Nama : Rizaldy Aulia Rachman

NIM : 2311104051

Kelas : S1SE-07-02

Praktikum Struktur Data

Asesmen Praktikum CLO 1

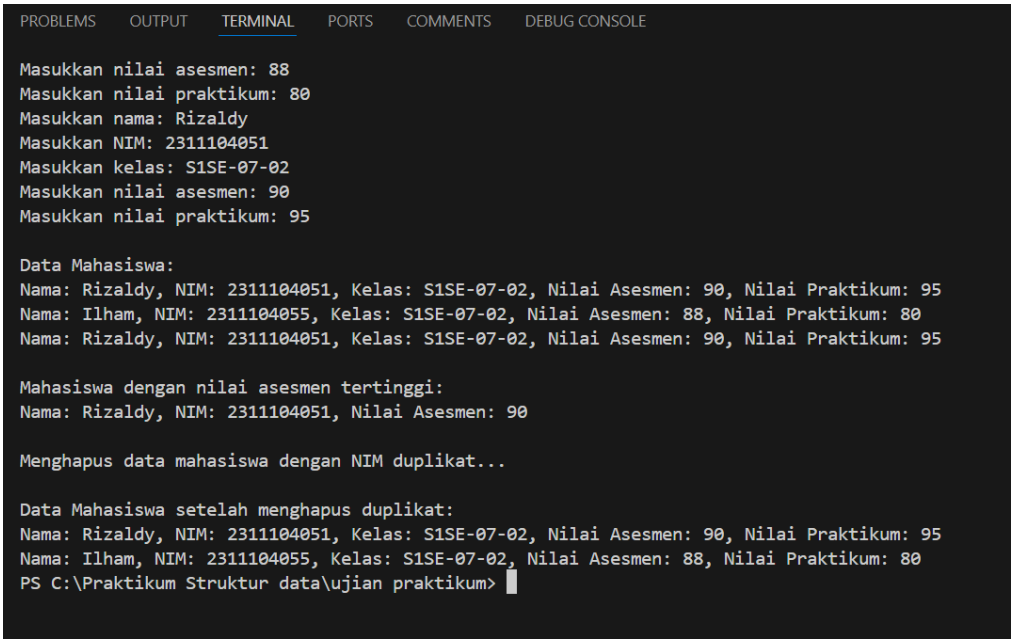
1. linked_list.h
   Code:

```cpp
#ifndef LINKED_LIST_H
#define LINKED_LIST_H

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


struct Mahasiswa {
    string name;
    string nim;
    string class_name;
    float assessment_score;
    float practical_score;
};

struct Node {
    Mahasiswa data;
    Node* next;
};

struct List {
    Node* head;
};

Node* newElement(Mahasiswa data) {
    Node* new_node = new Node;
    new_node->data = data;
    new_node->next = nullptr;
    return new_node;
}

List createNewList() {
    List list;
    list.head = nullptr;
    return list;
}

bool isEmpty(List list) {
    return list.head == nullptr;
}

void insertFirst(List& list, Node* node) {
    node->next = list.head;
    list.head = node;
}

void insertLast(List& list, Node* node) {
    if (isEmpty(list)) {
        list.head = node;
    } else {
        Node* temp = list.head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = node;
    }
}

void printList(List list) {
    if (isEmpty(list)) {
        cout << "List kosong." << endl;
        return;
    }

    Node* temp = list.head;
    while (temp != nullptr) {
        cout << "Nama: " << temp->data.name
             << ", NIM: " << temp->data.nim
             << ", Kelas: " << temp->data.class_name
             << ", Nilai Asesmen: " << temp->data.assessment_score
             << ", Nilai Praktikum: " << temp->data.practical_score
             << endl;
        temp = temp->next;
    }
}

Node* findHighestAssessment(List list) {
    if (isEmpty(list)) return nullptr;

    Node* temp = list.head;
    Node* highest = temp;
    while (temp != nullptr) {
        if (temp->data.assessment_score > highest->data.assessment_score) {
            highest = temp;
        }
        temp = temp->next;
    }
    return highest;
}

void removeDuplicates(List& list) {
    if (isEmpty(list)) return;

    Node* current = list.head;
    while (current != nullptr) {
        Node* temp = current;
        while (temp->next != nullptr) {
            if (temp->next->data.nim == current->data.nim) {
                Node* duplicate = temp->next;
                temp->next = duplicate->next;
                delete duplicate;
            } else {
                temp = temp->next;
            }
        }
        current = current->next;
    }
}

#endif
```

