LAPORAN PRAKTIKUM

POSTTEST 5

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Ananda Daffa Harahap (2409106050)

Kelas (B1 ‘24)

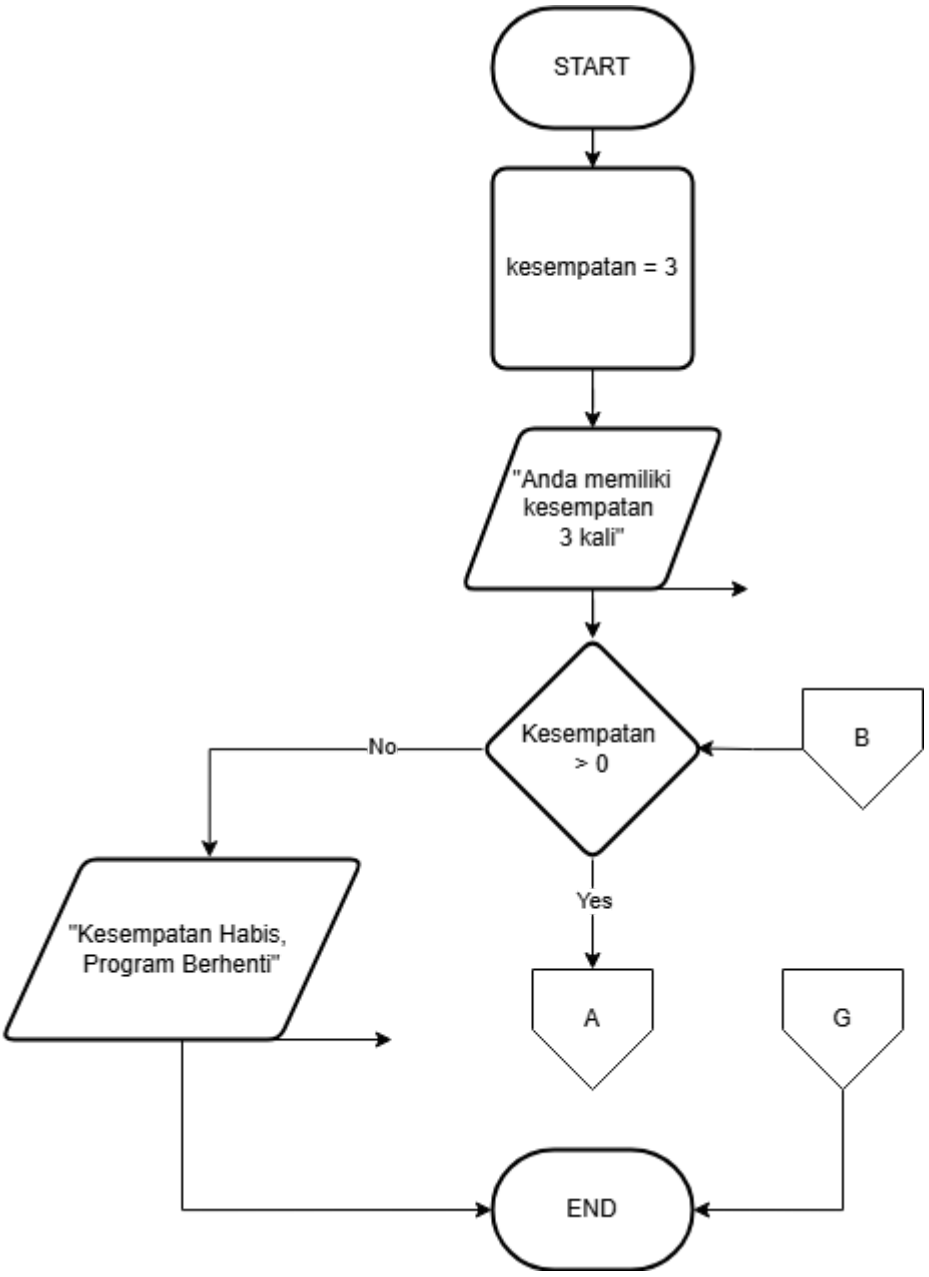
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

