LAPORAN UJIAN PRAKTIKUM

Nama :

Dimastian Aji Wibowo (2311104058)

Dosen :

Wahyu Andi Saputra, S.Pd.,
M.Eng.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. PROGRAM

doublelinkedlist.h

```h
//Dimastian Aji WIbowo
//2311104058
#ifndef DOUBLELINKEDLIST_H
#define DOUBLELINKEDLIST_H
#include <string>
using namespace std;

struct Mahasiswa{
    string nama;
    string nim;
    string kelas;
    int nilaiAsesmen;
    int nilaiPraktikum;
    Mahasiswa* next;
    Mahasiswa* prev;
};
Mahasiswa* newElement(string nama, string nim, string kelas, int nilaiAsesmen, int nilaiPraktikum);

Mahasiswa* createNewList();

bool isEmpty(Mahasiswa* list);

void insertLast(Mahasiswa*& list, Mahasiswa* p);

void deleteFirst(Mahasiswa*& list, Mahasiswa*& p);

void deleteLast(Mahasiswa*& list, Mahasiswa*& p);

int length(Mahasiswa* list);

Mahasiswa* findElement(Mahasiswa* list, string nim);

void printList(Mahasiswa* list);

void tambahData(Mahasiswa*& list, int N);

void tampilkanNilaiTertinggi(Mahasiswa* list);

void hapusDuplikat(Mahasiswa*& list);

#endif
```

doublelinkedlist.cpp

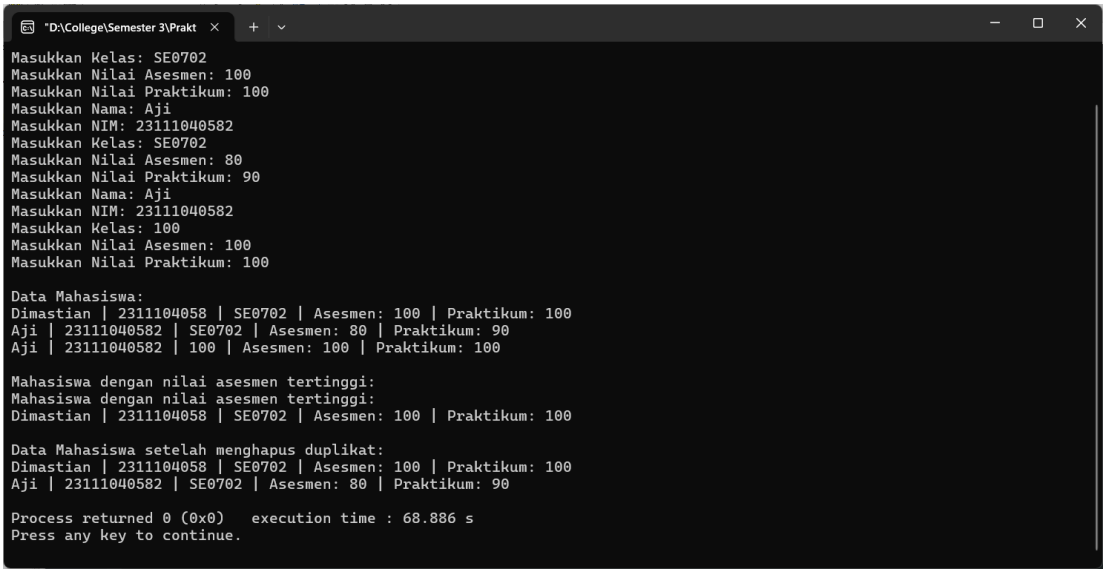
```cpp
//Dimastian Aji WIbowo
//2311104058
#include "doublelinkedlist.h"
#include <iostream>
using namespace std;

Mahasiswa* newElement(string nama, string nim, string kelas, int nilaiAsesmen, int nilaiPraktikum){
    Mahasiswa* p = new Mahasiswa;
    p->nama = nama;
    p->nim = nim;
    p->kelas = kelas;
    p->nilaiAsesmen = nilaiAsesmen;
    p->nilaiPraktikum = nilaiPraktikum;
    p->next = nullptr;
    p->prev = nullptr;
    return p;
}
Mahasiswa* createNewList(){
    return nullptr;
}
bool isEmpty(Mahasiswa* list){
    return list == nullptr;
}
void insertLast(Mahasiswa*& list, Mahasiswa* p){
    if(isEmpty(list)){
        list = p;
    }else{
        Mahasiswa* temp = list;
        while (temp->next != nullptr){
            temp = temp->next;
        }
        temp->next = p;
        p->prev = temp;
    }
}
void deleteFirst(Mahasiswa*& list, Mahasiswa*& p){
    if(!isEmpty(list)){
        p = list;
        list = list->next;
        if (list != nullptr) list->prev = nullptr;
        p->next = nullptr;
    }
}
void deleteLast(Mahasiswa*& list, Mahasiswa*& p){
    if(!isEmpty(list)){
        if(list->next == nullptr){
            p = list;
            list = nullptr;
        }else{
            Mahasiswa* temp = list;
            while(temp->next != nullptr){
                temp = temp->next;
            }
            p = temp;
            temp->prev->next = nullptr;
            temp->prev = nullptr;
        }
    }
}
int length(Mahasiswa* list){
    int count = 0;
    while (list != nullptr){
        count++;
        list = list->next;
    }
    return count;
}
Mahasiswa* findElement(Mahasiswa* list, string nim){
    while(list != nullptr){
        if (list->nim == nim) return list;
        list = list->next;
    }
    return nullptr;
}
void printList(Mahasiswa* list){
    while(list != nullptr){
        cout<<list->nama<<" | "<<list->nim<<" | "<<list->kelas
            <<" | Asesmen: "<<list->nilaiAsesmen
            <<" | Praktikum: "<<list->nilaiPraktikum<<endl;
        list = list->next;
    }
}
```

```cpp
void tambahData(Mahasiswa*& list, int N){
    cin.ignore();
    for(int i = 0; i < N; i++){
        string nama, nim, kelas;
        int nilaiAsesmen, nilaiPraktikum;

        cout<<"Masukkan Nama: ";
        getline(cin, nama);

        cout<<"Masukkan NIM: ";
        cin>>nim;

        cout<<"Masukkan Kelas: ";
        cin>>kelas;

        cout<<"Masukkan Nilai Asesmen: ";
        cin>>nilaiAsesmen;

        cout<<"Masukkan Nilai Praktikum: ";
        cin>>nilaiPraktikum;

        cin.ignore();

        Mahasiswa* p = newElement(nama, nim, kelas, nilaiAsesmen, nilaiPraktikum);
        insertLast(list, p);
    }
}
void tampilkanNilaiTertinggi(Mahasiswa* list){
    if(isEmpty(list)){
        cout<<"List kosong.\n";
        return;
    }
    Mahasiswa* max = list;
    Mahasiswa* temp = list;
    while(temp != nullptr){
        if(temp->nilaiAsesmen > max->nilaiAsesmen){
            max = temp;
        }
        temp = temp->next;
    }
    cout<<"Mahasiswa dengan nilai asesmen tertinggi:\n";
    cout<<max->nama<<" | "<<max->nim<<" | "<<max->kelas
        <<" | Asesmen: "<<max->nilaiAsesmen
        <<" | Praktikum: "<<max->nilaiPraktikum<<endl;
}
void hapusDuplikat(Mahasiswa*& list){
    if(isEmpty(list)) return;

    Mahasiswa* current = list;
    while(current != nullptr){
        Mahasiswa* runner = current->next;
        while(runner != nullptr){
            if(runner->nim == current->nim){
                Mahasiswa* temp = runner;
                if(runner->prev != nullptr)runner->prev->next = runner->next;
                if(runner->next != nullptr)runner->next->prev = runner->prev;
                runner = runner->next;
                delete temp;
            }else{
                runner = runner->next;
            }
        }
        current = current->next;
    }
}
```

