

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

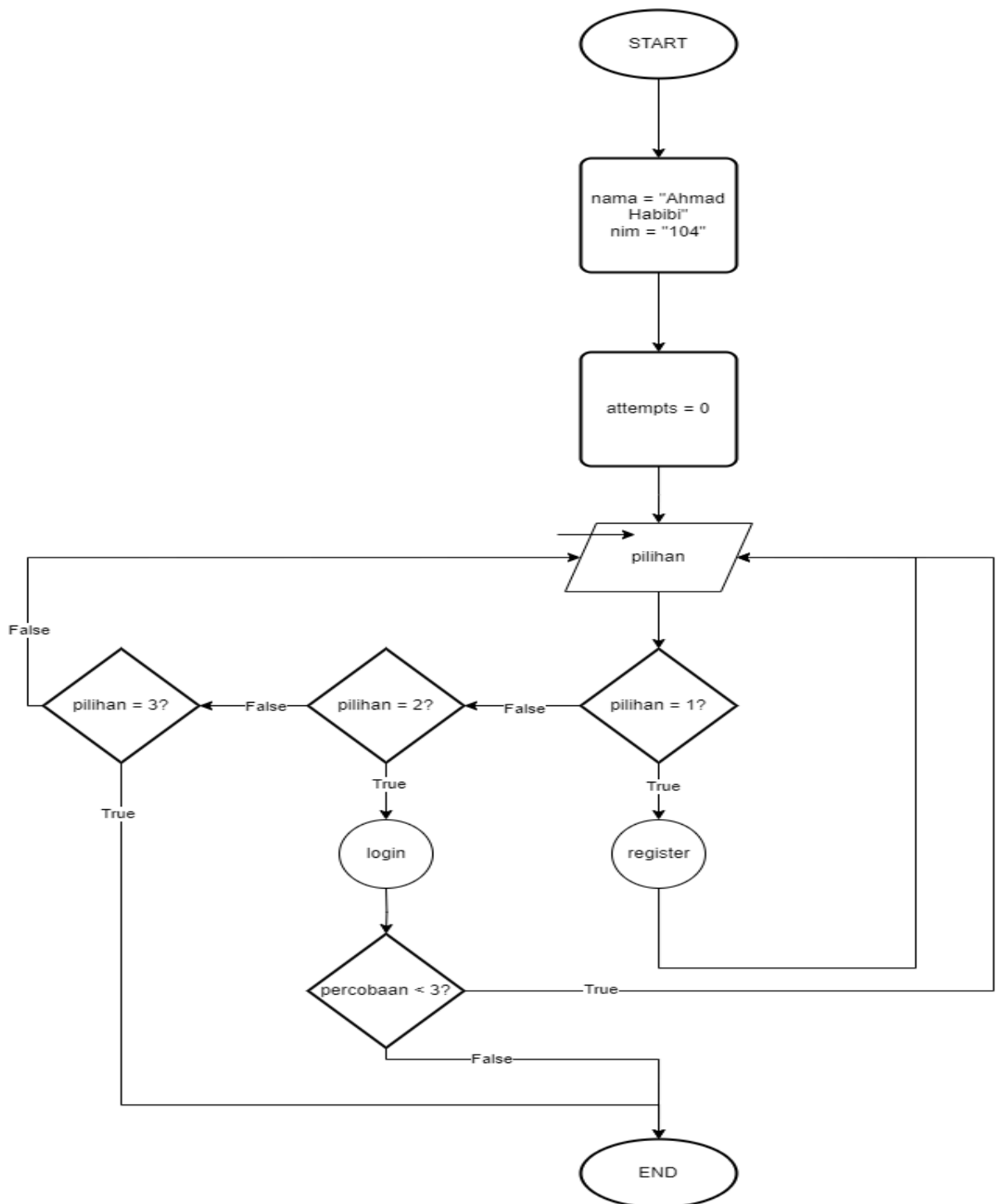


Disusun oleh:
Ahmad Habibi 2409106104
Kelas C1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

