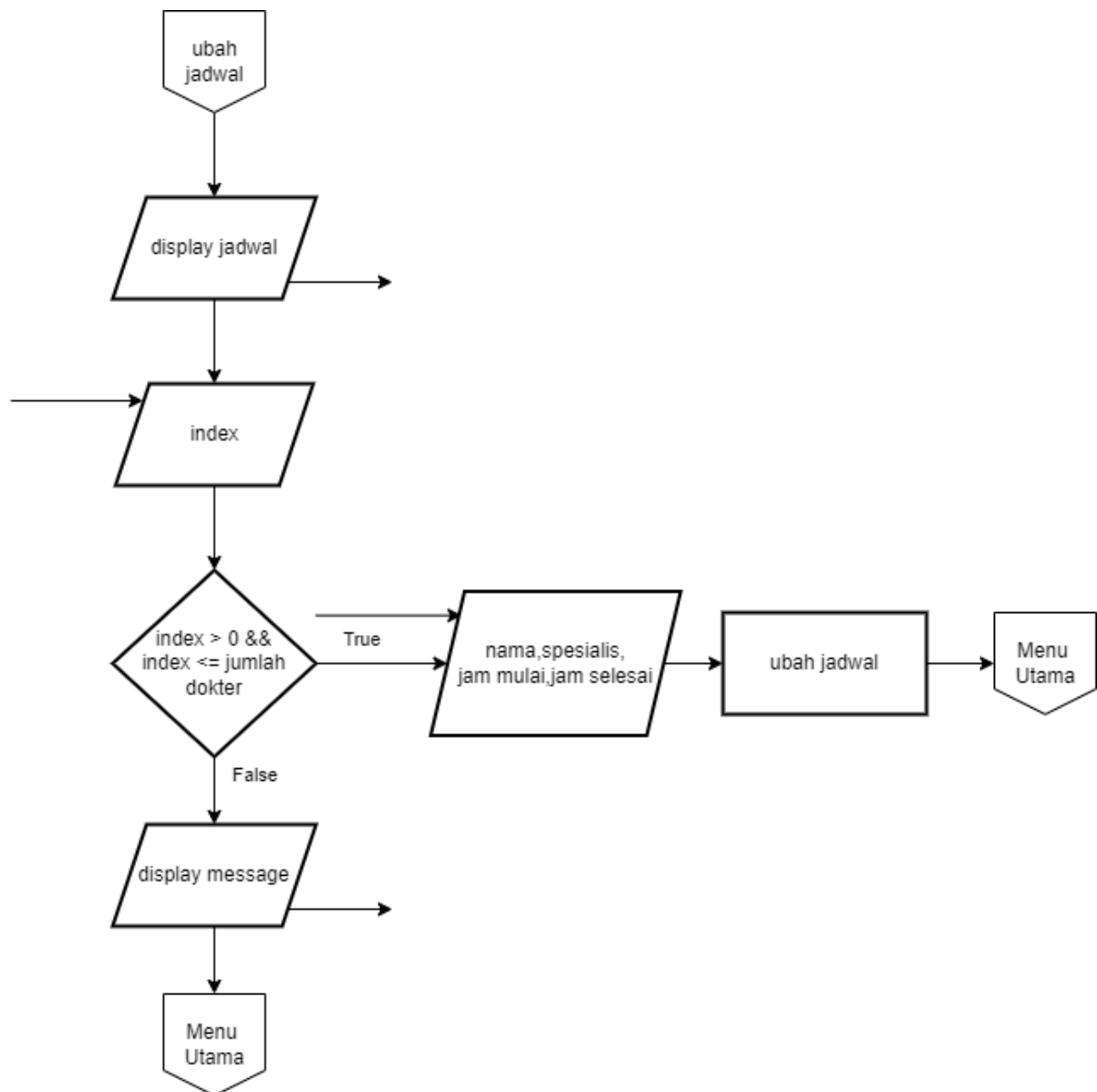
Gambar 1.3 Menu Read

1.4 Menu Create



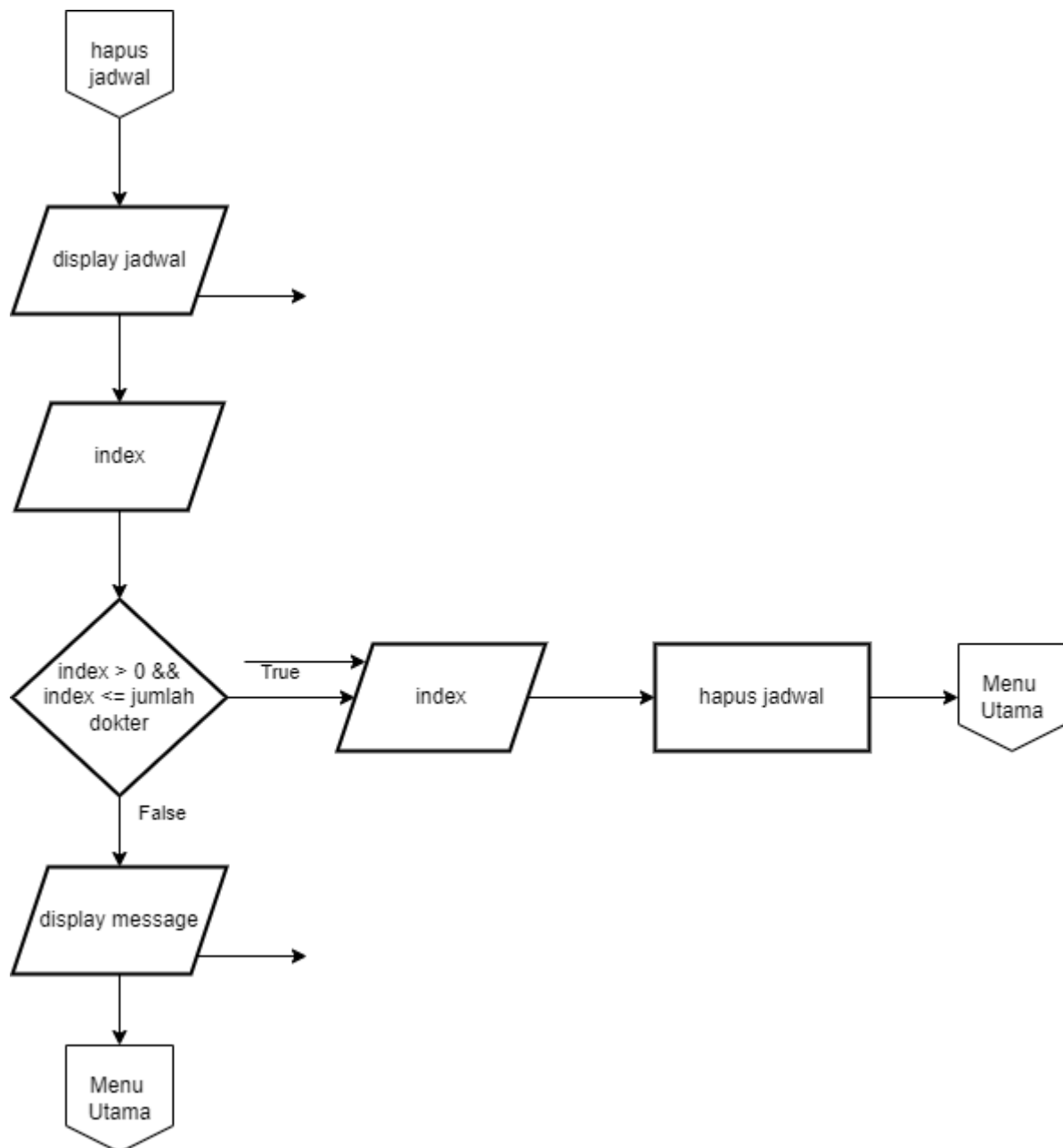
Gambar 1.4 Menu Create

1.5 Menu Update



Gambar 1.5 Menu Update

1.6 Menu Delete



Gambar 1.6 Menu Delete

2. Analisis Program

2.1 Deskripsi Singkat Program

Sebuah program yang berfungsi untuk memanage jadwal praktek dokter spesialis.