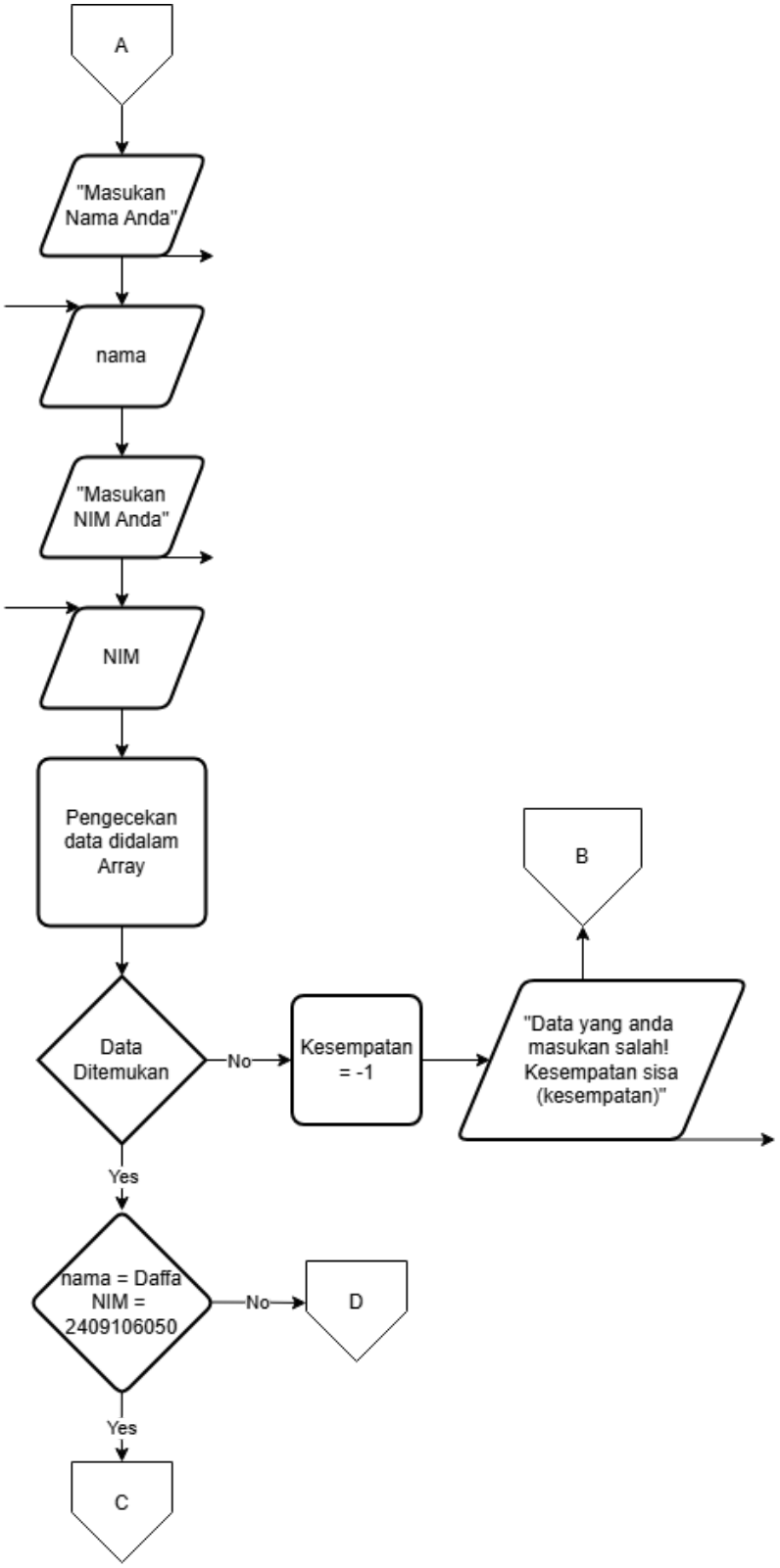
SAMARINDA

2025

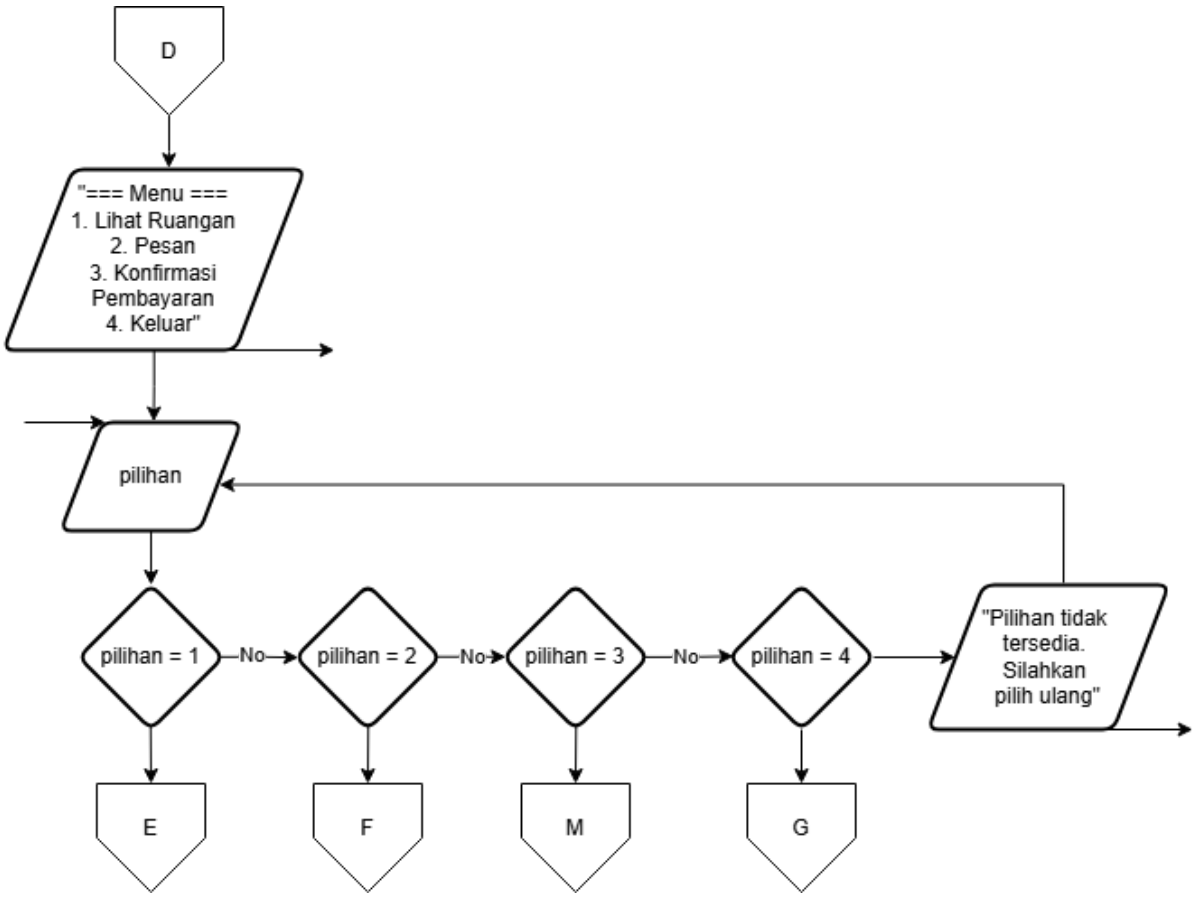
1. Flowchart



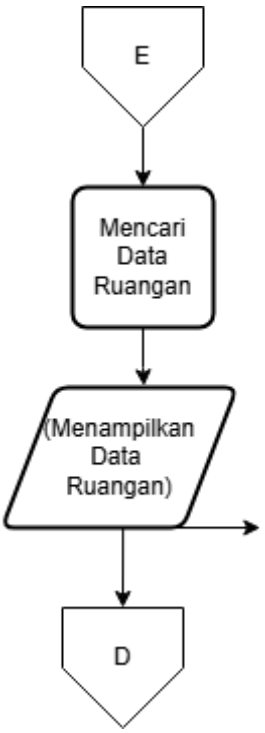
Gambar 1.1 *Flowchart* Bagian 1



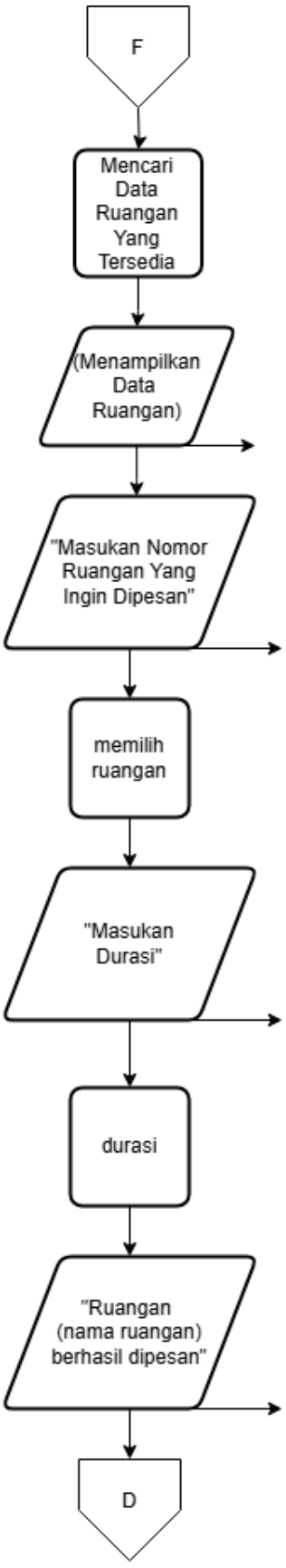
Gambar 1.2 *Flowchart* Bagian 2



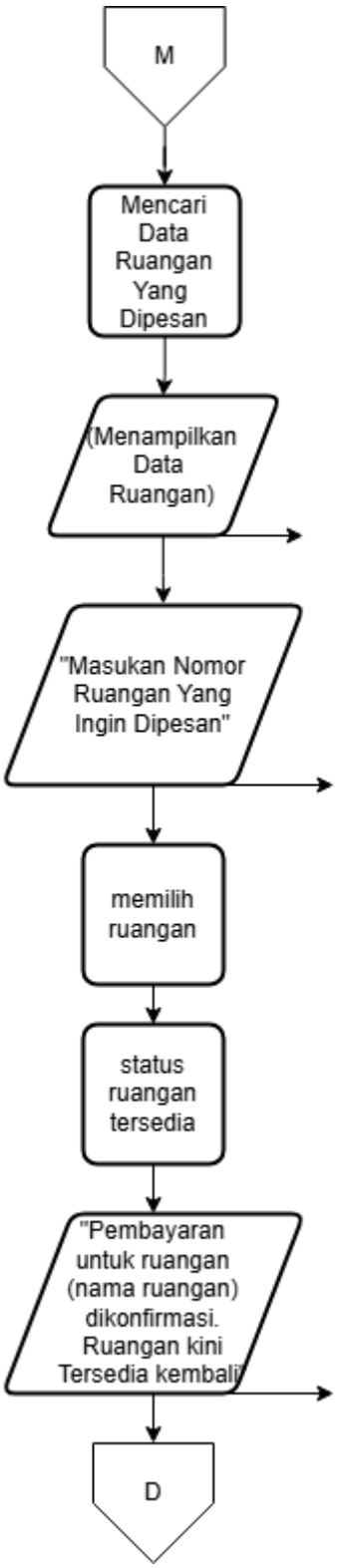
Gambar 1.3 *Flowchart* Bagian 3



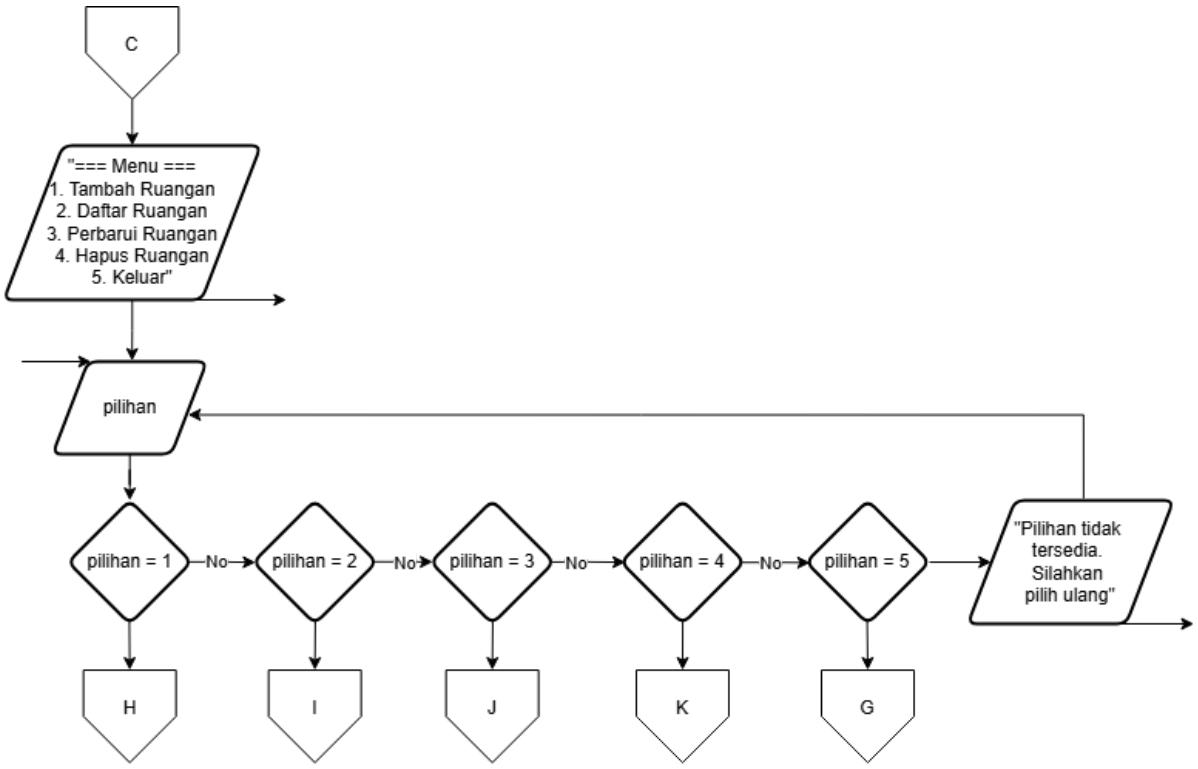
Gambar 1.4 *Flowchart* Bagian 4



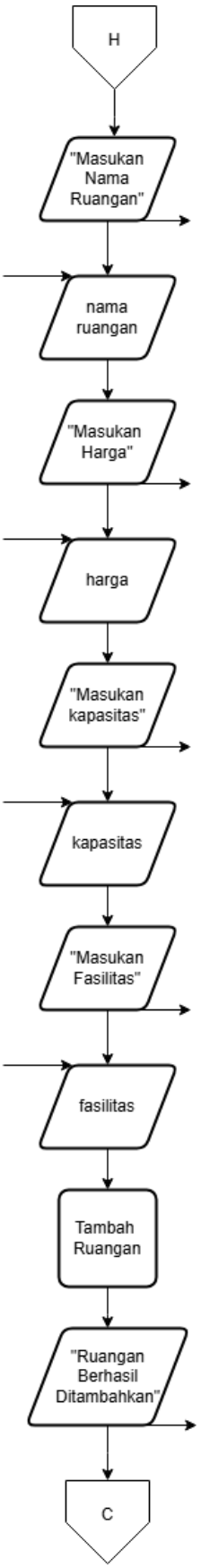
Gambar 1.5 *Flowchart* Bagian 5



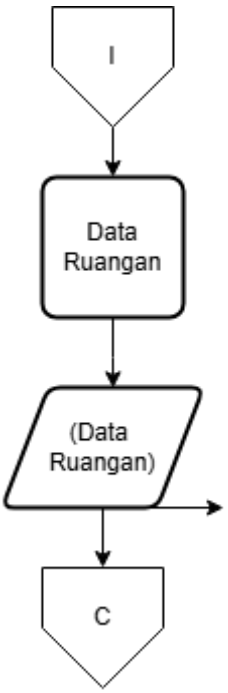
Gambar 1.6 *Flowchart* Bagian 6



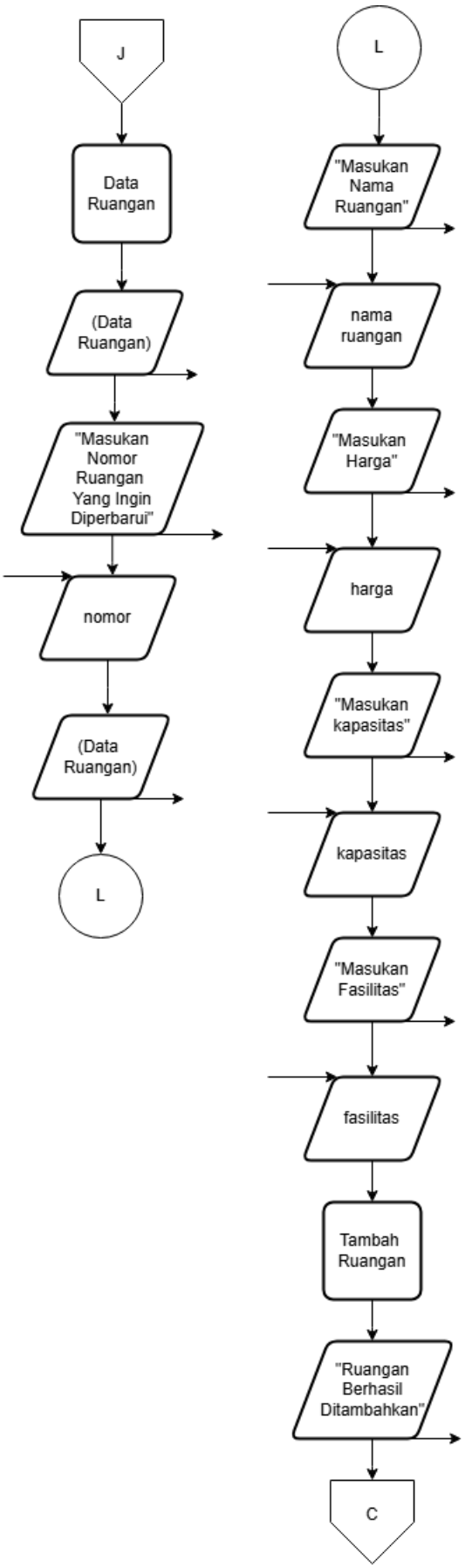
Gambar 1.7 *Flowchart* Bagian 7



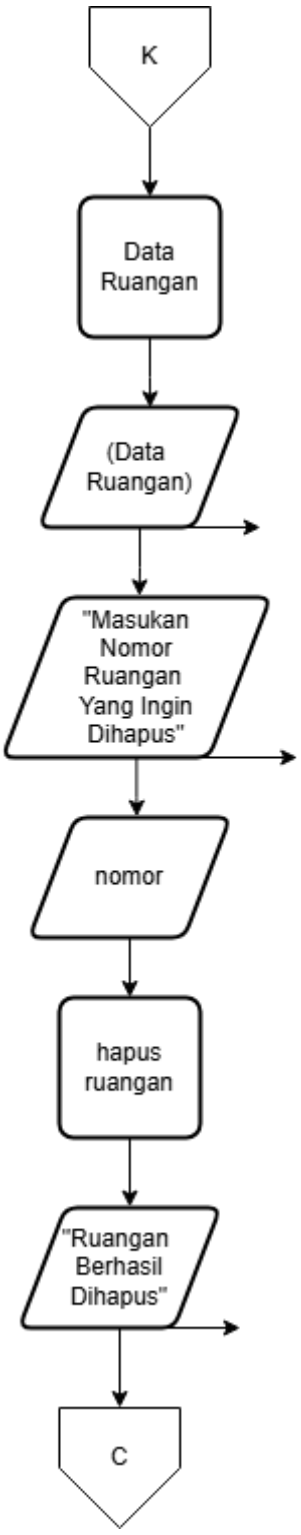
Gambar 1.8 *Flowchart* Bagian 8



