

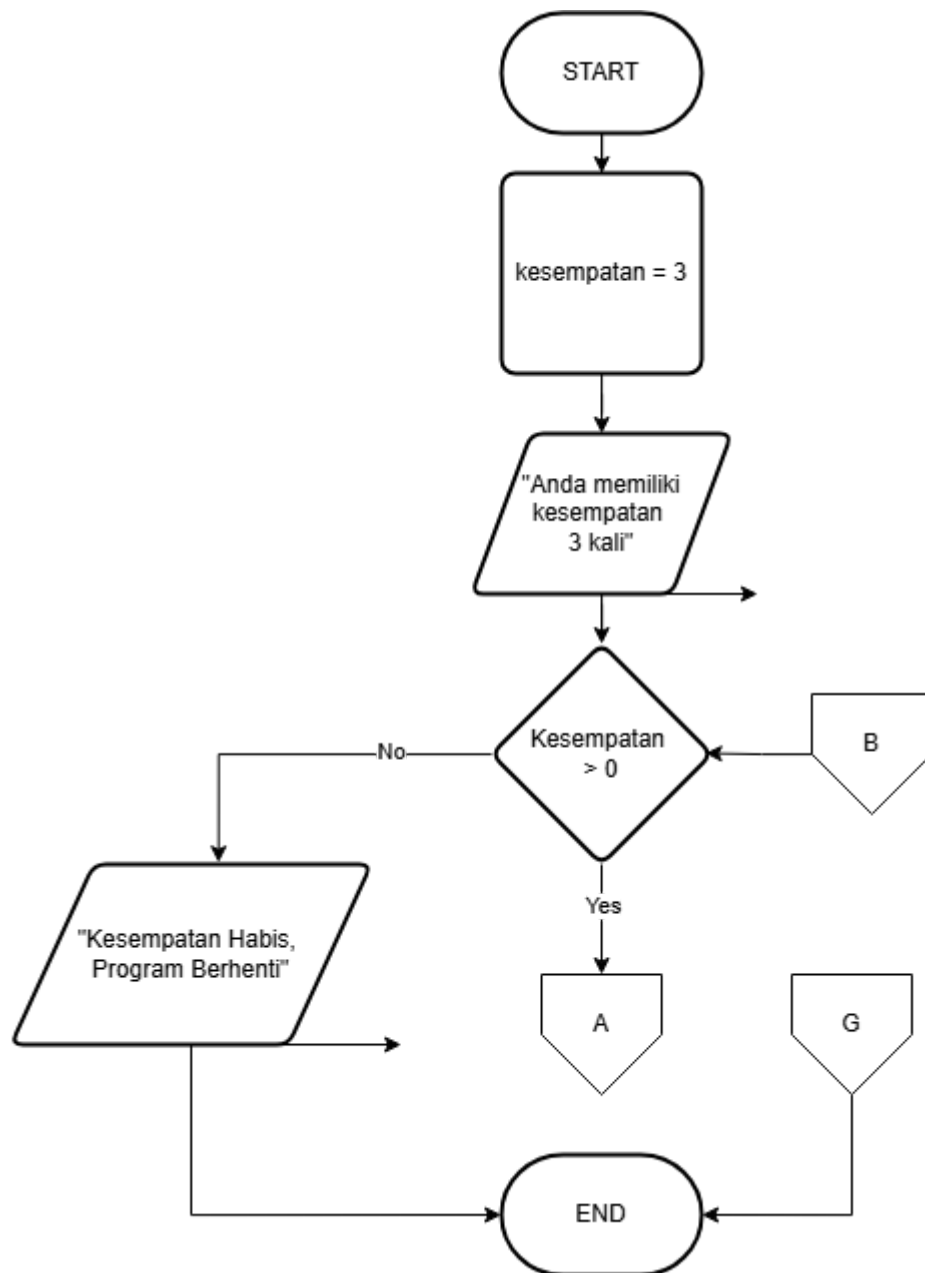
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



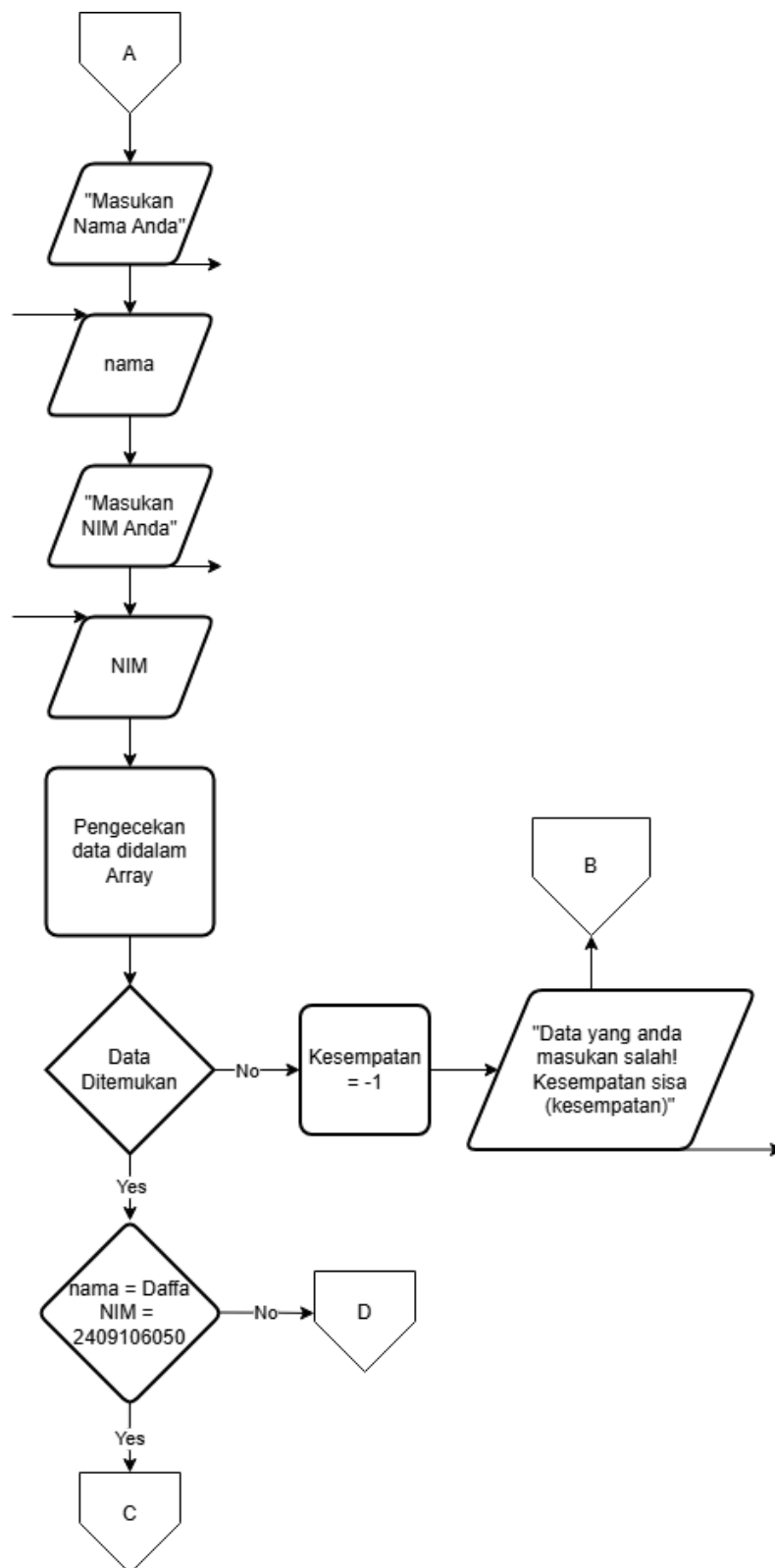
Disusun oleh:
Ananda Daffa Harahap (2409106050)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