2.2 Penjelasan Alur & Algoritma

2.2.1 Alur Program

1. Menampilkan Menu Login.
 - Pilihan 1: Login
 - Pilihan 2: Keluar
2. Meminta Pengguna Untuk Input :
 - a. Jika input adalah 1.
 - i. Maka pengguna diminta untuk memasukkan input berupa username dan password.
 - ii. Jika Berhasil lanjut tahap ke 3.
 - iii. Jika Gagal :
 1. Jika percobaan kurang dari 3 maka ulangi langkah pertama.
 2. Jika percobaan sama dengan 3 maka program dihentikan.
 - b. Jika input adalah 2.
 - i. Maka program dihentikan.
 - c. Jika input bukan 1 atau 2.
 - i. Maka Kembali ke tahap 1.
3. Menampilkan Menu Utama
 - Pilihan 1: Tampilkan Jadwal
 - Pilihan 2: Tambah Jadwal
 - Pilihan 3: Ubah Jadwal
 - Pilihan 4: Hapus Jadwal
 - Pilihan 5: Keluar

4. Meminta pengguna untuk input :
 - a. Jika input adalah 1.
 - i. Menampilkan jadwal.
 - b. Jika input adalah 2.
 - i. Melakukan proses menambahkan jadwal.
 - c. Jika input adalah 3.
 - i. Melakukan proses mengubah jadwal.
 - d. Jika input adalah 4.
 - i. Melakukan proses menghapus jadwal.
 - e. Jika input adalah 5 maka program dihentikan.

2.2.2 Algoritma Program

1. Mulai
2. Deklarasi Variabel :
 - string : username = "faiz", password = "031", login_username, login_password, pilih, pilih_konversi, nama_dokter[MAX_DOKTER], spesialis[MAX_DOKTER], jadwal_mulai[MAX_DOKTER], jadwal_selesai[MAX_DOKTER]
 - int : percobaan = 0, jumlah_dokter = 0, MAX_DOKTER = 10
3. Tampilkan Menu Login
4. Proses Login :

Input login_username and login_password.

 - a. Jika input sama dengan username dan password maka login berhasil dan lanjut ke menu utama.
 - b. Jika input salah maka percobaan + 1 dan kembali ke menu login.
 - c. Jika percobaan = 3 maka program dihentikan.
5. Tampilkan menu utama.

Input pilih

 - a. Jika pilih adalah 1 maka menampilkan jadwal.
 - b. Jika pilih adalah 2 maka menambahkan jadwal.
 - c. Jika pilih adalah 3 maka mengubah jadwal.
 - d. Jika pilih adalah 4 maka menghapus jadwal.

- e. Jika pilih adalah 5 maka program dihentikan.
- 6. Selesai

3. Source Code

A. Login

Fitur ini digunakan sebagai syarat untuk menjalankan program manajemen jadwal praktek dokter spesialis. Data yang diminta berupa username dan password.

Source Code:

```
int login(){
    int percobaan = 0;
    string username = "faiz", password = "031", login_user, login_password,
    pilih;
    while (percobaan < 3) {
        cout << "<==== Home Page =====>\n";
        cout << "1. Login\n";
        cout << "2. Keluar\n";
        cout << "Pilih Fitur: "; cin >> pilih;

        if(pilih == "1"){
            cout << "\n<==== Login Page =====>\n";
            cout << "Username: "; cin >> login_user;
            cout << "Password: "; cin >> login_password;
            system("cls");

            if (login_user == username and login_password == password) {
                cout << "Login berhasil!\n";
                return 0;
            } else {
                cout << "Username atau password salah!\n";
                percobaan++;
                if (percobaan == 3) {
                    cout << "Login gagal! Program Berhenti.\n";
                    exit(0);
                }
            }
        } else if(pilih == "2"){
            cout << "Keluar dari program.\n";
            exit(0);
        } else {
            cout << "Input Yang Anda Pilih Tidak Valid. Silahkan Coba
            lagi.\n\n";
        }
    }
    return 1;
}
```