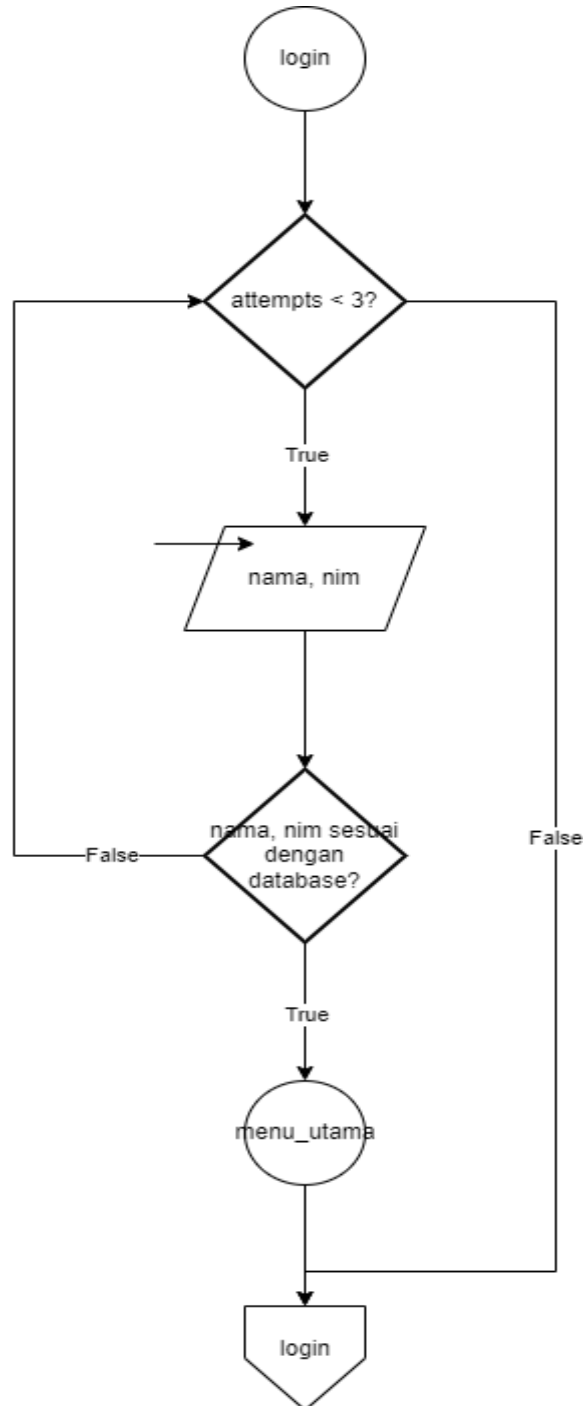
1. Flowchart

- Main



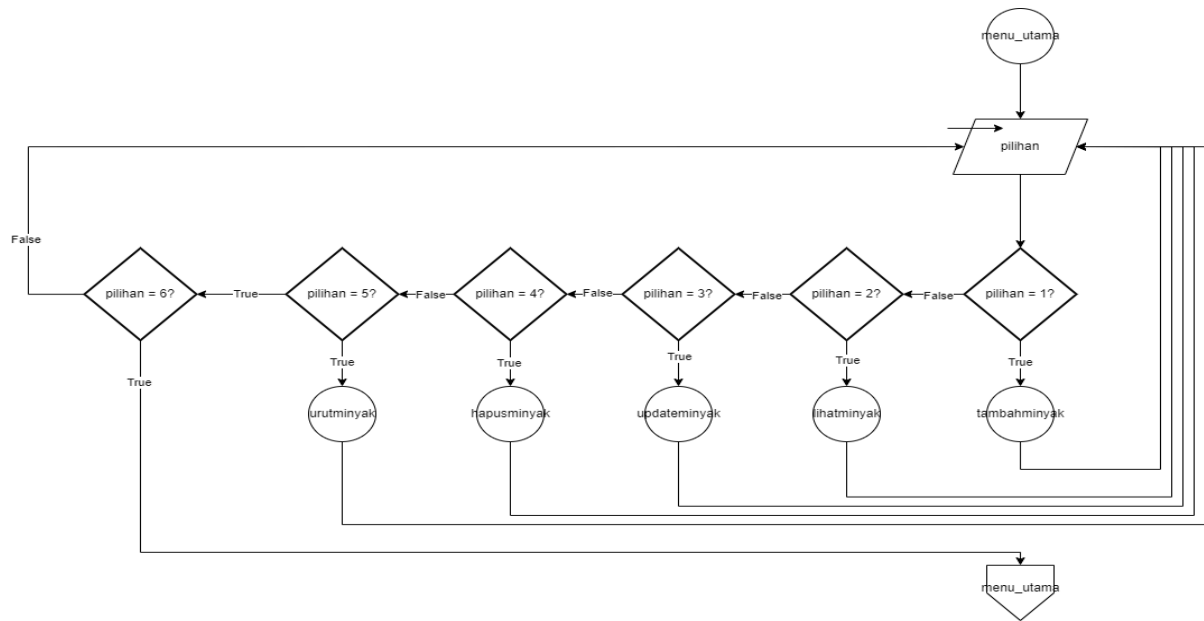
Gambar 2.1 FC-Main

- Login



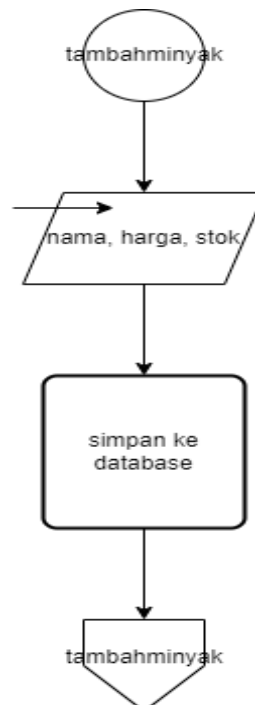
Gambar 2.2 FC-Login

- Menu Utama



Gambar 2.3 FC-Menu-Utama

- Tambah Minyak



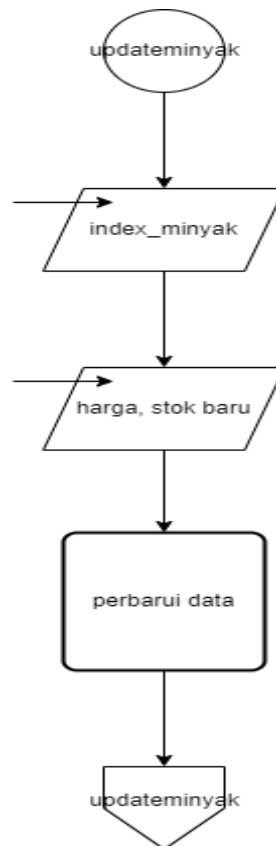
Gambar 2.4 FC-Tambah-Minyak

