

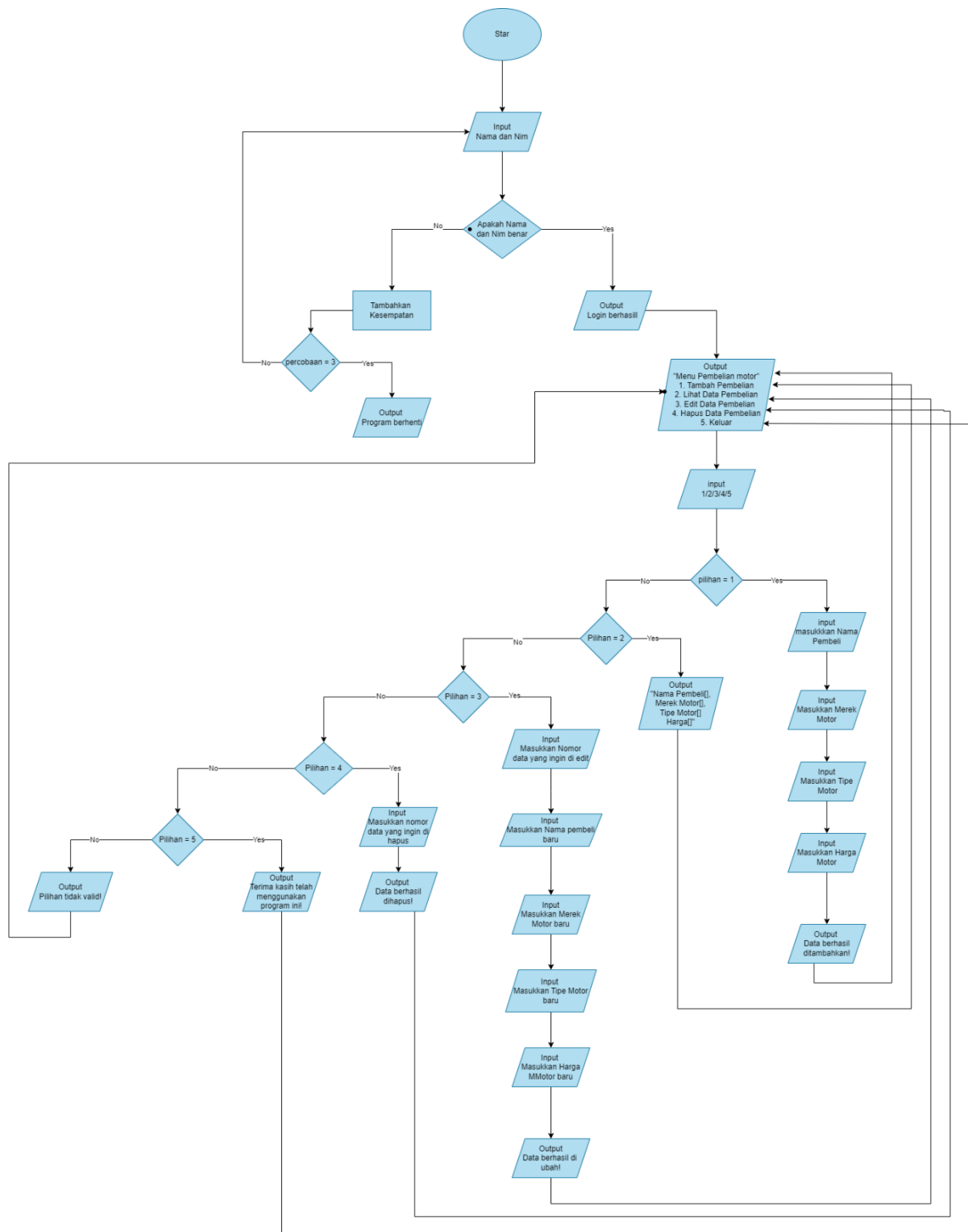
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Muhammad Fizriannur
2409106105
Kelas C1`24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini adalah sistem pembelian motor berbasis C++ yang memungkinkan pengguna untuk menyimpan, melihat, mengedit, dan menghapus data pembelian motor.

Fitur utama:

1. Login pengguna – Memerlukan input nama dan NIM untuk mengakses program (maksimal 3 kali percobaan).
2. CRUD Data Pembelian – Pengguna dapat menambah, menampilkan, mengedit, dan menghapus data pembelian motor yang disimpan dalam vector.
3. Tampilan Tabel – Data pembelian motor ditampilkan dalam format tabel untuk memudahkan pembacaan.
4. Looping Menu – Program terus berjalan hingga pengguna memilih opsi keluar (5).

