LAPORAN PRAKTIKUM

POSTTEST 2

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Nama (2409106104)

Kelas (C1‘24)

PROGRAM STUDI INFORMATIKA

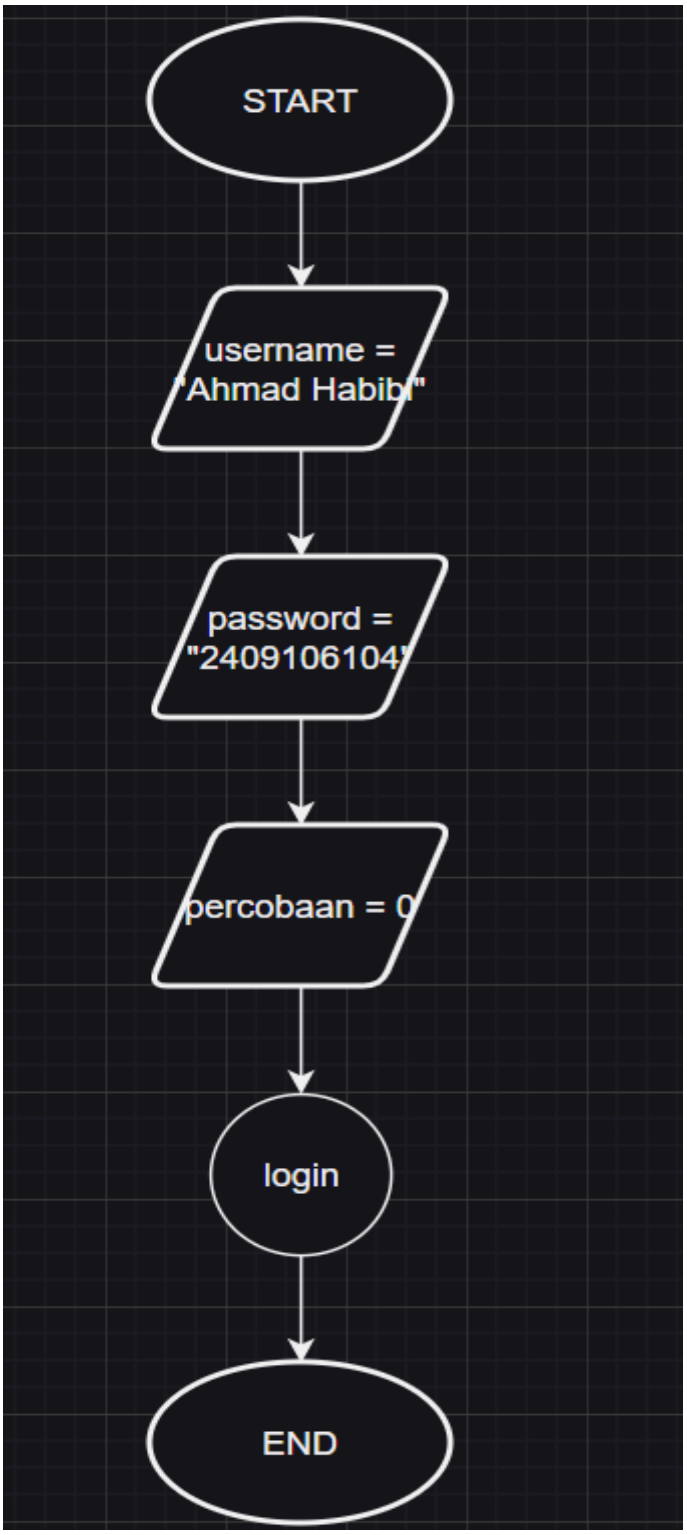
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