main.cpp

```cpp
//Dimastian Aji Wibowo
//2311104058
#include "doublelinkedlist.h"
#include <iostream>
using namespace std;

int main(){
    Mahasiswa* list = createNewList();
    int N;

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

    tambahData(list, N);

    cout<<"\nData Mahasiswa:\n";
    printList(list);

    cout<<"\nMahasiswa dengan nilai asesmen tertinggi:\n";
    tampilkanNilaiTertinggi(list);

    hapusDuplikat(list);
    cout<<"\nData Mahasiswa setelah menghapus duplikat:\n";
    printList(list);

    return 0;
}
```

II. OUTPUT


"D:\College\Semester 3\Prakt
Masukkan Kelas: SE0702
Masukkan Nilai Asesmen: 100
Masukkan Nilai Praktikum: 100
Masukkan Nama: Aji
Masukkan NIM: 23111040582
Masukkan Kelas: SE0702
Masukkan Nilai Asesmen: 80
Masukkan Nilai Praktikum: 90
Masukkan Nama: Aji
Masukkan NIM: 23111040582
Masukkan Kelas: 100
Masukkan Nilai Asesmen: 100
Masukkan Nilai Praktikum: 100
Data Mahasiswa:
Dimastian | 2311104058 | SE0702 | Asesmen: 100 | Praktikum: 100
Aji | 23111040582 | SE0702 | Asesmen: 80 | Praktikum: 90
Aji | 23111040582 | 100 | Asesmen: 100 | Praktikum: 100
Mahasiswa dengan nilai asesmen tertinggi:
Mahasiswa dengan nilai asesmen tertinggi:
Dimastian | 2311104058 | SE0702 | Asesmen: 100 | Praktikum: 100
Data Mahasiswa setelah menghapus duplikat:
Dimastian | 2311104058 | SE0702 | Asesmen: 100 | Praktikum: 100
Aji | 23111040582 | SE0702 | Asesmen: 80 | Praktikum: 90
Process returned 0 (0x0) execution time : 68.886 s
Press any key to continue.