

**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**

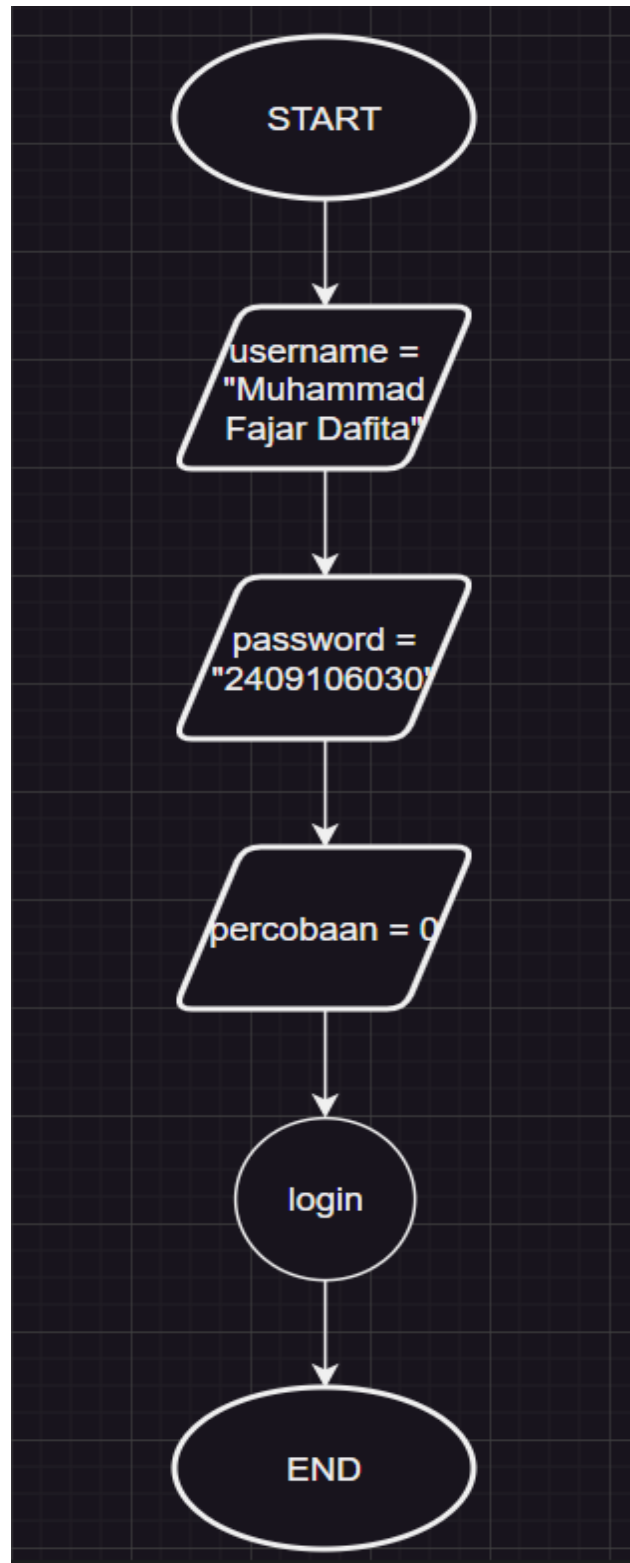


**Disusun oleh:**  
**Nama (2409106030)**  
**Kelas (A2 '24)**

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