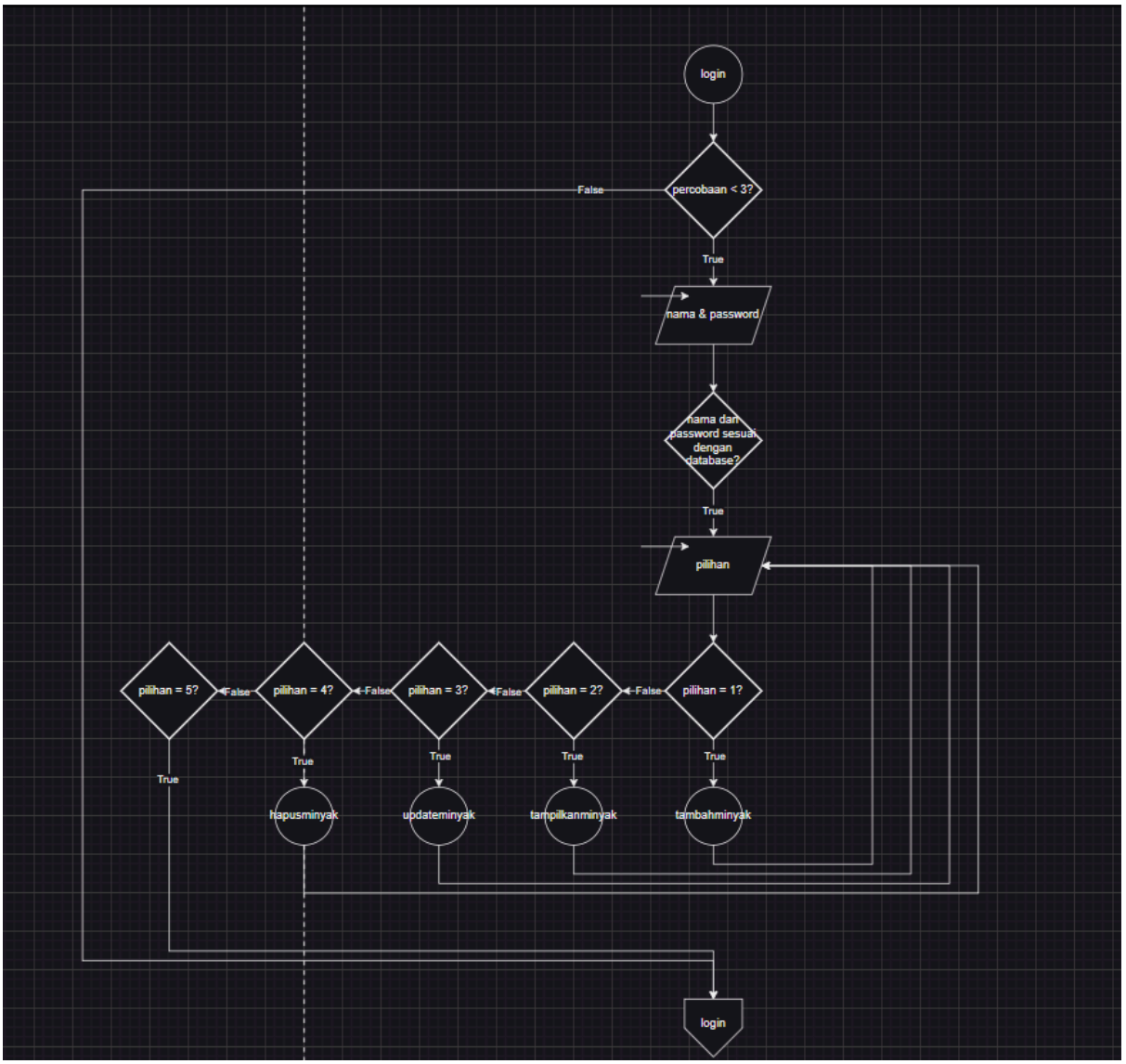
1. Flowchart

- **Main**



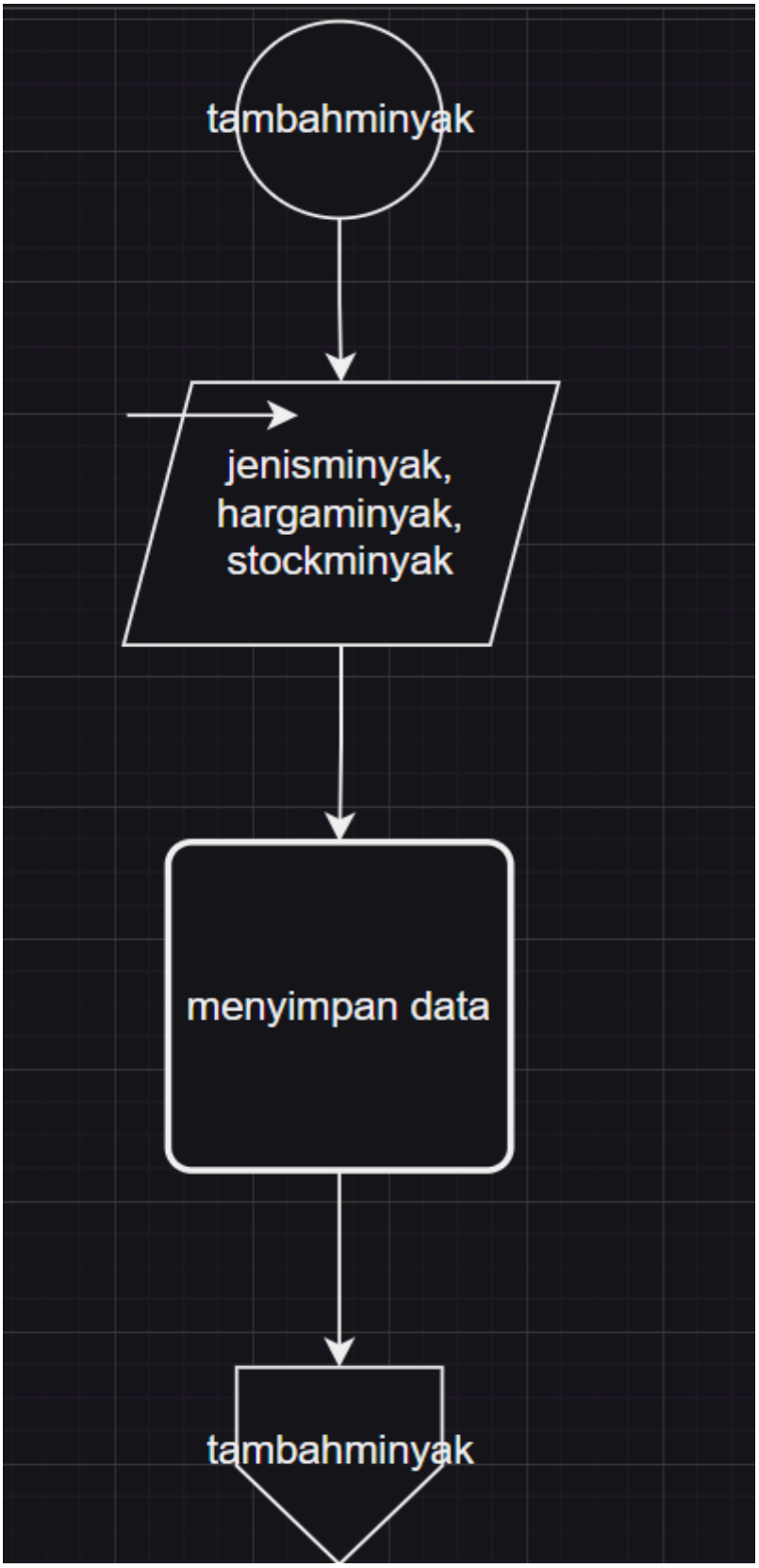
Gambar 1.1 FC-Main

- **Login**



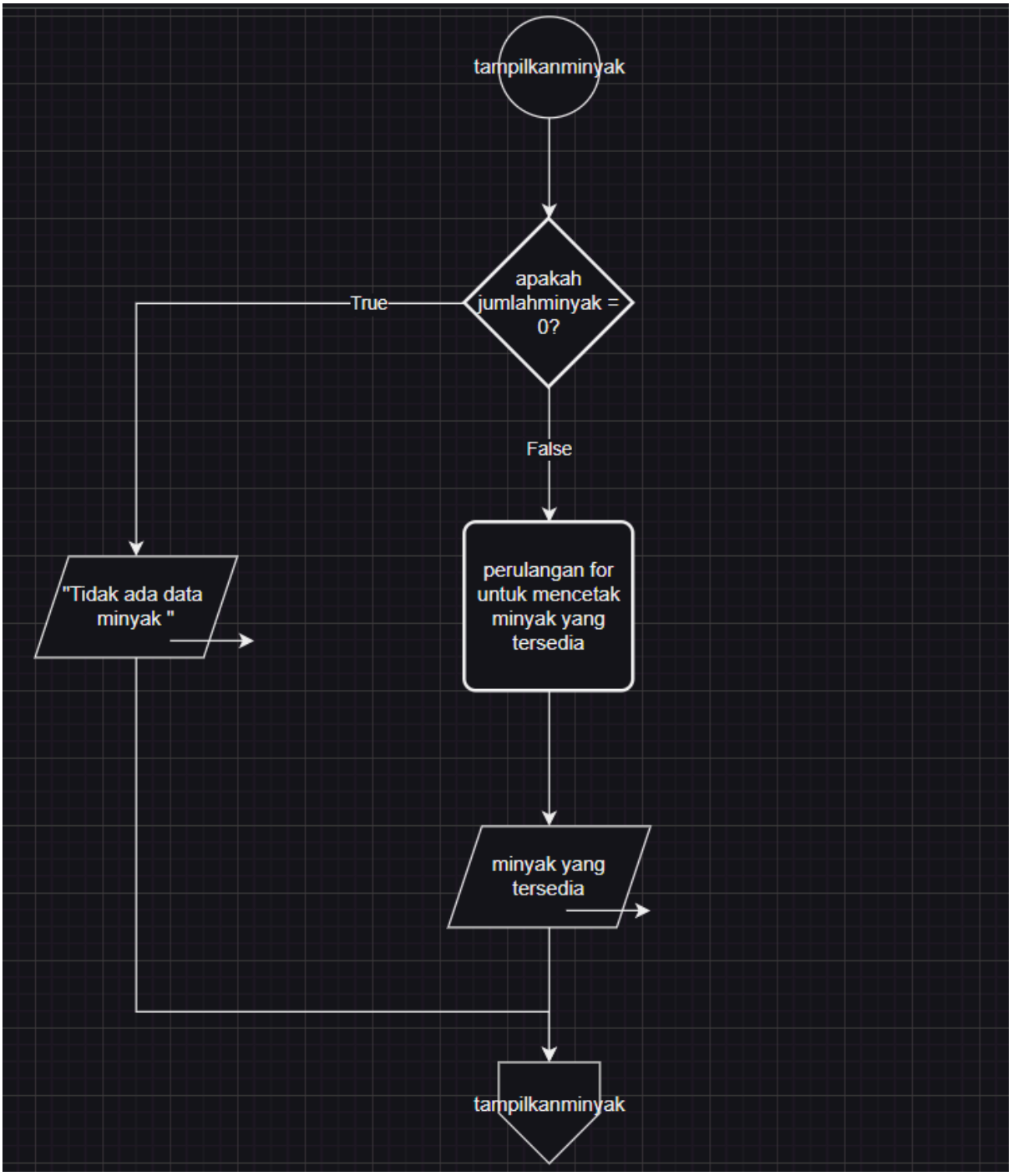
Gambar 1.2 FC-Login

- **Tambah Minyak**



