

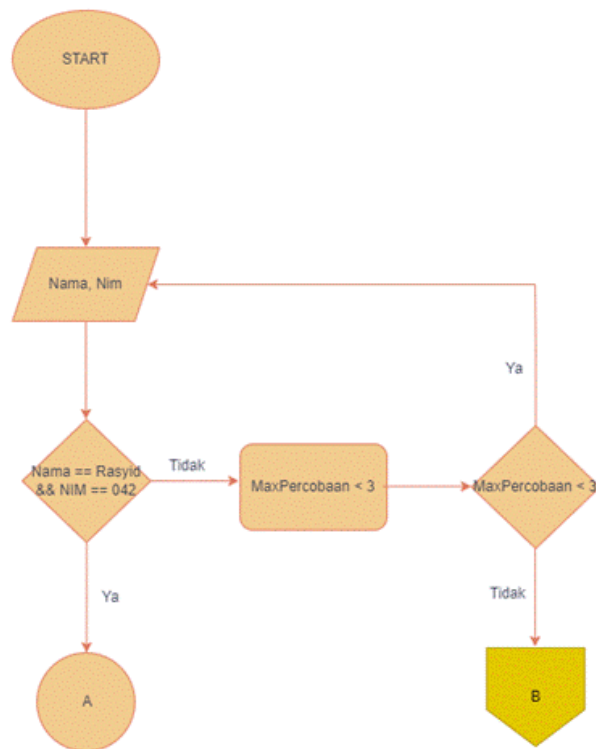
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



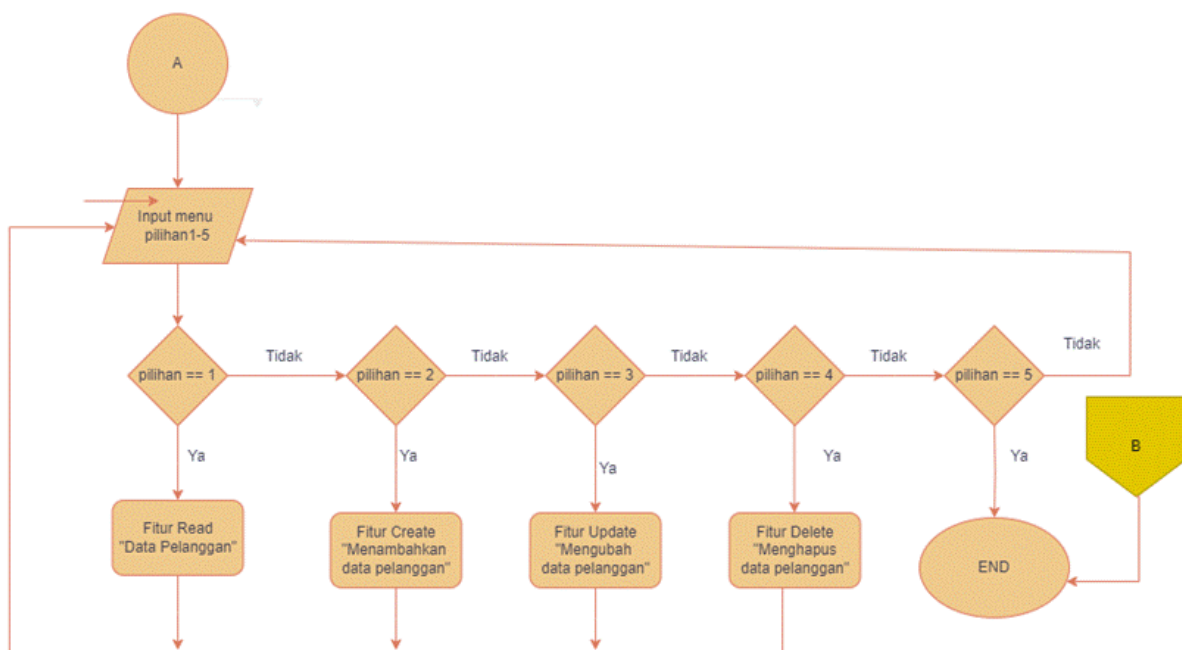
**Disusun oleh:**  
**Muhammad Rasyid (2409106042)**  
**Kelas (A2'24)**

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

## 1. Flowchart



*Gambar 1.1 Program Login*



*Gambar 1.2 Menu Program CRUD*

## **2. Analisis Program**

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.

### 3. Source Code

Login

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

#define MAX_KAMAR 100

struct Penghuni {
    string nama;
    string umur;
    string kamar;
};

struct Datakost {
    Penghuni penghuni[MAX_KAMAR];
    int panjang = 0;
};

bool login(string nama, string nim) {
    return nama == "rasyid" && nim == "042";
}
```

Menu Fungsi dan Prosedur

```
void tampilkanData(Datakost &data) {
    if (data.panjang == 0) {
        cout << "Belum ada pesanan\n" << endl;
    } else {
        cout << "No    Nama                Umur                Kamar" <<
endl;
        cout << "-----" <<
endl;
        for (int i = 0; i < data.panjang; i++) {
            cout << i + 1 << "    " << data.penghuni[i].nama;
            for (int j = data.penghuni[i].nama.length(); j < 20; j++) cout
<< " ";
            cout << data.penghuni[i].umur;
            for (int j = data.penghuni[i].umur.length(); j < 20; j++) cout
<< " ";
            cout << data.penghuni[i].kamar << endl;
        }
    }
}

void tambahData(Datakost &data) {
    if (data.panjang < MAX_KAMAR) {
        cout << "Masukkan nama: ";
    }
}
```