Gambar 1.9 *Flowchart* Bagian 9



Gambar 1.10 *Flowchart* Bagian 10



Gambar 1.11 *Flowchart* Bagian 11

2. Analisis Program

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.

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 memasukan 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, status;
    int kesempatan = 3;
    bool loginBerhasil = false;
    bool isAdmin = false;

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

        if (nama == daftarPengguna[0].nama && nim == daftarPengguna[0].nim) {
            loginBerhasil = true;
            isAdmin = (daftarPengguna[0].status == "Admin");
            break;
        } else if (nama == daftarPengguna[1].nama && nim ==
daftarPengguna[1].nim) {
            loginBerhasil = true;
            isAdmin = (daftarPengguna[1].status == "Admin");
            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 pertama-tama akan menampilkan daftar ruangan yang sudah ada sehingga admin mengetahui nama ruangan apa saja yang sudah ada didalam sistem kemudian sistem akan menjalankan perintah untuk menambahkan ruangan sesuai yang diinginkan oleh admin.

Source Code :

```
if (isAdmin) {
            if (pilihan == 1) {
                lihatRuangan();
                tambahRuangan();
            }
```

D. Admin (Pilihan 2)

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

Source Code :

```
else if (pilihan == 2) {
                lihatRuangan();
            }
```

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 konfirmasi apa saja yang ada seperti apakah ada AC? apakah ada Whiteboard? DLL.

Source Code :

```
else if (pilihan == 3) {
                lihatRuangan();
                perbaruiRuangan();
            }
```

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) {
                lihatRuangan();
                hapusRuangan();
            }
```

G. Kasir (Pilihan 1)

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

Source Kode :

```
if (pilihan == 1) {
                lihatRuangan();
            }
```

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) {
                lihatRuangan();
                pesanRuangan();
            }
```

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.meja << ", "
```

```
                            << (daftarRuangan[i].fasilitas.proyektor ?
"Proyektor, " : "")
                            << (daftarRuangan[i].fasilitas.whiteboard ?
"Whiteboard, " : "")
                            << (daftarRuangan[i].fasilitas.soundSystem ?
"Sound System, " : "")
                            << (daftarRuangan[i].fasilitas.ac ? "AC, " :
"")
                            << (daftarRuangan[i].fasilitas.wifi ? "Wifi" :
"") << 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;
```

K. Prosedur lihatRuangan

Saat dipanggil maka program akan otomatis menampilkan data ruangan.

Source Kode :

