

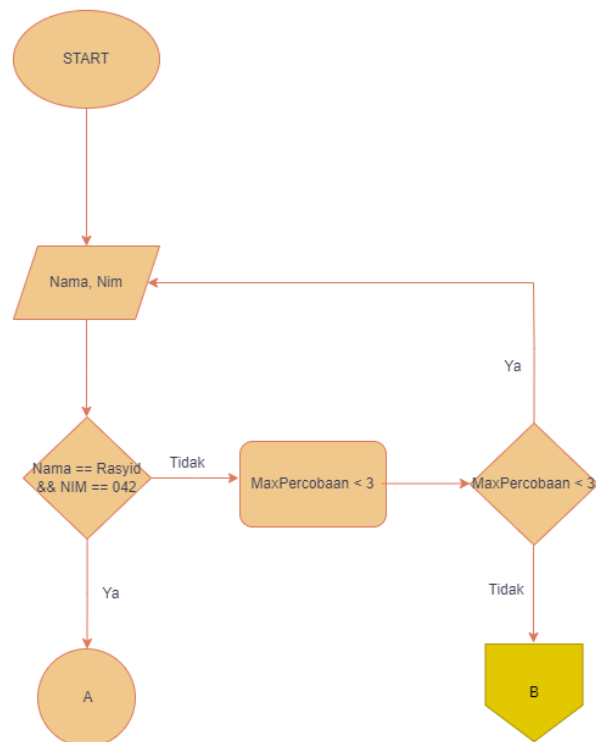
**LAPORAN PRAKTIKUM**  
**POSTTEST (2)**  
**ALGORITMA PEMROGRAMAN LANJUT**



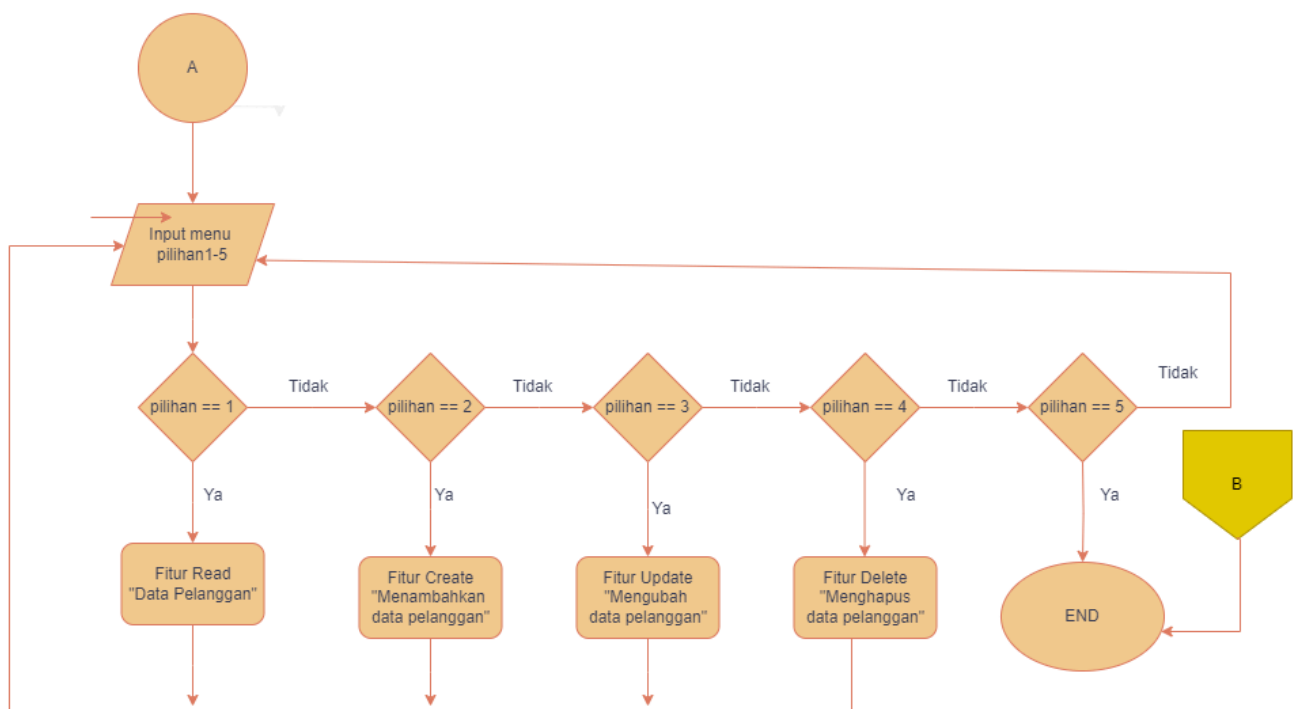
**Disusun oleh:**  
**Muhammad Rasyid (2409106042)**  
**Kelas (A2'24)**