```

        cin.ignore();
        getline(cin, data.penghuni[data.panjang].nama);
        cout << "Masukkan umur: ";
        getline(cin, data.penghuni[data.panjang].umur);
        cout << "Masukkan nomor kamar: ";
        getline(cin, data.penghuni[data.panjang].kamar);
        data.panjang++;
        cout << "Data pesanan berhasil ditambahkan\n" << endl;
    } else {
        cout << "Kapasitas penuh! Tidak bisa menambah penghuni lagi.\n" <<
endl;
    }
}

void ubahData(Datakost &data) {
    if (data.panjang == 0) {
        cout << "Belum ada kamar untuk diubah." << endl;
    } else {
        tampilkanData(data);
        int index;
        cout << "Masukkan nomor kamar yang akan diubah: ";
        cin >> index;
        if (index > 0 && index <= data.panjang) {
            cout << "Masukkan nama baru: ";
            cin.ignore();
            getline(cin, data.penghuni[index - 1].nama);
            cout << "Masukkan umur baru: ";
            getline(cin, data.penghuni[index - 1].umur);
            cout << "Masukkan nomor kamar baru: ";
            getline(cin, data.penghuni[index - 1].kamar);
            cout << "Kamar berhasil diubah" << endl;
        } else {
            cout << "Nomor kamar tidak valid" << endl;
        }
    }
}

void hapusData(Datakost &data) {
    if (data.panjang == 0) {
        cout << "Belum ada kamar untuk dihapus.\n" << endl;
    } else {
        tampilkanData(data);
        int index;
        cout << "Masukkan nomor kamar yang akan dihapus: ";
        cin >> index;
        if (index > 0 && index <= data.panjang) {
            for (int i = index - 1; i < data.panjang - 1; i++) {
                data.penghuni[i] = data.penghuni[i + 1];
            }
        }
    }
}

```

```

        }
        data.panjang--;
        cout << "Kamar berhasil dihapus\n" << endl;
    } else {
        cout << "Nomor kamar tidak valid\n" << endl;
    }
}
}
}

```

#### Lanjutan program Login dan Menu

```

int main() {
    Datakost data;
    string inputNama, inputNim;
    int maksimalpercobaan = 3;
    int pilihan;

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

        if (login(inputNama, inputNim)) {
            cout << "Login berhasil! Selamat datang, " << inputNama << endl;
            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/penghuni" << endl;
                cout << "5. Keluar Dari Program" << endl;
                cout << "Pilih menu: ";
                cin >> pilihan;

                switch (pilihan) {
                    case 1: tampilkanData(data); break;
                    case 2: tambahData(data); break;
                    case 3: ubahData(data); break;
                    case 4: hapusData(data); break;
                    case 5: 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

```

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/penghuni
5. Keluar Dari Program

Pilih menu: 2

Masukkan nama: rasyid

Masukkan umur: 19

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/penghuni
5. Keluar Dari Program

Pilih menu: 1

No	Nama	Umur	Kamar
1	rasyid	19	1

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

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

Pilih menu: █

Gambar 4.1 Menu Read dan Create

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/penghuni
5. Keluar Dari Program
Pilih menu: 3
No  Nama                Umur                Kamar
-----
1   rasyid              19                  1
2   zifa                18                  2
Masukkan nomor kamar yang akan diubah: 2
Masukkan nama baru: ahnaf
Masukkan umur baru: 20
Masukkan nomor kamar baru: 3
Kamar berhasil diubah

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/penghuni
5. Keluar Dari Program
Pilih menu: 1
No  Nama                Umur                Kamar
-----
1   rasyid              19                  1
2   ahnaf               20                  3

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

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

Gambar 4.2 Menu Update

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/penghuni
5. Keluar Dari Program
Pilih menu: 4
No  Nama                umur                Kamar
-----
1   rasyid              19                  1
2   ahnaf               20                  3
Masukkan nomor kamar yang akan dihapus: 2
Kamar berhasil dihapus

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Pelanggan/penghuni
3. Ubah Data Pelanggan/penghuni
4. Hapus Data Pelanggan/penghuni
5. Keluar Dari Program
Pilih menu: 1
No  Nama                Umur                Kamar
-----
1   rasyid              19                  1

MANAJEMEN PEMESANAN KAMAR KOST PUTRA

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

Gambar 4.3 Menu Delete dan Keluar dari program



## 5. Langkah-Langkah Git pada VSCode

```
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)
        modified:   .vscode/settings.json
        modified:   post-test/post-test-apl-3/2409106042-MuhammadRasyid-PT-3.exe
        new file:   post-test/post-test-apl-4/2409106042-MuhammadRasyid-PT-4.cpp
        new file:   post-test/post-test-apl-4/2409106042-MuhammadRasyid-PT-4.exe

PS C:\Users\Public\Documents\praktikum-apl> git commit -m "program"
[main 8e85a2c] program
 4 files changed, 141 insertions(+), 1 deletion(-)
 create mode 100644 post-test/post-test-apl-4/2409106042-MuhammadRasyid-PT-4.cpp
 create mode 100644 post-test/post-test-apl-4/2409106042-MuhammadRasyid-PT-4.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: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 677.30 KiB | 6.98 MiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/MRasyid18/praktikum-apl.git
 cd5a9d8..8e85a2c  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

Langkah git dan penjelasan:

- 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.