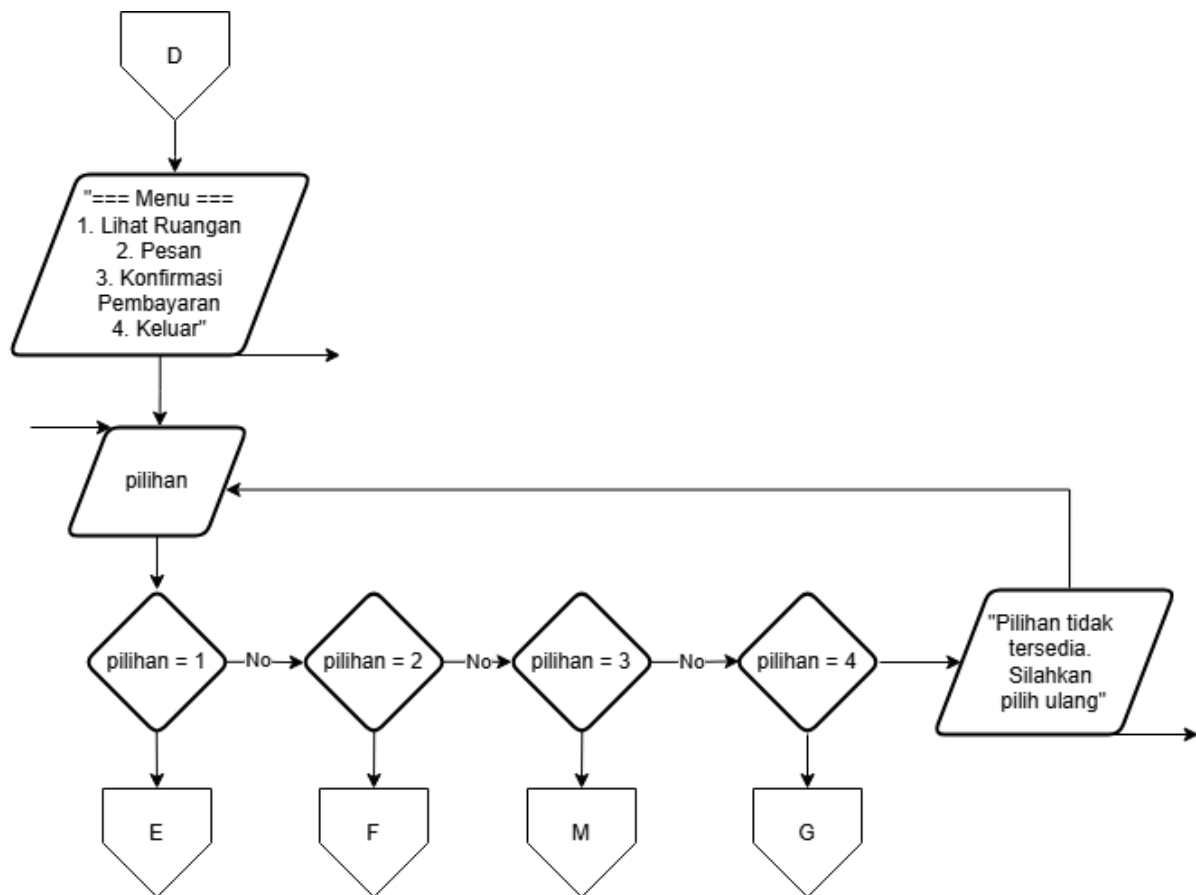
1. Flowchart



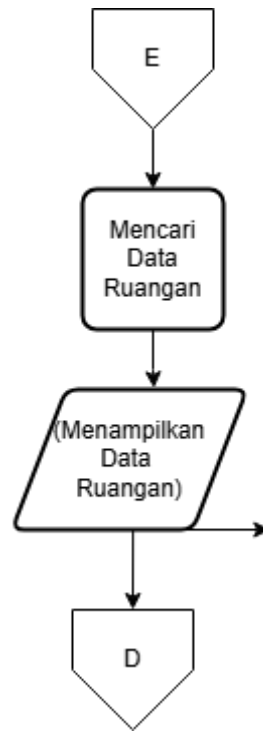
Gambar 1.1 *Flowchart* Bagian 1



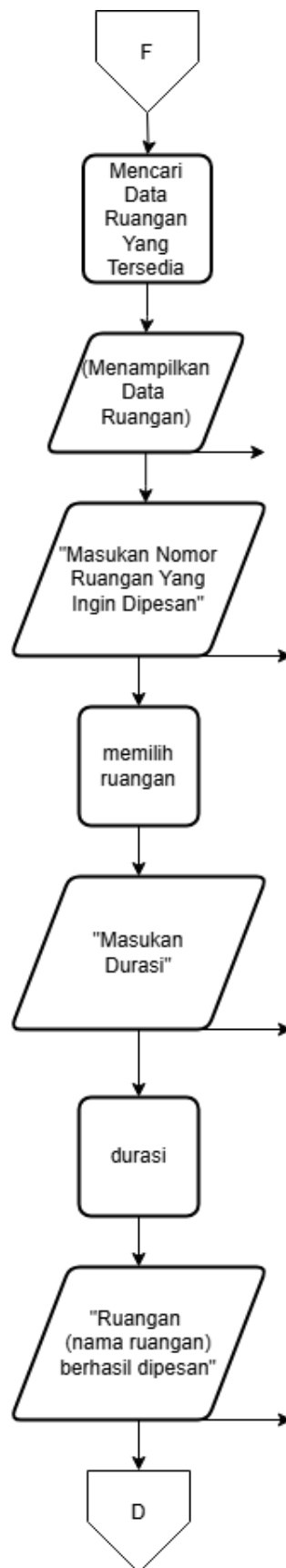
Gambar 1.2 *Flowchart* Bagian 2



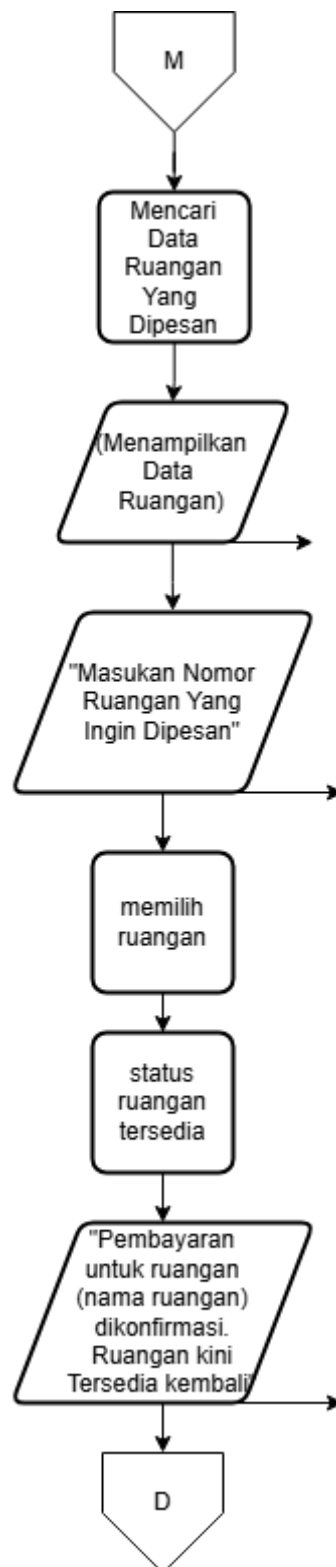
Gambar 1.3 *Flowchart* Bagian 3



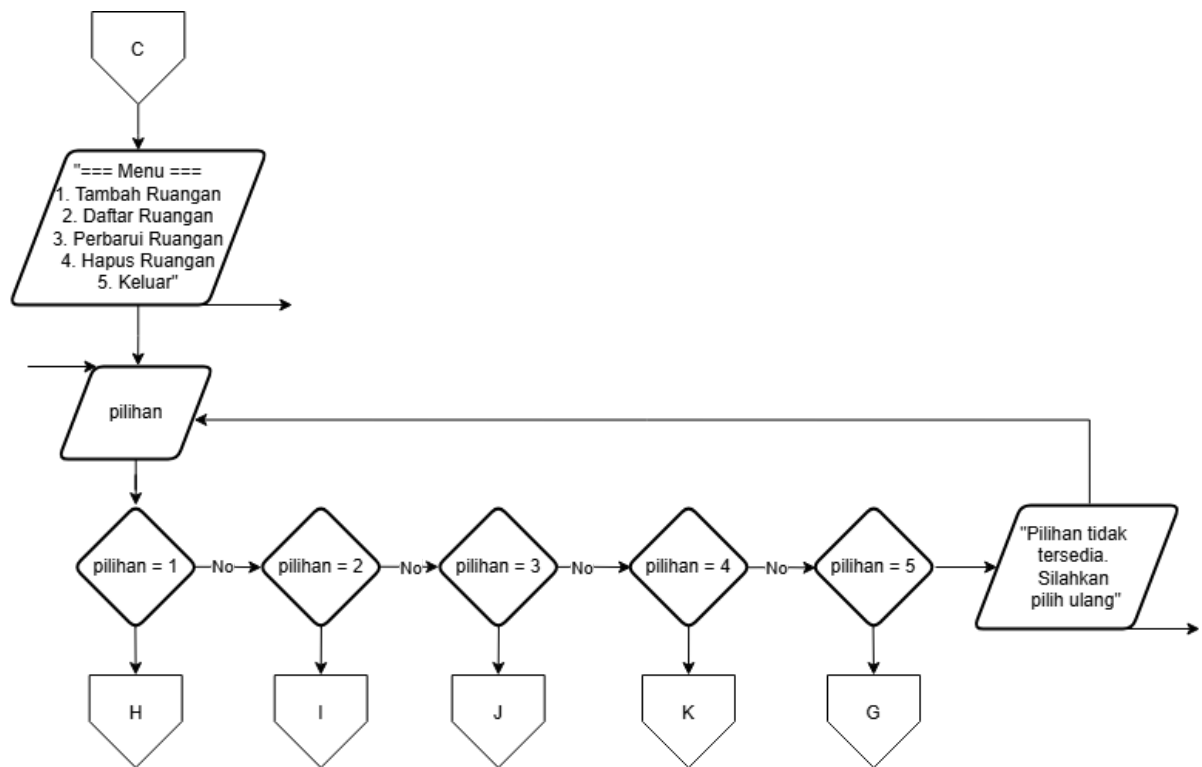
Gambar 1.4 *Flowchart* Bagian 4



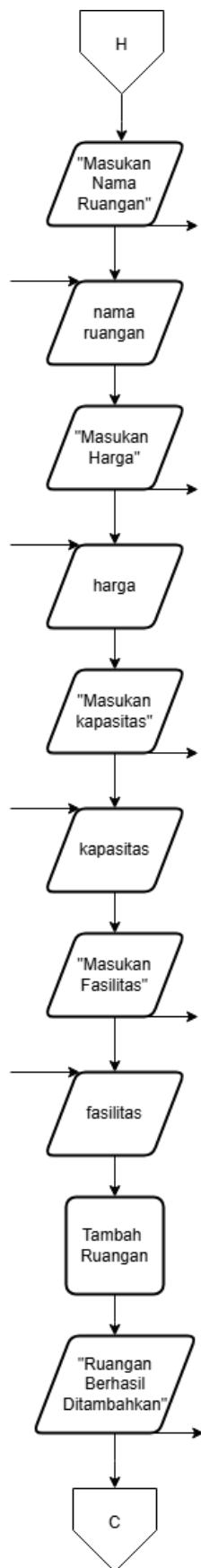