2. main.cpp
   Code:

```cpp
#include "linked_list.h"

int main() {
    List mahasiswaList = createNewList();

    int n;
    cout << "Masukkan jumlah data mahasiswa: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        Mahasiswa mahasiswa;
        cin.ignore();
        cout << "Masukkan nama: ";
        getline(cin, mahasiswa.name);
        cout << "Masukkan NIM: ";
        cin >> mahasiswa.nim;
        while (mahasiswa.nim.length() != 10 || !isdigit(mahasiswa.nim[0])) {
            cout << "NIM harus berupa 10 digit angka. Masukkan ulang: ";
            cin >> mahasiswa.nim;
        }
        cin.ignore();
        cout << "Masukkan kelas: ";
        getline(cin, mahasiswa.class_name);
        cout << "Masukkan nilai asesmen: ";
        cin >> mahasiswa.assessment_score;
        cout << "Masukkan nilai praktikum: ";
        cin >> mahasiswa.practical_score;

        Node* new_node = newElement(mahasiswa);
        if ((mahasiswa.nim[mahasiswa.nim.length() - 1] - '0') % 2 == 0) {
            insertLast(mahasiswaList, new_node);
        } else {
            insertFirst(mahasiswaList, new_node);
        }
    }

    cout << "\nData Mahasiswa:" << endl;
    printList(mahasiswaList);

    Node* highest = findHighestAssessment(mahasiswaList);
    if (highest != nullptr) {
        cout << "\nMahasiswa dengan nilai asesmen tertinggi:" << endl;
        cout << "Nama: " << highest->data.name
             << ", NIM: " << highest->data.nim
             << ", Nilai Asesmen: " << highest->data.assessment_score
             << endl;
    }

    cout << "\nMenghapus data mahasiswa dengan NIM duplikat..." << endl;
    removeDuplicates(mahasiswaList);

    cout << "\nData Mahasiswa setelah menghapus duplikat:" << endl;
    printList(mahasiswaList);

    return 0;
}
```

   Output:


PROBLEMS
OUTPUT
TERMINAL
PORTS
COMMENTS
DEBUG CONSOLE
Masukkan nilai asesmen: 88
Masukkan nilai praktikum: 80
Masukkan nama: Rizaldy
Masukkan NIM: 2311104051
Masukkan kelas: S1SE-07-02
Masukkan nilai asesmen: 90
Masukkan nilai praktikum: 95
Data Mahasiswa:
Nama: Rizaldy, NIM: 2311104051, Kelas: S1SE-07-02, Nilai Asesmen: 90, Nilai Praktikum: 95
Nama: Ilham, NIM: 2311104055, Kelas: S1SE-07-02, Nilai Asesmen: 88, Nilai Praktikum: 80
Nama: Rizaldy, NIM: 2311104051, Kelas: S1SE-07-02, Nilai Asesmen: 90, Nilai Praktikum: 95
Mahasiswa dengan nilai asesmen tertinggi:
Nama: Rizaldy, NIM: 2311104051, Nilai Asesmen: 90
Menghapus data mahasiswa dengan NIM duplikat...
Data Mahasiswa setelah menghapus duplikat:
Nama: Rizaldy, NIM: 2311104051, Kelas: S1SE-07-02, Nilai Asesmen: 90, Nilai Praktikum: 95
Nama: Ilham, NIM: 2311104055, Kelas: S1SE-07-02, Nilai Asesmen: 88, Nilai Praktikum: 80
PS C:\Praktikum Struktur data\ujian praktikum>