

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

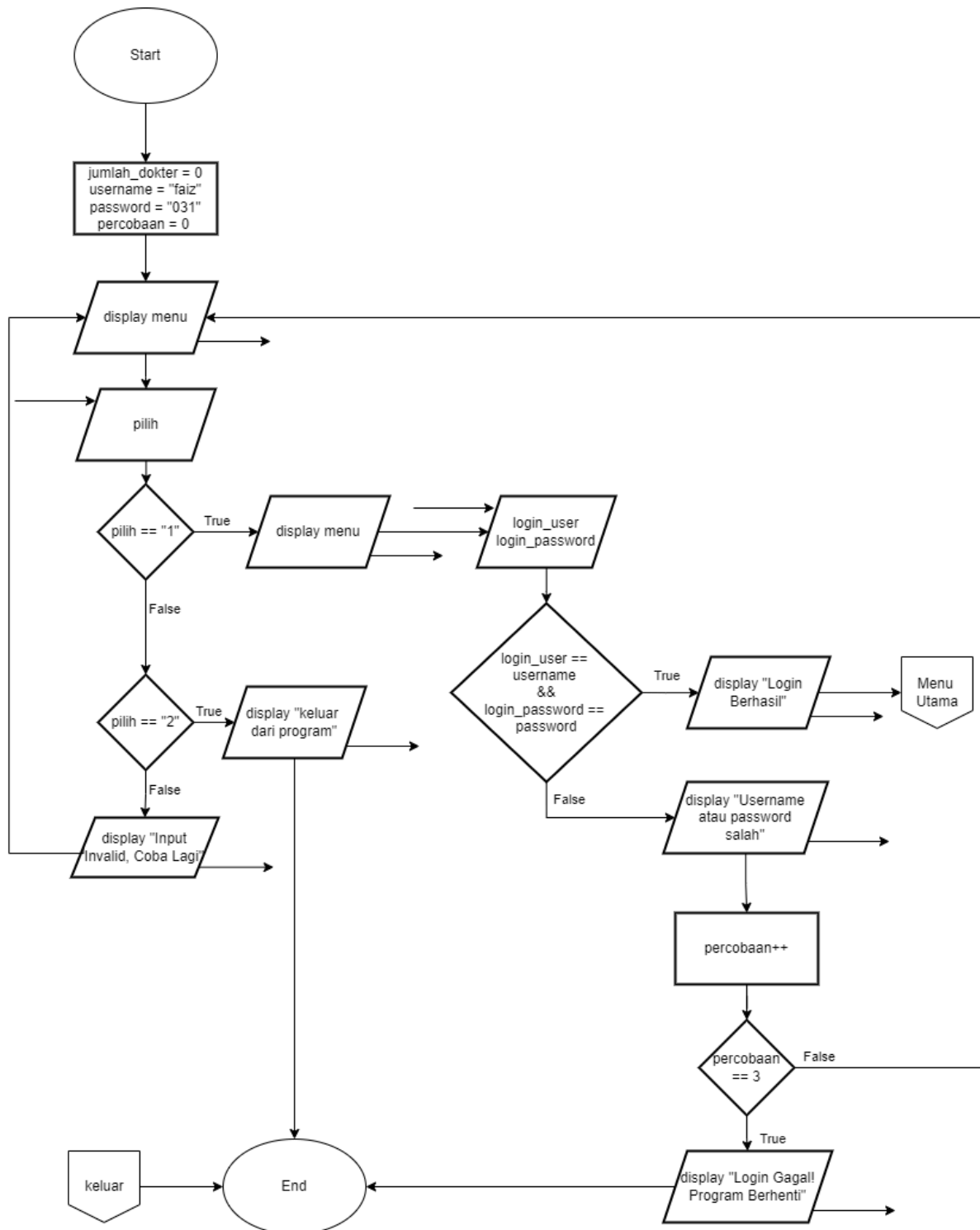


**Disusun oleh:**  
**Muhammad Faiz Lazuardi (2409106031)**  
**Kelas (A2 '24)**

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

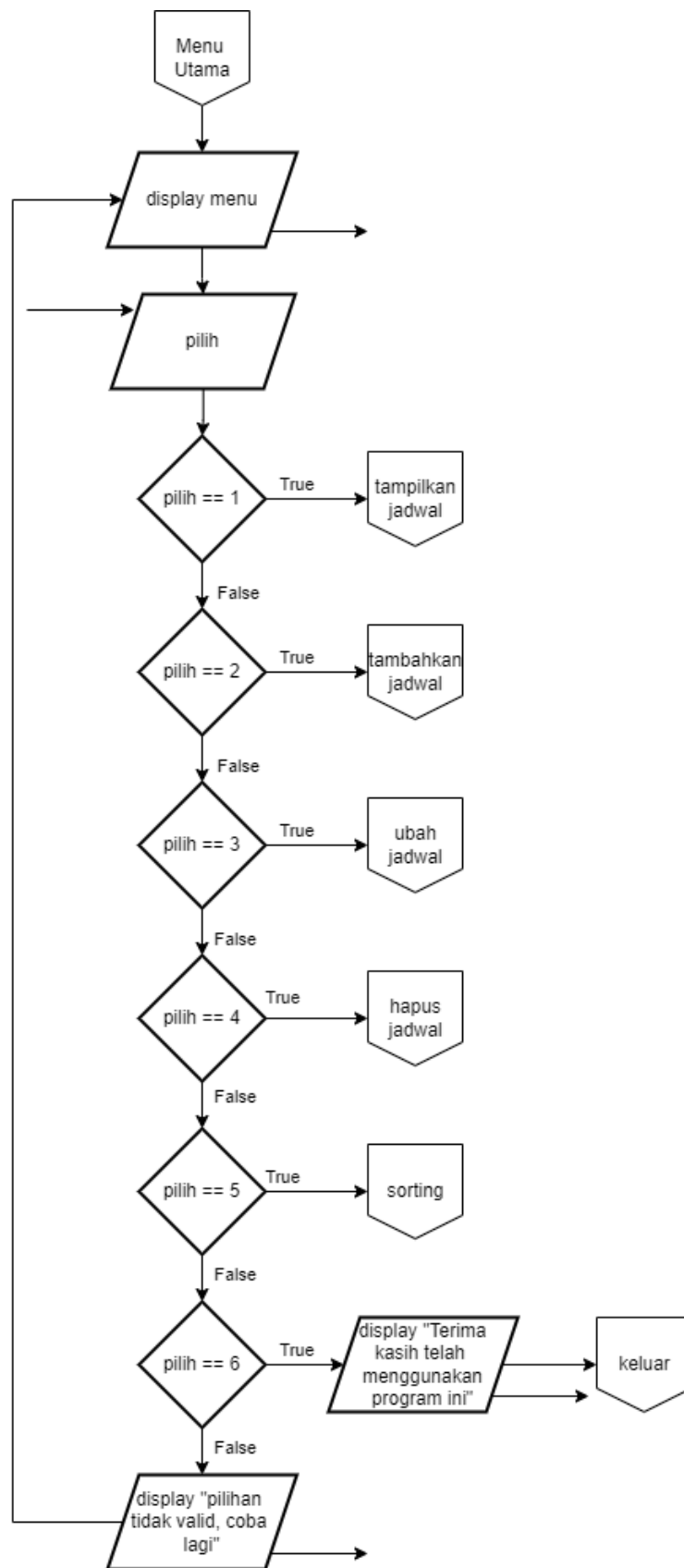
## 1. Flowchart

### 1.1 Menu Login



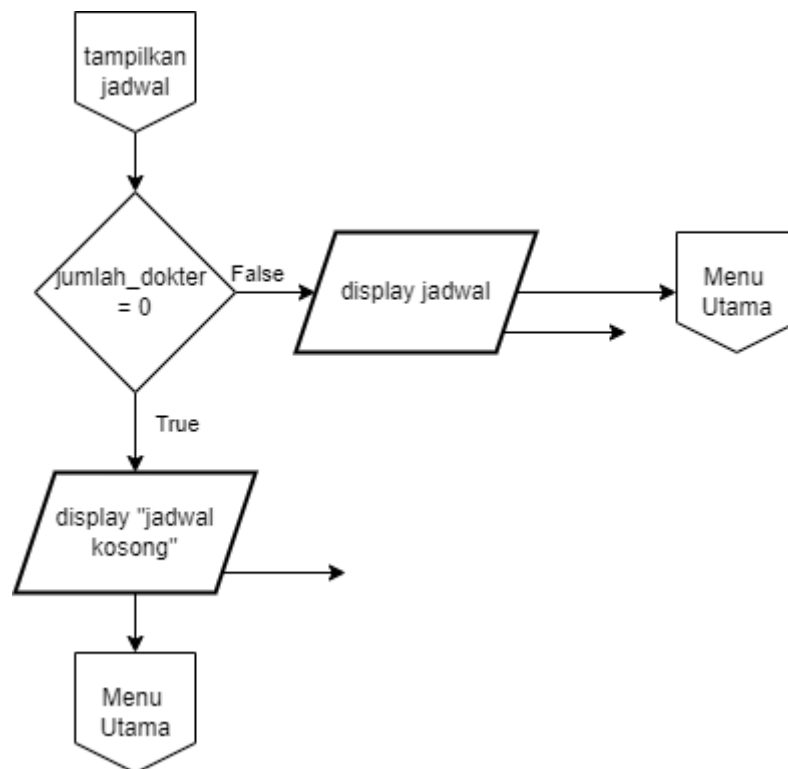
Gambar 1.1 Menu Login

## 1.2 Menu Utama



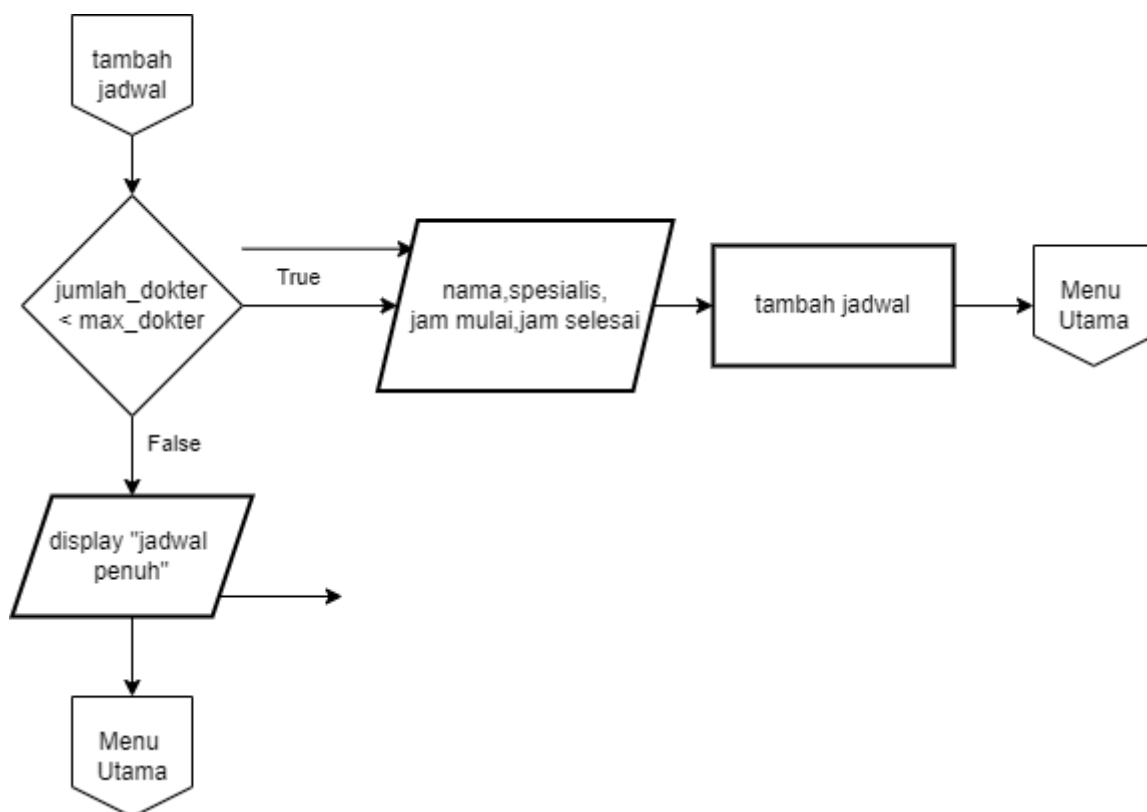
Gambar 1.2 Menu Utama

### 1.3 Menu Read



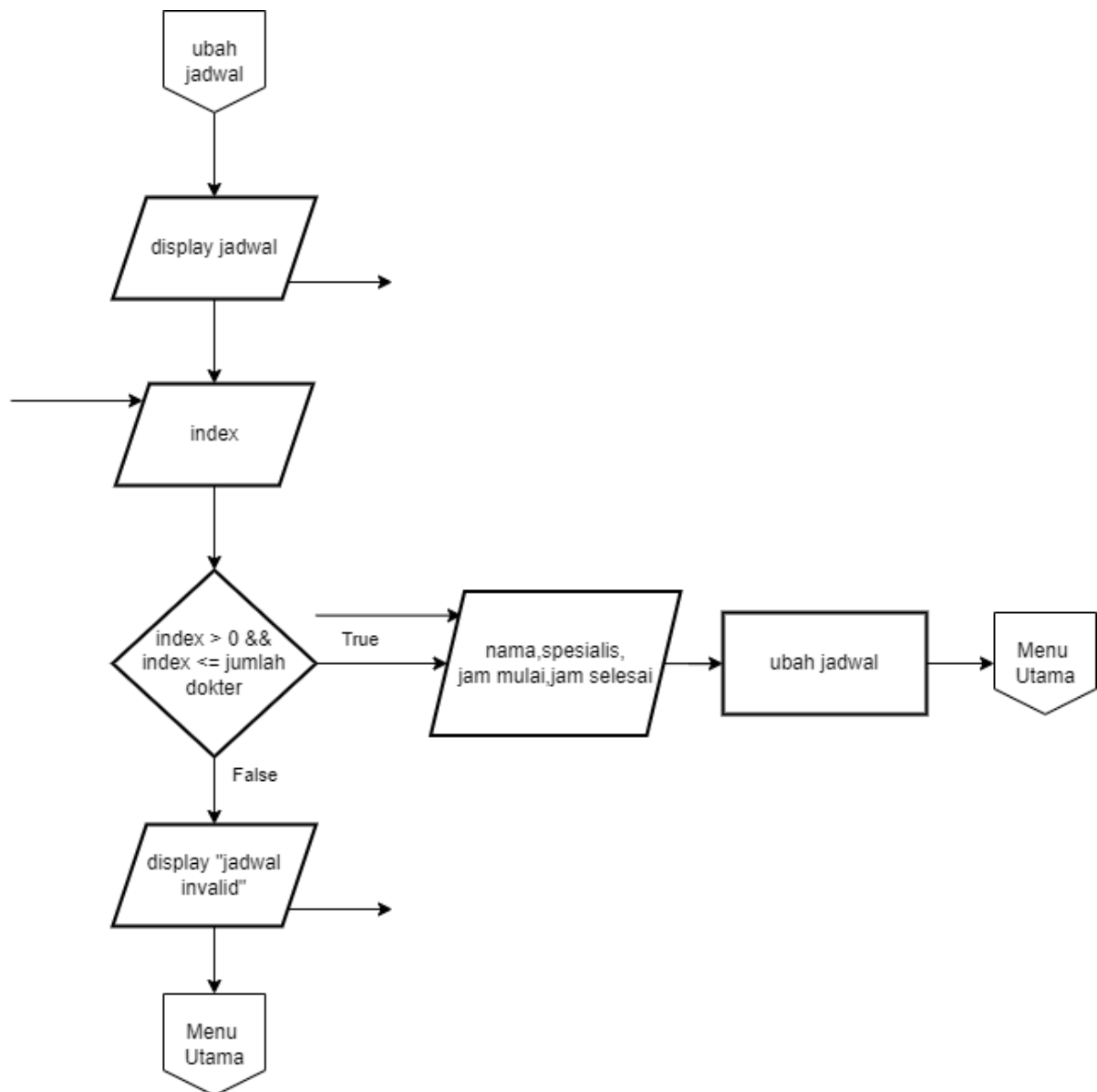
Gambar 1.3 Menu Read

### 1.4 Menu Create



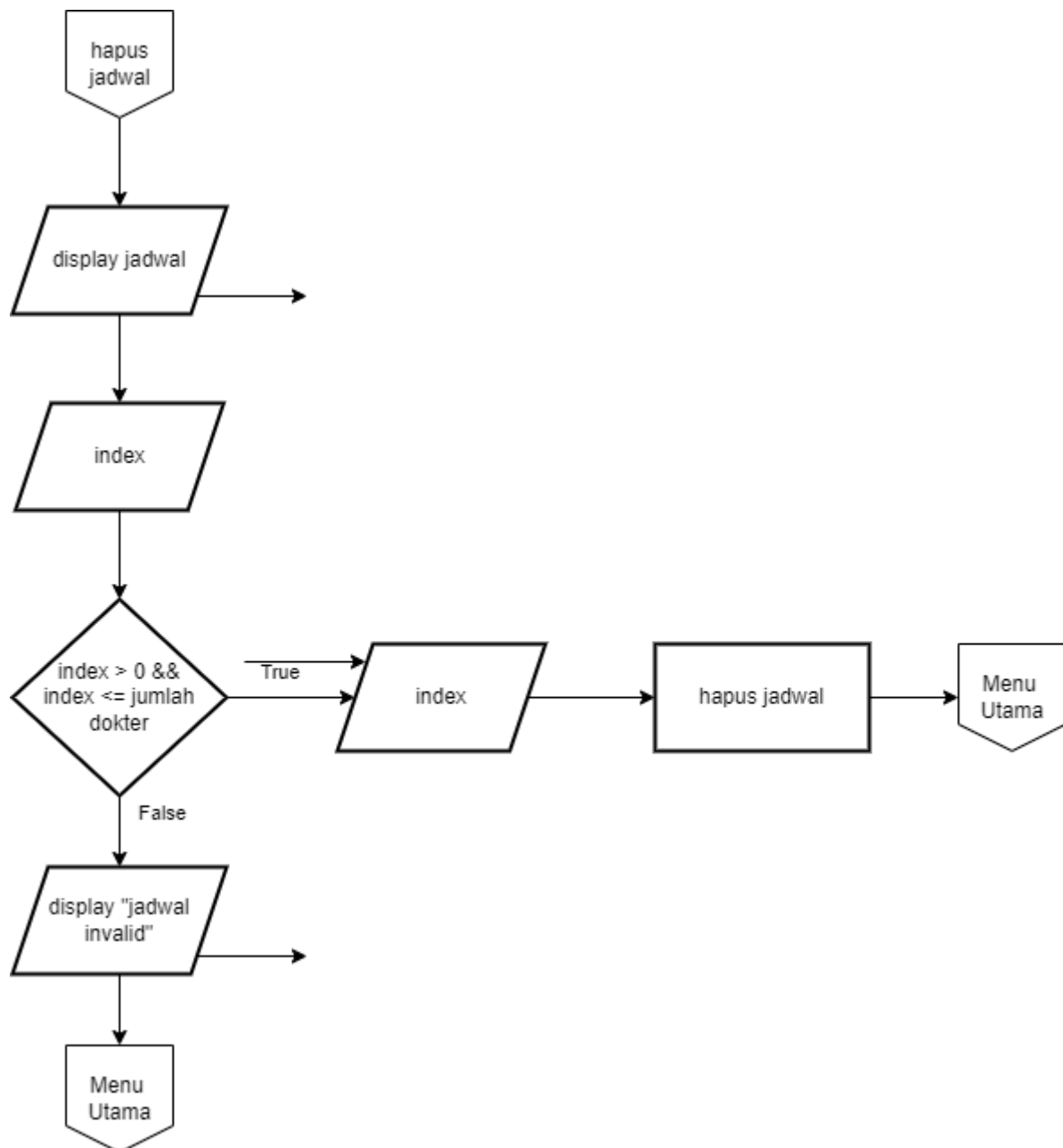
Gambar 1.4 Menu Create

## 1.5 Menu Update



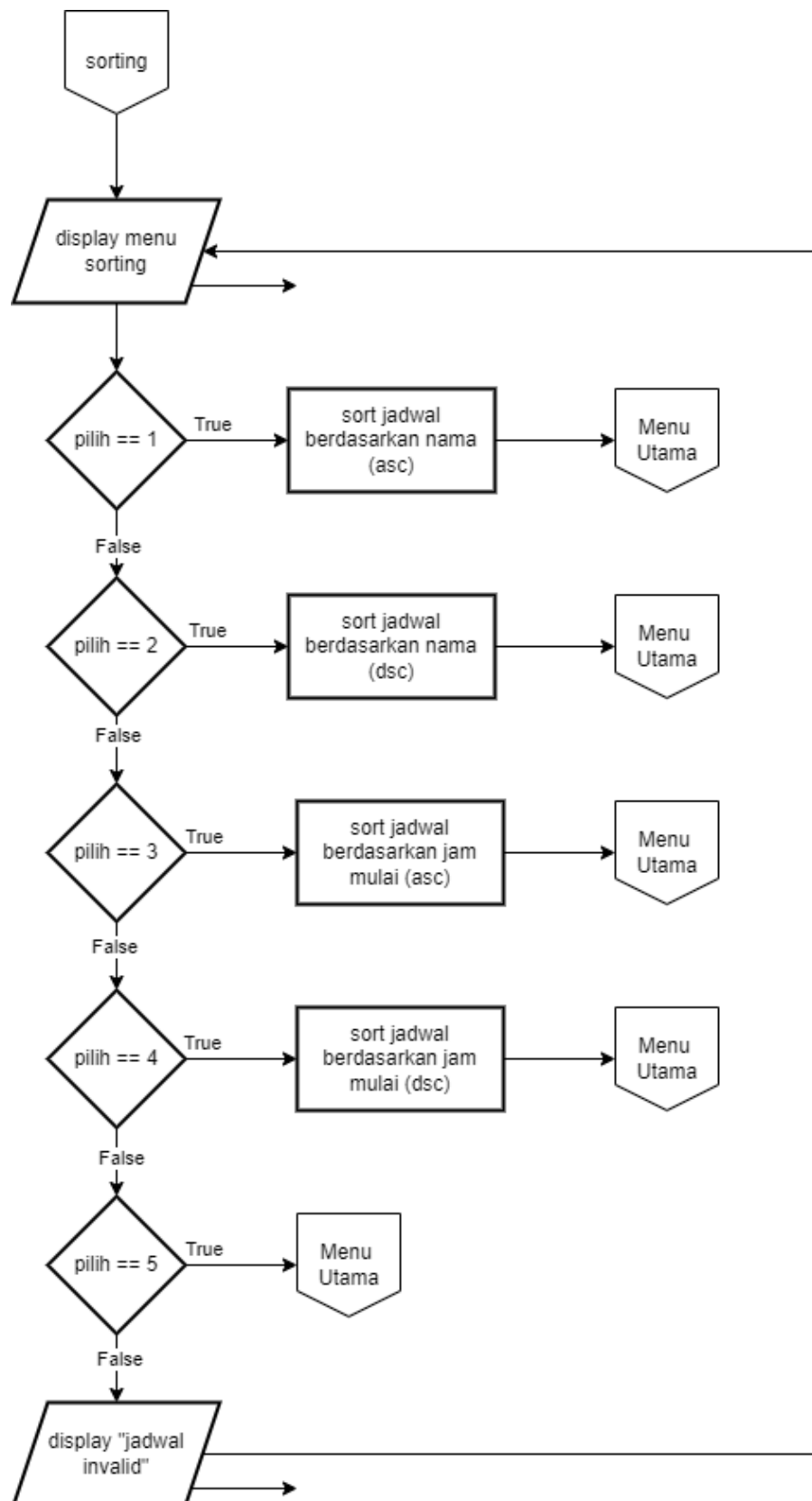
Gambar 1.5 Menu Update

## 1.6 Menu Delete



Gambar 1.6 Menu Delete

## 1.7 Sorting



Gambar 1.7 Menu Sorting

## 2. Analisis Program

Sebuah program yang berfungsi untuk manage jadwal praktek dokter spesialis.

## 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:**

```