2.2 Penjelasan Alur & Algoritma

Alur Program

1. Login Pengguna
 - a. Program meminta input Nama dan NIM pengguna sebagai data login.
 - b. Pengguna diberikan 3 kali kesempatan untuk login.
 - c. Jika gagal 3 kali, program langsung berhenti.
 - d. Jika berhasil, pengguna diarahkan ke menu utama.
2. Menampilkan Menu Utama
 - Setelah login berhasil, program menampilkan menu pilihan:
 - a. Tambah Data Pembelian Motor
 - b. Lihat Data Pembelian
 - c. Edit Data Pembelian
 - d. Hapus Data Pembelian
 - e. Keluar
 - Pengguna memasukkan angka sesuai pilihan.
3. Proses CRUD (Create, Read, Update, Delete)
 - (1) Tambah Pembelian
 - a. Pengguna menginput Nama Pembeli, Merek Motor, Tipe Motor, dan Harga Motor.
 - b. Data disimpan dalam vector daftarPembelian.
 - (2) Lihat Data Pembelian
 - a. Program menampilkan daftar pembelian dalam bentuk tabel.
 - (3) Edit Data Pembelian
 - a. Pengguna memilih nomor data yang ingin diedit.
 - b. Jika nomor valid, data bisa diperbarui.

(4) Hapus Data Pembelian

- a. Pengguna memilih nomor data yang ingin dihapus.
- b. Jika nomor valid, data akan dihapus dari **vector**.

(5) Keluar

- a. Program berhenti dengan pesan "Terima kasih telah menggunakan program ini!".

3. Source Code

a. Ini adalah fungsi program login dengan batas 3 x percobaan, menerima membaca string dari aliran input, dan menampilkan pesan jika gagal.

```
int main() {
    string namaUser, nimUser;
    string namaInput, nimInput;
    vector<PembelianMotor> daftarPembelian;
    int kesempatan = 3;

    cout << "=== Login Sistem Pembelian Motor ===" << endl;
    cout << "Masukkan Nama: ";
    getline(cin, namaUser);
    cout << "Masukkan NIM: ";
    getline(cin, nimUser);

    while (kesempatan > 0) {
        cout << "\nLogin - Masukkan Nama: ";
        getline(cin, namaInput);
        cout << "Login - Masukkan NIM: ";
        getline(cin, nimInput);

        if (namaInput == namaUser && nimInput == nimUser) {
            cout << "\nLogin berhasil!" << endl;
            break;
        } else {
            kesempatan--;
            cout << "Login gagal! Sisa kesempatan: " << kesempatan << "\n";
        }

        if (kesempatan == 0) {
            cout << "\nAnda telah gagal login 3 kali. Program berhenti.\n";
            return 0;
        }
    }
}
```

Gambar 3.1 Program login sederhana

B. ini adalah menu pilihan

```
int pilihan;
do {
    cout << "\n=== Menu Pembelian Motor ===\n";
    cout << "1. Tambah Pembelian\n";
    cout << "2. Lihat Data Pembelian\n";
    cout << "3. Edit Data Pembelian\n";
    cout << "4. Hapus Data Pembelian\n";
    cout << "5. Keluar\n";
    cout << "Pilih menu: ";
    cin >> pilihan;

    switch (pilihan) {
        case 1: {
            PembelianMotor pembelian;
            cout << "\nMasukkan Nama Pembeli: ";
            cin >> pembelian.namaPembeli;
            cout << "Masukkan Merek Motor: ";
            cin >> pembelian.merekMotor;
            cout << "Masukkan Tipe Motor: ";
            cin >> pembelian.tipeMotor;
            cout << "Masukkan Harga Motor: ";
            cin >> pembelian.hargaMotor;
            daftarPembelian.push_back(pembelian);
            cout << "\nData berhasil ditambahkan!\n";
            break;
        }
    }
}
```

Gambar 3.2 Menu pilihan

C. Data pembelian bertable

```
case 2: {
    if (daftarPembelian.empty()) {
        cout << "\nTidak ada data pembelian.\n";
    } else {
        cout << "\n=====\\n";
        cout << "| No | Nama Pembeli | Merek Motor | Tipe Motor | Harga |\\n";
        cout << "=====\\n";
        for (size_t i = 0; i < daftarPembelian.size(); ++i) {
            cout << "| " << setw(2) << i + 1 << " | "
                << setw(13) << daftarPembelian[i].namaPembeli << " | "
                << setw(12) << daftarPembelian[i].merekMotor << " | "
                << setw(10) << daftarPembelian[i].tipeMotor << " | "
                << setw(6) << daftarPembelian[i].hargaMotor << " |\\n";
        }
        cout << "=====\\n";
    }
    break;
}
```

