LAPORAN PRAKTIKUM

POSTTEST 6

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Muhammad Rasyid (2409106042)

Kelas (A2’24)

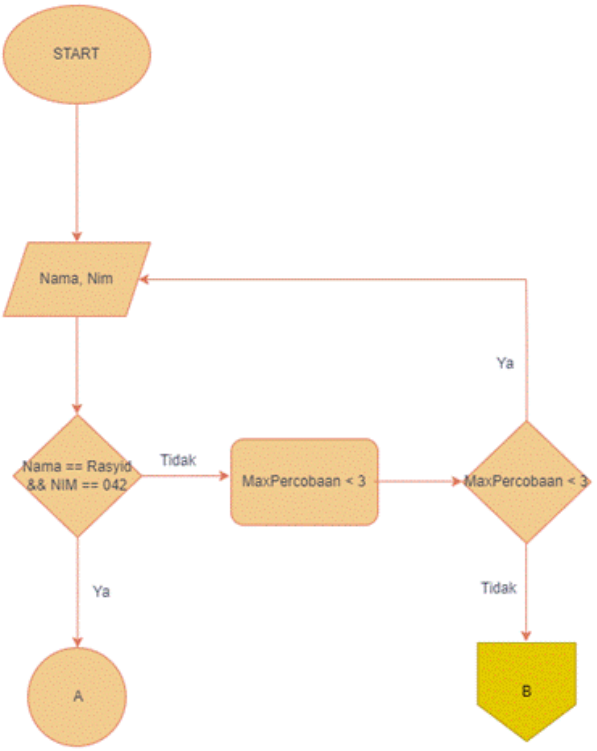
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

