

LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT

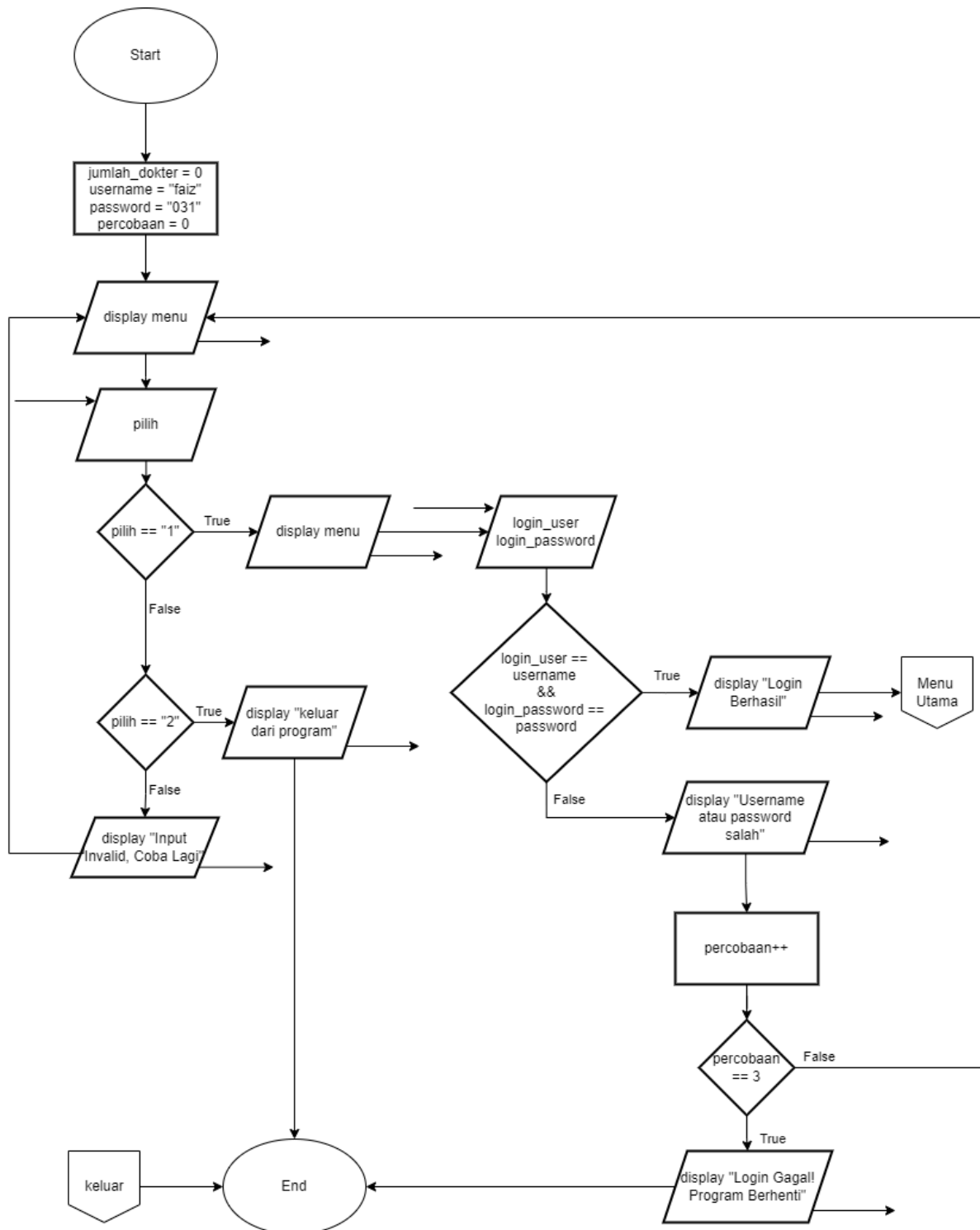


Disusun oleh:
Muhammad Faiz Lazuardi (2409106031)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