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

## 1. Flowchart



Gambar 1.1 Program Login



Gambar 1.2 Menu Program CRUD

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Membuat program manajemen pemesanan kamar kost memiliki manfaat yang signifikan, terutama dalam memudahkan pemilik kost untuk menyimpan data penghuni/pelanggan. Program ini sangat berguna dalam kehidupan sehari-hari, di mana pendataan data diri pelanggan sering diperlukan, seperti nama, nomor telepon, dan juga nomor kamar yang mereka tinggali. Dengan adanya program ini, pemilik kost tidak perlu menulis/mendata secara manual, sehingga menghemat waktu dan mengurangi risiko kesalahan pendataan.

### 2.2 Penjelasan Alur & Algoritma

#### 2.2.1 Sistem Login:

- a. Program dimulai dengan Login.
- b. Pertama Input Nama dan NIM.
- c. Jika nama yang di input “rasyid” dan NIM nya “042” maka login berhasil
- d. Jika salah menginput maka program akan berulang sampai 3 kali. Jika gagal login 3 kali maka program berhenti.

#### 2.2.2 Fitur Crud: Jika login berhasil, program akan masuk ke menu pilihan.

- Tampilkan Data Kamar.
- Tambah Data Pelanggan.
- Mengubah Data Pelanggan.
- Menghapus Data Pelanggan.
- Keluar dari program.

Input pilihan 1-5,

- a. Jika pengguna menginput 1:
  - Jika tidak ada data “panjang == 0”, program akan menampilkan output “Belum ada pesanan”.
  - Jika ada data tambahan, program akan menampilkan data penghuni yang berisi nama nomor telepon dan nomor kamar.
- b. Jika pengguna menginput 2:
  - Program meminta input nama, nomor telepon, dan nomor kamar.
  - Hasil inputan akan tersimpan kedalam variabel panjang yang akan di tampilkan di fitur read.
- c. Jika pengguna menginput 3:
  - Jika tidak ada data “panjang == 0”, program akan menampilkan output “Belum ada untuk diubah”.
  - Jika ada data tambahan, program akan meminta input index seberapa yang ingin di ubah.

- Jika sudah memilih data mana yg ingin di ubah, program akan meminta input nama, nomor telepon, dan kamar baru untuk di simpan dan di perbarui di fitur read.
- d. Jika pengguna menginput 4:
  - Jika tidak ada data “panjang == 0”, program akan menampilkan output “Belum ada untuk dihapus”.
  - Jika ada data tambahan, program akan meminta input index beberapa yang ingin di hapus.
- e. Jika pengguna menginput 5:
  - Keluar dari program.
- f. Jika pengguna menginput selain dari 1-5 maka program akan terus berulang di bagian menu pilihan sampai pengguna menginput yang benar.

### 3. Source Code

#### A. Login

Fitur ini digunakan untuk menambahkan data masuk pengguna baru sebagai keamanan. Data yang ditambahkan berupa nama dan 3 digit NIM.

#### Source Code:

```
#include <iostream>
using namespace std;

#define MAX_KAMAR 100

int main() {
    string kamar[MAX_KAMAR];
    string nama[MAX_KAMAR];
    string nim[MAX_KAMAR];
    string inputNama, inputNim;
    int panjang = 0;
    int maksimalpercobaan = 3;
    int pilihan, index;

    // Login system
    for (int percobaan = 1; percobaan <= maksimalpercobaan; percobaan++) {
        cout << "=== LOGIN ===" << endl;
        cout << "Masukkan nama: ";
        cin >> inputNama;
        cout << "Masukkan NIM (3 digit angka): ";
        cin >> inputNim;

        if (inputNama == "rasyid" && inputNim == "042") {
            cout << "Login berhasil! Selamat datang, " << inputNama << endl;
        }
    }
}
```

#### B. Menu

Fitur ini digunakan untuk memilih menu yang akan di input dan berisi fitur Create, Read, Update, dan Delete.