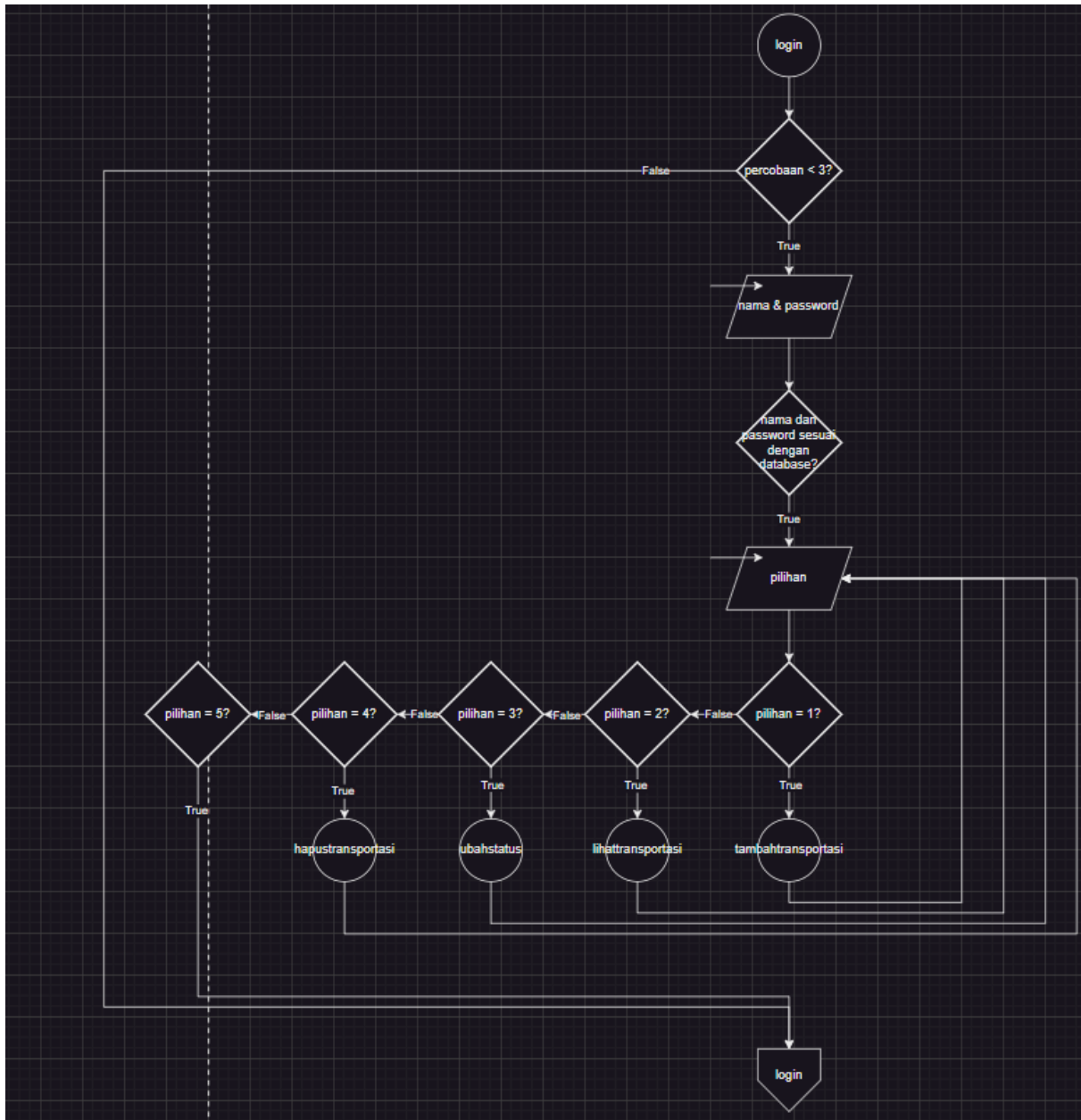
## 1. Flowchart

### a. Main



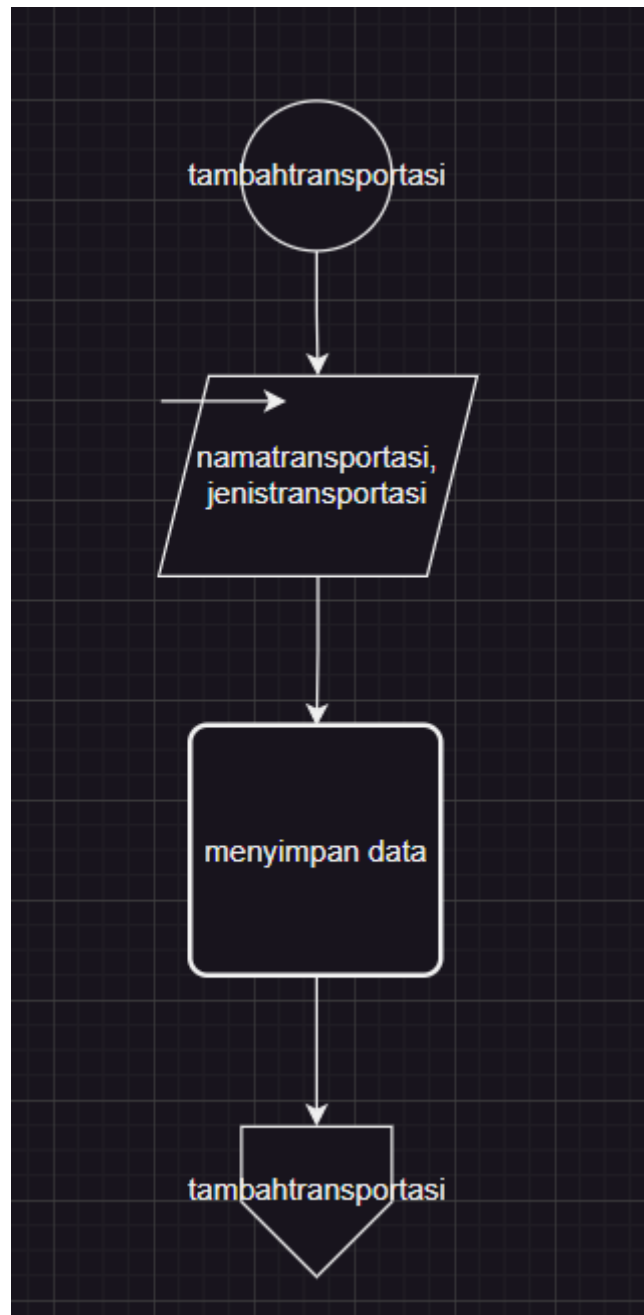
Gambar 1.1 FC-Main

## b. Login



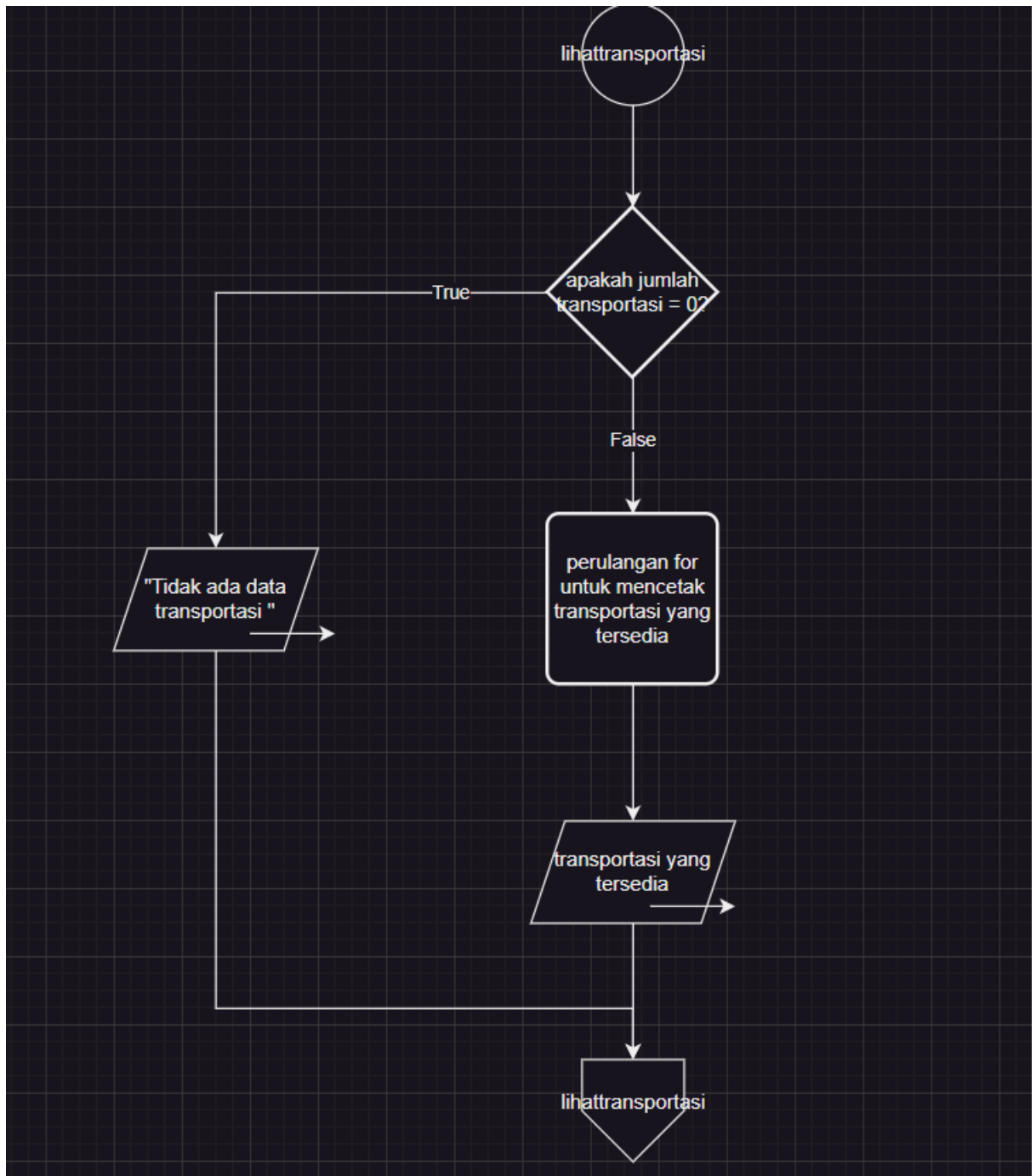