- Lihat Minyak



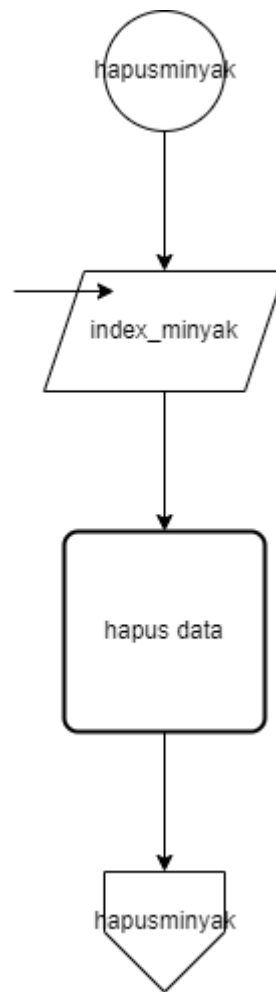
Gambar 2.5 FC-Lihat-Minyak

- Update Minyak



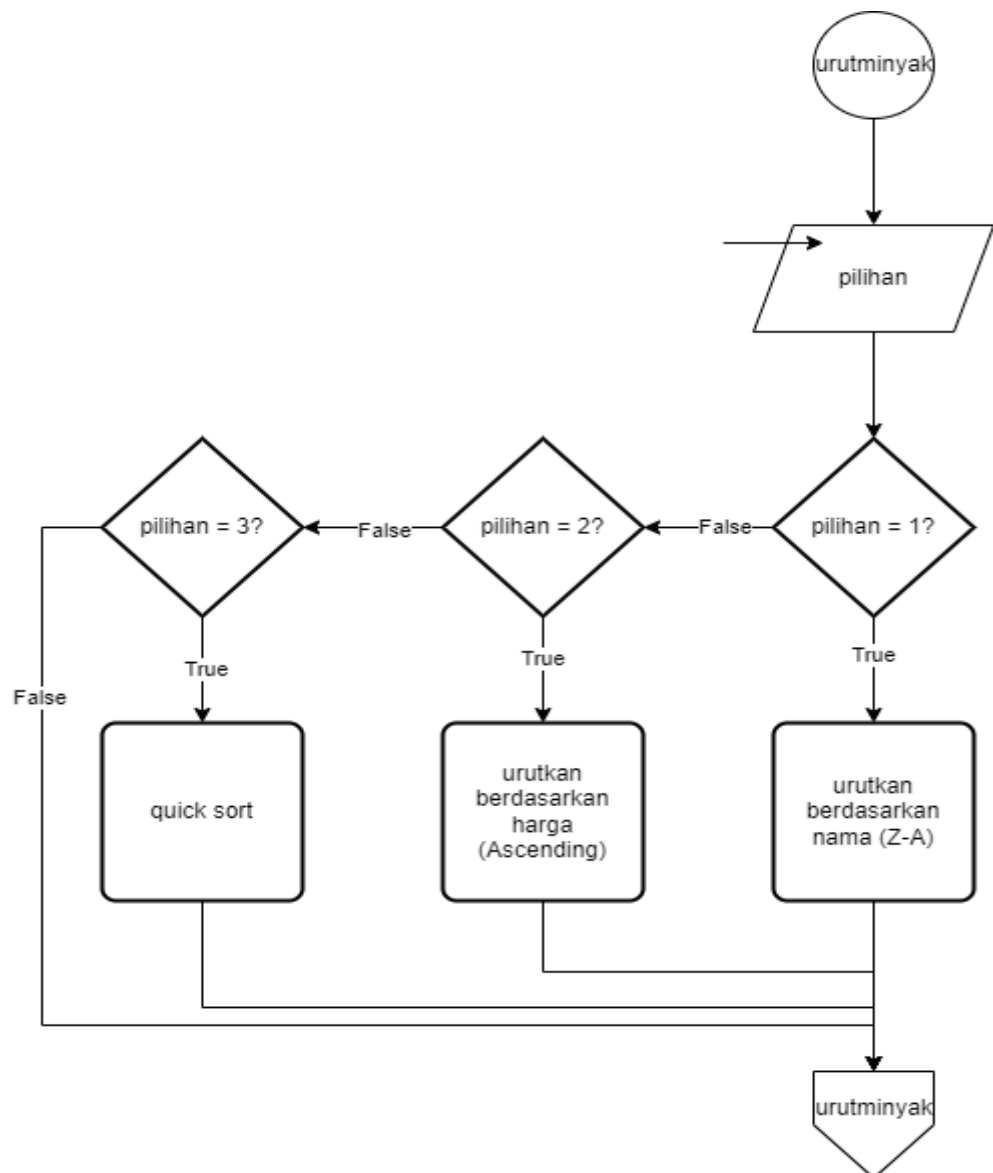
Gambar 2.6 FC-Update-Minyak

- Hapus Minyak



Gambar 2.7 FC-Hapus-Minyak

- Pengurutan Minyak



Gambar 2.8 FC-Pengurutan-Minyak

2. Analisis Program

Program berfungsi untuk mengolah data stok minyak dengan sistem login user. Setelah login, user bisa menambahkan jenis minyak, melihat daftar minyak, mengubah harga dan stok, menghapus data minyak, serta mengurutkan data minyak berdasarkan nama, harga, atau stok. Program ini juga menerapkan fungsi rekursif, pointer, reference, dan quick sort untuk memproses datanya.

