

LAPORAN PRAKTIKUM
“Asesmen Praktikum CLO 1”



Disusun Oleh:
Dewi Atika Muthi -2211104042
SE-07-02

Dosen:
Wahyu Andi Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Soal

Seorang dosen MK STD ingin mengimplementasikan struktur data linked list untuk menyimpan nilai mahasiswa. Berikut adalah data nilai mahasiswa yang akan disimpan: Nama mahasiswa, NIM, Kelas, Nilai Asesmen, Nilai Praktikum.

- a. Buatlah subprogram untuk menambahkan sebanyak N data baru kedalam list. List mungkin kosong.
Catatan : Mekanisme Insert First untuk NIM ganjil. Insert last untuk NIM genap.
- b. Tampilkan data mahasiswa yang memiliki nilai asesmen paling tinggi.
- c. Buatlah subprogram untuk menghapus data mahasiswa yang duplikat (Misal hapus data mahasiswa jika NIMnya duplikat).

2. Sourcecodde: main.cpp

```
#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa {
    string nama;
    string nim;
    string kelas;
    float nilaiAsesmen;
    float nilaiPraktikum;
};

struct Element {
    Mahasiswa data;
    Element* next;
    Element* prev;
};

struct List {
    Element* first;
    Element* last;
};

Element* newElement(Mahasiswa data) {
    Element* newEl = new Element;
    newEl->data = data;
    newEl->next = NULL;
    newEl->prev = NULL;
    return newEl;
}

List NewList() {
    List l;
    l.first = NULL;
```

```

        l.last = NULL;
        return l;
    }

    bool isEmpty(List l) {
        return l.first == NULL;
    }

    void insertLast(List& l, Element* p) {
        if (isEmpty(l)) {
            l.first = p;
            l.last = p;
        } else {
            l.last->next = p;
            p->prev = l.last;
            l.last = p;
        }
    }

    void deleteFirst(List& l, Element*& p) {
        if (!isEmpty(l)) {
            p = l.first;
            if (l.first == l.last) {
                l.first = NULL;
                l.last = NULL;
            } else {
                l.first = l.first->next;
                l.first->prev = NULL;
            }
            p->next = NULL;
        }
    }

    void deleteLast(List& l, Element*& p) {
        if (!isEmpty(l)) {
            p = l.last;
            if (l.first == l.last) {
                l.first = NULL;
                l.last = NULL;
            } else {
                l.last = l.last->prev;
                l.last->next = NULL;
            }
            p->prev = NULL;
        }
    }

    int length(List l) {
        int count = 0;
        Element* current = l.first;
        while (current != NULL) {
            count++;
            current = current->next;
        }
        return count;
    }
}

```

```

void printList(List l) {
    Element* current = l.first;
    int nomor = 1;
    while (current != NULL) {
        cout << nomor << ". ";
        cout << "Nama: " << current->data.nama
            << ", NIM: " << current->data.nim
            << ", Kelas: " << current->data.kelas
            << ", Nilai Asesmen: " << current->data.nilaiAsesmen
            << ", Nilai Praktikum: " << current->data.nilaiPraktikum <<
endl;
        current = current->next;
        nomor++;
    }
}

void tambahDataMahasiswa(List& l, int N) {
    for (int i = 0; i < N; i++) {
        Mahasiswa mhs;
        cout << endl;
        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);
        cout << "Nilai Asesmen: "; cin >> mhs.nilaiAsesmen;
        cout << "Nilai Praktikum: "; cin >> mhs.nilaiPraktikum;

        Element* newEl = newElement(mhs);

        insertLast(l, newEl);
    }
}

void hapusDuplikat(List& l) {
    if (isEmpty(l)) return;

    Element* current = l.first;
    while (current != NULL) {
        Element* checker = current->next;
        while (checker != NULL) {
            if (checker->data.nim == current->data.nim) {
                Element* temp = checker;
                checker = checker->next;

                if (temp == l.last) {
                    deleteLast(l, temp);
                } else {
                    temp->prev->next = temp->next;
                    temp->next->prev = temp->prev;
                    delete temp;
                }
            } else {
                checker = checker->next;
            }
        }
        current = current->next;
    }
}

```

```

    }
}

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;
}

int main()
{
    cout << "Nama: Dewi Atika Muthi \nNim: 2211104042" << endl << endl;

    List daftarMahasiswa = NewList();

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

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

    cout << "\nDaftar Mahasiswa:" << endl;
    printList(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;
    }

    return 0;
}

```

3. **Penjelasan code**

Program ini dimulai dengan menginisialisasi daftar kosong untuk menyimpan data mahasiswa menggunakan *doubly linked list* (nim genap). Pengguna diminta untuk memasukkan data mahasiswa, yang kemudian disimpan dalam daftar (dengan `insertLast`). Setelah itu, program menghapus entri duplikat berdasarkan NIM dan menampilkan daftar mahasiswa yang unik (dengan `hapusDuplikat`). Selanjutnya, program mencari mahasiswa dengan nilai asesmen tertinggi dan menampilkan hasilnya(`cariNilaiTertinggi`) . Program ini memanfaatkan fungsi untuk menambahkan, menghapus, dan mencari elemen dalam daftar.

untuk menampilkan seluruh daftar menggunakan (`printList(daftarMahasiswa)`).