Gambar 1.2 FC-Login

c. Tambah Transportasi



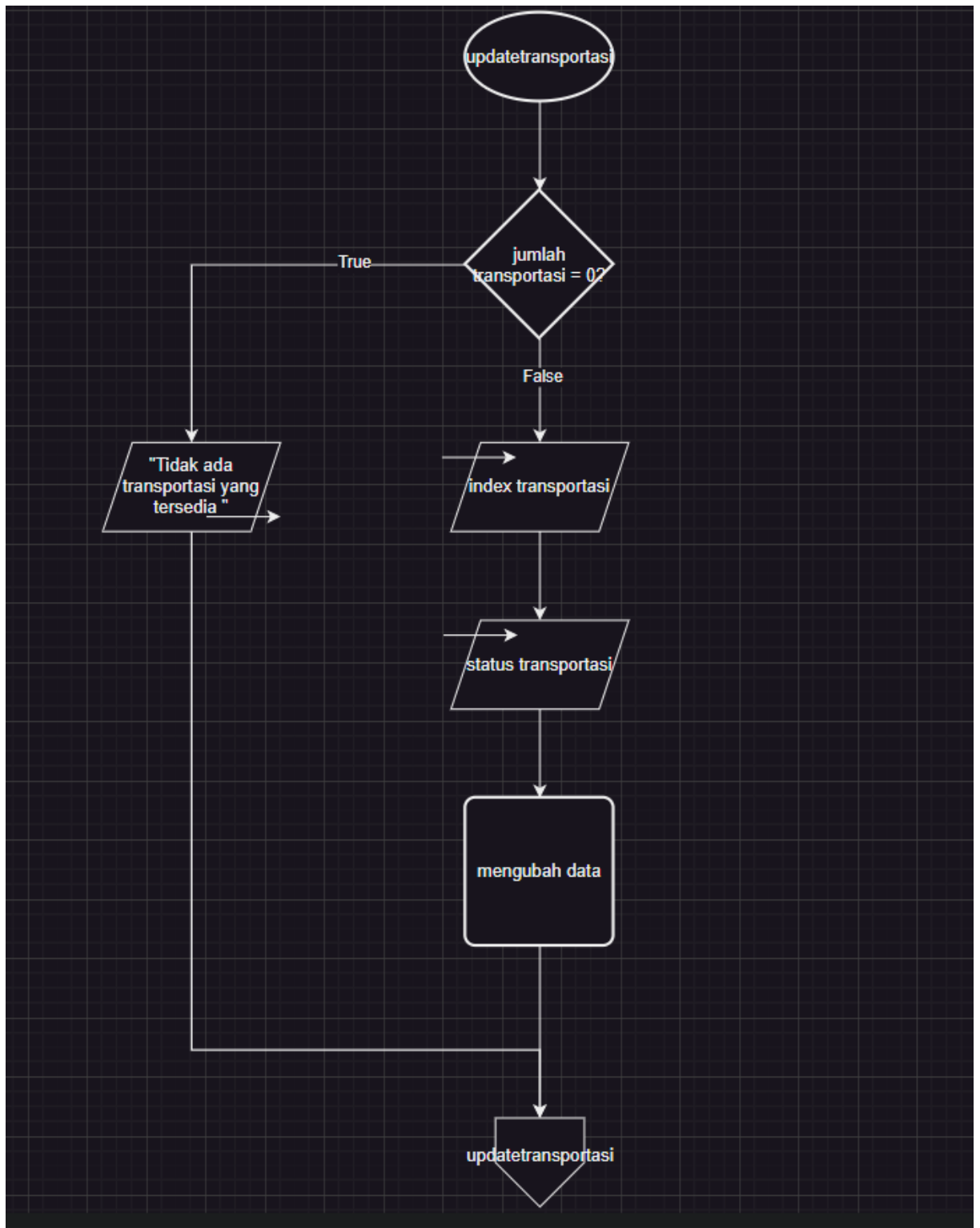
Gambar 1.3 FC-Tambah-Transportasi

#### d. Lihat Transportasi



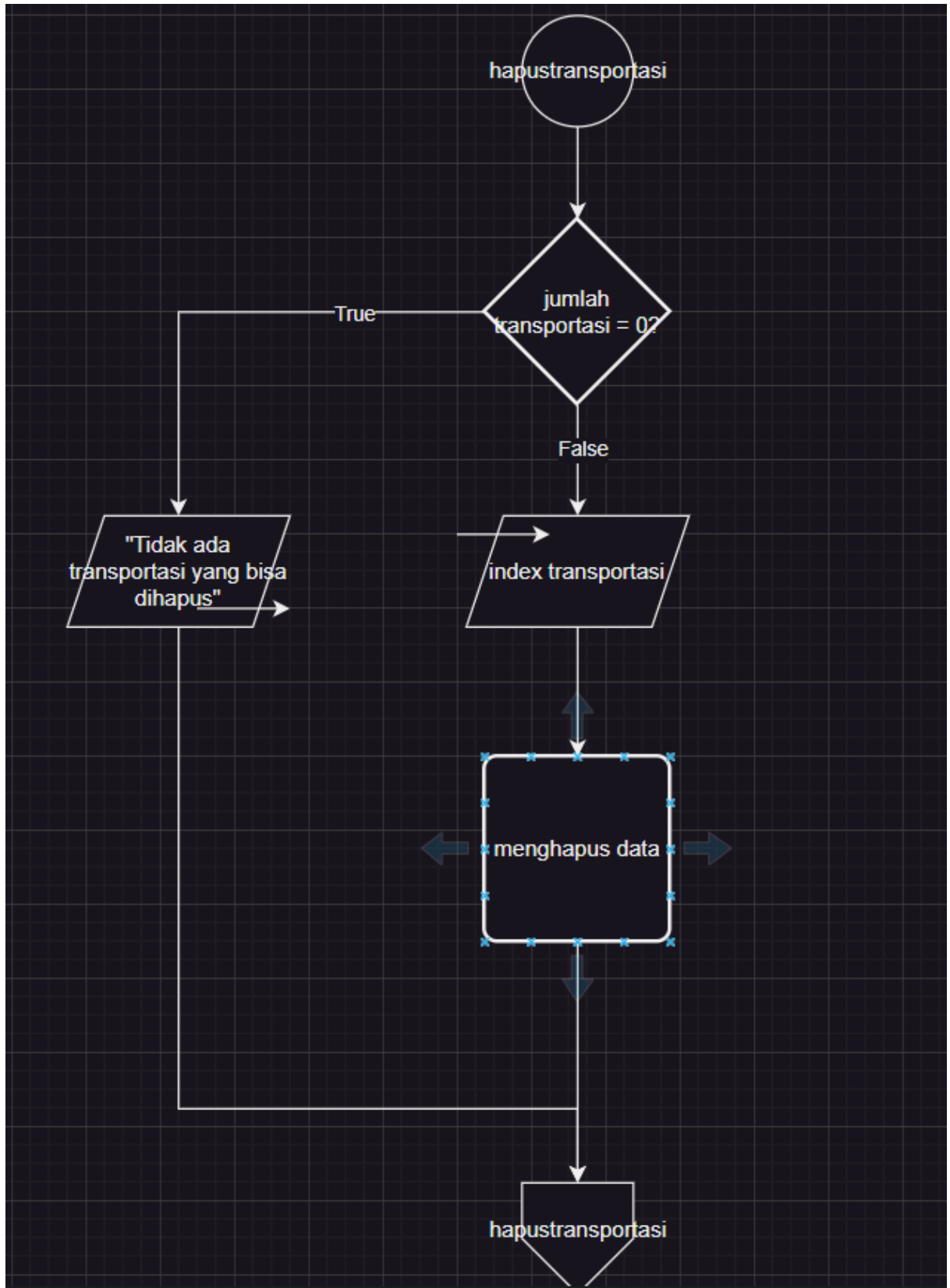
Gambar 1.4 FC-Lihat-Transportasi

e. Update Transportasi



Gambar 1.5 FC-Update-Transportasi

f. Hapus Transportasi



Gambar 1.7 FC-Hapus-Transportasi

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini berfungsi untuk mendata transportasi dari sebuah perusahaan yang menyediakan kendaraan untuk para wisatawan, program bisa memberikan status kepada kendaraan yang ada (Tersedia/Tidak Tersedia).