3. Source Code

```
#include <iostream>
#include <iomanip>
#include <algorithm>
using namespace std;

struct User {
    string nama;
    string nim;
};

struct Minyak {
    string jenis;
    int harga;
    int stok;
};

struct Data {
    User users[100];
    Minyak daftarMinyak[100];
    int jumlahUser = 0;
    int jumlahMinyak = 0;
};

void quickSortStok(Minyak arr[], int low, int high);
int partition(Minyak arr[], int low, int high);
bool compareNamaDesc(const Minyak &a, const Minyak &b);
bool compareHargaAsc(const Minyak &a, const Minyak &b);

void tampilkanMinyakRekursif(Minyak minyak[], int i, int total) {
    if (i >= total) return;
    cout << "| " << setw(3) << i + 1 << " | "
         << setw(20) << left << minyak[i].jenis << " | Rp."
         << setw(15) << minyak[i].harga << " | "
         << setw(15) << minyak[i].stok << " |\n";
    tampilkanMinyakRekursif(minyak, i + 1, total);
}

void tambahStokDenganPointer(Minyak* minyak, int tambahan) {
    minyak->stok += tambahan;
    cout << "Stok berhasil ditambah menjadi: " << minyak->stok << endl;
}

void ubahHargaDenganAddressOf(Minyak &minyak, int hargaBaru) {
    minyak.harga = hargaBaru;
    cout << "Harga berhasil diubah menjadi: " << minyak.harga << endl;
}
```

```

void registerUser(Data &data) {
    if (data.jumlahUser >= 100) {
        cout << "Pendaftaran penuh!\n";
        return;
    }
    cout << "Masukkan Nama: ";
    getline(cin, data.users[data.jumlahUser].nama);
    cout << "Masukkan NIM: ";
    getline(cin, data.users[data.jumlahUser].nim);
    data.jumlahUser++;
    cout << "Registrasi berhasil!\n";
}

bool login(Data &data, string nama, string nim) {
    for (int i = 0; i < data.jumlahUser; i++) {
        if (data.users[i].nama == nama && data.users[i].nim == nim) {
            return true;
        }
    }
    return false;
}

void tambahMinyak(Data &data) {
    if (data.jumlahMinyak >= 100) {
        cout << "Data minyak penuh!\n";
        return;
    }
    cout << "Masukkan nama minyak: ";
    getline(cin, data.daftarMinyak[data.jumlahMinyak].jenis);
    cout << "Masukkan harga minyak: ";
    cin >> data.daftarMinyak[data.jumlahMinyak].harga;
    cout << "Masukkan stok minyak: ";
    cin >> data.daftarMinyak[data.jumlahMinyak].stok;
    cin.ignore();
    data.jumlahMinyak++;
    cout << "Data minyak berhasil ditambahkan!\n";
}

void lihatMinyak(Data &data) {
    if (data.jumlahMinyak == 0) {
        cout << "Tidak ada data minyak.\n";
        return;
    }
    cout <<
    "\n===== \n";
    cout << "| No | Nama Minyak | Harga Minyak | Stok Minyak | \n";

