

LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT

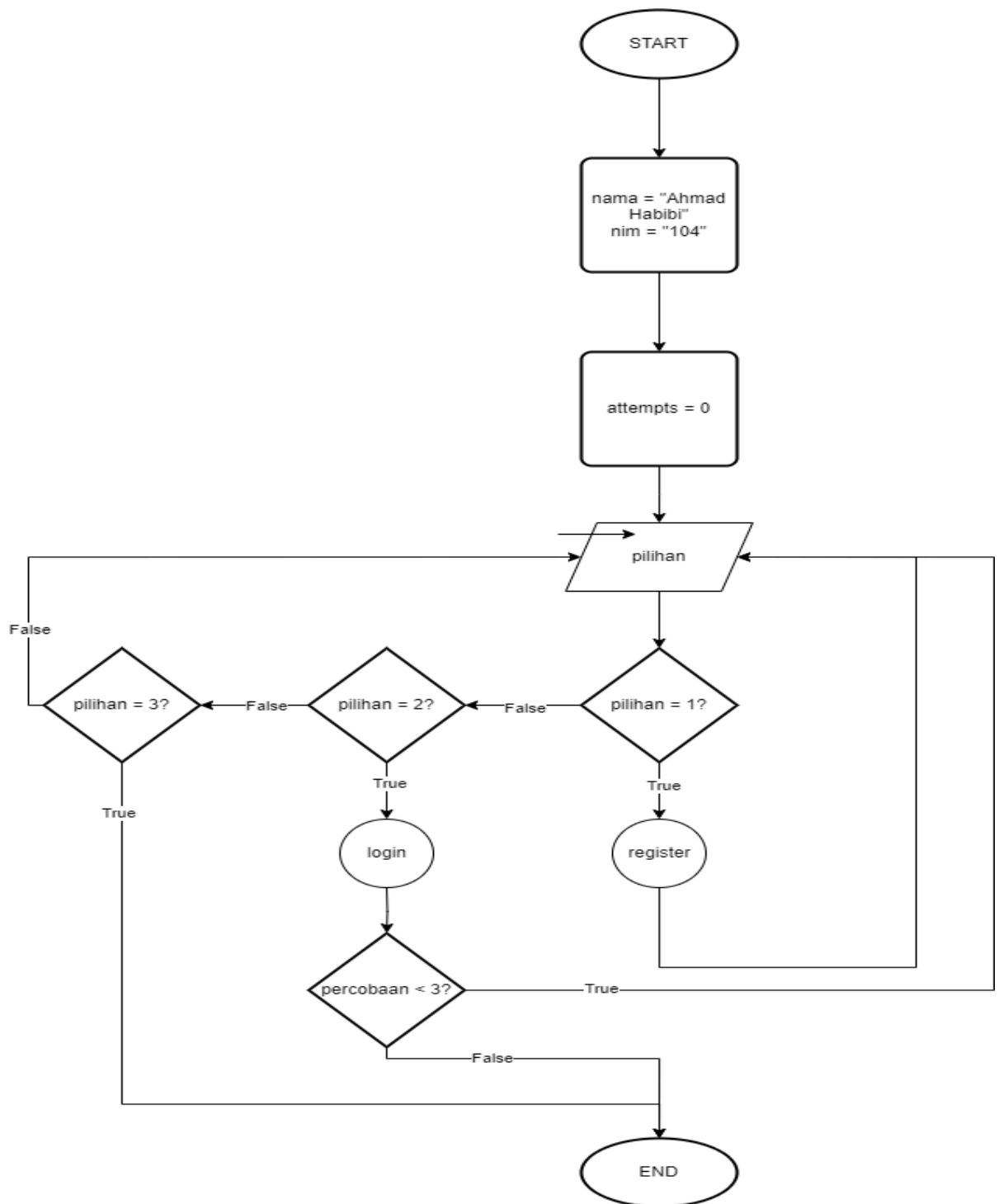


Disusun oleh:
Ahmad Habibi 2409106104
Kelas C1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