B. Menu Utama

Menu Utama berisi beberapa pilihan fitur. Pilihan 1 sampai 4 merupakan fitur CRUD dan pilihan 5 merupakan fitur keluar dari program.

Source Code:

```
cout << "\n<==== Jadwal Praktik Dokter <====>\n";
cout << "1. Tampilkan Jadwal\n";
cout << "2. Tambah Jadwal\n";
cout << "3. Ubah Jadwal\n";
cout << "4. Hapus Jadwal\n";
cout << "5. Keluar\n";
cout << "Pilih Fitur: "; cin >> pilih;
```

C. Pilihan 1

Menampilkan jadwal.

Source Code:

```
int tampilkan_jadwal() {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
        return 0;
    }
    for(int i = 0; i < jumlah_dokter; ++i){
        cout << i + 1 << ". Nama: " << nama_dokter[i] << endl;
        cout << "    " << "Spesialis: " << spesialis[i] << endl;
        cout << "    " << "Jam Mulai: " << jadwal_mulai[i] << endl;
        cout << "    " << "Jam Selesai: " << jadwal_selesai[i] << endl;
        cout << "    " << endl;
    }
    return 1;
}
```

D. Pilihan 2

Menambahkan jadwal.

Source Code:

```
bool tambah_jadwal() {
    if (jumlah_dokter < MAX_DOKTER) {
        cout << "Masukkan Nama Dokter: "; cin >> nama_dokter[jumlah_dokter];
        cout << "Masukkan Spesialis: "; cin >> spesialis[jumlah_dokter];
        cout << "Masukkan Jam Mulai: "; cin >> jadwal_mulai[jumlah_dokter];
        cout << "Masukkan Jam Selesai: "; cin >> jadwal_selesai[jumlah_dokter];
        jumlah_dokter++;
        cout << "Jadwal berhasil ditambahkan!\n";

    } else {
        cout << "Jadwal Penuh.\n";
    }
    return 1;
}
```

E. Pilihan 3

Mengubah jadwal.

Source Code:

```
bool ubah_jadwal() {
    tampilkan_jadwal();
    int index;
    cout << "Pilih Jadwal Yang Ingin Di Ubah: ";
    cin >> index;
    if (index > 0 && index <= jumlah_dokter) {
        cout << "Masukkan Nama Dokter baru: "; cin >> nama_dokter[index - 1];
        cout << "Masukkan Spesialis baru: "; cin >> spesialis[index - 1];
        cout << "Masukkan Jam Mulai baru: "; cin >> jadwal_mulai[index - 1];
        cout << "Masukkan Jam Selesai baru: "; cin >> jadwal_selesai[index - 1];
        cout << "Jadwal berhasil diperbarui!\n";

    } else {
        cout << "Jadwal Tidak Valid!\n";
    }
    return 1;
}
```

F. Pilihan 4

Menghapus jadwal.

Source Code:

```
bool hapus_jadwal() {
    int index;
    tampilkan_jadwal();
    if(jumlah_dokter == 0) {
        return 0;
    } else {
        cout << "Pilih Jadwal Yang Ingin Di Hapus: ";
        cin >> index;
        if (index > 0 && index <= jumlah_dokter) {
            for (int i = index - 1; i < jumlah_dokter - 1; i++) {
                nama_dokter[i] = nama_dokter[i + 1];
                spesialis[i] = spesialis[i + 1];
                jadwal_mulai[i] = jadwal_mulai[i + 1];
                jadwal_selesai[i] = jadwal_selesai[i + 1];
            }
            jumlah_dokter--;
            cout << "Jadwal berhasil dihapus!\n";
        } else {
            cout << "Jadwal Tidak Valid!\n";
        }
    }
    return 1;
}
```

G. Program

Keseluruhan program.