3.3 Gambar data pembelian

D. Fitur ini menangani edit data pembelian motor dalam sebuah menu.

```
case 3: {
    if (daftarPembelian.empty()) {
        cout << "\nTidak ada data pembelian untuk diedit.\n";
        break;
    }
    cout << "\nMasukkan nomor data yang ingin diedit: ";
    int index;
    cin >> index;
    if (index > 0 && index <= daftarPembelian.size()) {
        cout << "Masukkan Nama Pembeli baru: ";
        cin >> daftarPembelian[index - 1].namaPembeli;
        cout << "Masukkan Merek Motor baru: ";
        cin >> daftarPembelian[index - 1].merekMotor;
        cout << "Masukkan Tipe Motor baru: ";
        cin >> daftarPembelian[index - 1].tipeMotor;
        cout << "Masukkan Harga Motor baru: ";
        cin >> daftarPembelian[index - 1].hargaMotor;
        cout << "\nData berhasil diubah!\\n";
    } else {
        cout << "Nomor tidak valid!\\n";
    }
    break;
}
```

Gambar 3.4 Kode fitur edit data

E. Fitur ini menangani penghapusan data pembelian motor di dalam sebuah menu

```
case 4: {
    if (daftarPembelian.empty()) {
        cout << "\nTidak ada data pembelian untuk dihapus.\n";
        break;
    }
    cout << "\nMasukkan nomor data yang ingin dihapus: ";
    int index;
    cin >> index;
    if (index > 0 && index <= daftarPembelian.size()) {
        daftarPembelian.erase(daftarPembelian.begin() + index - 1);
        cout << "\nData berhasil dihapus!\n";
    } else {
        cout << "Nomor tidak valid!\n";
    }
    break;
}
case 5:
    cout << "\nTerima kasih telah menggunakan program ini!\n";
    break;
default:
    cout << "Pilihan tidak valid!\n";
    break;
}
} while (pilihan != 5);

return 0;
}
```

Gambar 3.5 program penghapus data

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1 = Yang pertama masuk dulu ke sistem pembelian lalu masukkan nama Muhammad Fizriannur dan nim 2409106105, lalu login masukkan lagi nama dan nim sama seperti nama yang disistem pembelian, maka akan berhasil dan masuk ke menu pembelian.

Skenario 2 = Yang pertama masuk dulu ke sistem pembeliannya lalu masukkan nama Muhammad Fizriannur dan nim 2409106105, lalu login memasukkan nama dan nim jika tidak sama seperti namadisistem pembelian maka akan gagal, dan batas kesempatannya cuman 3 kali jika salah program akan terhenti.

4.2 Hasil Output

Jika login berhasil

```
=== Login Sistem Pembelian Motor ===
Masukkan Nama: Muhammad Fizriannur
Masukkan NIM: 2409106105

Login - Masukkan Nama: Muhammad Fizriannur
Login - Masukkan NIM: 2409106105

Login berhasil!
```

Gambar 4.1 Berhasil

Jika login tidak berhasil

```
=== Login Sistem Pembelian Motor ===
Masukkan Nama: Muhammad Fizriannur
Masukkan NIM: 2409106105

Login - Masukkan Nama: fizri
Login - Masukkan NIM: 2409106105
Login gagal! Sisa kesempatan: 2

Login - Masukkan Nama: Muhammad
Login - Masukkan NIM: 2409106105
Login gagal! Sisa kesempatan: 1

Login - Masukkan Nama: Muhammad Fizriannur
Login - Masukkan NIM: 240909106
Login gagal! Sisa kesempatan: 0

Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 4. 2 Program Gagal

5. Uji Coba Dan Hasil Output

5.1 Uji Coba

Skenario 1 = Saya masuk ke menu pembelian motor, lalu memilih no.1 dan ada pilihan 1 -5. Jika saya memilih no.1 maka masukkan nam pembeli : muhammad riski, merek motor : nmax, harga motor : 30000000.

Skenario 2 = Tambahkan lagi menu pembelian motor, masukkan nama pembeli : udin, merek motor : vario 150, harga motor : 14000000.

Skenario 3 = Tmbahkan lagi menu pembelian motor, masukkan nama : william, merek motor : mio 125, harga motor : 200000.

Skenario 4 = Pilih no.2, lalu ada table data – data yang tadi kita masukkan.

Skenario 5 = Saya memilih no.3 karena ada data yang salah, masukkan no yang mau di edit : 1, lalu masukkan nama baru : riski, merek motor : nmax 155, harga motor : 3000000, lalu data berhasil di ubah.

Skenario 6 = Saya memilih no.4 karena ingin menghapus data yang ada, lalu no data yang ingin di hapus : 2,lalu data berhasil di hapus.

Skenario 7 = Saya ingin melihat data yang tersisa dengan memilih no.2, dan sisa riski dan william saja yang ada.

Skenario 8 = Dan jika memilih no.5 maka program akan keluar dan menghasilkan output “terima kasih telah menggunakan program ini!”.

5.2 Hasil Output

Skenario 1.

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 1

Masukkan Nama Pembeli: muhammad riski
Masukkan Merek Motor: Masukkan Tipe Motor: nmax
Masukkan Harga Motor: 30000000

Data berhasil ditambahkan!
```