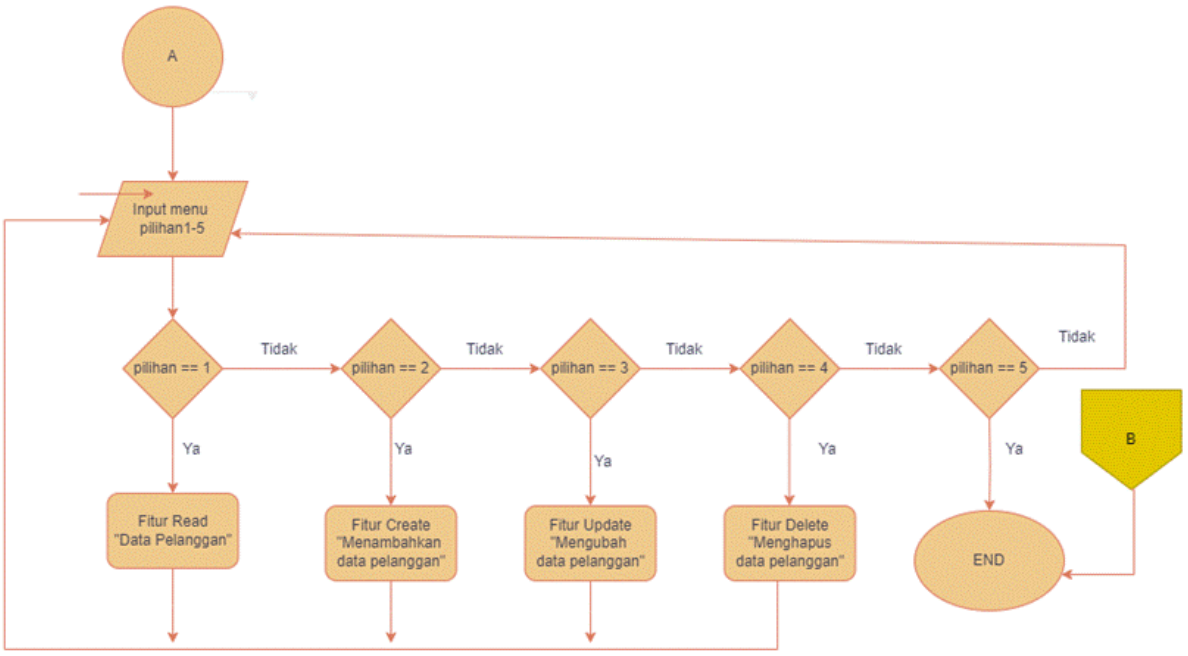
SAMARINDA

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. Pada posttest kali ini saya menambahkan fitur sorting dengan berbagai pilihan seperti sorting berdasarkan nama, umur dan nomor kamar yang akan memudahkan pemilik untuk membaca data berdasarkan sorting yang di inginkan. 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;
};

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

Fungsi dari menu menu pilihan

```
// Fungsi menampilkan data
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;
        }
    }
}

// Fungsi menambah data
```

```
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;
    }
}

// Fungsi mengubah data
void ubahData(Datakost *data) {
    if (data->panjang == 0) {
        cout << "Belum ada kamar untuk diubah." << endl;
    } else {
        tampilkanData(data);
        int index;
        cout << "Masukkan nomor data 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 << "Data berhasil diubah" << endl;
        } else {
            cout << "Nomor tidak valid" << endl;
        }
    }
}

// Fungsi menghapus data
void hapusData(Datakost *data) {
    if (data->panjang == 0) {
        cout << "Belum ada kamar untuk dihapus.\n" << endl;
    } else {
        tampilkanData(data);
        int index;
```

```
        cout << "Masukkan nomor data 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 << "Data berhasil dihapus\n" << endl;
        } else {
            cout << "Nomor tidak valid\n" << endl;
        }
    }
}

// Fungsi reset panjang
void resetPanjang(Datakost &data) {
    data.panjang = 0;
    cout << "Semua data penghuni telah direset!" << endl;
}

// Sorting Nama menggunakan SELECTION SORT (Descending)
void sortNamaDescending(Datakost *data) {
    for (int i = 0; i < data->panjang - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < data->panjang; j++) {
            if (data->penghuni[j].nama > data->penghuni[maxIdx].nama) {
                maxIdx = j;
            }
        }
        if (maxIdx != i) {
            swap(data->penghuni[i], data->penghuni[maxIdx]);
        }
    }
    cout << "Data telah diurutkan berdasarkan Nama (Z-A) menggunakan
Selection Sort." << endl;
}

// Sorting Umur menggunakan INSERTION SORT (Ascending)
void sortUmurAscending(Datakost *data) {
    for (int i = 1; i < data->panjang; i++) {
        Penghuni key = data->penghuni[i];
        int j = i - 1;
        while (j >= 0 && stoi(data->penghuni[j].umur) > stoi(key.umur)) {
            data->penghuni[j + 1] = data->penghuni[j];
            j--;
        }
        data->penghuni[j + 1] = key;
    }
```

```
    cout << "Data telah diurutkan berdasarkan Umur (Ascending) menggunakan
Insertion Sort." << endl;
}

// Sorting Kamar menggunakan BUBBLE SORT (Ascending)
void sortKamarAscending(Datakost *data) {
    for (int i = 0; i < data->panjang - 1; i++) {
        for (int j = 0; j < data->panjang - i - 1; j++) {
            if (data->penghuni[j].kamar > data->penghuni[j + 1].kamar) {
                swap(data->penghuni[j], data->penghuni[j + 1]);
            }
        }
    }
    cout << "Data telah diurutkan berdasarkan Nomor Kamar (Ascending)
menggunakan Bubble Sort." << endl;
}

// Main Program
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 Penghuni" << endl;
                cout << "3. Ubah Data Penghuni" << endl;
                cout << "4. Hapus Data Penghuni" << endl;
                cout << "5. Reset Semua Data" << endl;
                cout << "6. Sorting Nama (Descending)" << endl;
                cout << "7. Sorting Umur (Ascending)" << endl;
                cout << "8. Sorting Kamar (Ascending)" << endl;
                cout << "9. 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: resetPanjang(data); break;
                    case 6:
                        sortNamaDescending(&data);
                        tampilkanData(&data);
                        break;
                    case 7:
                        sortUmurAscending(&data);
                        tampilkanData(&data);
                        break;
                    case 8:
                        sortKamarAscending(&data);
                        tampilkanData(&data);
                        break;
                    case 9: cout << "Keluar dari program" << endl; break;
                    default: cout << "Pilihan tidak valid" << endl; break;
                }
            } while (pilihan != 9);

            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

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Penghuni
3. Ubah Data Penghuni
4. Hapus Data Penghuni
5. Reset Semua Data
6. Sorting Nama (Descending)
7. Sorting Umur (Ascending)
8. Sorting Kamar (Ascending)
9. Keluar Dari Program
Pilih menu: 1
No  Nama                Umur                Kamar
-----------------------------------------------------
1   rasyid              18                  1
2   ahnaps              19                  2
3   zifa                20                  3
```