Gambar 1.5 *Flowchart* Bagian 5



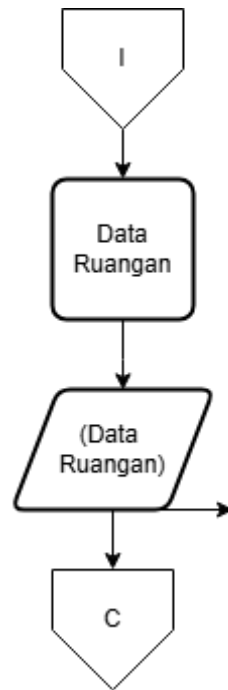
Gambar 1.6 *Flowchart* Bagian 6



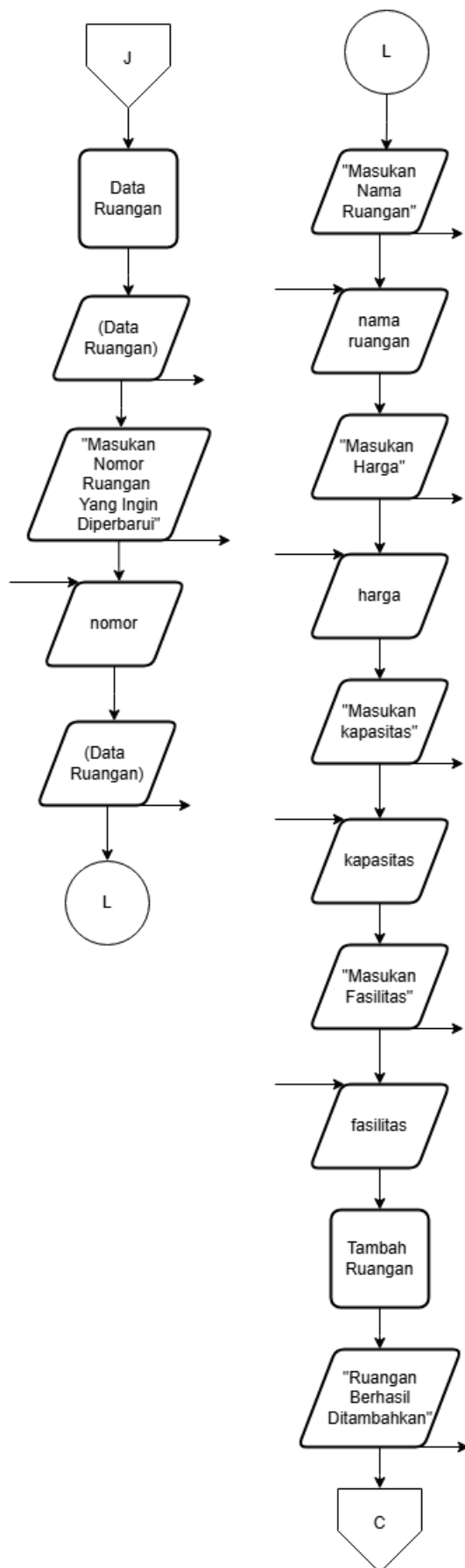
Gambar 1.7 Flowchart Bagian 7



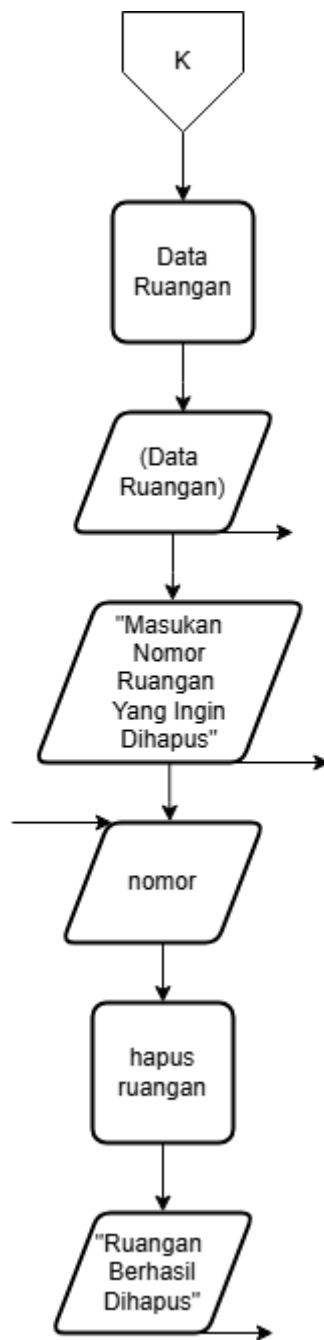
Gambar 1.8 *Flowchart* Bagian 8



Gambar 1.9 *Flowchart* Bagian 9



Gambar 1.10 *Flowchart* Bagian 10



Gambar 1.11 *Flowchart* Bagian 11

2. Analisis Program

2.1 Deskripsi Singkat Program

.Program ini adalah sistem manajemen penyewaan ruangan meeting. Program memungkinkan pengguna untuk melihat daftar ruangan yang tersedia, memesan ruangan, serta mengelola status ruangan yang telah dipesan. Selain itu, terdapat fitur untuk menampilkan daftar ruangan yang telah dipesan dan mengubah statusnya menjadi tersedia kembali setelah pembayaran dikonfirmasi.