void login() {
    int percobaan = 0;
    string username = "faiz", password = "031", login_user, login_password,
    pilih;
    bool login_berhasil = false;

    while (percobaan < 3) {
        cout << "\n<==== Login Page >====>\n";
        cout << "Username: "; cin >> login_user;
        cout << "Password: "; cin >> login_password;

        if (login_user == username and login_password == password) {
            cout << "Login berhasil!\n";
            login_berhasil = true;
            break;
        } else {
            cout << "Username atau password salah!\n";
            percobaan++;
        }
    }

    if (!login_berhasil){
        cout << "Login gagal! Program Berhenti.\n";
        login_gagal = true;
        return;
    }

    while (login_berhasil){
        menu_utama();
        cin >> pilih;
        cin.ignore();
        clear();
    }
}
```



```

        if (pilih == "1"){
            int index = 0;
            lihat_jadwal(index);
        } else if (pilih == "2"){
            tambah_jadwal();
        } else if (pilih == "3"){
            ubah_jadwal();
        } else if (pilih == "4"){
            hapus_jadwal();
        } else if (pilih == "5"){
            sorting_jadwal();
        } else if (pilih == "6"){
            login_berhasil = false;
        } else {
            cout << "Pilihan tidak valid! Coba lagi.\n";
        }
    }
}

```

## B. Menu Utama

Menu Utama berisi beberapa pilihan fitur. Pilihan 1 sampai 4 merupakan fitur CRUD dan pilihan 5 merupakan fitur untuk kembali ke halaman login.

### Source Code:

```

void menu_utama() {
    cout << "\n<===== Jadwal Praktek Dokter =====>\n";
    cout << "1. Tampilkan Jadwal\n";
    cout << "2. Tambah Jadwal\n";
    cout << "3. Ubah Jadwal\n";
    cout << "4. Hapus Jadwal\n";
    cout << "5. Sorting Jadwal\n";
    cout << "6. Logout\n";
    cout << "Pilih Fitur: ";
}

```

### C. Melihat Jadwal

Menampilkan jadwal.

#### Source Code:

```