Gambar 4.1 Sebelum di sorting

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Penghuni
3. Ubah Data Penghuni
4. Hapus Data Penghuni
5. Reset Semua Data
6. Sorting Nama (Descending)
7. Sorting Umur (Ascending)
8. Sorting Kamar (Ascending)
9. Keluar Dari Program
Pilih menu: 6
Data telah diurutkan berdasarkan Nama (Z-A) menggunakan Selection Sort.
No  Nama                Umur                Kamar
-----------------------------------------------------
1   zifa                20                  3
2   rasyid              18                  1
3   ahnaps              19                  2
```

Gambar 4.2 Menu sorting berdasarkan Nama secara Descending menggunakan Selection Sort

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Penghuni
3. Ubah Data Penghuni
4. Hapus Data Penghuni
5. Reset Semua Data
6. Sorting Nama (Descending)
7. Sorting Umur (Ascending)
8. Sorting Kamar (Ascending)
9. Keluar Dari Program
Pilih menu: 7
Data telah diurutkan berdasarkan Umur (Ascending) menggunakan Insertion Sort.
No  Nama                Umur                 Kamar
-------------------------------------------------
1   rasyid              18                   1
2   ahnaps              19                   2
3   zifa                20                   3
```

Gambar 4.3 Menu sorting berdasarkan Umur secara Ascending menggunakan Insertion Sort

```
MANAJEMEN PEMESANAN KAMAR KOST PUTRA

1. Tampilkan Data Kost
2. Tambah Data Penghuni
3. Ubah Data Penghuni
4. Hapus Data Penghuni
5. Reset Semua Data
6. Sorting Nama (Descending)
7. Sorting Umur (Ascending)
8. Sorting Kamar (Ascending)
9. Keluar Dari Program
Pilih menu: 8
Data telah diurutkan berdasarkan Nomor Kamar (Ascending) menggunakan Bubble Sort.
No  Nama                Umur                 Kamar
-------------------------------------------------
1   rasyid              18                   1
2   ahnaps              19                   2
3   zifa                20                   3
```

Gambar 4.3 Menu sorting berdasarkan Nomor Kamar secara Ascending menggunakan Bubble Sort

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)
        new file:   post-test/post-test-apl-6/2409106042-MuhammadRasyid-PT-6.cpp
        new file:   post-test/post-test-apl-6/2409106042-MuhammadRasyid-PT-6.exe

PS C:\Users\Public\Documents\praktikum-apl> git commit -m "program"
[main dd51df7] program
 2 files changed, 210 insertions(+)
 create mode 100644 post-test/post-test-apl-6/2409106042-MuhammadRasyid-PT-6.cpp
 create mode 100644 post-test/post-test-apl-6/2409106042-MuhammadRasyid-PT-6.exe
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% (6/6), done.
Writing objects: 100% (6/6), 679.33 KiB | 8.18 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/MRasyid18/praktikum-apl.git
   de74425..dd51df7  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 repositori.

• 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 defaulf.