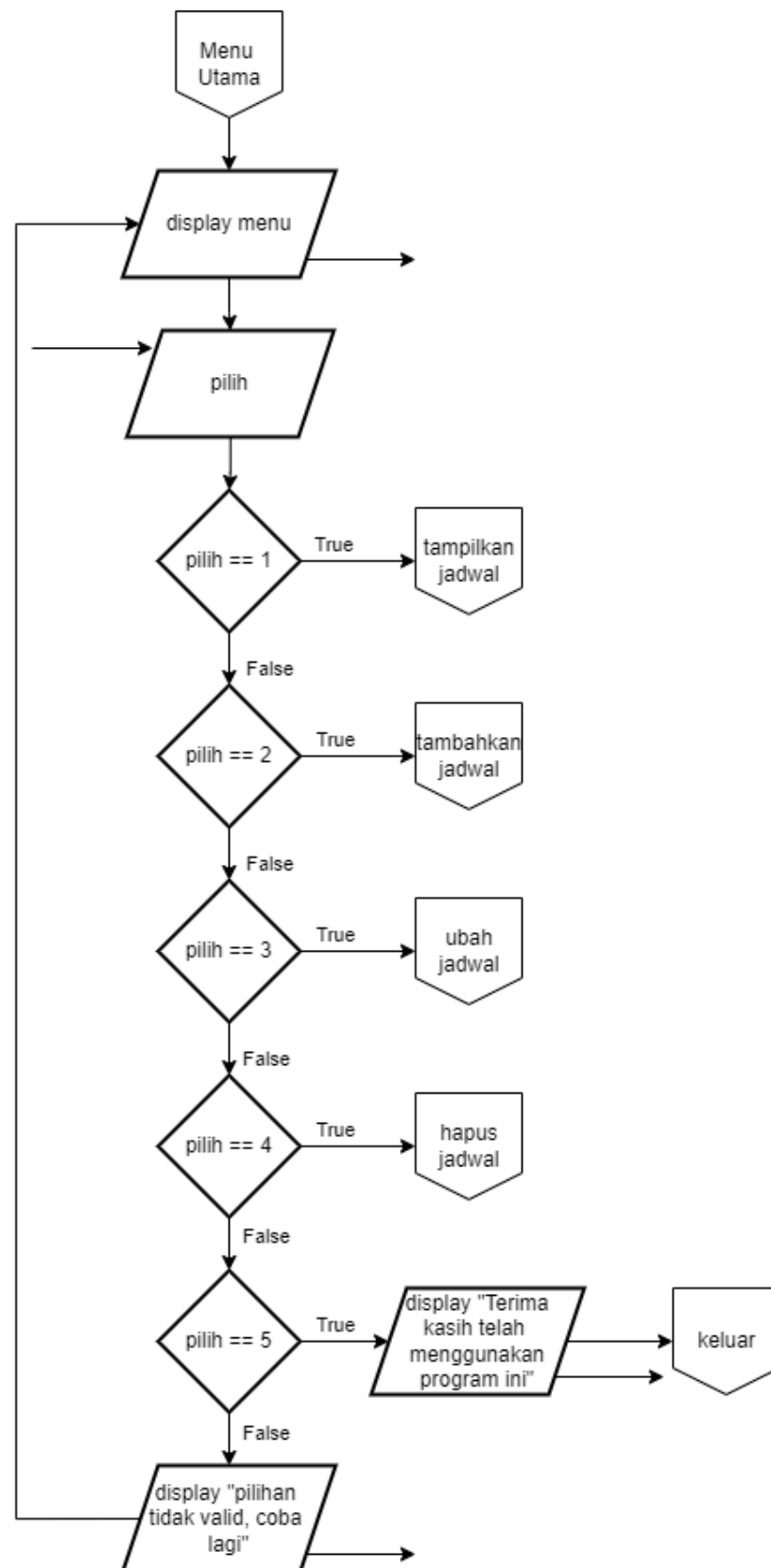
1. Flowchart

1.1 Menu Login



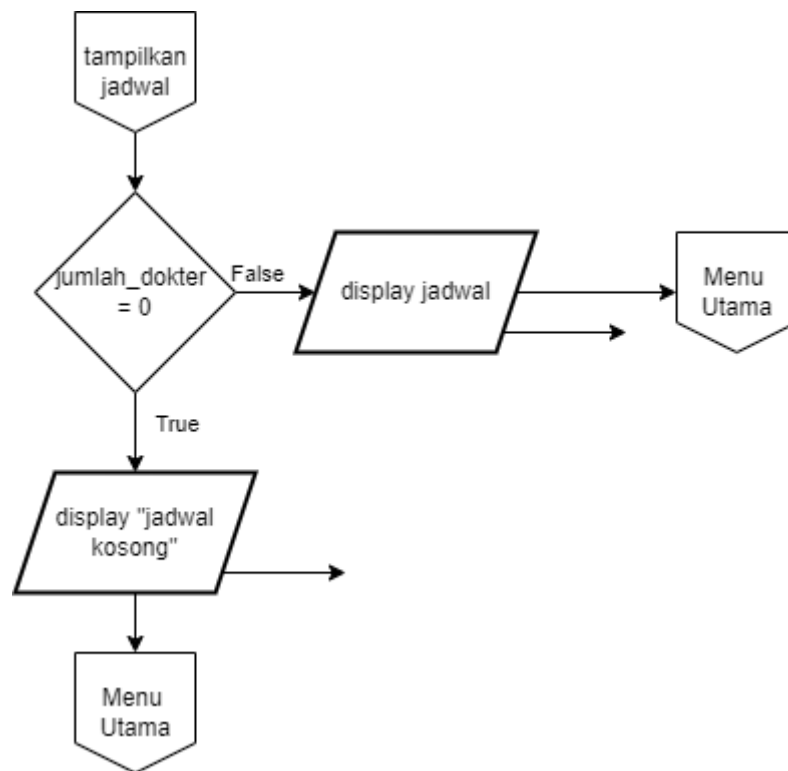
Gambar 1.1 Menu Login

1.2 Menu Utama



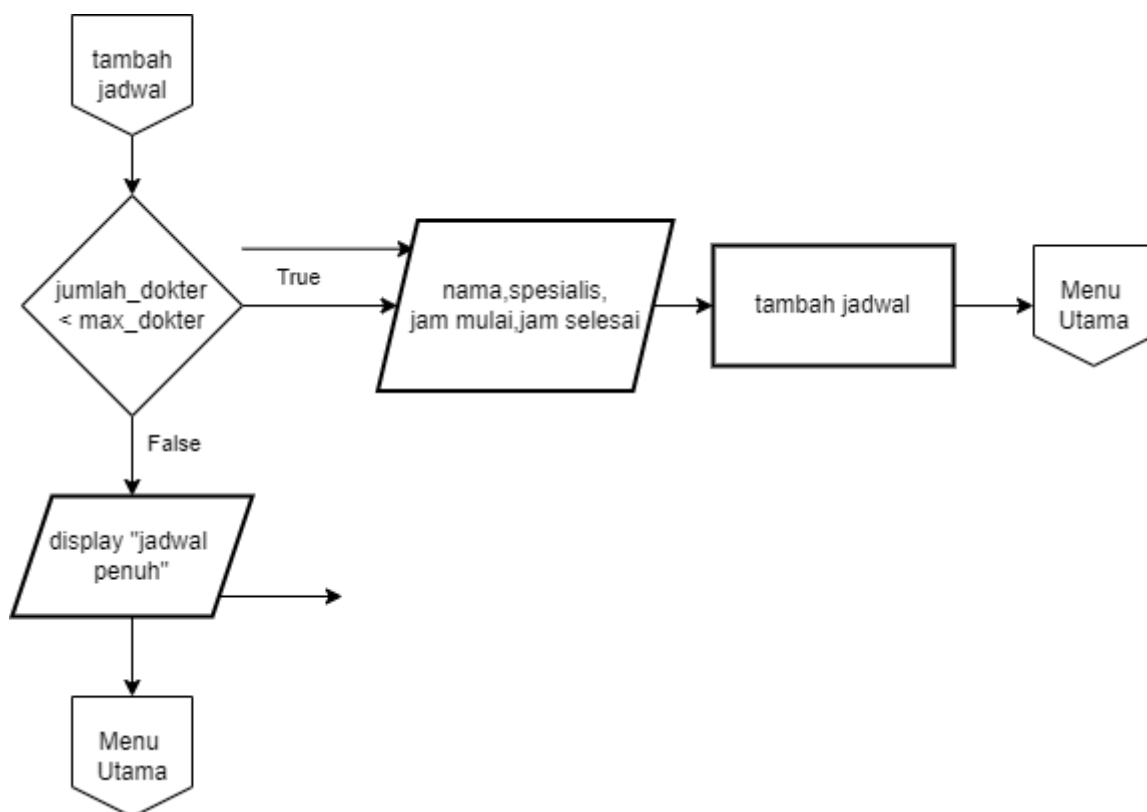
Gambar 1.2 Menu Utama

1.3 Menu Read