2.2 Penjelasan Alur & Algoritma

1. Program pertama-tama akan meminta pengguna untuk memasukan nama dan password, apabila yang dimasukan adalah nama Daffa dan password NIM maka akan masuk sebagai admin, tetapi jika namanya daffa maka akan masuk sebagai kasir.
2. Jika pengguna salah memasukan NIM program akan memunculkan pesan yang memberitahukan pengguna bahwa NIM yang mereka masukan salah dan kesempatan mereka berkurang 1 setelah itu program akan langsung selesai jika pengguna salah memasukan NIM sebanyak 3 kali. Dan jika pengguna berhasil memasukan NIM yang benar maka program akan masuk ke bagian menu sesuai masuk sebagai admin atau kasir.
3. Saat login sebagai kasir, program akan menampilkan 4 pilihan yaitu lihat ruangan, pesan, konfirmasi pembayaran, dan keluar. Kasir akan diminta untuk memasukkan sesuai kebutuhan, apabila lebih dari 1-4 maka program akan menampilkan 4 pilihan kembali.
4. apabila kasir memilih 1 yaitu lihat ruangan maka program akan mencari data ruangan lalu akan ditampilkan.
5. Apabila memilih 2 maka program akan menampilkan list ruangan lalu akan meminta kasir memasukan nomor ruangan yang ingin dipesan, program secara otomatis akan memilih ruangan yang dimaksud. Program akan meminta durasi setelah selesai maka program akan menampilkan pesan sukses.
6. Apabila memilih 3, maka program akan menampilkan list ruangan yang dipesan atau tidak tersedia lalu program akan meminta kita memasukan nomor ruangan lalu program akan mengembalikan status ruangan ke “Tersedia” dan kemudian memunculkan pesan berhasil.

7. Sedangkan apabila login sebagai admin, program akan menampilkan 5 pilihan berbeda yaitu tambah ruangan, daftar ruangan, perbarui, ruangan, hapus ruangan, dan keluar apabila lebih dari 1-5 maka program akan kembali menampilkan pilihan.
8. Apabila admin memilih 1, yaitu tambah ruangan maka program akan langsung meminta admin memasukan nama ruangan, harga, kapasitas, dan fasilitas. Setelah itu program akan menambahkan ruangan dan memberikan pesan berhasil.
9. Jika admin memilih 2, Daftar ruangan maka program akan mencari dan menampilkan seluruh daftar ruangan
10. jika admin memilih 3, yaitu perbarui ruangan maka akan ditampilkan dulu ruangan yang ada, lalu akan diminta untuk memasukan nomor ruangan yang ingin diubah. Setelah itu akan diminta untuk memasukan nama ruangan, harga, kapasitas, dan fasilitas dan pesan berhasil.
11. Terakhir adalah 4 yaitu hapus ruangan, program akan menampilkan daftar ruangan dan akan meminta admin memasukan nomor ruangan yang ingin dihapus. Setelah itu program akan menghapus dan menampilkan pesan berhasil.

3. Source Code

A. Login

Pertama-tama program akan memberikan kesempatan sebanyak 3 kali kepada pengguna, lalu program akan meminta pengguna untuk melakukan login dengan cara memasukkan nama dan NIM yang adalah *password* dari program yang kita buat, apabila *password* salah maka program akan mengurangi kesempatan yang dimiliki pengguna sebanyak 1, apabila pengguna menghabiskan kesempatan yang dimilikinya maka program akan berhenti. Jika pengguna login menggunakan nama Daffa maka akan login sebagai admin, tetapi jika daffa maka akan sebagai kasir.

Source Code :

```
int main() {
    string nama, nim;
    int kesempatan = 3;
    bool loginBerhasil = false;
    bool isAdmin = false;

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

        if (nama == "Daffa" && nim == "2409106050") {
            loginBerhasil = true;
            isAdmin = true;
            break;
        } else if (nama == "daffa" && nim == "2409106050") {
            loginBerhasil = true;
            isAdmin = false;
            break;
        }
        cout << "Login gagal! Kesempatan tersisa: " << kesempatan << endl;
    }

    if (!loginBerhasil) {
        cout << "Anda telah gagal login 3 kali. Program berhenti." << endl;
        return 0;
    }
}
```

B. Menu

Disini program akan menampilkan menu yang tersedia sesuai peran yang digunakan (admin atau kasir) dan akan meminta pengguna untuk memilih menu yang ingin digunakan.

Source Code :

```
do {  
    if (isAdmin) {  
        cout << "\nMenu Admin:\n";  
        cout << "1. Tambah Ruangan\n";  
        cout << "2. Lihat Ruangan\n";  
        cout << "3. Perbarui Ruangan\n";  
        cout << "4. Hapus Ruangan\n";  
        cout << "5. Keluar\n";  
    } else {  
        cout << "\nMenu Kasir:\n";  
        cout << "1. Lihat Ruangan\n";  
        cout << "2. Pesan Ruangan\n";  
        cout << "3. Konfirmasi Pembayaran\n";  
        cout << "4. Keluar\n";  
    }  
    cout << "Pilih menu: ";  
    cin >> pilihan;  
    cin.ignore();  
}
```


C. Admin (Pilihan 1)

Jika admin memilih 1 maka program akan menjalankan perintah untuk menambahkan ruangan sesuai yang diinginkan oleh admin.