## Source Code:

```
do {
    cout << "\nMANAJEMEN PEMESANAN KAMAR KOST PUTRA\n" << endl;
    cout << "1. Tampilkan Data Kost" << endl;
    cout << "2. Tambah Data Pelanggan/penghuni" << endl;
    cout << "3. Ubah Data Pelanggan/penghuni" << endl;
    cout << "4. Hapus Data Pelanggan/peghuni" << endl;
    cout << "5. Keluar Dari Program" << endl;
    cout << "Pilih menu: ";
    cin >> pilihan;

    switch (pilihan) {
        case 1: // FITUR READ Melihat data penghuni kamar
            if (panjang == 0) {
                cout << "Belum ada pesanan\n" << endl;
            } else {
                cout << "No    Nama                                No.Tlp
Kamar" << endl;

                cout <<
                "-----" << endl;
                for (int i = 0; i < panjang; i++) {
                    cout << i + 1 << "    " << nama[i];
                    for (int j = nama[i].length(); j < 20; j++)
                        cout << " ";
                    cout << nim[i];
                    for (int j = nim[i].length(); j < 20; j++)
                        cout << " ";
                    cout << kamar[i] << endl;
                }
            }
            break;

        case 2: // FITUR CREATE Menambahkan data penghuni dan
kamar

            if (panjang < MAX_KAMAR) {
                cout << "Masukkan nama: ";
                cin.ignore();
                getline(cin, nama[panjang]);
                cout << "Masukkan nomor telepon: ";
                getline(cin, nim[panjang]);
```

```

        cout << "Masukkan nomor kamar: ";
        getline(cin, kamar[panjang]);
        panjang++;
        cout << "Data pesanan berhasil ditambahkan\n" <<
endl;

    } else {
        cout << "Kapasitas penuh! Tidak bisa menambah
penghuni lagi.\n" << endl;
    }
    break;

case 3: // FITUR UPDATE Mengubah data penghuni
    if (panjang == 0) {
        cout << "Belum ada kamar untuk diubah." << endl;
    } else {
        cout << "No    Nama                                NIM
Kamar" << endl;

        cout <<
"-----" << endl;
        for (int i = 0; i < panjang; i++) {
            cout << i + 1 << "    " << nama[i];
            for (int j = nama[i].length(); j < 20; j++)
cout << " ";

            cout << nim[i];
            for (int j = nim[i].length(); j < 20; j++)
cout << " ";

            cout << kamar[i] << endl;
        }
        cout << "Masukkan nomor kamar yang akan diubah:
";

        cin >> index;
        if (index > 0 && index <= panjang) {
            cout << "Masukkan nama baru: ";
            cin.ignore();
            getline(cin, nama[index - 1]);
            cout << "Masukkan NIM baru: ";
            getline(cin, nim[index - 1]);
            cout << "Masukkan nomor kamar baru: ";
            getline(cin, kamar[index - 1]);
            cout << "Kamar berhasil diubah" << endl;
        } else {

```

```

        cout << "Nomor kamar tidak valid" << endl;
    }
}
break;

case 4: // FITUR DELETE Menghapus data penghuni
    if (panjang == 0) {
        cout << "Belum ada kamar untuk dihapus.\n" <<
endl;

        } else {
            cout << "No    Nama                NIM
Kamar" << endl;

            cout <<
"-----" << endl;
            for (int i = 0; i < panjang; i++) {
                cout << i + 1 << "    " << nama[i];
                for (int j = nama[i].length(); j < 20; j++)
cout << " ";

                cout << nim[i];
                for (int j = nim[i].length(); j < 20; j++)
cout << " ";

                cout << kamar[i] << endl;
            }
            cout << "Masukkan nomor kamar yang akan dihapus:
";

            cin >> index;
            if (index > 0 && index <= panjang) {
                for (int i = index - 1; i < panjang - 1;
i++) {

                    nama[i] = nama[i + 1];
                    nim[i] = nim[i + 1];
                    kamar[i] = kamar[i + 1];
                }
                panjang--;
                cout << "Kamar berhasil dihapus\n" << endl;
            } else {
                cout << "Nomor kamar tidak valid\n" << endl;
            }
        }
    }
break;

```



```

        case 5: // Keluar dari program
            cout << "Keluar dari program" << endl;
            break;

        default:
            cout << "Pilihan tidak valid" << endl;
            break;
    }
} while (pilihan != 5);

return 0;
} else {
    cout << "\nNama atau NIM yang anda masukkan salah! Percobaan
tersisa: " << maksimalpercobaan - percobaan << endl;
}
}
cout << "\nAnda telah gagal login 3 kali. Program berhenti." << endl;
return 0;
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Skenario yang saya gunakan ada 3 yaitu :

1. Skenario 1 Salah menginput nama dan nim pada saat login sebanyak 3 kali percobaan.
2. Skenario 2 Berhasil login tetapi salah menginput program menu
3. Skenario 3 Berhasil login dan berhasil menginput program menu dengan berulang lalu keluar dari program.

## 4.2 Hasil Output