```

```

        cout <<
"=====\\n";
        tampilkanMinyakRekursif(data.daftarMinyak, 0, data.jumlahMinyak);
        cout <<
"=====\\n";
    }

void updateMinyak(Data &data) {
    if (data.jumlahMinyak == 0) {
        cout << "Tidak ada minyak yang bisa diupdate!\\n";
        return;
    }
    int index, tambahan, hargaBaru;
    cout << "Pilih nomor minyak yang ingin diupdate: ";
    cin >> index;
    if (index < 1 || index > data.jumlahMinyak) {
        cout << "Nomor minyak tidak valid!\\n";
        return;
    }
    index--;
    cout << "Masukkan harga baru: ";
    cin >> hargaBaru;
    ubahHargaDenganAddressOf(data.daftarMinyak[index], hargaBaru);
    cout << "Masukkan tambahan stok: ";
    cin >> tambahan;
    tambahStokDenganPointer(&data.daftarMinyak[index], tambahan);
    cin.ignore();
    cout << "Data minyak berhasil diupdate!\\n";
}

void hapusMinyak(Data &data) {
    if (data.jumlahMinyak == 0) {
        cout << "Tidak ada minyak yang bisa dihapus!\\n";
        return;
    }
    int index;
    cout << "Pilih nomor minyak yang ingin dihapus: ";
    cin >> index;
    if (index < 1 || index > data.jumlahMinyak) {
        cout << "Nomor minyak tidak valid!\\n";
        return;
    }
    for (int i = index - 1; i < data.jumlahMinyak - 1; i++) {
        data.daftarMinyak[i] = data.daftarMinyak[i + 1];
    }
    data.jumlahMinyak--;
    cin.ignore();
    cout << "Data minyak berhasil dihapus!\\n";
}

```

```

}

bool compareNamaDesc(const Minyak &a, const Minyak &b) {
    return a.jenis > b.jenis;
}

bool compareHargaAsc(const Minyak &a, const Minyak &b) {
    return a.harga < b.harga;
}

int partition(Minyak arr[], int low, int high) {
    int pivot = arr[high].stok;
    int i = low - 1;

    for (int j = low; j <= high - 1; j++) {
        if (arr[j].stok <= pivot) {
            i++;
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return i + 1;
}

void quickSortStok(Minyak arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSortStok(arr, low, pi - 1);
        quickSortStok(arr, pi + 1, high);
    }
}

void sortMenu(Data &data) {
    if (data.jumlahMinyak == 0) {
        cout << "Tidak ada data minyak untuk diurutkan.\n";
        return;
    }

    int choice;
    cout << "\n===== MENU PENGURUTAN =====\n";
    cout << "1. Urutkan berdasarkan Nama (Z-A)\n";
    cout << "2. Urutkan berdasarkan Harga (Murah-Mahal)\n";
    cout << "3. Urutkan berdasarkan Stok (Quick Sort)\n";
    cout << "Pilih metode pengurutan: ";
    cin >> choice;
    cin.ignore();

    switch (choice) {

```

```

        case 1:
            sort(data.daftarMinyak, data.daftarMinyak + data.jumlahMinyak,
compareNamaDesc);
            cout << "Data telah diurutkan berdasarkan Nama (Z-A).\n";
            break;
        case 2:
            sort(data.daftarMinyak, data.daftarMinyak + data.jumlahMinyak,
compareHargaAsc);
            cout << "Data telah diurutkan berdasarkan Harga
(Murah-Mahal).\n";
            break;
        case 3:
            quickSortStok(data.daftarMinyak, 0, data.jumlahMinyak - 1);
            cout << "Data telah diurutkan berdasarkan Stok (Quick Sort).\n";
            break;
        default:
            cout << "Pilihan tidak valid!\n";
    }
}

int main() {
    Data data;
    data.users[0] = {"Ahmad Habibi", "104"};
    data.jumlahUser = 1;
    int pilihan;
    string nama, nim;
    int percobaan;
    while (true) {
        cout << "\n===== MENU UTAMA =====\n";
        cout << "1. Register User\n";
        cout << "2. Login User\n";
        cout << "3. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();
        switch (pilihan) {
            case 1:
                registerUser(data);
                break;
            case 2:
                percobaan = 0;
                while (percobaan < 3) {
                    cout << "Masukkan Nama: ";
                    getline(cin, nama);
                    cout << "Masukkan NIM: ";
                    getline(cin, nim);
                    if (login(data, nama, nim)) {
                        cout << "Login Berhasil!\n";

