

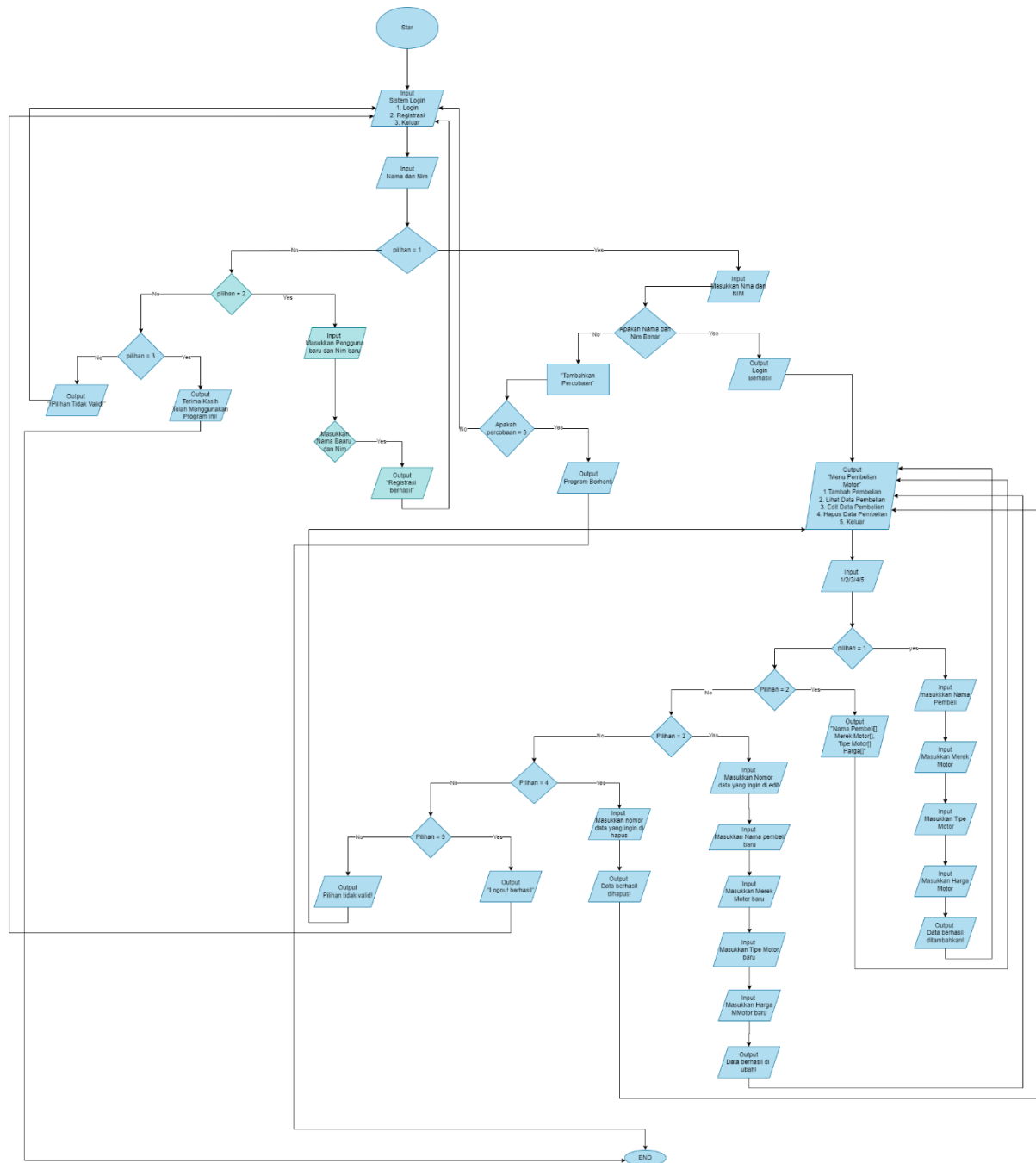
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Muhammad Fizriannur
2409106105
C1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1

2. Analisis Program

Program ini bertujuan untuk mengelola sistem pembelian motor dengan fitur login dan registrasi pengguna, serta memungkinkan pengguna untuk mencatat, melihat, mengedit, dan menghapus data pembelian motor.