Source Code :

```
if (isAdmin) {
    if (pilihan == 1) {
        if (jumlahRuangan < MAX_RUANGAN) {
            cout << "Masukkan nama ruangan : ";
            getline(cin, daftarRuangan[jumlahRuangan].nama);
            cout << "Masukkan harga : ";
            cin >> daftarRuangan[jumlahRuangan].harga;
            cout << "Masukkan kapasitas ruangan : ";
            cin >> daftarRuangan[jumlahRuangan].kapasitas;
            cin.ignore();
            cout << "Masukkan fasilitas : ";
            getline(cin, daftarRuangan[jumlahRuangan].fasilitas);
            daftarRuangan[jumlahRuangan].tersedia = true;
            jumlahRuangan++;
            cout << "Ruangan berhasil ditambahkan!\n";
        } else {
            cout << "Kapasitas ruangan penuh!\n";
        }
    }
}
```

D. Admin (Pilihan 2)

Pada bagian ini admin meminta untuk daftar ruangan sehingga program akan menampilkan data ruangan.

Source Code :

```
else if (pilihan == 2) {
    cout << "\nDaftar Ruangan:\n";
    cout <<
    "-----\n";
    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
    setw(15) << "Status"
        << setw(10) << "Harga" << setw(10) << "Kapasitas" <<
    setw(10) << "Durasi"
        << "Fasilitas" << endl;
    for (int i = 0; i < jumlahRuangan; i++) {
        cout << left << setw(10) << (i + 1) << setw(15) <<
    daftarRuangan[i].nama
        << setw(15) << (daftarRuangan[i].tersedia ?
    "Tersedia" : "Dipesan")
        << setw(10) << daftarRuangan[i].harga
        << setw(10) << daftarRuangan[i].kapasitas
        << setw(10) << (daftarRuangan[i].durasi > 0 ?
    to_string(daftarRuangan[i].durasi) + " jam" : "-")
        << daftarRuangan[i].fasilitas << endl;
    }
    cout <<
    "-----\n";
}
```

E. Admin (Pilihan 3)

pada bagian ini admin ingin memperbarui ruangan sehingga akan muncul daftar ruangan yang ada beserta data-data seperti nama, kapasitas, dll. Setelah muncul maka program akan meminta admin ruangan manakah yang ingin diperbarui, setelah admin memasukan ruangan yang ingin diperbarui, program lanjut meminta user untuk memasukan nama, harga, kapasitas, dan fasilitas

Source Code :

```
else if (pilihan == 3) {

    cout << "\nDaftar Ruangan:\n";
    cout <<
    "-----\n";
    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
    setw(15) << "Status"
        << setw(10) << "Harga" << setw(10) << "Kapasitas" <<
    setw(10) << "Durasi"
        << "Fasilitas" << endl;
    for (int i = 0; i < jumlahRuangan; i++) {
        cout << left << setw(10) << (i + 1) << setw(15) <<
    daftarRuangan[i].nama
        << setw(15) << (daftarRuangan[i].tersedia ?
    "Tersedia" : "Dipesan")
        << setw(10) << daftarRuangan[i].harga
        << setw(10) << daftarRuangan[i].kapasitas
        << setw(10) << (daftarRuangan[i].durasi > 0 ?
    to_string(daftarRuangan[i].durasi) + " jam" : "-")
        << daftarRuangan[i].fasilitas << endl;
    }
    cout <<
    "-----\n";

    cout << "Masukkan nomor ruangan yang ingin diperbarui : ";
    int pilih;
    cin >> pilih;
    cin.ignore();
    if (pilih >= 1 && pilih <= jumlahRuangan) {
        cout << "Masukkan nama : ";
        getline(cin, daftarRuangan[pilih - 1].nama);
```

```

        cout << "Masukkan harga : ";
        cin >> daftarRuangan[pilih - 1].harga;
        cout << "Masukkan kapasitas : ";
        cin >> daftarRuangan[pilih - 1].kapasitas;
        cin.ignore();
        cout << "Masukkan fasilitas : ";
        getline(cin, daftarRuangan[pilih - 1].fasilitas);
        cout << "Ruangan berhasil diperbarui!\n";
    } else {
        cout << "Nomor ruangan tidak valid!\n";
    }
}

```

F. Admin (Pilihan 4)

Pada bagian ini admin ingin menghapus ruangan sehingga program akan memunculkan data ruangan dan meminta admin memasukan nomor ruangan yang ingin dihapus.

Source Code :

```

else if (pilihan == 4) {

    cout << "\nDaftar Ruangan:\n";
    cout <<
    "-----\n";
    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
    setw(15) << "Status"
        << setw(10) << "Harga" << setw(10) << "Kapasitas" <<
    setw(10) << "Durasi"
        << "Fasilitas" << endl;
    for (int i = 0; i < jumlahRuangan; i++) {
        cout << left << setw(10) << (i + 1) << setw(15) <<
    daftarRuangan[i].nama
        << setw(15) << (daftarRuangan[i].tersedia ?
    "Tersedia" : "Dipesan")
        << setw(10) << daftarRuangan[i].harga
        << setw(10) << daftarRuangan[i].kapasitas
        << setw(10) << (daftarRuangan[i].durasi > 0 ?
    to_string(daftarRuangan[i].durasi) + " jam" : "-")
    }
}

```

```

        << daftarRuangan[i].fasilitas << endl;
    }
    cout <<
    "-----\n";

    cout << "Masukkan nomor ruangan yang ingin dihapus : ";
    int pilih;
    cin >> pilih;
    if (pilih >= 1 && pilih <= jumlahRuangan) {
        for (int i = pilih - 1; i < jumlahRuangan - 1; i++) {
            daftarRuangan[i] = daftarRuangan[i + 1];
        }
        jumlahRuangan--;
        cout << "Ruangan berhasil dihapus!\n";
    } else {
        cout << "Nomor ruangan tidak valid!\n";
    }
}

```

G. Kasir (Pilihan 1)

Disini kasir meminta program untuk menampilkan daftar ruangan, sehingga program akan menampilkan semua data ruangan.