Source Code:

```
int main() {
    login();
    while (true) {
        int pilih;
        cout << "\n<==== Jadwal Praktik Dokter >====>\n";
        cout << "1. Tampilkan Jadwal\n";
        cout << "2. Tambah Jadwal\n";
        cout << "3. Ubah Jadwal\n";
        cout << "4. Hapus Jadwal\n";
        cout << "5. Keluar\n";
        cout << "Pilih Fitur: "; cin >> pilih;
        system("cls");

        switch (pilih) {
            case 1:
                tampilkan_jadwal();
                break;
            case 2:
                tambah_jadwal();
                break;
            case 3:
                ubah_jadwal();
                break;
            case 4:
                hapus_jadwal();
                break;
            case 5:
                cout << "Terima kasih telah menggunakan program ini!\n";
                exit(0);
            default:
                cout << "Pilihan tidak valid! Coba lagi.\n";
        }
    }
    return 0;
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

4.1.1 Skenario 1 :

- Percobaan Login Gagal
- Program dihentikan

4.1.2 Skenario 2 :

- Percobaan Login Berhasil
- Input 2 (Menambah jadwal)
- Input data
- Input 1 (Menampilkan jadwal)
- Input 3 (Mengubah jadwal)
- Input 1 (Memilih index)
- Input data baru
- Input 1 (Menampilkan jadwal)
- Input 4 (Menghapus jadwal)
- Input 1 (Memilih index)
- Input 5 (Keluar dari program)

4.2 Hasil Output

4.2.1 Skenario 1 :

```
<===== Home Page =====>
1. Login
2. Keluar
Pilih Fitur: 1

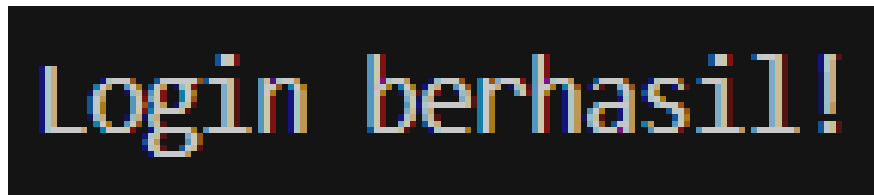
<===== Login Page =====>
Username: 1
Password: 1
```

Gambar 4.1 Percobaan Login

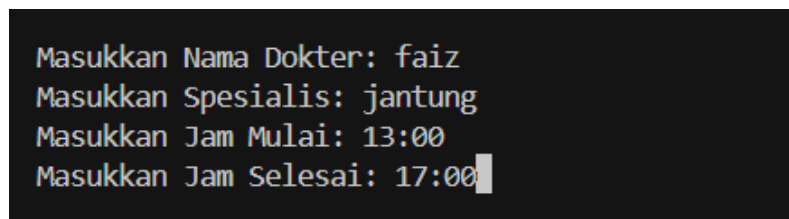
```
Username atau password salah!
Login gagal! Program Berhenti.
```

Gambar 4.2 Program Dihentikan

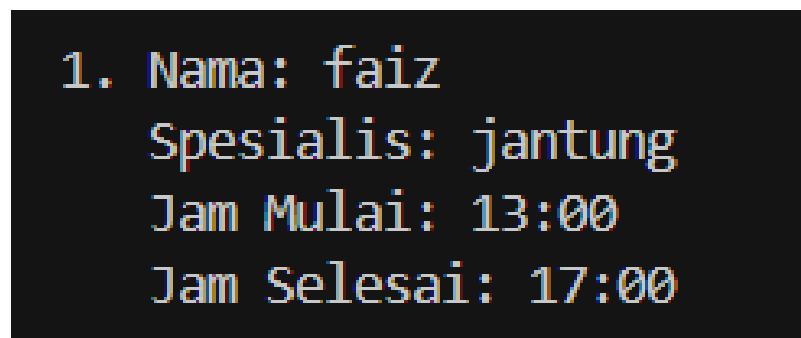
4.2.2 Skenario 2 :



Gambar 4.3 Login Berhasil



Gambar 4.4 Menambahkan jadwal



Gambar 4.5 Menampilkan Jadwal