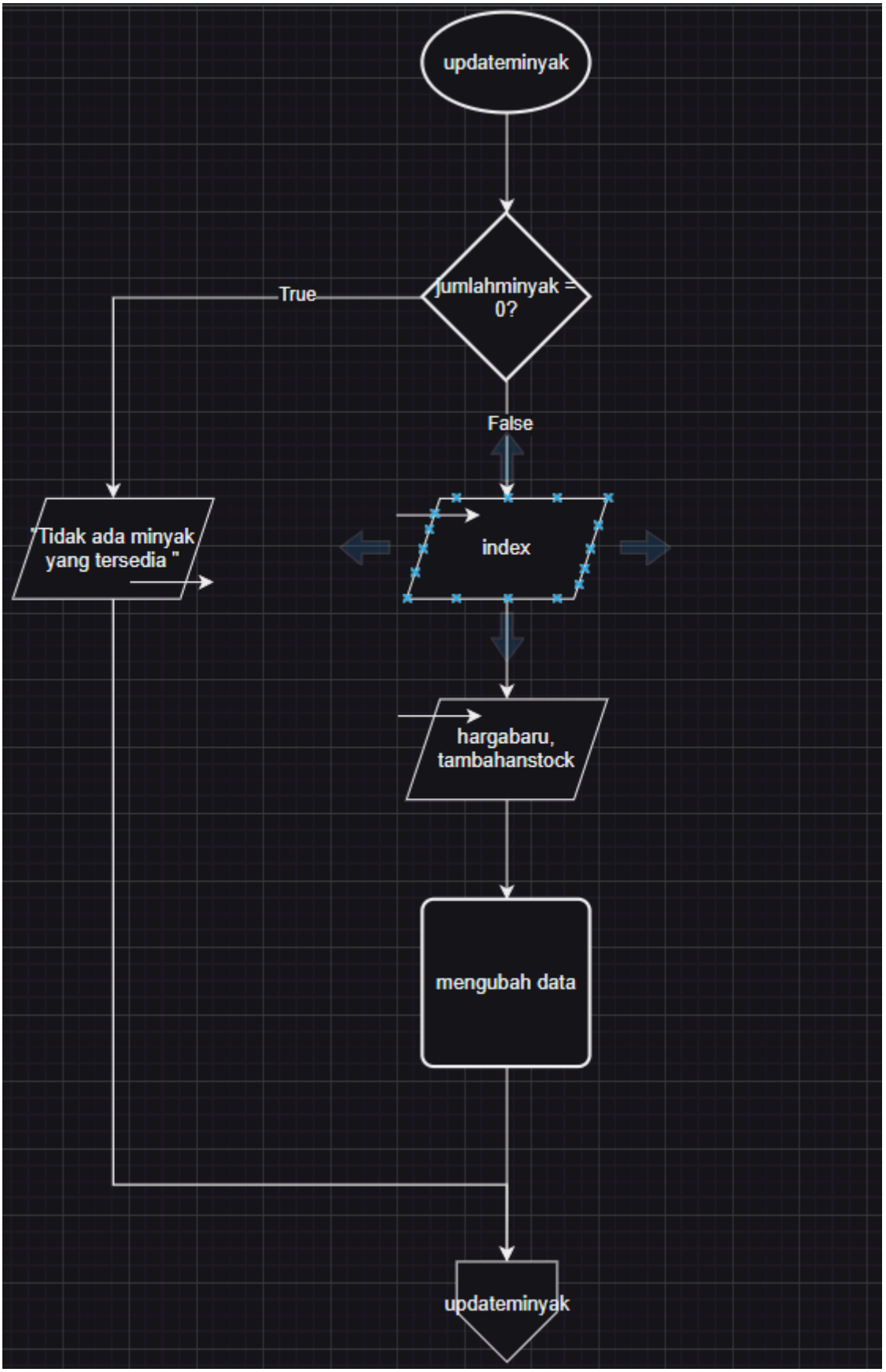
Gambar 1.3 FC-Tambah-Minyak

- **Tampilkan Minyak**



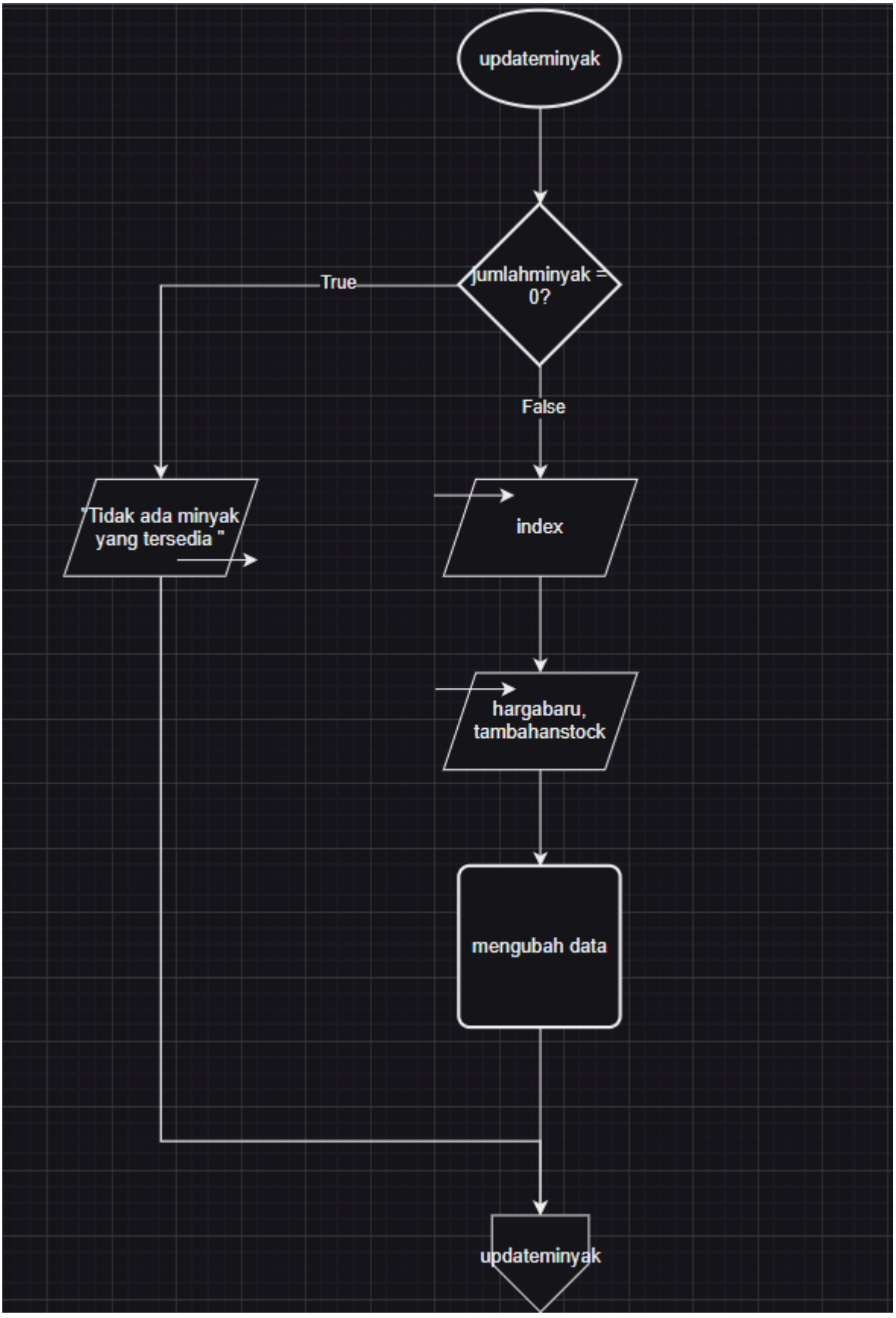
Gambar 1.4 FC-Tampilkan-Minyak

- **Update Minyak**



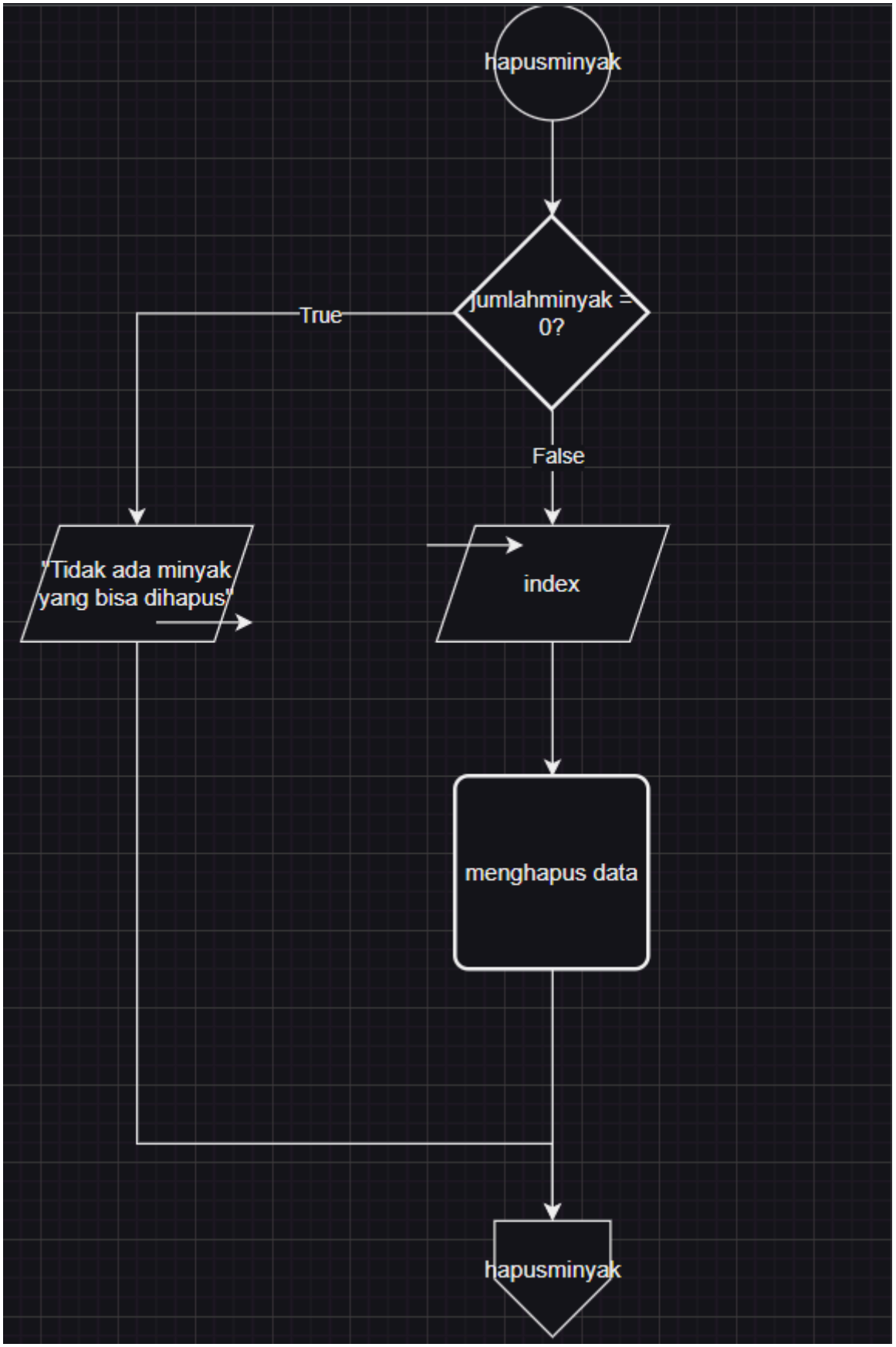
Gambar 1.5 FC-Update-Minyak

- **Update Minyak**



Gambar 1.6 FC-Update-Minyak

- **Hapus Minyak**



Gambar 1.7 FC-Hapus-Minyak

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini memiliki fungsi untuk mendata usaha perminyakan, program bisa menyimpan nama minyak, harga minyak, dan stocknya, Harga dan Stocknya dapat diubah kapan saja, Program juga bisa menampilkan semua minyak yang ada. Program juga bisa menghapus data minyak berdasarkan index yang diinput.