Gambar 5.2 Menu Pembelian

Skenario 2

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 1

Masukkan Nama Pembeli: udin
Masukkan Merek Motor: vario 150
Masukkan Tipe Motor: Masukkan Harga Motor: 14000000

Data berhasil ditambahkan!
```

Gambar 5.3 Tambah pembelian

Skenario 3

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 1

Masukkan Nama Pembeli: william
Masukkan Merek Motor: mio 125
Masukkan Tipe Motor: Masukkan Harga Motor: 20000000

Data berhasil ditambahkan!
```

Gambar 5.4 Tambah pembelian

Skenario 4

```
=====
| No | Nama Pembeli | Merek Motor | Tipe Motor | Harga |
=====
| 1 | muhammad | riski | nmax | 30000000 |
| 2 | udin | vario | 150 | 14000000 |
| 3 | william | mio | 125 | 2000000 |
=====
```

Gambar 5.5 Taable pembelian

Skenario 5

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 3

Masukkan nomor data yang ingin diedit: 1
Masukkan Nama Pembeli baru: riski
Masukkan Merek Motor baru: nmax 155
Masukkan Tipe Motor baru: Masukkan Harga Motor baru: 30000000

Data berhasil diubah!
```

Gambar 5.6 Edit data pembelian

Skenario 6

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 4

Masukkan nomor data yang ingin dihapus: 2

Data berhasil dihapus!
```

Gambar 5.7 Hapus Data

Skenario 7

```
=== Menu Pembelian Motor ===
1. Tambah Pembelian
2. Lihat Data Pembelian
3. Edit Data Pembelian
4. Hapus Data Pembelian
5. Keluar
Pilih menu: 2

=====
| No | Nama Pembeli | Merek Motor | Tipe Motor | Harga |
=====
| 1 | riski | nmax | 155 | 30000000 |
| 2 | william | mio | 125 | 20000000 |
=====
```

Gambar 5.8 table

Skenario 8

```
=== Menu Pembelian Motor ===  
1. Tambah Pembelian  
2. Lihat Data Pembelian  
3. Edit Data Pembelian  
4. Hapus Data Pembelian  
5. Keluar  
Pilih menu: 5  
  
Terima kasih telah menggunakan program ini!
```

Gambar 5.9 Keluar

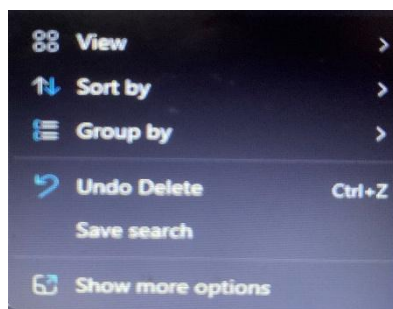
6. Sistem GitHub

1. Membuat folder



Gambar 6.1 Folder

2. Klik kanan di mouse dan klic show more options dan klik open git bash here.



Gambar 6.2 git

3. Langkah – langkah Git

1. Pertama klik kanan pada folder lalu pilih show open witch lalu pilih open git bash
2. Kedua ketika sudah di terminal ketik git config –global user.email”sesuaikan dengan email yang ada di akun github”
3. Ketiga git init untuk menginisiasi repository git.
4. Keempat ketik git add untuk menambah file yang akan di komit dan git comit –m “Update” untuk membuat check point
5. Kelima ketik git remove untuk menghubungkan repository yang ada dilokal computer dengan cara mengcopy link yang ada di github.

```
MINGW64:/c/Github/Praktikum-Apl

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl
$ git config --global user.email "fizri907@gmail.com"

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl
$ git init
Initialized empty Git repository in C:/Github/Praktikum-Apl/.git/

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (master)
$ git add .

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (master)
$ git commit -m "update fizri"
[master (root-commit) 6211ddf] update fizri
1 file changed, 167 insertions(+)
create mode 100644 Post-test/Post-test-1/2409106105-Muhammad Fizriannur-PT-1.cp
p

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (master)
$ git branch -M main

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (main)
$ git remote add origin https://github.com/fizri907/Praktikum-Apl.git

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.43 KiB | 1.43 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/fizri907/Praktikum-Apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

fizri@LAPTOP-2D98KBHL MINGW64 /c/Github/Praktikum-Apl (main)
$ |
```

Gambar 6.1 Langkah – Langkah Git