```
1. Nama: faiz
   Spesialis: jantung
   Jam Mulai: 13:00
   Jam Selesai: 17:00

Pilih Jadwal Yang Ingin Di Ubah: 1
Masukkan Nama Dokter baru: joko
Masukkan Spesialis baru: jantung
Masukkan Jam Mulai baru: 08:30
Masukkan Jam Selesai baru: 10:45
Jadwal berhasil diperbarui!
```

Gambar 4.6 Mengubah Jadwal

```
1. Nama: joko
   Spesialis: jantung
   Jam Mulai: 08:30
   Jam Selesai: 10:45

Pilih Jadwal Yang Ingin Di Hapus: 1
Jadwal berhasil dihapus!
```

Gambar 4.7 Menghapus Jadwal

```
Terima kasih telah menggunakan program ini!
PS C:\Users\ASUS\Desktop\praktikum-apl\post-test\post-test-2> █
```

Gambar 4.8 Keluar Dari Program

5. Git

5.1 Git Init

Git init merupakan command yang berfungsi untuk menginisiasi repository yang ada pada file lokal dan berlokasi di folder .git. git branch -m master main merupakan command yang berfungsi untuk merubah branch repository dari master ke main.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop
$ mkdir praktikum-apl

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop
$ cd praktikum-apl/

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl
$ pwd
/c/Users/ASUS/Desktop/praktikum-apl

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl
$ git init
Initialized empty Git repository in C:/Users/ASUS/Desktop/praktikum-apl/.git/

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl (master)
$ git branch -m master main

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl (main)
```

Gambar 5.1 Git init dan Change branch

5.2 Git Add

Menambahkan file yang ingin di commit.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-2 (main)
$ git add 2409106031-MuhammadFaizLazuardi-PT-2.cpp
```

Gambar 5.2 Membuat File dan Git add

5.3 Git Commit

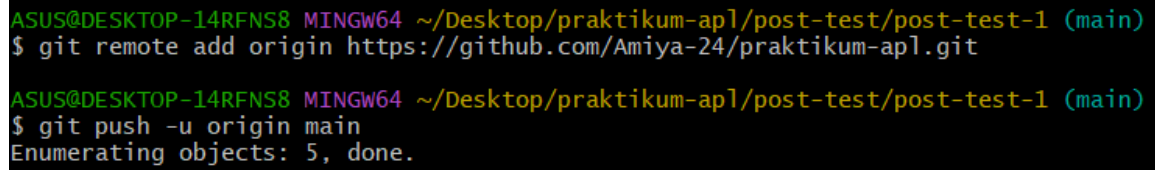
Menyimpan perubahan yang disimpan pada folder lokal dengan sebuah pesan.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "first commit"
[main 8e6c032] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 post-test/post-test-2/2409106031-MuhammadFaizLazuardi-PT-2.cpp
```

Gambar 5.3 Git Commit

5.4 Git Remote dan Git Push

Git remote berfungsi untuk menghubungkan repository yang ada di lokal dengan repository cloud pada github. Git push berfungsi untuk mengupload semua yang ada di lokal ke github.

A screenshot of a Windows command prompt window showing two Git commands and their output. The first command is 'git remote add origin https://github.com/Amiya-24/praktikum-apl.git'. The second command is 'git push -u origin main', which outputs 'Enumerating objects: 5, done.'.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (main)
$ git remote add origin https://github.com/Amiya-24/praktikum-apl.git

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (main)
$ git push -u origin main
Enumerating objects: 5, done.
```

Gambar 5.4 Git Remote dan Git Push