Source Kode :

```

if (pilihan == 1) {
    cout << "\nDaftar Ruangan:\n";

    cout <<
    "-----\n";

    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
    setw(15) << "Status"
        << setw(10) << "Harga" << setw(10) << "Kapasitas" <<
    setw(10) << "Durasi"
        << "Fasilitas" << endl;
    for (int i = 0; i < jumlahRuangan; i++) {
        cout << left << setw(10) << (i + 1) << setw(15) <<
    daftarRuangan[i].nama
        << setw(15) << (daftarRuangan[i].tersedia ?
    "Tersedia" : "Dipesan")
        << setw(10) << daftarRuangan[i].harga

```

```

        << setw(10) << daftarRuangan[i].kapasitas
        << setw(10) << (daftarRuangan[i].durasi > 0 ?
to_string(daftarRuangan[i].durasi) + " jam" : "-")
        << daftarRuangan[i].fasilitas << endl;
    }

    cout <<
"-----\n";

}

```

H. Kasir (Pilihan 2)

Disini kasir meminta program untuk melakukan pemesanan ruangan, sehingga program akan menampilkan semua data ruangan terlebih dahulu lalu program akan meminta nomor ruangan yang ingin dipesan yang tersedia setelah itu program meminta durasi apabila berhasil akan ada pesan berhasil.

Source Kode :

```

else if (pilihan == 2) {

    cout << "\nDaftar Ruangan :\n";

    cout <<
"-----\n";

    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
setw(15) << "Status"
        << setw(10) << "Harga" << setw(10) << "Kapasitas" <<
setw(10) << "Durasi"
        << "Fasilitas" << endl;
    for (int i = 0; i < jumlahRuangan; i++) {
        cout << left << setw(10) << (i + 1) << setw(15) <<
daftarRuangan[i].nama
        << setw(15) << (daftarRuangan[i].tersedia ?
"Tersedia" : "Dipesan")
        << setw(10) << daftarRuangan[i].harga
        << setw(10) << daftarRuangan[i].kapasitas
        << setw(10) << (daftarRuangan[i].durasi > 0 ?
to_string(daftarRuangan[i].durasi) + " jam" : "-")
        << daftarRuangan[i].fasilitas << endl;
    }

    cout <<
"-----\n";
}

```

```

        cout << "Masukkan nomor ruangan yang ingin dipesan : ";
        int pilih, durasi;
        cin >> pilih;
        cout << "Masukkan durasi : ";
        cin >> durasi;

        if (pilih >= 1 && pilih <= jumlahRuangan &&
daftarRuangan[pilih - 1].tersedia) {
            daftarRuangan[pilih - 1].tersedia = false;
            daftarRuangan[pilih - 1].durasi = durasi;
            cout << "Ruangan " << daftarRuangan[pilih - 1].nama << "
berhasil dipesan.\n";
        } else {
            cout << "Ruangan tidak tersedia atau nomor salah!\n";
        }
    }
}

```

I. Kasir (Pilihan 3)

Disini kasir ingin melakukan konfirmasi pembayaran dimana pembayaran dilakukan secara offline sehingga setelah dikonfirmasi oleh kasir maka program akan mengubah status ruangan dari Dipesan menjadi Tersedia.

Source Kode :

```

else if (pilihan == 3) {
    cout << "\nDaftar Ruangan yang Dipesan :\n";

    cout <<
"-----\n";
    cout << left << setw(10) << "No" << setw(15) << "Nama" <<
setw(10) << "Harga"
        << setw(10) << "Kapasitas" << setw(10) << "Durasi" <<
"Fasilitas" << endl;

    int count = 0;
    for (int i = 0; i < jumlahRuangan; i++) {
        if (!daftarRuangan[i].tersedia) {
            cout << left << setw(10) << (i + 1) << setw(15) <<
daftarRuangan[i].nama
                << setw(10) << daftarRuangan[i].harga
                << setw(10) << daftarRuangan[i].kapasitas

```

```

        << setw(10) << daftarRuangan[i].durasi
        << daftarRuangan[i].fasilitas << endl;
        count++;
    }
}

if (count == 0) {
    cout << "Tidak ada ruangan yang dipesan.\n";
} else {
    cout <<
    "-----\n";
    cout << "Masukkan nomor ruangan yang ingin dikonfirmasi
pembayarannya : ";
    int pilih;
    cin >> pilih;

    if (pilih >= 1 && pilih <= jumlahRuangan &&
!daftarRuangan[pilih - 1].tersedia) {
        daftarRuangan[pilih - 1].tersedia = true;
        daftarRuangan[pilih - 1].durasi = 0;
        cout << "Pembayaran untuk ruangan " <<
daftarRuangan[pilih - 1].nama << " dikonfirmasi.\n";
        cout << "Ruangan kini tersedia kembali.\n";
    } else {
        cout << "Nomor ruangan tidak valid atau belum
dipesan!\n";
    }
}
}
}

```


J. Keluar

Disini pengguna baik admin maupun kasir akan keluar dari aplikasi sesuai dengan nomor di menu masing-masing.

Source Kode :

```
} while ((isAdmin && pilihan != 5) || (!isAdmin && pilihan != 4));  
  
    cout << "Terima kasih telah menggunakan layanan kami!\n";  
    return 0;
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Kasir sedang melayani pelanggan dimana pelanggan bertanya ada ruangan kosong apa saja dan melakukan pemesanan setelah selesai pelanggan melakukan pembayaran.
2. Admin melakukan pengecekan ruangan apa saja yang sedang tersedia saat ini dan memperbarui sebuah ruangan.
3. kasir sedang terburu-buru dan ia memasukkan password yang salah sebanyak 3 kali.

4.2 Hasil Output

A. Skenario Pertama

```
Masukkan Nama: daffa
Masukkan NIM: 2409106050

Menu Kasir:
1. Lihat Ruangan
2. Pesan Ruangan
3. Konfirmasi Pembayaran
4. Keluar
Pilih menu: █
```

Gambar 4.1 Login Dan Menu Kasir