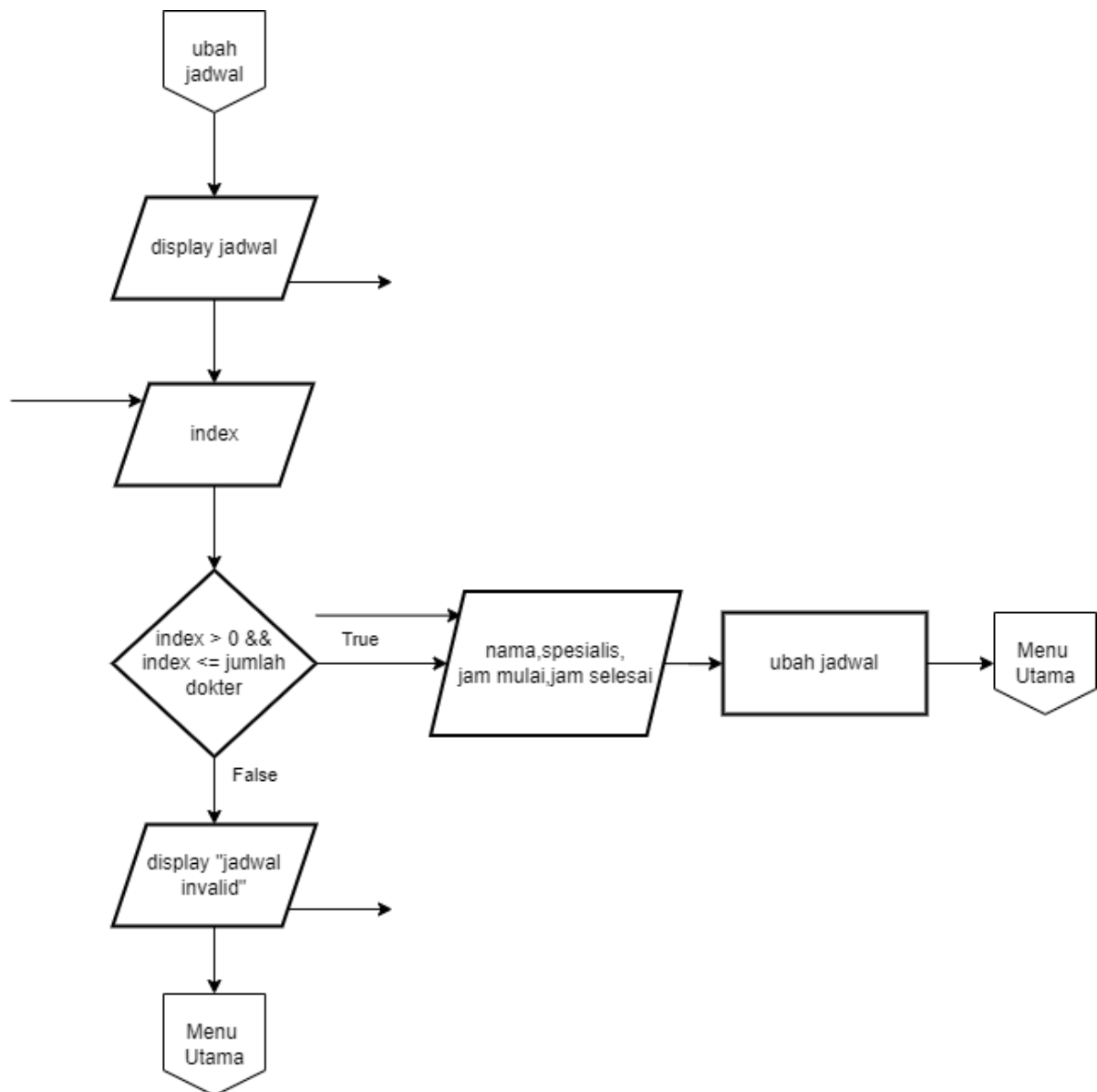
Gambar 1.3 Menu Read

1.4 Menu Create



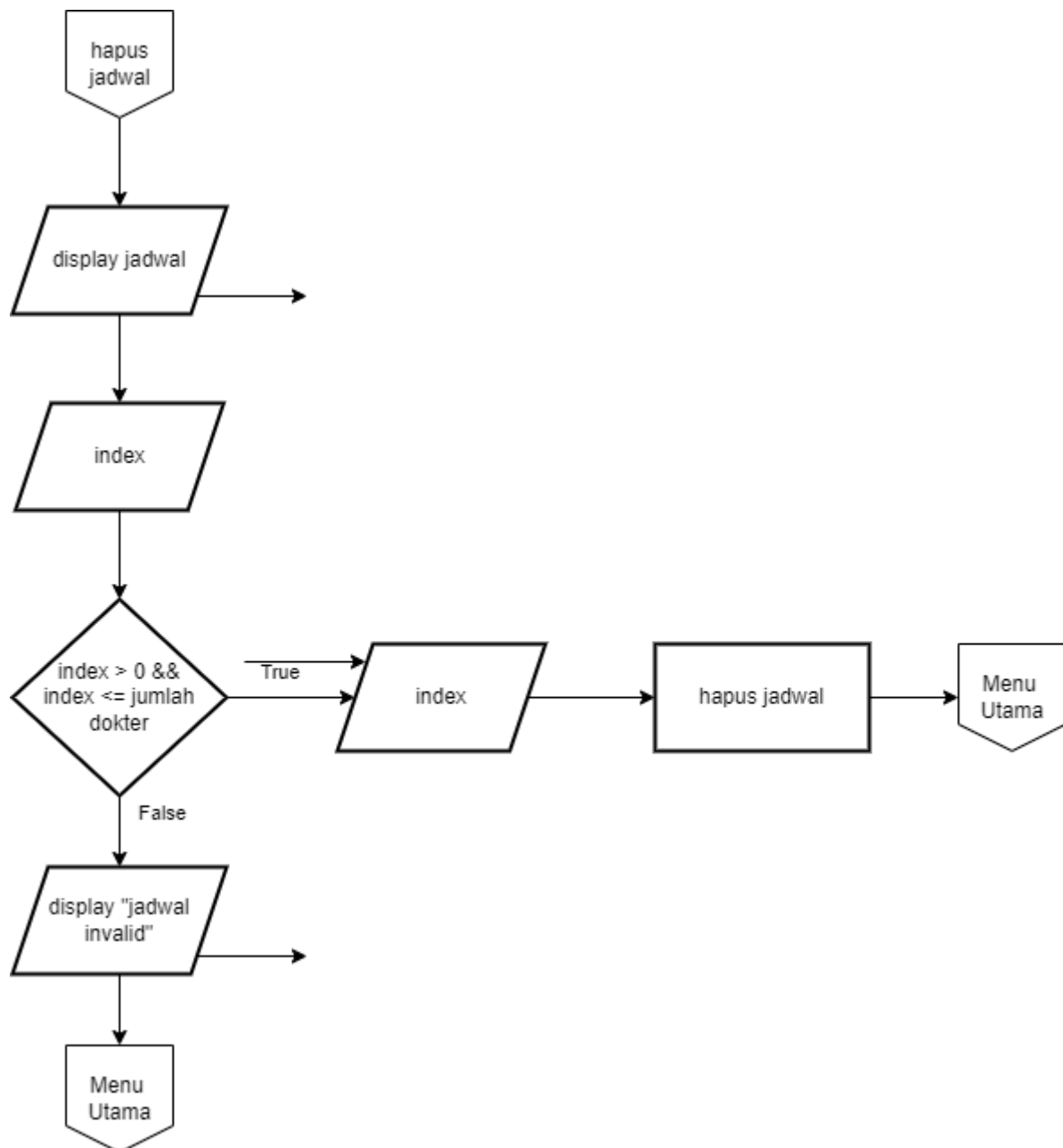
Gambar 1.4 Menu Create

1.5 Menu Update



Gambar 1.5 Menu Update

1.6 Menu Delete



Gambar 1.6 Menu Delete

2. Analisis Program

Sebuah program yang berfungsi untuk memanage 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"){
            lihat_jadwal();
        } else if (pilih == "2"){
            tambah_jadwal();
        }
    }
}
```

```

    } else if (pilih == "3"){
        ubah_jadwal();
    } else if (pilih == "4"){
        hapus_jadwal();
    } else if (pilih == "5"){
        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. 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;
    }