```

```

        int menu;
        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. Urutkan Data Minyak\n";
            cout << "6. Logout\n";
            cout << "Pilih menu: ";
            cin >> menu;
            cin.ignore();
            switch (menu) {
                case 1: tambahMinyak(data); break;
                case 2: lihatMinyak(data); break;
                case 3: updateMinyak(data); break;
                case 4: hapusMinyak(data); break;
                case 5: sortMenu(data); break;
                case 6: cout << "Logout berhasil.\n"; break;
                default: cout << "Pilihan tidak valid!\n";
            }
        } while (menu != 6);
        break;
    }
    cout << "Nama atau NIM salah. Coba lagi.\n";
    percobaan++;
}
if (percobaan == 3) {
    cout << "Percobaan melebihi batas, program terhenti.\n";
    return 0;
}
break;
case 3:
    cout << "Program selesai.\n";
    return 0;
default:
    cout << "Pilihan tidak valid!\n";
}
}
return 0;
}

```

4. Uji Coba dan Hasil Output

- Register, Login dan Tambah Minyak

```
===== MENU UTAMA =====
1. Register User
2. Login User
3. Keluar
Pilih menu: 1
Masukkan Nama: Bimota
Masukkan NIM: 22
Registrasi berhasil!

===== MENU UTAMA =====
1. Register User
2. Login User
3. Keluar
Pilih menu: 2
Masukkan Nama: Bimota
Masukkan NIM: 22
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. Urutkan Data Minyak
6. Logout
Pilih menu: 1
Masukkan nama minyak: Bimoli
Masukkan harga minyak: 23
Masukkan stok minyak: 4
Data minyak berhasil ditambahkan!
```

Gambar 4.1 Percobaan-Register-Login-Tambah-Minyak

- Lihat Minyak dan Update Minyak

```
===== 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. Urutkan Data Minyak
6. Logout
Pilih menu: 2

=====
| No | Nama Minyak | Harga Minyak | Stok Minyak |
=====
| 1 | ww | Rp.2 | 2 |
| 2 | ww | Rp.22 | 22 |
| 3 | z | Rp.1 | 23 |
| 4 | Bimoli | Rp.23 | 4 |
=====
```

Gambar 4.2 Lihat-Minyak-Update-Minyak

- Hapus Minyak dan Lihat Minyak


```

===== 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. Urutkan Data Minyak
6. Logout
Pilih menu: 4
Pilih nomor minyak yang ingin dihapus: 1
Data minyak berhasil dihapus!

```

```

===== 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. Urutkan Data Minyak
6. Logout
Pilih menu: 2

```

=====			
No	Nama Minyak	Harga Minyak	Stok Minyak
=====			
1	ww	Rp.22	22
2	z	Rp.1	23
3	Bimoli	Rp.23	4
=====			

Gambar 4.3 Hapus-Minyak-Lihat-Minyak

- Urutkan data Minyak

```

===== MENU PENGURUTAN =====
1. Urutkan berdasarkan Nama (Z-A)
2. Urutkan berdasarkan Harga (Murah-Mahal)
3. Urutkan berdasarkan Stok (Quick Sort)
Pilih metode pengurutan: 2
Data telah diurutkan berdasarkan Harga (Murah-Mahal).

```

```

===== 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. Urutkan Data Minyak
6. Logout
Pilih menu: 2

```

No	Nama Minyak	Harga Minyak	Stok Minyak
1	z	Rp.1	23
2	ww	Rp.22	22
3	Bimoli	Rp.23	4

Gambar 4.5 Urutkan-Data-Minyak

- Logout

```

===== 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. Urutkan Data Minyak
6. Logout
Pilih menu: 6
Logout berhasil.

===== MENU UTAMA =====
1. Register User
2. Login User
3. Keluar
Pilih menu: 3
Program selesai.

```

Gambar 4.6 Logout

5. Langkah-Langkah Git pada VSCode