

**LAPORAN PRAKTIKUM
UJIAN PRAKTIKUM 1**



**Disusun Oleh:
Zivana Afra Yulianto -2211104039
SE-07-02**

**Dosen:
Wahyu Andi Saputra**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024**

UJIAN PRAKTIKUM

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 (current->data == checker->data)
            {
                checker->next = checker->next->next;
            }
            checker = checker->next;
        }
        current = current->next;
    }
}
```

```

        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: Zivana Afra Yulianto" << endl;
cout << "NIM: 2211104039" << endl;

return 0;
}

```

Screenshoot output :

```

Masukkan jumlah mahasiswa: 3
Masukkan data mahasiswa ke-1:
Nama: A
NIM: 1
Kelas: 1
Nilai Asesmen (>= 0): 1
Nilai Praktikum (>= 0): 1
Masukkan data mahasiswa ke-2:
Nama: B
NIM: 2
Kelas: 2
Nilai Asesmen (>= 0): 2
Nilai Praktikum (>= 0): 2
Masukkan data mahasiswa ke-3:
Nama: A
NIM: 1
Kelas: 1
Nilai Asesmen (>= 0): 1
Nilai Praktikum (>= 0): 1

Daftar Mahasiswa (Insert First, Cetak Terbalik):
Nama: B, NIM: 2, Kelas: 2, Nilai Asesmen: 2, Nilai Praktikum: 2
Nama: A, NIM: 1, Kelas: 1, Nilai Asesmen: 1, Nilai Praktikum: 1

Mahasiswa dengan Nilai Asesmen Tertinggi:
Nama: B, NIM: 2, Nilai Asesmen: 2

Program dibuat oleh:
Nama: Zivana Afra Yulianto
NIM: 2211104039

Process returned 0 (0x0)   execution time : 24.025 s
Press any key to continue.

```

Deskripsi :

Program ini menggunakan **linked list** untuk mengelola data mahasiswa dengan fitur:

- **Insert First:** Menambahkan data di awal list.
- **Hapus Duplikat:** Menghapus data dengan NIM yang sama.
- **Cetak Terbalik:** Menampilkan data dari input pertama ke terakhir menggunakan rekursi.
- **Cari Nilai Tertinggi:** Menemukan mahasiswa dengan nilai asesmen tertinggi.

Pengguna memasukkan data mahasiswa, program mencetak daftar mahasiswa (urut input) dan informasi mahasiswa dengan nilai tertinggi. Identitas pembuat juga ditampilkan di akhir