    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;
        }
    }

    if (index >= jumlah_dokter) {
        return true;
    }
    return lihat_jadwal(index + 1);
}
```

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

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 = 30;
17 jadwal_dokter jadwal_dokter[MAX_DOKTER];
18 int jumlah_dokter = 0;
19 bool login_gagal = false;
20
21 void tampilkan();
22 void menu_utama();
23 void menu_utama();
24 void login();
25 bool lihat_jadwal(int index = 0);
26 void tambah_jadwal();
27 bool tambah_jadwal(double jam_mulai, double jam_selesai);
28 bool tambah_jadwal(string nama_dokter, string spesialis);
29 void ubah_jadwal();
30 void hapus_jadwal();
31
32 void clear() {
33     system("cls");
34 }
35
36 void menu_utama() {
37     cout << "===== Menu Utama =====\n";
38     cout << "1. Login\n";
39     cout << "2. Keluar\n";
40     cout << "Pilih Menu : ";
41 }
42
43 void menu_utama() {
44     cout << "===== Jadwal Praktek Dokter =====\n";
45     cout << "1. Tampilkan Jadwal\n";
46     cout << "2. Tambah Jadwal\n";
47     cout << "3. Ubah Jadwal\n";
48     cout << "4. Hapus Jadwal\n";
49     cout << "5. Login\n";
50     cout << "Pilih Menu : ";
51 }
52
53 void login() {
54     int percobaan = 0;
55     string username = "user", password = "1234", login_user, login_password, pilih;
56     bool login_berhasil = false;
57
58     while (percobaan < 3) {
59         cout << "===== Login Page =====\n";
60         cout << "Username : "; cin >> login_user;
61         cout << "Password : "; cin >> login_password;
62
63         if (login_user == username and login_password == password) {
64             cout << "Login berhasil!\n";
65             login_berhasil = true;
66             break;
67         } else {
68             cout << "Username atau password salah!\n";
69             percobaan++;
70         }
71     }
72
73     if (!login_berhasil) {
74         cout << "Login gagal! Program Berakhir!\n";
75         login_gagal = true;
76         return;
77     }
78
79     while (!login_berhasil) {
80         menu_utama();
81         cin >> pilih;
82         cin.ignore();
83         clear();
84     }
85 }
```

Gambar 4.1 Code Program

```

87     if (pilih == "1"){
88         lihat_jadwal();
89     } else if (pilih == "2"){
90         tambah_jadwal();
91     } else if (pilih == "3"){
92         ubah_jadwal();
93     } else if (pilih == "4"){
94         hapus_jadwal();
95     } else if (pilih == "5"){
96         login_berhasil = false;
97     } else {
98         cout << "Pilihan tidak valid! Coba lagi.\n";
99     }
100 }
101 }
102
103 bool lihat_jadwal(int index) {
104     if (jumlah_dokter == 0) {
105         cout << "Jadwal kosong" << endl;
106     }
107
108     if (index == 0) {
109         for(int i = 0; i < jumlah_dokter; ++i){
110             cout << i + 1 << ". Nama: " << jadwal_praktek[i].nama_dokter << endl;
111             cout << " " << "Spesialis: " << jadwal_praktek[i].spesialis << endl;
112             cout << " " << "Jam Mulai: " << jadwal_praktek[i].jam_mulai << endl;
113             cout << " " << "Jam Selesai: " << jadwal_praktek[i].jam_selesai << endl;
114             cout << " " << endl;
115         }
116     }
117
118     if (index == jumlah_dokter) {
119         return true;
120     }
121     return lihat_jadwal(index + 1);
122 }
123
124 void tambah_jadwal() {
125     if (jumlah_dokter > MAX_DOKTER) {
126         cout << "Jadwal Penuh!\n";
127     }
128
129     double jam_mulai, jam_selesai;
130     string nama_dokter, spesialis;
131
132     cout << "Masukkan Nama Dokter: ";
133     cin >> nama_dokter;
134     cout << "Masukkan Spesialis: ";
135     cin >> spesialis;
136     cin.ignore();
137     cout << "Masukkan Jam Mulai: ";
138     cin >> jam_mulai;
139     cout << "Masukkan Jam Selesai: ";
140     cin >> jam_selesai;
141
142     if (tambah_jadwal(nama_dokter, spesialis) && tambah_jadwal(jam_mulai, jam_selesai)) {
143         cout << "Jadwal berhasil ditambahkan!\n";
144     }
145     else {
146         cout << "Gagal menambahkan jadwal!\n";
147     }
148 }
149
150 bool tambah_jadwal(string nama_dokter, string spesialis) {
151     if (jumlah_dokter > MAX_DOKTER) return false;
152     jadwal_praktek[jumlah_dokter].nama_dokter = nama_dokter;
153     jadwal_praktek[jumlah_dokter].spesialis = spesialis;
154     return true;
155 }
156
157 bool tambah_jadwal(double jam_mulai, double jam_selesai) {
158     if (jumlah_dokter > MAX_DOKTER) return false;
159     jadwal_praktek[jumlah_dokter].jam_mulai = jam_mulai;
160     jadwal_praktek[jumlah_dokter].jam_selesai = jam_selesai;
161     jumlah_dokter++;
162     return true;
163 }
164
165 void ubah_jadwal() {
166     if (jumlah_dokter == 0) {
167         cout << "Jadwal Kosong" << endl;
168     }
169     else {
170         lihat_jadwal();
171         int index;
172         cout << "Pilih Jadwal Yang Ingin Di Ubah: ";
173         cin >> index;
174         if (index > 0 && index < jumlah_dokter) {
175             cout << "Masukkan Nama Dokter baru: "; cin >> jadwal_praktek[index - 1].nama_dokter;
176             cout << "Masukkan Spesialis baru: "; cin >> jadwal_praktek[index - 1].spesialis;
177             cout << "Masukkan Jam Mulai baru: "; cin >> jadwal_praktek[index - 1].jam_mulai;
178             cout << "Masukkan Jam Selesai baru: "; cin >> jadwal_praktek[index - 1].jam_selesai;
179             cout << "Jadwal baru akan diubah!\n";
180         }
181         else {
182             cout << "Jadwal tidak valid!\n";
183         }
184     }
185 }