Fungsi/Manfaat Utama:

1. Registrasi dan Login Pengguna:

- a. Memungkinkan pengguna baru untuk mendaftar.
- b. Mengetahui pengguna dengan nama dan NIM sebelum mengakses sistem.

2. Manajemen Data Pembelian Motor:

- a. Menambah Pembelian: Pengguna dapat mencatat pembelian motor dengan nama pembeli, merek, tipe, dan harga.
- b. Melihat Data Pembelian: Menampilkan daftar pembelian yang telah dilakukan.
- c. Mengedit Data Pembelian: Memungkinkan pengguna memperbarui informasi pembelian.
- d. Menghapus Data Pembelian: Memberikan opsi untuk menghapus data pembelian yang tidak diperlukan.

3. Keamanan Akses:

- a. Batasan login dengan maksimal 3 kali percobaan untuk mencegah akses tidak sah.

3. Source Code

1. Memvalidasi login pengguna berdasarkan nama dan Nim agar hanya pengguna yang sudah terdaftar yang bisa mengakses sistem.

```
struct SistemPembelian {
    user;
    PembelianMotor daftarPembelian[100];
    int jumlahPembelian = 0;
};

SistemPembelian sistem[10];
int jumlahUser = 0;

int login(string nama, string nim) {
    int i = 0;
    while (i < jumlahUser) {
        if (sistem[i].user.nama == nama && sistem[i].user.nim == nim) {
            return i;
        }
        i++;
    }
    return -1;
}
```

Gambar 3.1

2. Sistem ini untuk pengguna baru mendaftar ke sistem selama kapasitas belum penuh, dan kapasitas ini sampai 10, sehingga mereka dapat menggunakan fitur dalam sistem pembelian motor.

```
int registrasi() {
    if (jumlahUser >= 10) {
        cout << "Kapasitas user penuh!\n";
        return 0;
    }
    cout << "Masukkan Nama: ";
    cin >> sistem[jumlahUser].user.nama;
    cout << "Masukkan NIM: ";
    cin >> sistem[jumlahUser].user.nim;
    jumlahUser++;
    cout << "Registrasi berhasil!\n";
    return 1;
}
```

Gambar 3.2

3. Ini fungsi untuk mengelola menu utama sistem pembelian motor, memproses login dengan batas 3 kali percobaan, dan jika gagal 3 kali program bakal terhenti.

```
int main() {
    int pilihan = 0;
    while (pilihan != 3) {
        cout << "\n=== Sistem Pembelian Motor ===\n";
        cout << "1. Login\n";
        cout << "2. Registrasi\n";
        cout << "3. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;

        if (pilihan == 1) {
            string nama, nim;
            int kesempatan = 3;
            int userIndex = -1;

            while (kesempatan > 0) {
                cout << "Masukkan Nama: ";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;

                userIndex = login(nama, nim);
                if (userIndex != -1) {
                    cout << "\nLogin berhasil!\n";
                    break;
                } else {
                    kesempatan--;
                    cout << "\nLogin gagal! Sisa kesempatan: " << kesempatan << "\n";
                }
            }

            if (userIndex == -1) {
                cout << "\nAnda telah gagal login 3 kali. Program berhenti.\n";
                return 0;
            }
        }
    }
}
```

Gambar 3.3

4. Struct data ini gunanya untuk menyimpan info-info data dari pengguna di sistem pembelian motor.

```
struct User {
    string nama;
    string nim;
};
```