### 2.2 Penjelasan Alur & Algoritma

#### A. Login

Pada bagian ini, program menentukan apakah percobaan pengguna kurang dari 3? jika tidak maka program akan otomatis berhenti, jika tidak maka program akan meminta user memasukan nama dan NIM, jika salah maka program akan meminta ulang input tapi percobaan akan bertambah 1. Jika user menginput nama dan NIM benar maka, akan lanjut ke menu utama.

```
string nama, input_nim;
while (percobaan < 3) {
    cout << "Masukkan Nama: ";
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    getline(cin, input_nim);
    if (nama == username && input_nim == nim) {
        cout << "Login Berhasil!\n";
        break;
    } else {
        cout << "Nama atau NIM salah! Coba lagi.\n";
        percobaan++;
    }
}
if (percobaan == 3) {
    cout << "Percobaan login habis. Program berhenti.\n";
    return 0;
}
```



## B. Tambah Transportasi

sebelum ini user akan diminta untuk menentukan pilihan pada menu utama, jika user memilih untuk tambah transportasi maka program akan mengecek apakah transportasi yang ada sudah 100? jika iya maka data sudah penuh. Jika tidak maka user akan diminta memasukkan nama transportasi dan jenisnya. Lalu data akan di masukkan ke dalam database.

```
switch (pilihan) {
    case 1: {
        if (jumlahTransportasi >= MAX_TRANSPORT) {
            cout << "Data penuh! Tidak bisa menambah transportasi
lagi.\n";
            break;
        }
        cout << "Masukkan nama transportasi: ";
        getline(cin, namaTransportasi[jumlahTransportasi]);
        cout << "Masukkan jenis transportasi: ";
        getline(cin, jenisTransportasi[jumlahTransportasi]);
        statusTransportasi[jumlahTransportasi] = "Tersedia";
        jumlahTransportasi++;
        cout << "Transportasi berhasil ditambahkan!\n";
        break;
    }
}
```

## C. Lihat Transportasi

Pada pilihan ini program akan mengecek apakah transportasi ada pada database, jika tidak maka program akan memunculkan output “Tidak ada data transportasi”. Jika ada maka program akan menampilkannya menggunakan perulangan for.

```
case 2: {
    if (jumlahTransportasi == 0) {
        cout << "Tidak ada data transportasi.\n";
        break;
    }
    cout <<
"\n===== \n";
    cout << "| No | Nama Transportasi | Jenis |"
Status | \n";
    cout <<
"===== \n";
    for (int i = 0; i < jumlahTransportasi; i++) {
        cout << "| " << setw(3) << i + 1 << " | "
<< setw(20) << left << namaTransportasi[i] << " | "
}
```

```

        << setw(18) << jenisTransportasi[i] << " | "
        << setw(12) << statusTransportasi[i] << " |\n";
    }
    cout <<
    "=====\\n";
    break;
}

```

#### D. Update Transportasi

Pada bagian ini program akan mengecek apakah transportasi ada pada database, jika tidak maka tidak bisa mengedit, jika ada maka program akan meminta user untuk memasukkan index untuk mengedit status transportasi tersebut, setelah itu program akan memprosesnya.

```

case 3: {
    if (jumlahTransportasi == 0) {
        cout << "Tidak ada transportasi untuk diupdate.\n";
        break;
    }
    int index;
    cout << "Pilih nomor transportasi yang ingin diupdate: ";
    cin >> index;
    cin.ignore();
    if (index < 1 || index > jumlahTransportasi) {
        cout << "Nomor tidak valid!\n";
        break;
    }
    cout << "Masukkan status baru (Tersedia/Tidak Tersedia): ";
    getline(cin, statusTransportasi[index - 1]);
    cout << "Status transportasi berhasil diperbarui!\n";
    break;
}

```

#### E. Hapus Transportasi

Seperti biasa, program akan mengecek apakah transportasi ada, jika ada maka program akan meminta user untuk memasukkan index transportasi yang ingin dihapus, lalu program akan menghapus transportasi tersebut, lalu mengupdate databasenya.