```

Gambar 4.2 Code Program

```

186 void hapus_jadwal() {
187     if (jumlah_siswa == 0) {
188         cout << "Tidak Ada Siswa" << endl;
189     } else {
190         int index;
191         cout << "Pilih Jadwal Yang Ingin Di Hapus: ";
192         cin >> index;
193         if (index > 0 && index <= jumlah_siswa) {
194             for (int i = index - 1; i < jumlah_siswa - 1; i++) {
195                 jadwal_siswa[i] = jadwal_siswa[i + 1];
196             }
197             jumlah_siswa--;
198             cout << "Jadwal berhasil dihapus!\n";
199         } else {
200             cout << "Jadwal tidak valid!\n";
201         }
202     }
203 }
204
205 int main() {
206     clear();
207     string pilih;
208     while(true) {
209         menu_siswa();
210         cin >> pilih;
211         cin.ignore();
212         if (pilih == "1") {
213             login();
214             if (login_gagal)
215                 break;
216         } else if (pilih == "2") {
217             cout << "Selamat datang, program!\n";
218             break;
219         } else {
220             cout << "Pilih Yang Anda Pilih Tidak Valid. Silakan Coba lagi!\n";
221         }
222     }
223     return 0;
224 }

```

Gambar 4.3 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

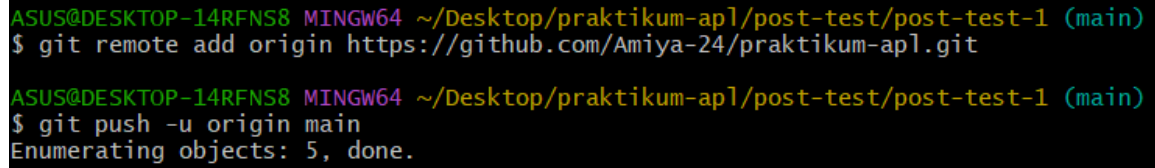
Menyimpan perubahan yang disimpan pada folder lokal dengan sebuah pesan.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test (main)
$ git commit -m "sementara"
[main e559f41] sementara
2 files changed, 230 insertions(+)
create mode 100644 post-test/post-test-4/2409106031-MuhammadFaizLazuardi-PT-4.c
pp
create mode 100644 post-test/post-test-4/2409106031-MuhammadFaizLazuardi-PT-4.e
xe
```

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 with a black background and green text. The prompt is 'ASUS@DESKTOP-14RFNS8 MINGW64 ~'. 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', followed by the output '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