

LAPORAN UJIAN PRAKTIKUM 1



Nama :

Fahmi Hasan Asagaf (2311104074)

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

Code

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

// Struktur data mahasiswa
struct Mahasiswa
{
    string nama;
    string nim;
    string kelas;
    float nilaiAsesmen;
    float nilaiPraktikum;
};

// Definisi elemen list
struct Element
{
    Mahasiswa data;
    Element *next;
};

// Definisi List
struct List
{
    Element *first;
};

// Fungsi membuat elemen baru
Element *newElement(Mahasiswa data)
{
    Element *newEl = new Element;
    newEl->data = data;
    newEl->next = NULL;
    return newEl;
}
```

```

// Fungsi membuat list baru
List createNewList()
{
    List l;
    l.first = NULL;
    return l;
}

// Fungsi pengecekan list kosong
bool isEmpty(List l)
{
    return l.first == NULL;
}

// Prosedur insert first
void insertFirst(List &l, Element *p)
{
    p->next = l.first;
    l.first = p;
}

// Fungsi rekursif untuk mencetak data dari elemen terakhir ke pertama
void printReverse(Element *current)
{
    if (current == NULL)
        return;

    // Rekursi ke elemen berikutnya
    printReverse(current->next);

    // Cetak data setelah rekursi kembali ke elemen ini
    cout << "Nama: " << current->data.nama
        << ", NIM: " << current->data.nim
        << ", Kelas: " << current->data.kelas
        << ", Nilai Asesmen: " << current->data.nilaiAsesmen
        << ", Nilai Praktikum: " << current->data.nilaiPraktikum << endl;
}

```

```
// Prosedur hapus mahasiswa duplikat
```

```
void hapusDuplikat(List &l)
```

```
{
```

```
    if (isEmpty(l))
```

```
        return;
```

```
    Element *current = l.first;
```

```
    while (current != NULL)
```

```
    {
```

```
        Element *checker = current;
```

```
        while (checker->next != NULL)
```

```
        {
```

```
            if (checker->next->data.nim == current->data.nim)
```

```
            {
```

```
                // Hapus elemen duplikat (elemen setelah elemen pertama yang ditemukan)
```

```
                Element *temp = checker->next;
```

```
                checker->next = temp->next;
```

```
                delete temp;
```

```
            }
```

```
            else
```

```
            {
```

```
                checker = checker->next;
```

```
            }
```

```
        }
```

```
        current = current->next;
```

```
    }
```

```
}
```

```
// Fungsi mencari mahasiswa dengan nilai asesmen tertinggi
```

```
Mahasiswa cariNilaiTertinggi(List l)
```

```
{
```

```
    if (isEmpty(l))
```

```
        throw runtime_error("List kosong");
```

```
    Mahasiswa tertinggi = l.first->data;
```

```
    Element *current = l.first;
```

```

while (current != NULL)
{
    if (current->data.nilaiAsesmen > tertinggi.nilaiAsesmen)
    {
        tertinggi = current->data;
    }
    current = current->next;
}

return tertinggi;
}

// Prosedur mencetak list secara terbalik
void printListReverse(List l)
{
    printReverse(l.first);
}

void tambahDataMahasiswa(List &l, int N)
{
    for (int i = 0; i < N; i++)
    {
        Mahasiswa mhs;
        cout << "Masukkan data mahasiswa ke-" << i + 1 << ":" << endl;
        cout << "Nama: ";
        getline(cin >> ws, mhs.nama);
        cout << "NIM: ";
        getline(cin, mhs.nim);
        cout << "Kelas: ";
        getline(cin, mhs.kelas);

        do
        {
            cout << "Nilai Asesmen (>= 0): ";
            cin >> mhs.nilaiAsesmen;
        } while (mhs.nilaiAsesmen < 0);

        do
        {
            cout << "Nilai Praktikum (>= 0): ";

```

```

cin >> mhs.nilaiPraktikum;
    } while (mhs.nilaiPraktikum < 0);

    Element *newEl = newElement(mhs);
    insertFirst(l, newEl); // Semua elemen ditambahkan di awal daftar
}
}
int main()
{
    List daftarMahasiswa = createNewList();
    int N;

    cout << "Masukkan jumlah mahasiswa: ";
    cin >> N;

    tambahDataMahasiswa(daftarMahasiswa, N);
    hapusDuplikat(daftarMahasiswa);

    cout << "\nDaftar Mahasiswa (Insert First, Cetak Terbalik):" << endl;
    printListReverse(daftarMahasiswa);

    try
    {
        Mahasiswa tertinggi = cariNilaiTertinggi(daftarMahasiswa);
        cout << "\nMahasiswa dengan Nilai Asesmen Tertinggi:" << endl;
        cout << "Nama: " << tertinggi.nama
            << ", NIM: " << tertinggi.nim
            << ", Nilai Asesmen: " << tertinggi.nilaiAsesmen << endl;
    }
    catch (const exception &e)
    {
        cout << e.what() << endl;
    }

    cout << "\nProgram dibuat oleh:" << endl;
    cout << "Nama: Fahmi Hasan Asagaf " << endl;
    cout << "NIM: 2311104074" << endl;

    return 0;
}

```

Output

```
Masukkan jumlah mahasiswa: 2
Masukkan data mahasiswa ke-1:
Nama: Fahmi
NIM: 2311104074
Kelas: SE 07 02
Nilai Asesmen (>= 0): 99
Nilai Praktikum (>= 0): 100
Masukkan data mahasiswa ke-2:
Nama: Laila
NIM: C0B023045
Kelas: SE 07 02
Nilai Asesmen (>= 0): 100
Nilai Praktikum (>= 0): 100

Daftar Mahasiswa (Insert First, Cetak Terbalik):
Nama: Fahmi, NIM: 2311104074, Kelas: SE 07 02, Nilai Asesmen: 99, Nilai Praktikum: 100
Nama: Laila , NIM: C0B023045, Kelas: SE 07 02, Nilai Asesmen: 100, Nilai Praktikum: 100

Mahasiswa dengan Nilai Asesmen Tertinggi:
Nama: Laila , NIM: C0B023045, Nilai Asesmen: 100
```

Penjelasan

Program di atas mengimplementasikan pengelolaan data mahasiswa menggunakan struktur data linked list dengan berbagai operasi seperti penambahan data, penghapusan elemen duplikat, pencarian nilai tertinggi, dan pencetakan data secara terbalik (reverse). Setiap mahasiswa direpresentasikan sebagai elemen yang menyimpan informasi seperti nama, NIM, kelas, nilai asesmen, dan nilai praktikum. Data mahasiswa diinput oleh pengguna dan ditambahkan ke list menggunakan metode *insert first*, sehingga elemen baru selalu berada di awal list.

Fitur utama program meliputi penghapusan elemen duplikat berdasarkan NIM, pencarian mahasiswa dengan nilai asesmen tertinggi, dan mencetak daftar mahasiswa secara terbalik menggunakan rekursi. Program juga dilengkapi validasi input nilai (tidak boleh negatif) untuk memastikan data valid. Di akhir, program menampilkan mahasiswa dengan nilai asesmen tertinggi serta mencetak daftar mahasiswa dengan urutan terbalik dari penambahan awal.