```
Daftar Ruangan:
-----
No      Nama      Status      Harga      Kapasitas Durasi      Fasilitas
1       Ruang A    Tersedia    25000      4          -          Meja biasa, Proyektor, WiFi
2       Ruang B    Tersedia    30000      4          -          Meja biasa, Proyektor, AC, WiFi
3       Ruang C    Tersedia    35000      6          -          Meja diskusi, Whiteboard, AC, Wifi
4       Ruang D    Tersedia    40000      8          -          Meja Rapat, Proyektor, Mic, WiFi
5       Ruang E    Tersedia    50000      10         -          Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi
-----

Menu Kasir:
1. Lihat Ruangan
2. Pesan Ruangan
3. Konfirmasi Pembayaran
4. Keluar
Pilih menu: █
```

Gambar 4.2 Lihat Ruangan

Daftar Ruangan :

No	Nama	Status	Harga	Kapasitas	Durasi	Fasilitas
1	Ruang A	Tersedia	25000	4	-	Meja biasa, Proyektor, Wifi
2	Ruang B	Tersedia	30000	4	-	Meja biasa, Proyektor, AC, Wifi
3	Ruang C	Tersedia	35000	6	-	Meja diskusi, Whiteboard, AC, Wifi
4	Ruang D	Tersedia	40000	8	-	Meja Rapat, Proyektor, Mic, Wifi
5	Ruang E	Tersedia	50000	10	-	Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi

Masukkan nomor ruangan yang ingin dipesan : 3
 Masukkan durasi : 2
 Ruangan Ruang C berhasil dipesan.

Gambar 4.3 Pesan Ruangan

Daftar Ruangan yang Dipesan :

No	Nama	Harga	Kapasitas	Durasi	Fasilitas
3	Ruang C	35000	6	2	Meja diskusi, Whiteboard, AC, Wifi

Masukkan nomor ruangan yang ingin dikonfirmasi pembayarannya : 3
 Pembayaran untuk ruangan Ruang C dikonfirmasi.
 Ruangan kini tersedia kembali.

Gambar 4.4 Konfirmasi Pembayaran

B. Skenario Kedua

Masukkan Nama: Daffa
 Masukkan NIM: 2409106050

Menu Admin:

1. Tambah Ruangan
2. Lihat Ruangan
3. Perbarui Ruangan
4. Hapus Ruangan
5. Keluar

Pilih menu: █

Gambar 4.5 Login Dan Menu Admin

Daftar Ruangan:

No	Nama	Status	Harga	Kapasitas	Durasi	Fasilitas
1	Ruang A	Tersedia	25000	4	-	Meja biasa, Proyektor, Wifi
2	Ruang B	Tersedia	30000	4	-	Meja biasa, Proyektor, AC, Wifi
3	Ruang C	Tersedia	35000	6	-	Meja diskusi, Whiteboard, AC, Wifi
4	Ruang D	Tersedia	40000	8	-	Meja Rapat, Proyektor, Mic, Wifi
5	Ruang E	Tersedia	50000	10	-	Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi

Gambar 4.6 Lihat Ruangan

Daftar Ruangan:

No	Nama	Status	Harga	Kapasitas	Durasi	Fasilitas
1	Ruang A	Tersedia	25000	4	-	Meja biasa, Proyektor, Wifi
2	Ruang B	Tersedia	30000	4	-	Meja biasa, Proyektor, AC, Wifi
3	Ruang C	Tersedia	35000	6	-	Meja diskusi, Whiteboard, AC, Wifi
4	Ruang D	Tersedia	40000	8	-	Meja Rapat, Proyektor, Mic, Wifi
5	Ruang E	Tersedia	50000	10	-	Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi

Masukkan nomor ruangan yang ingin diperbarui : 4
 Masukkan nama : Ruang H
 Masukkan harga : 45000
 Masukkan kapasitas : 10
 Masukkan fasilitas : Meja Rapat x2, Proyektor, Mic, Wifi
 Ruangan berhasil diperbarui!

Gambar 4.7 Perbarui Ruangan

C. Skenario ketiga

```
Masukkan Nama: daffa
Masukkan NIM: 203123
Login gagal! Kesempatan tersisa: 2
Masukkan Nama: daddaw
Masukkan NIM: 321312
Login gagal! Kesempatan tersisa: 1
Masukkan Nama: daffa
Masukkan NIM: 2409106051
Login gagal! Kesempatan tersisa: 0
Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 4.8 Kesalahan Login

5. Langkah-langkah Git

A. GIT Add

```
PS C:\Users\ASUS\Desktop\Praktikum-APL> git add .
```

Gambar 5.1 GIT Add

GITAdd untuk memasukan file-file yang telah diubah atau dimodif. Disini saya menggunakan . karena untuk semua file sekaligus.

B. GIT Commit

```
PS C:\Users\ASUS\Desktop\Praktikum-APL> git commit -m "Finish Post Test 2"
[main f001f02] Finish Post Test 2
2 files changed, 268 insertions(+)
create mode 100644 Post-Test/Post-Test-2/main.cpp
create mode 100644 Post-Test/Post-Test-2/main.exe
```

Gambar 5.2 GIT Commit

Git Commit bertujuan agar file-file yang sudah di Git Add tadi masuk kedalam stage sehingga dapat melakukan Git Push untuk memasukan perubahan yang dilakukan kedalam github yang terhubung ke file kita.

C. GIT Push

```
PS C:\Users\ASUS\Desktop\Praktikum-APL> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 682.45 KiB | 6.38 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Dapaaaaaa/Praktikum-APL.git
82a4e02..f001f02 main -> main
PS C:\Users\ASUS\Desktop\Praktikum-APL>
```

Gambar 5.3 GIT Push

GIT Push dilakukan agar file-file yang telah diubah atau dimodif akan masuk kedalam repositori GITHUB kita sehingga saat melakukan pull maka sudah akan terupdate menjadi file yang telah diperbarui atau dimodif, disini menggunakan git push tanpa origin main karena apabila melakukan perintah git push origin main maka saat melakukan git push akan juga push ke main.