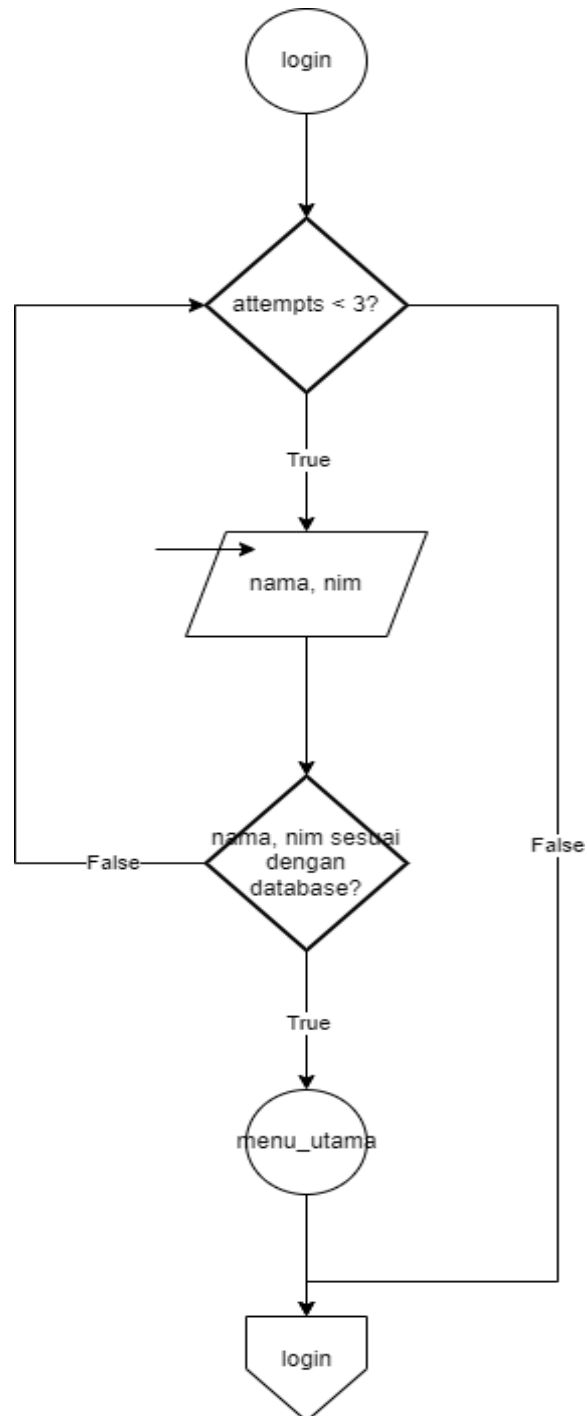
1. Flowchart

- Main



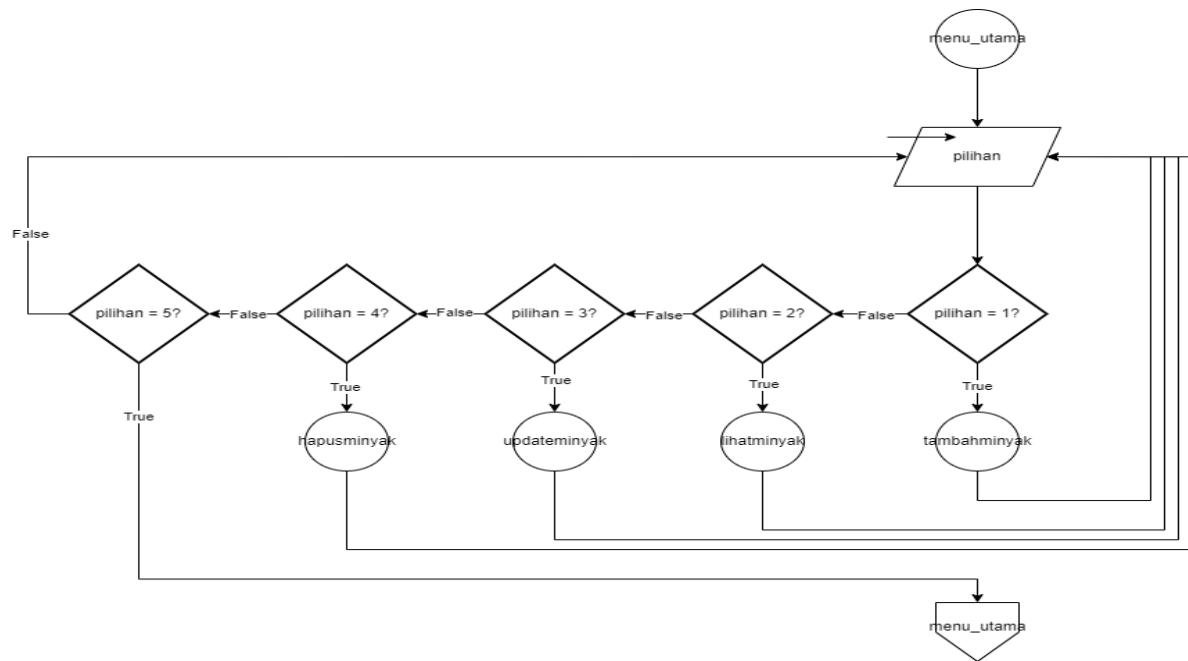
Gambar 2.1 FC-Main

- Login



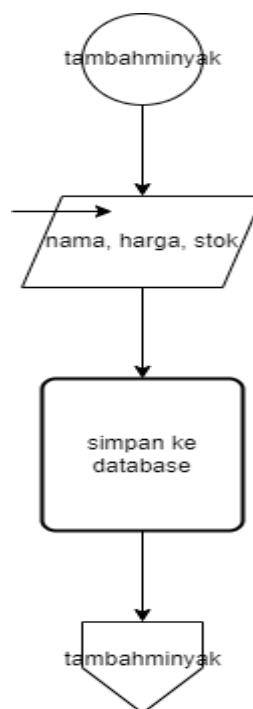
Gambar 2.2 FC-Login

- Menu Utama



Gambar 2.3 FC-Menu-Utama

- Tambah Minyak



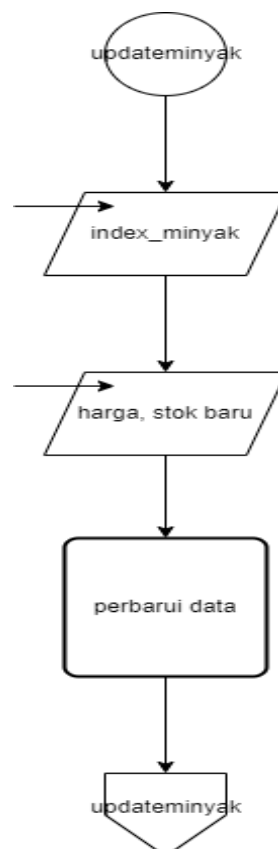
Gambar 2.4 FC-Tambah-Minyak

- Lihat Minyak



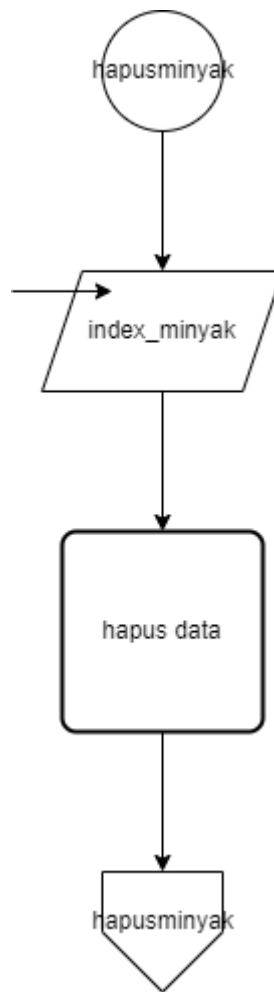
Gambar 2.5 FC-Lihat-Minyak

- Update Minyak



Gambar 2.6 FC-Update-Minyak

- Hapus Minyak



Gambar 2.7 FC-Hapus-Minyak

2. Analisis Program

Program ini adalah sebuah sistem manajemen data minyak goreng yang dibuat menggunakan bahasa pemrograman C++. Tujuan dari program ini adalah untuk mengatur data pengguna dan data berbagai jenis minyak secara interaktif sesuai dengan menu - menu yang ada di program.

Program dimulai dengan menampilkan menu utama yang memberikan tiga pilihan kepada pengguna, yaitu registrasi akun baru, login, dan keluar dari program. Pada bagian registrasi, pengguna diminta untuk menginput nama dan NIM, yang kemudian disimpan dalam array struktur User.