bool lihat_jadwal(int& index) {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
        return false;
    }

    if (index == 0) {
        for(int i = 0; i < jumlah_dokter; ++i){
            cout << i + 1 << ". Nama: " << jadwal_praktek[i].nama_dokter <<
endl;
            cout << " " << "Spesialis: " << jadwal_praktek[i].spesialis <<
endl;
            cout << " " << "Jam Mulai: " << jadwal_praktek[i].jadwal.jam_mulai
<< endl;
            cout << " " << "Jam Selesai: " <<
jadwal_praktek[i].jadwal.jam_selesai << endl;
            cout << " " << endl;
            index++;
        }
    }

    if (index >= jumlah_dokter) {
        return true;
    }
    return lihat_jadwal(index);
}
```

## D. Menambahkan Jadwal

Menambahkan jadwal.

### Source Code:

```
void tambah_jadwal() {
    if (jumlah_dokter >= MAX_DOKTER) {
        cout << "Jadwal Penuh.\n";
    }

    double jam_mulai, jam_selesai;
    string nama_dokter, spesialis;

    cout << "Masukkan Nama Dokter: ";
    cin >> nama_dokter;
    cout << "Masukkan Spesialis: ";
    cin >> spesialis;
    cin.ignore();
    cout << "Masukkan Jam Mulai: ";
    cin >> jam_mulai;
    cout << "Masukkan Jam Selesai: ";
    cin >> jam_selesai;

    if (tambah_jadwal(&nama_dokter, &spesialis) && tambah_jadwal(&jam_mulai,
&jam_selesai)) {
        cout << "Jadwal berhasil ditambahkan!\n";
    } else {
        cout << "Gagal menambahkan jadwal.\n";
    }
}

bool tambah_jadwal(string* nama_dokter, string* spesialis) {
    if (jumlah_dokter >= MAX_DOKTER) return false;
    jadwal_praktek[jumlah_dokter].nama_dokter = *nama_dokter;
    jadwal_praktek[jumlah_dokter].spesialis = *spesialis;
    return true;
}

bool tambah_jadwal(double* jam_mulai, double* jam_selesai) {
    if (jumlah_dokter >= MAX_DOKTER) return false;
    jadwal_praktek[jumlah_dokter].jadwal.jam_mulai = *jam_mulai;
    jadwal_praktek[jumlah_dokter].jadwal.jam_selesai = *jam_selesai;
    jumlah_dokter++;
    return true;
}
```

## E. Mengubah Jadwal

Mengubah jadwal.