```
void lihatRuangan() {
    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.meja << ", "
                         << (daftarRuangan[i].fasilitas.proyektor ?
"Proyektor, " : "")
                         << (daftarRuangan[i].fasilitas.whiteboard ?
"Whiteboard, " : "")
                         << (daftarRuangan[i].fasilitas.soundSystem ? "Sound
System, " : "")
                         << (daftarRuangan[i].fasilitas.ac ? "AC, " : "")
                         << (daftarRuangan[i].fasilitas.wifi ? "Wifi" : "")
<< endl;
                }
                cout <<
"-------------------------------------------------------------------------\n";
}
```

L. Deference dan Address-Of

```
void perbaruiHargaRuangan(int* harga) {
    cout << "Masukkan harga baru: ";
    cin >> *harga;
}

bool cekKetersediaanRuangan(Ruangan* ruangan) {
    return ruangan->tersedia;
}
```

M. Prosedur tambahRuangan

Saat dipanggil maka program akan otomatis menampilkan output yang diperlukan untuk menambahkan ruangan.

Source Kode :

```
void tambahRuangan() {
    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 Meja : ";
        getline(cin, daftarRuangan[jumlahRuangan].fasilitas.meja);

        char pilihan;
        cout << "Apakah ada proyektor? (y/n) : ";
        cin >> pilihan;
        daftarRuangan[jumlahRuangan].fasilitas.proyektor = (pilihan == 'y');

        cout << "Apakah ada whiteboard? (y/n) : ";
        cin >> pilihan;
        daftarRuangan[jumlahRuangan].fasilitas.whiteboard = (pilihan ==
'y');

        cout << "Apakah ada sound system? (y/n) : ";
        cin >> pilihan;
        daftarRuangan[jumlahRuangan].fasilitas.soundSystem = (pilihan ==
'y');
```

```
        cout << "Apakah ada AC? (y/n) : ";
        cin >> pilihan;
        daftarRuangan[jumlahRuangan].fasilitas.ac = (pilihan == 'y');

        cout << "Apakah ada wifi? (y/n) : ";
        cin >> pilihan;
        daftarRuangan[jumlahRuangan].fasilitas.wifi = (pilihan == 'y');
        daftarRuangan[jumlahRuangan].durasi = 0;
        daftarRuangan[jumlahRuangan].tersedia = true;
        jumlahRuangan++;

        cout << "Ruangan berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas ruangan penuh!\n";
    }
```

N. Prosedur perbaruiRuangan

Saat dipanggil maka program akan otomatis menampilkan output yang diperlukan untuk memperbarui data ruangan.

Source Kode :

```
void perbaruiRuangan() {
    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);

        perbaruiHargaRuangan(&daftarRuangan[pilih - 1].harga);

        cout << "Masukkan kapasitas : ";
        cin >> daftarRuangan[pilih - 1].kapasitas;
        cin.ignore();

        cout << "Masukkan jenis meja: ";
        getline(cin, daftarRuangan[pilih - 1].fasilitas.meja);

        char pilihan;
        cout << "Apakah ada proyektor? (y/n): ";
        cin >> pilihan;
        daftarRuangan[pilih - 1].fasilitas.proyektor = (pilihan == 'y');

        cout << "Apakah ada whiteboard? (y/n): ";
```

```
        cin >> pilihan;
        daftarRuangan[pilih - 1].fasilitas.whiteboard = (pilihan == 'y');

        cout << "Apakah ada sound system? (y/n): ";
        cin >> pilihan;
        daftarRuangan[pilih - 1].fasilitas.soundSystem = (pilihan == 'y');

        cout << "Apakah ada AC? (y/n): ";
        cin >> pilihan;
        daftarRuangan[pilih - 1].fasilitas.ac = (pilihan == 'y');

        cout << "Apakah ada WiFi? (y/n): ";
        cin >> pilihan;
        daftarRuangan[pilih - 1].fasilitas.wifi = (pilihan == 'y');

        cin.ignore();
        cout << "Ruangan berhasil diperbarui!\n";
    } else {
        cout << "Nomor ruangan tidak valid!\n";
    }
}
        daftarRuangan[pilih - 1].fasilitas.wifi = (pilihan == 'y');

        cin.ignore();
        cout << "Ruangan berhasil diperbarui!\n";
    } else {
        cout << "Nomor ruangan tidak valid!\n";
    }
}
```

O. Prosedur hapusRuangan

Saat dipanggil maka program akan otomatis menampilkan output untuk menghapus data ruangan.

Source Kode :