2.2 Penjelasan Alur & Algoritma

A. Login

Pada bagian ini dipakai untuk memastikan bahwa yang login harus sesuai dengan database, jika pengguna program melakukan kesalahan input username ataupun password melebihi dari 3 kali, maka program akan langsung berhenti otomatis, Jika pengguna menginput username dan password benar maka program akan lanjut ke menu utama.

```
string nama, pw;
    while (percobaan < 3) {
        cout << "Masukkan username: ";
        getline(cin, nama);
        cout << "Masukkan password: ";
        getline(cin, pw);
        if (nama == username && pw == password) {
            cout << "Login Berhasil\n";
            break;
        } else {
            cout << "Username atau password salah. Coba lagi.\n";
            percobaan++;
        }
    }
    if (percobaan == 3) {
        cout << "Percobaan melebihi batas, program terhenti.\n";
        return 0;
    }
```

B. Tambah Minyak

fungsi ini digunakan untuk menambahkan data minyak baru ke dalam sistem. Sebelum menambahkan data, program terlebih dahulu memeriksa apakah jumlah minyak yang tersimpan sudah mencapai batas maksimum yang ditentukan oleh MAX_MINYAK yaitu 100, jika batas sudah tercapai maka program otomatis tidak menambahkan data baru tersebut. Jika data masih ada yang tersisa maka program akan meminta input nama, harga, dan stok minyak yang ingin ditambahkan.

```
switch (pilihan) {
            case 1: {
                if (jumlahMinyak >= MAX_MINYAK) {
                    cout << "Data minyak penuh!\n";
                    break;
                }
                cout << "Masukkan nama minyak: ";
                getline(cin, jenisMinyak[jumlahMinyak]);
                cout << "Masukkan harga minyak: ";
                cin >> hargaMinyak[jumlahMinyak];
                cout << "Masukkan stok minyak: ";
                cin >> stokMinyak[jumlahMinyak];
                jumlahMinyak++;
                cout << "Data minyak berhasil ditambahkan!\n";
                break;
            }
```

C. Tampilkan Minyak

Pada fungsi ini minyak ditampilkan semua data minyak yang tersedia, tapi program akan mengecek dulu apakah ada minyak yang tersedia, jika tidak maka program akan memunculkan output “Tidak ada data minyak”. Jika ada maka program akan menampilkan dalam bentuk tabel.

```
case 2: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada data minyak.\n";
                    break;
                }
                cout <<
"\n=================================================================\n";
                cout << "| No  | Nama Minyak         | Harga Minyak      |
Stok Minyak       |\n";
```

```
                cout <<
"=================================================================\n";
                for (int i = 0; i < jumlahMinyak; i++) {
                    cout << "| " << setw(3) << i + 1 << " | "
                        << setw(20) << left << jenisMinyak[i] << " | Rp."
                        << setw(15) << hargaMinyak[i] << " | "
                        << setw(15) << stokMinyak[i] << " |\n";
                }
                cout <<
"=================================================================\n";
                break;
            }
```

D. Update Minyak

Pertama - tama program akan mengecek apakah ada data minyak, jika tidak maka program akan memberikan output “Tidak ada minyak yang bisa di update”. Jika ada maka program akan memunculkan data yang ada menggunakan fungsi Tampilkan Minyak. Program akan meminta pengguna untuk memasukkan index yang ingin di update, jika index tersebut ada maka program akan meminta input harga, dan tambahan stok.

```
case 4: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada minyak yang bisa dihapus!\n";
                    break;
                }
                int index;
                cout << "Pilih nomor minyak yang ingin dihapus: ";
                cin >> index;
                if (index < 1 || index > jumlahMinyak) {
                    cout << "Nomor minyak tidak valid!\n";
                    break;
                }
                index--;
                for (int i = index; i < jumlahMinyak - 1; i++) {
                    jenisMinyak[i] = jenisMinyak[i + 1];
                    hargaMinyak[i] = hargaMinyak[i + 1];
                    stokMinyak[i] = stokMinyak[i + 1];
                }
                jumlahMinyak--;
                cout << "Data minyak berhasil dihapus!\n";
                break;
            }
```

E. Hapus Minyak

Fungsi ini akan mengecek dulu apakah ada data minyak, Jika tidak maka program akan memunculkan output “Tidak ada minyak yang dapat dihapus”. Jika ada maka program akan memunculkan data yang ada dengan fungsi Tampilkan Minyak. Program akan meminta user untuk input index, jika index ada dalam data maka program akan menghapus data yang sesuai dengan index yang diinput.

```
case 4: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada minyak yang bisa dihapus!\n";
                    break;
                }
                int index;
                cout << "Pilih nomor minyak yang ingin dihapus: ";
                cin >> index;
                if (index < 1 || index > jumlahMinyak) {
                    cout << "Nomor minyak tidak valid!\n";
                    break;
                }
                index--;
                for (int i = index; i < jumlahMinyak - 1; i++) {
                    jenisMinyak[i] = jenisMinyak[i + 1];
                    hargaMinyak[i] = hargaMinyak[i + 1];
                    stokMinyak[i] = stokMinyak[i + 1];
                }
                jumlahMinyak--;
                cout << "Data minyak berhasil dihapus!\n";
                break;
            }
```

3. Source Code

A. Login

Fungsi login() digunakan untuk memastikan bahwa hanya pengguna yang memiliki kredensial yang benar yang dapat mengakses sistem. Program meminta pengguna untuk memasukkan username dan password. Jika input yang diberikan sesuai dengan database, program akan menampilkan pesan "Login Berhasil" dan pengguna akan masuk ke dalam menu utama. Jika input salah, pengguna akan diberikan kesempatan hingga tiga kali untuk mencoba kembali. Jika pengguna gagal dalam tiga kali percobaan, program akan menampilkan pesan "Percobaan melebihi batas, program terhenti." dan langsung keluar.