```
=== LOGIN ===  
Masukkan nama: rasyid  
Masukkan NIM (3 digit angka): 000  
  
Nama atau NIM yang anda masukkan salah! Percobaan tersisa: 2  
=== LOGIN ===  
Masukkan nama: rsd  
Masukkan NIM (3 digit angka): 042  
  
Nama atau NIM yang anda masukkan salah! Percobaan tersisa: 1  
=== LOGIN ===  
Masukkan nama: tes  
Masukkan NIM (3 digit angka): 120  
  
Nama atau NIM yang anda masukkan salah! Percobaan tersisa: 0  
  
Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 4.1 output skenario 1

```
=== LOGIN ===  
Masukkan nama: rasyid  
Masukkan NIM (3 digit angka): 042  
Login berhasil! Selamat datang, rasyid  
  
MANAJEMEN PEMESANAN KAMAR KOST PUTRA  
  
1. Tampilkan Data Kost  
2. Tambah Data Pelanggan/penghuni  
3. Ubah Data Pelanggan/penghuni  
4. Hapus Data Pelanggan/peghuni  
5. Keluar Dari Program  
Pilih menu: 6  
Pilihan tidak valid  
  
MANAJEMEN PEMESANAN KAMAR KOST PUTRA  
  
1. Tampilkan Data Kost  
2. Tambah Data Pelanggan/penghuni  
3. Ubah Data Pelanggan/penghuni  
4. Hapus Data Pelanggan/peghuni  
5. Keluar Dari Program  
Pilih menu: |
```

Gambar 4.1 output skenario 2

```

=== LOGIN ===
Masukkan nama: rasyid
Masukkan NIM (3 digit angka): 042
Login berhasil! Selamat datang, rasyid

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/peghuni
5. Keluar Dari Program
Pilih menu: 2
Masukkan nama: rasyid
Masukkan nomor telepon: 0812345
Masukkan nomor kamar: 1
Data pesanan berhasil ditambahkan

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/peghuni
5. Keluar Dari Program
Pilih menu: 1

```

No	Nama	No.Tlp	Kamar
1	rasyid	0812345	1

```

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/peghuni
5. Keluar Dari Program
Pilih menu: 5
Keluar dari program
PS C:\Users\Public\Documents\praktikum-apl\post-test\post-test-apl-2>

```

Gambar 4.1 output skenario 3

## 5. GIT

```
PS C:\Users\Public\Documents\praktikum-apl> git init
Initialized empty Git repository in C:/Users/Public/Documents/praktikum-apl/.git/
PS C:\Users\Public\Documents\praktikum-apl> git add .
PS C:\Users\Public\Documents\praktikum-apl> git status

On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   post-test/post-test-apl-2/2409106042-MuhammadRasyid-PT-2.cpp
    new file:   post-test/post-test-apl-2/2409106042-MuhammadRasyid-PT-2.exe

PS C:\Users\Public\Documents\praktikum-apl> git commit -m "program"
[main 7df39dc] program
 2 files changed, 144 insertions(+)
 create mode 100644 post-test/post-test-apl-2/2409106042-MuhammadRasyid-PT-2.cpp
 create mode 100644 post-test/post-test-apl-2/2409106042-MuhammadRasyid-PT-2.exe
PS C:\Users\Public\Documents\praktikum-apl> git remote add origin https://github.com/MRasyid18/praktikum-apl.git
error: remote origin already exists.
PS C:\Users\Public\Documents\praktikum-apl> git branch -M main
PS C:\Users\Public\Documents\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 676.78 KiB | 7.20 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MRasyid18/praktikum-apl.git
   ec20712..7df39dc  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Public\Documents\praktikum-apl> 
```

Gambar 5.1 Kode untuk masuk ke github

Selain github untuk mengupload tugas kali ini bisa menggunakan software tambahan untuk mempermudah dalam mengontrol perubahan pada repository github. Berikut langkah langkah yang bisa di gunakan dalam mengakses repository dengan github adalah sebagai berikut:

- git init : digunakan di terminal vs code untuk menginisialisasi repository Git di dalam folder proyek. Setelah menjalankan perintah ini, Git akan membuat folder
- git add . : digunakan untuk menambahkan file ke dalam staging area, sehingga siap untuk dikomit.
- git status : perintah ini berfungsi untuk mengecek proses file yang akan di upload ke dalam repository.
- git commit : berfungsi menyimpan perubahan dengan pesan commit.
- git remote add origin : berfungsi menghubungkan repository lokal ke GitHub.
- git push -u origin main : perintah mengunggah kode dari repository lokal ke repository remote (GitHub) dan menetapkan branch default.