### Source Code:

```
void ubah_jadwal() {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
    } else {
        int index = 0;
        lihat_jadwal(index);
        int pilih;
        cout << "Pilih Jadwal Yang Ingin Di Ubah: ";
        cin >> pilih;
        if (pilih > 0 && pilih <= jumlah_dokter) {
            cout << "Masukkan Nama Dokter baru: "; cin >> jadwal_praktek[pilih - 1].nama_dokter;
            cout << "Masukkan Spesialis baru: "; cin >> jadwal_praktek[pilih - 1].spesialis;
            cout << "Masukkan Jam Mulai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_mulai;
            cout << "Masukkan Jam Selesai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_selesai;
            cout << "Jadwal berhasil diperbarui!\n";

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

## F. Menghapus Jadwal

Menghapus jadwal.

### Source Code:

```
void hapus_jadwal() {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
    } else {
        int index = 0;
        lihat_jadwal(index);
        int pilih;
        cout << "Pilih Jadwal Yang Ingin Di Hapus: ";
        cin >> pilih;
        if (pilih > 0 && pilih <= jumlah_dokter) {
            for (int i = pilih - 1; i < jumlah_dokter - 1; i++) {
                jadwal_praktek[i] = jadwal_praktek[i + 1];
            }
            jumlah_dokter--;
            cout << "Jadwal berhasil dihapus!\n";
        } else {
            cout << "Jadwal Tidak Valid!\n";
        }
    }
}
```

## G. Sorting

Mengurutkan Jadwal.

### Source Code:

```
void sorting_jadwal() {
    while (true) {
        if (jumlah_dokter == 0) {
            cout << "Jadwal Kosong" << endl;
            return;
        }

        cout << "<==== Menu Sorting =====>\n";
        cout << "1. Berdasarkan Nama ascending\n";
        cout << "2. Berdasarkan Nama Descending\n";
        cout << "3. Berdasarkan Jam Mulai ascending\n";
        cout << "4. Berdasarkan Jam Mulai Descending\n";
        cout << "5. Kembali\n";
    }
}
```

```

        cout << "Pilih Fitur: ";

        int pilih;
        cin >> pilih;
        cin.ignore();
        clear();

        switch (pilih) {
            case 1:
                sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, true);
                cout << "Produk diurutkan berdasarkan nama (ascending).\n";
                break;
            case 2:
                sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, false);
                cout << "Produk diurutkan berdasarkan nama (descending).\n";
                break;
            case 3:
                sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, true);
                cout << "Produk diurutkan berdasarkan Jam Mulai (ascending).\n";
                break;
            case 4:
                sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, false);
                cout << "Produk diurutkan berdasarkan Jam Mulai
(descending).\n";
                break;
            case 5:
                clear();
                return;
            default:
                cout << "Pilihan tidak valid!\n";
                continue;
        }
        return;
    }
}

void tukar_jadwal(jadwal_dokter &a, jadwal_dokter&b) {
    jadwal_dokter temp = a;
    a = b;
    b = temp;
}

void sort_jam(jadwal_dokter arr[], int low, int high, bool asc) {
    if (low < high) {
        int pivot = low;
        int i = low;
        int j = high;
        while (i <= j) {
            if (asc) {
                while (arr[i].jadwal.jam_mulai <= arr[pivot].jadwal.jam_mulai &&
i <= high) i++;

```

```

        while (arr[j].jadwal.jam_mulai > arr[pivot].jadwal.jam_mulai &&
j >= low) j--;
    } else {
        while (arr[i].jadwal.jam_mulai >= arr[pivot].jadwal.jam_mulai &&
i <= high) i++;
        while (arr[j].jadwal.jam_mulai < arr[pivot].jadwal.jam_mulai &&
j >= low) j--;
    }
    if (i < j) tukar_jadwal(arr[i], arr[j]);
}
tukar_jadwal(arr[j], arr[pivot]);
sort_jam(arr, low, j - 1, asc);
sort_jam(arr, j + 1, high, asc);
}
}

void sort_nama(jadwal_dokter arr[], int low, int high, bool asc) {
    if (low < high) {
        int pivot = low;
        int i = low;
        int j = high;
        while (i <= j) {
            if (asc) {
                while (arr[i].nama_dokter <= arr[pivot].nama_dokter && i <=
high) i++;
                while (arr[j].nama_dokter > arr[pivot].nama_dokter && j >= low)
j--;
            } else {
                while (arr[i].nama_dokter >= arr[pivot].nama_dokter && i <=
high) i++;
                while (arr[j].nama_dokter < arr[pivot].nama_dokter && j >= low)
j--;
            }
            if (i < j) tukar_jadwal(arr[i], arr[j]);
        }
        tukar_jadwal(arr[j], arr[pivot]);
        sort_nama(arr, low, j - 1, asc);
        sort_nama(arr, j + 1, high, asc);
    }
}
}

```

#### 4. Screenshot Program

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct jam_praktek {
6      double jam_mulai;
7      double jam_selesai;
8  };
9
10 struct jadwal_dokter {
11     string nama_dokter;
12     string spesialis;
13     jam_praktek jadwal;
14 };
15
16 const int MAX_DOKTER = 10;
17 jadwal_dokter jadwal_praktek[MAX_DOKTER] = {
18     {"faiz", "gigi", {07.30, 09.00 }},
19     {"fathur", "bedah", {13.30, 15.45 }},
20     {"zaky", "jantung", {09.30, 11.00 }},
21     {"zeydan", "mata", {11.30, 13.00 }},
22 };
23
24 int jumlah_dokter = 4;
25 bool login_gagal = false;
26
27 void clear();
28 void menu_awal();
29 void menu_utama();
30 void login();
31 bool lihat_jadwal(int& index);
32 void tambah_jadwal();
33 bool tambah_jadwal(double* jam_mulai, double* jam_selesai);
34 bool tambah_jadwal(string* nama_dokter, string* spesialis);
35 void ubah_jadwal();
36 void hapus_jadwal();
37 void sorting_jadwal();
38 void sort_nama(jadwal_dokter arr[], int low, int high, bool asc);
39 void sort_jam(jadwal_dokter arr[], int low, int high, bool asc);
40
41
42 void clear() {
43     system("cls");
44 }
45
46 void menu_awal() {
47     cout << "<==== Home Page =====>\n";
48     cout << "1. Login\n";
49     cout << "2. Keluar\n";
50     cout << "Pilih Fitur: ";
51 }
```