B. Tambah Minyak

Fungsi tambahMinyak() digunakan untuk menambahkan data minyak baru ke dalam sistem. Sebelum menambahkan data, program terlebih dahulu memeriksa apakah jumlah minyak yang tersimpan sudah mencapai batas maksimum yang ditentukan oleh MAX_MINYAK. Jika batas sudah tercapai, fungsi akan mengembalikan nilai 0 sebagai tanda bahwa data minyak penuh. Jika masih ada ruang untuk menambahkan data, program akan meminta pengguna untuk memasukkan nama minyak, harga minyak, dan jumlah stok minyak. Data yang dimasukkan akan disimpan dalam jenisMinyak, hargaMinyak, dan stokMinyak.

C. Tampilkan Minyak

Fungsi tampilkanMinyak() bertugas untuk menampilkan semua data minyak yang telah dimasukkan ke dalam sistem. Fungsi ini terlebih dahulu mengecek apakah ada data minyak yang tersimpan dengan memeriksa nilai jumlahMinyak. Jika tidak ada data, program akan menampilkan pesan "Tidak ada data minyak.". Jika data tersedia, fungsi akan mencetak daftar minyak dalam bentuk tabel yang mencakup nomor urut, nama minyak, harga minyak, dan stok minyak yang tersedia. Untuk merapikan tampilan tabel, program menggunakan manipulasi keluaran seperti setw() agar setiap kolom memiliki lebar yang seragam.

D. Update Minyak

Fungsi updateMinyak() digunakan untuk mengubah harga minyak dan menambah stok minyak yang sudah tersimpan. Fungsi ini pertama - tama menampilkan daftar minyak yang tersedia dengan memanggil tampilkanMinyak(). Jika tidak ada minyak yang tersimpan, fungsi akan menampilkan output “Tidak ada minyak yang dapat diupdate!”. Jika daftar minyak tersedia, pengguna akan diminta untuk memilih nomor minyak yang ingin diperbarui. Setelah memilih nomor, program meminta pengguna untuk memasukkan harga baru dan

tambahan stok minyak. Harga minyak akan langsung diperbarui, sementara stok minyak akan ditambah dengan jumlah yang baru dimasukkan.

E. Hapus Minyak

Fungsi hapusMinyak() memungkinkan pengguna untuk menghapus data minyak dari daftar yang tersimpan. Seperti fungsi update, fungsi ini pertama-tama menampilkan daftar minyak yang tersedia dengan memanggil tampilkanMinyak(). Jika tidak ada minyak dalam sistem, fungsi langsung menampilkan output "Tidak ada minyak yang dapat dihapus!". Jika ada minyak, program meminta pengguna memasukkan index minyak yang ingin dihapus. Setelah index diinput, program akan menggeser seluruh data minyak yang ada setelahnya ke posisi sebelumnya dalam array, menimpa data yang dihapus. Pergeseran ini dilakukan dengan perulangan yang menggantikan nilai elemen saat ini dengan elemen berikutnya. Setelah proses ini selesai, jumlahMinyak dikurangi satu untuk memperbarui jumlah total minyak yang tersimpan.

```
int main() {
    string nama, pw;
    while (percobaan < 3) {
        cout << "Masukkan username: ";
        getline(cin, nama);
        cout << "Masukkan password: ";
        getline(cin, pw);
        if (nama == username && pw == password) {
            cout << "Login Berhasil\n";
            break;
        } else {
            cout << "Username atau password salah. Coba lagi.\n";
            percobaan++;
        }
    }
    if (percobaan == 3) {
        cout << "Percobaan melebihi batas, program terhenti.\n";
        return 0;
    }
    int pilihan;
    do {
        cout << "\n===== MENU CRUD MINYAK =====\n";
        cout << "1. Tambah Tipe Minyak (Create)\n";
        cout << "2. Lihat Stok Minyak (Read)\n";
        cout << "3. Ubah Harga & Tambah Stok Minyak (Update)\n";
```

```
        cout << "4. Hapus Minyak (Delete)\n";
        cout << "5. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();

        switch (pilihan) {
            case 1: {
                if (jumlahMinyak >= MAX_MINYAK) {
                    cout << "Data minyak penuh!\n";
                    break;
                }
                cout << "Masukkan nama minyak: ";
                getline(cin, jenisMinyak[jumlahMinyak]);
                cout << "Masukkan harga minyak: ";
                cin >> hargaMinyak[jumlahMinyak];
                cout << "Masukkan stok minyak: ";
                cin >> stokMinyak[jumlahMinyak];
                jumlahMinyak++;
                cout << "Data minyak berhasil ditambahkan!\n";
                break;
            }
            case 2: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada data minyak.\n";
                    break;
                }
                cout <<
"\n============================================================
======\n";
                cout << "| No  | Nama Minyak         | Harga
Minyak      | Stok Minyak       |\n";
                cout <<
"============================================================
====\n";
                for (int i = 0; i < jumlahMinyak; i++) {
                    cout << "| " << setw(3) << i + 1 << " | "
                        << setw(20) << left << jenisMinyak[i] <<
" | Rp."
                        << setw(15) << hargaMinyak[i] << " | "
                        << setw(15) << stokMinyak[i] << " |\n";
                }
```

```
                cout <<
"================================================================
====\n";
                break;
            }
            case 3: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada minyak yang bisa
diupdate!\n";
                    break;
                }
                int index;
                cout << "Pilih nomor minyak yang ingin diupdate:
";
                cin >> index;
                if (index < 1 || index > jumlahMinyak) {
                    cout << "Nomor minyak tidak valid!\n";
                    break;
                }
                index--;
                cout << "Masukkan harga baru: ";
                cin >> hargaMinyak[index];
                cout << "Masukkan tambahan stok: ";
                int tambahan;
                cin >> tambahan;
                stokMinyak[index] += tambahan;
                cout << "Data minyak berhasil diupdate!\n";
                break;
            }
            case 4: {
                if (jumlahMinyak == 0) {
                    cout << "Tidak ada minyak yang bisa
dihapus!\n";
                    break;
                }
                int index;
                cout << "Pilih nomor minyak yang ingin dihapus:
";
                cin >> index;
                if (index < 1 || index > jumlahMinyak) {
                    cout << "Nomor minyak tidak valid!\n";
                    break;
```

```
                }
                index--;
                for (int i = index; i < jumlahMinyak - 1; i++) {
                    jenisMinyak[i] = jenisMinyak[i + 1];
                    hargaMinyak[i] = hargaMinyak[i + 1];
                    stokMinyak[i] = stokMinyak[i + 1];
                }
                jumlahMinyak--;
                cout << "Data minyak berhasil dihapus!\n";
                break;
            }
            case 5:
                cout << "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 program di jalankan seperti biasa untuk menambahkan data lalu memunculkan data minyak.