Gambar 3.4

5. Untuk menambahkan data pembelian motor ke dalam sistem, yang nantinya dapat dilihat, diedit, atau dihapus sesuai kebutuhan.

```
if (pilihanMenu == 1) {
    cout << "\nMasukkan Nama Pembeli: ";
    cin >> sistem[userIndex].daftarPembelian[sistem[userIndex].jumlahPembelian].namaPembeli;
    cout << "Masukkan Merek Motor: ";
    cin >> sistem[userIndex].daftarPembelian[sistem[userIndex].jumlahPembelian].merekMotor;
    cout << "Masukkan Tipe Motor: ";
    cin >> sistem[userIndex].daftarPembelian[sistem[userIndex].jumlahPembelian].tipeMotor;
    cout << "Masukkan Harga Motor: ";
    cin >> sistem[userIndex].daftarPembelian[sistem[userIndex].jumlahPembelian].hargaMotor;
    sistem[userIndex].jumlahPembelian++;
    cout << "\nData berhasil ditambahkan!\n";
}
```

Gambar 3.5

6. Untuk melihat daftar pembelian motor yang sudah di tambahkan dalam sistem.

```
} else if (pilihanMenu == 2) {
    if (sistem[userIndex].jumlahPembelian == 0) {
        cout << "\nTidak ada data pembelian.\n";
    } else {
        cout << "\n=== Data Pembelian ===\n";
        for (int i = 0; i < sistem[userIndex].jumlahPembelian; i++) {
            cout << i + 1 << ". Nama Pembeli: " << sistem[userIndex].daftarPembelian[i].namaPembeli
                << " | Merek: " << sistem[userIndex].daftarPembelian[i].merekMotor
                << " | Tipe: " << sistem[userIndex].daftarPembelian[i].tipeMotor
                << " | Harga: " << sistem[userIndex].daftarPembelian[i].hargaMotor << endl;
        }
    }
}
```

Gambar 3.6

7. Ini fitur untuk mengedit data pembelian motor di sistem.

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

Gambar 3.7

8. Ini fitur untuk menghapus data pembelian motor yang ada di sistem.

```
} else if (pilihanMenu == 4) {  
    if (sistem[userIndex].jumlahPembelian == 0) {  
        cout << "\nTidak ada data pembelian untuk dihapus.\n";  
    } else {  
        cout << "\nMasukkan nomor data yang ingin dihapus: ";  
        int index;  
        cin >> index;  
        if (index > 0 && index <= sistem[userIndex].jumlahPembelian) {  
            for (int i = index - 1; i < sistem[userIndex].jumlahPembelian - 1; i++) {  
                sistem[userIndex].daftarPembelian[i] = sistem[userIndex].daftarPembelian[i + 1];  
            }  
            sistem[userIndex].jumlahPembelian--;  
            cout << "\nData berhasil dihapus!\n";  
        } else {  
            cout << "Nomor tidak valid!\n";  
        }  
    }  
}
```

Gambar 3. 8

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Uji coba login:

1. Langkah pertama saya registrasi dulu, masukkan nama dan nim.
 - a. Nama: Fizri
 - b. Nim: 2409106105
2. Lalu masuk ke menu login dengan memasukkan nama dan nim di atas
3. Jika salah masukkan nama atau nim yang beda dari data yang sudah di registrasi, ke login maka akan seperti;
 - a. Nama: fizri
 - b. Nim: 2323
Login gagal!sisa kesempatan 2
 - a. Nama: Zri
 - b. Nim: 2409106105
Login gagal!sisa kesempatan 1
 - a. Nama: sisa
 - b. Nim: 2323
Anda telah gagal login 3 kali. Program berhenti.
4. Masuk ke Menu Pembelian Motor, dan pilih no.1 seperti:
 - a. Masukkan Nama Pembeli: riski
 - b. Masukkan Merek Motor: nmax
 - c. Masukkan Tipe Motor: 160
 - d. Masukkan Harga Motor: 400000
5. Jika memilih no.2 maka data yang tadi bakal muncul di sini.
6. Jika memilih no.3 maka bakal muncul;
 - a. Masukkan nomor data yang ingin diedit: 1
 - b. Masukkan Nama Pembeli baru: iwan
 - c. Masukkan Merek Motor baru: vario
 - d. Masukkan Tipe Motor baru: 150
 - e. Masukkan Harga Motor baru: 200000
7. Jika memilih no.4 maka data yang tadi akan hilang.
8. Dan kita liat apakah masih ada data setelah kita hapus.
9. Jika memilih no.5 maka sistem akan kembali ke sistem pembeli motor.