Gambar 4.1 Code Program



```

1 void menu_utama() {
2     cout << "\n<==== Jadwal Praktek Dokter >====>\n";
3     cout << "1. Tampilkan Jadwal\n";
4     cout << "2. Tambah Jadwal\n";
5     cout << "3. Ubah Jadwal\n";
6     cout << "4. Hapus Jadwal\n";
7     cout << "5. Sorting Jadwal\n";
8     cout << "6. Logout\n";
9     cout << "Pilih Fitur: ";
10 }
11
12 void login() {
13     int percobaan = 0;
14     string username = "faiz", password = "031", login_user, login_password, pilih;
15     bool login_berhasil = false;
16
17     while (percobaan < 3) {
18         cout << "\n<==== Login Page >====>\n";
19         cout << "Username: "; cin >> login_user;
20         cout << "Password: "; cin >> login_password;
21
22         if (login_user == username and login_password == password) {
23             cout << "Login berhasil!\n";
24             login_berhasil = true;
25             break;
26
27         } else {
28             cout << "Username atau password salah!\n";
29             percobaan++;
30         }
31     }
32
33     if (!login_berhasil){
34         cout << "Login gagal! Program Berhenti.\n";
35         login_gagal = true;
36         return;
37     }
38
39     while (login_berhasil){
40         menu_utama();
41         cin >> pilih;
42         cin.ignore();
43         clear();
44
45         if (pilih == "1"){
46             int index = 0;
47             lihat_jadwal(index);
48         } else if (pilih == "2"){
49             tambah_jadwal();
50         } else if (pilih == "3"){
51             ubah_jadwal();
52         } else if (pilih == "4"){
53             hapus_jadwal();
54         } else if (pilih == "5"){
55             sorting_jadwal();
56         } else if (pilih == "6"){
57             login_berhasil = false;
58         } else {
59             cout << "Pilihan tidak valid! Coba lagi.\n";
60         }
61     }
62 }

```

Gambar 4.2 Code Program

```

1
2 bool lihat_jadwal(int& index) {
3     if (jumlah_dokter == 0) {
4         cout << "Jadwal Kosong" << endl;
5         return false;
6     }
7
8     if (index == 0) {
9         for(int i = 0; i < jumlah_dokter; ++i){
10             cout << i + 1 << ". Nama: " << jadwal_praktek[i].nama_dokter << endl;
11             cout << " " << "Spesialis: " << jadwal_praktek[i].spesialis << endl;
12             cout << " " << "Jam Mulai: " << jadwal_praktek[i].jadwal.jam_mulai << endl;
13             cout << " " << "Jam Selesai: " << jadwal_praktek[i].jadwal.jam_selesai << endl;
14             cout << " " << endl;
15             index++;
16         }
17     }
18
19     if (index >= jumlah_dokter) {
20         return true;
21     }
22     return lihat_jadwal(index);
23 }
24
25 void tambah_jadwal() {
26     if (jumlah_dokter >= MAX_DOKTER) {
27         cout << "Jadwal Penuh.\n";
28     }
29
30     double jam_mulai, jam_selesai;
31     string nama_dokter, spesialis;
32
33     cout << "Masukkan Nama Dokter: ";
34     cin >> nama_dokter;
35     cout << "Masukkan Spesialis: ";
36     cin >> spesialis;
37     cin.ignore();
38     cout << "Masukkan Jam Mulai: ";
39     cin >> jam_mulai;
40     cout << "Masukkan Jam Selesai: ";
41     cin >> jam_selesai;
42
43     if (tambah_jadwal(&nama_dokter, &spesialis) && tambah_jadwal(&jam_mulai, &jam_selesai)) {
44         cout << "Jadwal berhasil ditambahkan!\n";
45     }
46     else {
47         cout << "Gagal menambahkan jadwal.\n";
48     }
49 }
50
51 bool tambah_jadwal(string* nama_dokter, string* spesialis) {
52     if (jumlah_dokter >= MAX_DOKTER) return false;
53     jadwal_praktek[jumlah_dokter].nama_dokter = *nama_dokter;
54     jadwal_praktek[jumlah_dokter].spesialis = *spesialis;
55     return true;
56 }
57
58 bool tambah_jadwal(double* jam_mulai, double* jam_selesai) {
59     if (jumlah_dokter >= MAX_DOKTER) return false;
60     jadwal_praktek[jumlah_dokter].jadwal.jam_mulai = *jam_mulai;
61     jadwal_praktek[jumlah_dokter].jadwal.jam_selesai = *jam_selesai;
62     jumlah_dokter++;
63     return true;
64 }

```

Gambar 4.3 Code Program

```

1 void ubah_jadwal() {
2     if (jumlah_dokter == 0) {
3         cout << "Jadwal Kosong" << endl;
4     } else {
5         int index = 0;
6         lihat_jadwal(index);
7         int pilih;
8         cout << "Pilih Jadwal Yang Ingin Di Ubah: ";
9         cin >> pilih;
10        if (pilih > 0 && pilih <= jumlah_dokter) {
11            cout << "Masukkan Nama Dokter baru: "; cin >> jadwal_praktek[pilih - 1].nama_dokter;
12            cout << "Masukkan Spesialis baru: "; cin >> jadwal_praktek[pilih - 1].spesialis;
13            cout << "Masukkan Jam Mulai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_mulai;
14            cout << "Masukkan Jam Selesai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_selesai;
15            cout << "Jadwal berhasil diperbarui!\n";
16        } else {
17            cout << "Jadwal Tidak Valid!\n";
18        }
19    }
20 }
21 }
22
23 void hapus_jadwal() {
24     if (jumlah_dokter == 0) {
25         cout << "Jadwal Kosong" << endl;
26     } else {
27         int index = 0;
28         lihat_jadwal(index);
29         int pilih;
30         cout << "Pilih Jadwal Yang Ingin Di Hapus: ";
31         cin >> pilih;
32         if (pilih > 0 && pilih <= jumlah_dokter) {
33             for (int i = pilih - 1; i < jumlah_dokter - 1; i++) {
34                 jadwal_praktek[i] = jadwal_praktek[i + 1];
35             }
36             jumlah_dokter--;
37             cout << "Jadwal berhasil dihapus!\n";
38         } else {
39             cout << "Jadwal Tidak Valid!\n";
40         }
41     }
42 }
43 }