```
Masukkan username: Ahmad Habibi
Masukkan password: 2409106104
Login Berhasil

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 1
Masukkan nama minyak = Bimoli
Masukkan harga minyak: 23000
Masukkan stok minyak: 23
Data minyak berhasil ditambahkan!

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 2

=============================================================
| No  | Nama Minyak         | Harga Minyak      | Stok Minyak       |
=============================================================
|   1 | Bimoli              | Rp.23000          | 23                |
=============================================================
```

Gambar 4.1 Skenario-1

2. Skenario 2

User salah menginputkan username ataupun password sebanyak 3 kali sehingga program otomatis terhenti.

```
Masukkan username: a
Masukkan password: a
Username atau password salah. Coba lagi.
Masukkan username: Ahmad Habibi
Masukkan password: 23
Username atau password salah. Coba lagi.
Masukkan username: Praktikum APL
Masukkan password: susah
Username atau password salah. Coba lagi.
Percobaan melebihi batas, program terhenti.
Program selesai karena username atau password salah lebih dari 3 kali.
PS D:\dump tugas\project 0\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.2 Skenario-2

3. Skenario 3

Pada skenario ini pengguna menginput index yang salah pada saat ingin update dan menghapus minyak.

```
Masukkan nama minyak = Bimoli
Masukkan harga minyak: 23000
Masukkan stok minyak: 5
Data minyak berhasil ditambahkan!

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 3

=============================================================================
| No  | Nama Minyak          | Harga Minyak       | Stok Minyak        |
=============================================================================
|   1 | Bimoli               | Rp.23000           | 5                  |
=============================================================================
Pilih nomor minyak yang ingin diupdate: 2
Tidak ada minyak yang bisa diupdate!

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 4

=============================================================================
| No  | Nama Minyak          | Harga Minyak       | Stok Minyak        |
=============================================================================
| 1   | Bimoli               | Rp.23000           | 5                  |
=============================================================================
Pilih nomor minyak yang ingin dihapus: 2
Tidak ada minyak yang bisa dihapus!
```

Gambar 4.3 Skenario-3

4.2 Hasil Output

Contoh output program untuk tambah minyak.

```
Masukkan username: Ahmad Habibi
Masukkan password: 2409106104
Login Berhasil

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 1
Masukkan nama minyak = Bimoli
Masukkan harga minyak: 23000
Masukkan stok minyak: 20
Data minyak berhasil ditambahkan!

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 2

=============================================================
| No  | Nama Minyak         | Harga Minyak      | Stok Minyak       |
=============================================================
|   1 | Bimoli              | Rp.23000          | 20                |
=============================================================

===== MENU CRUD MINYAK =====
1. Tambah Tipe Minyak (Create)
2. Lihat Stok Minyak (Read)
3. Ubah Harga & Tambah Stok Minyak (Update)
4. Hapus Minyak (Delete)
5. Keluar
Pilih menu: 5
Program selesai.
PS D:\dump tugas\project 0\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.4 Output

5. Git

5.1 Add dan commit

Git add untuk memindahkan file ke dalam staging area dan siap untuk di commit. Penggunaan tanda "." setelah add menunjukkan bahwa semua file akan di masukkan ke dalam staging area.
Git commit untuk menaikkan file dari staging area ke dalam reposity github, memberikan commit massage "Finish Post Test 2".

```
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl> git add .
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl> git commit -m "Finish Post Test 2"
[main 5d61e2a] Finish Post Test 2
 3 files changed, 144 insertions(+)
 create mode 100644 post-test/post-test-apl-2/24091060104-AhmadHabibi-PT-2.cpp
 create mode 100644 post-test/post-test-apl-2/24091060104-AhmadHabibi-PT-2.exe
 create mode 100644 post-test/post-test-apl-2/24091060104-AhmadHabibi-PT-2.pdf
```

Gambar 5.1 Add-dan-Commit.

5.2 Push github

Menaikkan file yang sudah di commit untuk ditampilkan di dalam repository github.

```
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl> git push -u origin main
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), 1.30 MiB | 263.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/aeBeeBoy/praktikum-apl.git
   10fed59..5d61e2a  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl>
```

Gambar 5.2 Push