4.2 Hasil Output

1. Hasil nomor 1

```
=== Sistem Pembelian Motor ===  
1. Login  
2. Registrasi  
3. Keluar  
Pilih menu: 2  
Masukkan Nama: Fizri  
Masukkan NIM: 2409106105  
Registrasi berhasil!
```

Gambar 4.1

2. Hasil nomor 2

```
=== Sistem Pembelian Motor ===  
1. Login  
2. Registrasi  
3. Keluar  
Pilih menu: 1  
Masukkan Nama: fizri  
Masukkan NIM: 2323  
  
Login gagal! Sisa kesempatan: 2
```

Gambar 4.2

```
Masukkan Nama: Zri  
Masukkan NIM: 2409106105  
  
Login gagal! Sisa kesempatan: 1
```

Gambar 4.3

```
Masukkan Nama: sisa  
Masukkan NIM: 2323  
  
Login gagal! Sisa kesempatan: 0  
  
Anda telah gagal login 3 kali. Program berhenti.  
PS C:\Github\Praktikum-Apl\Post-test\post-test-3> █
```

Gambar 4.4

3. Hasil nomor 3

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

Masukkan Nama Pembeli: riski
Masukkan Merek Motor: nmax
Masukkan Tipe Motor: 160
Masukkan Harga Motor: 400000

Data berhasil ditambahkan!
```

Gambar 4.5

4. Hasil nomor 4

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

=== Data Pembelian ===
1. Nama Pembeli: riski | Merek: nmax | Tipe: 160 | Harga: 400000
```

Gambar 4.6

5. Hasil nomor 5

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

Masukkan nomor data yang ingin diedit: 1
Masukkan Nama Pembeli baru: iwan
Masukkan Merek Motor baru: vario
Masukkan Tipe Motor baru: 150
Masukkan Harga Motor baru: 200000

Data berhasil diubah!
```

Gambar 4.7

6. Hasil nomor 6

```
=== Menu Pembelian Motor ===  
1. Tambah Pembelian  
2. Lihat Data Pembelian  
3. Edit Data Pembelian  
4. Hapus Data Pembelian  
5. Logout  
Pilih menu: 4  
  
Masukkan nomor data yang ingin dihapus: 1  
  
Data berhasil dihapus!
```

Gambar 4.8

7. Hasil nomor 7

```
=== Menu Pembelian Motor ===  
1. Tambah Pembelian  
2. Lihat Data Pembelian  
3. Edit Data Pembelian  
4. Hapus Data Pembelian  
5. Logout  
Pilih menu: 2  
  
Tidak ada data pembelian.
```

Gambar 4.8

8. Hasil nomor 8

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

Gambar 4.9

5. Langkah-Langkah Git pada VSCode

Langkah-Langkah Git

Langkah 1: Download Git nya dlu di website resminya Git.

Langkah 2: Kalian installasi Git nya.

Langkah 3: Kalian buka Git nya dan kalian ketik ini “git config –global user.email “ini email kalian yah yang terhubung di GitHub”” trus enter.

Langkah 4: ketik ini Git add untuk menambah file yang akan di komit dan Git comit –m “Update” untuk membuat check point.

Langkah 5: Terus kalian buka GitHub biasanya kalau sudah manambahkan file atau folder yang ada di langkah 4 itu jika sudah terhubung biasanya ada bacaanya tuh jika sudah nemu yang bacaan “git remote add origin gitu lah” seperti ini.

Langkah 6: terus ketik “git push -u origin main”

```

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)
$ |

```