```

Gambar 4.4 Code Program

```

1 void sorting_jadwal() {
2     while (true) {
3         if (jumlah_dokter == 0) {
4             cout << "Jadwal Kosong" << endl;
5             return;
6         }
7
8         cout << "<==== Menu Sorting =====>\n";
9         cout << "1. Berdasarkan Nama ascending\n";
10        cout << "2. Berdasarkan Nama Descending\n";
11        cout << "3. Berdasarkan Jam Mulai ascending\n";
12        cout << "4. Berdasarkan Jam Mulai Descending\n";
13        cout << "5. Kembali\n";
14        cout << "Pilih Fitur: ";
15
16        int pilih;
17        cin >> pilih;
18        cin.ignore();
19        clear();
20
21        switch (pilih) {
22            case 1:
23                sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, true);
24                cout << "Produk diurutkan berdasarkan nama (ascending).\n";
25                break;
26            case 2:
27                sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, false);
28                cout << "Produk diurutkan berdasarkan nama (descending).\n";
29                break;
30            case 3:
31                sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, true);
32                cout << "Produk diurutkan berdasarkan Jam Mulai (ascending).\n";
33                break;
34            case 4:
35                sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, false);
36                cout << "Produk diurutkan berdasarkan Jam Mulai (descending).\n";
37                break;
38            case 5:
39                clear();
40                return;
41            default:
42                cout << "Pilihan tidak valid!\n";
43                continue;
44        }
45        return;
46    }
47 }
48
49 void tukar_jadwal(jadwal_dokter &a, jadwal_dokter&b) {
50     jadwal_dokter temp = a;
51     a = b;
52     b = temp;
53 }
54
55 void sort_jam(jadwal_dokter arr[], int low, int high, bool asc) {
56     if (low < high) {
57         int pivot = low;
58         int i = low;
59         int j = high;
60         while (i <= j) {
61             if (asc) {
62                 while (arr[i].jadwal.jam_mulai <= arr[pivot].jadwal.jam_mulai && i <= high) i++;
63                 while (arr[j].jadwal.jam_mulai > arr[pivot].jadwal.jam_mulai && j >= low) j--;
64             } else {
65                 while (arr[i].jadwal.jam_mulai >= arr[pivot].jadwal.jam_mulai && i <= high) i++;
66                 while (arr[j].jadwal.jam_mulai < arr[pivot].jadwal.jam_mulai && j >= low) j--;
67             }
68             if (i < j) tukar_jadwal(arr[i], arr[j]);
69         }
70         tukar_jadwal(arr[j], arr[pivot]);
71         sort_jam(arr, low, j - 1, asc);
72         sort_jam(arr, j + 1, high, asc);
73     }
74 }

```

Gambar 4.5 Code Program

```

1 void sort_nama(jadwal_dokter arr[], int low, int high, bool asc) {
2     if (low < high) {
3         int pivot = low;
4         int i = low;
5         int j = high;
6         while (i <= j) {
7             if (asc) {
8                 while (arr[i].nama_dokter <= arr[pivot].nama_dokter && i <= high) i++;
9                 while (arr[j].nama_dokter > arr[pivot].nama_dokter && j >= low) j--;
10            } else {
11                while (arr[i].nama_dokter >= arr[pivot].nama_dokter && i <= high) i++;
12                while (arr[j].nama_dokter < arr[pivot].nama_dokter && j >= low) j--;
13            }
14            if (i < j) tukar_jadwal(arr[i], arr[j]);
15        }
16        tukar_jadwal(arr[j], arr[pivot]);
17        sort_nama(arr, low, j - 1, asc);
18        sort_nama(arr, j + 1, high, asc);
19    }
20 }
21
22 int main() {
23     clear();
24     string pilih;
25     while(true) {
26         menu_awal();
27         cin >> pilih;
28         cin.ignore();
29         if (pilih == "1") {
30             login();
31             if (login_gagal)
32                 break;
33
34         } else if (pilih == "2"){
35             cout << "Keluar dari program.\n";
36             break;
37
38         } else {
39             cout << "Input Yang Anda Pilih Tidak Valid. Silahkan Coba lagi.\n\n";
40         }
41     }
42     return 0;
43 }

```

Gambar 4.6 Code 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 (main)
$ git add .
```

Gambar 5.2 Membuat File dan Git add

### 5.3 Git Commit

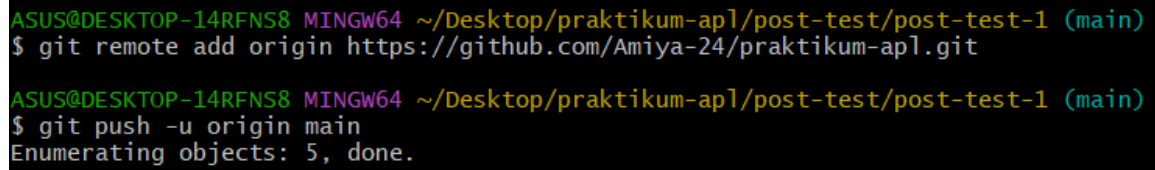
Menyimpan perubahan yang disimpan pada folder lokal dengan sebuah pesan.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-6 (main)
$ git commit -m "Finish Post Test 6"
[main 8f766b8] Finish Post Test 6
2 files changed, 6 insertions(+), 6 deletions(-)
```

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 terminal 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