Jika pengguna memilih login, mereka harus memasukkan nama dan NIM yang sesuai dengan data yang telah didaftarkan sebelumnya. Program memberikan batas maksimum tiga kali percobaan login, dan jika gagal tiga kali berturut-turut, maka program akan berhenti secara otomatis.

Setelah berhasil login, pengguna akan masuk ke menu untuk mengelola data minyak goreng. Dalam menu ini, pengguna dapat melakukan beberapa operasi:

- Menambahkan jenis minyak baru beserta harga dan stoknya,
- Melihat seluruh data minyak yang tersedia dalam bentuk tabel,
- Memperbarui harga dan menambahkan stok minyak tertentu,
- Menghapus data minyak dari daftar.

Data minyak disimpan dalam bentuk array dari struktur Minyak, dan untuk menampilkan data, digunakan fungsi rekursif tampilkanMinyakRekursif yang memproses dan menampilkan satu per satu data minyak dari indeks pertama hingga akhir.

Program juga menerapkan konsep fungsi overloading pada lihatMinyak, yaitu satu versi fungsi yang menerima parameter Data &data, dan versi lainnya menerima array minyak serta jumlahnya secara langsung.

Selain itu, program ini juga menerapkan konsep pointer dan address-of:

- Fungsi tambahStokDenganPointer menggunakan parameter pointer (*) untuk menambahkan stok minyak dengan cara mengakses dan memodifikasi data secara langsung melalui alamat memori.
- Fungsi ubahHargaDenganAddressOf menggunakan parameter reference (&) untuk mengubah harga minyak tanpa membuat salinan objek.

3. Source Code

```
#include <iostream>
#include <iomanip>
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 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";
}

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. 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: cout << "Logout berhasil.\n"; break;
                        default: cout << "Pilihan tidak valid!\n";
                    }
                } while (menu != 5);
                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: habiboy
Masukkan NIM: 104
Registrasi berhasil!

===== MENU UTAMA =====
1. Register User
2. Login User
3. Keluar
Pilih menu: 2
Masukkan Nama: habiboy
Masukkan NIM: 104
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. Logout
Pilih menu: 1
Masukkan nama minyak: Bimota
Masukkan harga minyak: 24500
Masukkan stok minyak: 5
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. Logout
Pilih menu: 2

=====
| No | Nama Minyak          | Harga Minyak      | Stok Minyak      |
=====
|  1 | Bimota                | Rp.24500          | 5                |
=====

===== 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. Logout
Pilih menu: 3
Pilih nomor minyak yang ingin diupdate: 1
Masukkan harga baru: 20000
Harga berhasil diubah menjadi: 20000
Masukkan tambahan stok: 3
Stok berhasil ditambah menjadi: 8
Data minyak berhasil diupdate!

```

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. Logout
Pilih menu: 4
Pilih nomor minyak yang ingin dihapus: 1
Data minyak berhasil dihapus!

```

Gambar 4.3 Hapus-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. Logout  
Pilih menu: 2  
Tidak ada data minyak.
```

Gambar 4.4 Lihat-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. Logout  
Pilih menu: 5  
Logout berhasil.  
  
===== MENU UTAMA =====  
1. Register User  
2. Login User  
3. Keluar  
Pilih menu: 3  
Program selesai.
```

Gambar 4.5 Logout

5. Langkah-Langkah Git pada VSCode

- Git add

```
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl\post-test\post-test-apl-5> git add .
```

Gambar 5.1 Git-add

- Git Commit

```
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl\post-test\post-test-apl-5> git commit -m "Finish Posttest 5"
[main 00186fc] Finish Posttest 5
3 files changed, 201 insertions(+)
create mode 100644 post-test/post-test-apl-5/2409106104-AhmadHabibi-PT-5.cpp
create mode 100644 post-test/post-test-apl-5/2409106104-AhmadHabibi-PT-5.exe
create mode 100644 post-test/post-test-apl-5/2409106104-AhmadHabibi-PT-5.pdf
```

Gambar 5.2 Git-commit

- Git Push

```
PS D:\dump tugas\project 0\APL\praktikum-apl\praktikum-apl\post-test\post-test-apl-5> git push
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), 1.10 MiB | 915.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aeBeeBoy/praktikum-apl.git
00168c5..00186fc main -> main
```

Gambar 5.3 Git-push