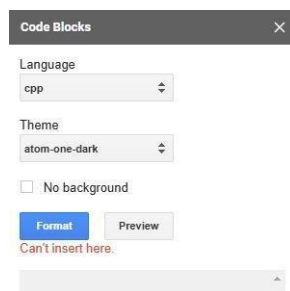
```
case 4: {
    if (jumlahTransportasi == 0) {
        cout << "Tidak ada transportasi untuk dihapus.\n";
        break;
    }
    int index;
    cout << "Pilih nomor transportasi yang ingin dihapus: ";
    cin >> index;
    if (index < 1 || index > jumlahTransportasi) {
        cout << "Nomor tidak valid!\n";
        break;
    }
    for (int i = index - 1; i < jumlahTransportasi - 1; i++) {
        namaTransportasi[i] = namaTransportasi[i + 1];
        jenisTransportasi[i] = jenisTransportasi[i + 1];
        statusTransportasi[i] = statusTransportasi[i + 1];
    }
    jumlahTransportasi--;
    cout << "Transportasi berhasil dihapus!\n";
    break;
}
```

### 3. Source Code

*(Pada bagian ini, sertakan hanya bagian kode yang berisi fitur penting dari program. Tidak perlu memasukkan seluruh kode, cukup bagian yang relevan dengan fungsionalitas utama.)*

#### Catatan:

- **Gunakan ekstensi Code Blocks di Google Docs** untuk menampilkan kode dengan format yang lebih rapi.
- Unduh dan instal ekstensi di sini:  
[Code Blocks - Google Workspace Marketplace](#)
- **Jangan gunakan tabel** untuk menampilkan kode program.
- Untuk **Source Code Font Uk : 10**
- Gunakan **“Language”** dan **“Theme”** yang sudah ditentukan



#### Contoh Format dalam Laporan:

##### A. Fitur login

Pada fitur ini program akan mengecek total percobaan dari user, jika kurang dari 3 maka akan lanjut ke menu utama.

##### B. Fitur Tambah Transportasi

Pada fitur ini program akan meminta input nama dan jenis transportasi lalu akan dimasukkan ke dalam database.

##### C. Fitur Lihat Transportasi

Pada fitur ini program akan melakukan perulangan for untuk mencetak database untuk ditampilkan ke pengguna.

#### D. Fitur Update Transportasi

Pada fitur ini program akan meminta user untuk memasukkan index, setelah index ada di database maka program akan meminta untuk memasukkan status transportasi tersebut.

#### E. Fitur Hapus Transportasi

Pada program ini user akan diminta memasukkan index, setelah berhasil program akan menghapus transportasi berdasarkan index dari database.

#### Source Code :

```
int main() {
    string nama, input_nim;
    while (percobaan < 3) {
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, input_nim);
        if (nama == username && input_nim == nim) {
            cout << "Login Berhasil!\n";
            break;
        } else {
            cout << "Nama atau NIM salah! Coba lagi.\n";
            percobaan++;
        }
    }
    if (percobaan == 3) {
        cout << "Percobaan login habis. Program berhenti.\n";
        return 0;
    }
    int pilihan;
    do {
        cout << "\n==== Manajemen Transportasi Wisata =====\n";
        cout << "1. Tambah Transportasi\n";
        cout << "2. Lihat Daftar Transportasi\n";
        cout << "3. Ubah Status Transportasi\n";
        cout << "4. Hapus Transportasi\n";
        cout << "5. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();

        switch (pilihan) {
            case 1: {
                if (jumlahTransportasi >= MAX_TRANSPORT) {
                    cout << "Data penuh! Tidak bisa menambah transportasi";
                }
            }
        }
    } while (true);
}
```

```

lagi.\n";
        break;
    }
    cout << "Masukkan nama transportasi: ";
    getline(cin, namaTransportasi[jumlahTransportasi]);
    cout << "Masukkan jenis transportasi: ";
    getline(cin, jenisTransportasi[jumlahTransportasi]);
    statusTransportasi[jumlahTransportasi] = "Tersedia";
    jumlahTransportasi++;
    cout << "Transportasi berhasil ditambahkan!\n";
    break;
}
case 2: {
    if (jumlahTransportasi == 0) {
        cout << "Tidak ada data transportasi.\n";
        break;
    }
    cout <<
"\n===== \n";
    cout << "| No | Nama Transportasi | Jenis |"
Status | \n";
    cout <<
"===== \n";
    for (int i = 0; i < jumlahTransportasi; i++) {
        cout << "| " << setw(3) << i + 1 << " | "
        << setw(20) << left << namaTransportasi[i] << " | "
        << setw(18) << jenisTransportasi[i] << " | "
        << setw(12) << statusTransportasi[i] << " |\n";
    }
    cout <<
"===== \n";
    break;
}
case 3: {
    if (jumlahTransportasi == 0) {
        cout << "Tidak ada transportasi untuk diupdate.\n";
        break;
    }
    int index;
    cout << "Pilih nomor transportasi yang ingin diupdate: ";
    cin >> index;
    cin.ignore();
    if (index < 1 || index > jumlahTransportasi) {
        cout << "Nomor tidak valid!\n";
        break;
    }
    cout << "Masukkan status baru (Tersedia/Tidak Tersedia): ";
    getline(cin, statusTransportasi[index - 1]);