```
void hapusRuangan() {
    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";
    }
}
```

P. Prosedur pesanRuangan

Saat dipanggil maka program akan otomatis menampilkan output untuk memesan ruangan.

Source Kode :

```
void pesanRuangan() {
    cout << "Masukkan nomor ruangan yang ingin dipesan : ";
    int pilih, durasi;
    cin >> pilih;
    cout << "Masukkan durasi (Jam) : ";
    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";
    }
}
```

4. Uji Coba dan Hasil Output

```
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

```
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, Sound System, Wifi
5       Ruang E       Tersedia      50000     10        -         Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi
----------------------------------------------------------------------
Masukkan nama ruangan : Ruang F
Masukkan harga : 20000
Masukkan kapasitas ruangan : 2
Masukkan Meja : Meja Biasa
Apakah ada proyektor? (y/n) : n
Apakah ada whiteboard? (y/n) : y
Apakah ada sound system? (y/n) : n
Apakah ada AC? (y/n) : y
Apakah ada wifi? (y/n) : y
Ruangan berhasil ditambahkan!
```

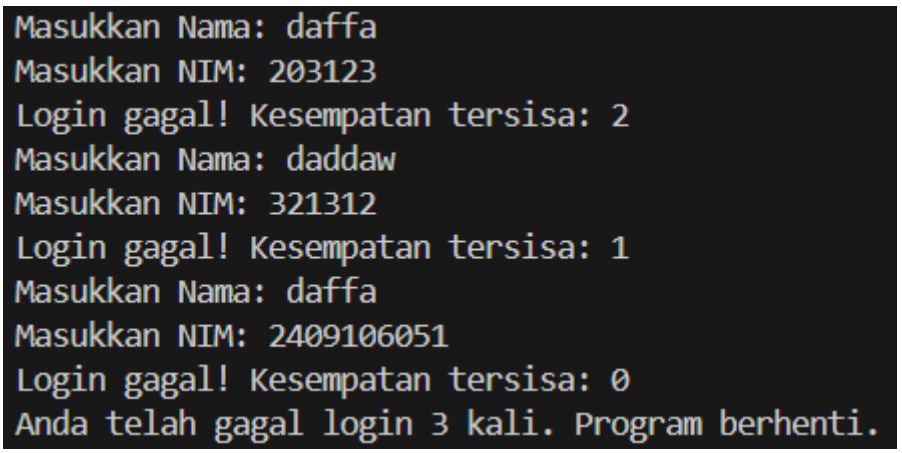
Gambar 4.7 Tambah Ruangan

```
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, Sound System, Wifi
5       Ruang E       Tersedia      50000     10        -         Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi
6       Ruang F       Tersedia      20000     2         -         Meja Biasa, Whiteboard, AC, Wifi
----------------------------------------------------------------------
Masukkan nomor ruangan yang ingin diperbarui : 6
Masukkan nama : Ruang Hemat
Masukkan harga : 20000
Masukkan kapasitas : 2
Masukkan jenis meja: Meja Biasa
Apakah ada proyektor? (y/n): n
Apakah ada whiteboard? (y/n): n
Apakah ada sound system? (y/n): n
Apakah ada AC? (y/n): y
Apakah ada WiFi? (y/n): y
Ruangan berhasil diperbarui!
```

Gambar 4.8 Perbarui 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, Sound System, Wifi
5       Ruang E       Tersedia      50000     10        -         Meja Rapat, Proyektor, Whiteboard, Sound System, Wifi
6       Ruang Hemat   Tersedia      20000     2         -         Meja Biasa, AC, Wifi
----------------------------------------------------------------------
```

Gambar 4.9 Lihat Ruangan Setelah Diperbarui



Gambar 4.10 Kesalahan Login

5. Langkah-langkah Git

A. GIT Add

```
PS F:\Kuliah\Praktikum\Algoritma Pemograman Lanjut> 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 5"
[main d7442b6] Finish Post Test 5
 3 files changed, 318 insertions(+)
 create mode 100644 Post-Test/Post-Test-5/2409106050-AnandaDaffaHarahap-PT-5.cpp
 create mode 100644 Post-Test/Post-Test-5/2409106050-AnandaDaffaHarahap-PT-5.exe
 create mode 100644 Post-Test/Post-Test-5/2409106050-AnandaDaffaHarahap-PT-5.pdf
PS C:\Users\ASUS\Desktop\Praktikum-APL>
```

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: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.49 MiB | 264.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Dapaaaaaa/Praktikum-APL.git
   dd28ac8..d7442b6  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.