```

```

        cout << "Status transportasi berhasil diperbarui!\n";
        break;
    }
    case 4: {
        if (jumlahTransportasi == 0) {
            cout << "Tidak ada transportasi untuk dihapus.\n";
            break;
        }
        int index;
        cout << "Pilih nomor transportasi yang ingin dihapus: ";
        cin >> index;
        if (index < 1 || index > jumlahTransportasi) {
            cout << "Nomor tidak valid!\n";
            break;
        }
        for (int i = index - 1; i < jumlahTransportasi - 1; i++) {
            namaTransportasi[i] = namaTransportasi[i + 1];
            jenisTransportasi[i] = jenisTransportasi[i + 1];
            statusTransportasi[i] = statusTransportasi[i + 1];
        }
        jumlahTransportasi--;
        cout << "Transportasi 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 user memasukkan semua input dengan benar dan menjalankan program dengan normal, program menampilkan output program secara normal.

#### **2. Skenario 2**

Pada skenario ini user salah menginput username hingga 3 kali salah, sehingga program akan otomatis terhenti.



## 4.2 Hasil Output

```
Masukkan Nama: Muhammad Fajar Dafita
Masukkan NIM: 2409106030
Login Berhasil!

===== Manajemen Transportasi Wisata =====
1. Tambah Transportasi
2. Lihat Daftar Transportasi
3. Ubah Status Transportasi
4. Hapus Transportasi
5. Keluar
Pilih menu: 1
Masukkan nama transportasi: HI-01B
Masukkan jenis transportasi: Bus
Transportasi berhasil ditambahkan!

===== Manajemen Transportasi Wisata =====
1. Tambah Transportasi
2. Lihat Daftar Transportasi
3. Ubah Status Transportasi
4. Hapus Transportasi
5. Keluar
Pilih menu: 2

=====
| No | Nama Transportasi | Jenis | Status |
=====
| 1 | HI-01B | Bus | Tersedia |
=====

===== Manajemen Transportasi Wisata =====
1. Tambah Transportasi
2. Lihat Daftar Transportasi
3. Ubah Status Transportasi
4. Hapus Transportasi
5. Keluar
Pilih menu: 5
Program selesai.
```

Gambar 4.1 Output-1

```
Masukkan Nama: A
Masukkan NIM: A
Nama atau NIM salah! Coba lagi.
Masukkan Nama: a
Masukkan NIM: a
Nama atau NIM salah! Coba lagi.
Masukkan Nama: Project0
Masukkan NIM: www
Nama atau NIM salah! Coba lagi.
Percobaan login habis. Program berhenti.
```

Gambar 4.2 Output-2

## 5. Git

### A. Git add

Git add digunakan untuk menaikkan file ke dalam stagin area.

```
PS D:\praktikum-apl> git add .
```

Gambar 5.1 Git-Add

### B. Git Commit

Git Commit digunakan untuk menaikkan file dari staging area ke github dan siap untuk di push ke github, dan memberikan pesan "Finish Post Test 2"

```
PS D:\praktikum-apl> git commit -m "Finish Post Test 2"
[main 414fec] Finish Post Test 2
3 files changed, 126 insertions(+)
create mode 100644 post-test/post-test-apl-2/2409106030-MuhammadFajarDafita-PT-2.cpp
create mode 100644 post-test/post-test-apl-2/2409106030-MuhammadFajarDafita-PT-2.exe
create mode 100644 post-test/post-test-apl-2/2409106030-MuhammadFajarDafita-PT-2.pdf
```

Gambar 5.2 Git-Commit

### C. Git Push

Git push digunakan untuk menaikkan file ke github agar bisa dilihat publik.

```
PS D:\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
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), 1.22 MiB | 952.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/fajarisme/praktikum-apl.git
 88dca59..414fec main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Git-Push