

**LAPORAN PRAKTIKUM  
“UJIAN PRAKTIKUM”**



**Disusun Oleh:  
Satria Ariq Adelard Dompas -  
2211104033 S1SE-07-2**

**Assisten Praktikum :  
Aldi Putra  
Andini Nur Hidayah**

**Dosen :  
Wahyu Andi Saputra, S.Pd., M.Eng**

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

## A. Program

### Source Code

Main.cpp :

```
//SATRIA ARIQ ADELARD DOMPAS
//2211104033
//SE 07 02

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

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

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

struct List {
    Node* first;
};

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

List newList() {
    List l;
    l.first = nullptr;
    return l;
}

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

```

void insertFirst(List& l, Node* p) {
    p->next = l.first;
    l.first = p;
}

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

void printList(List l) {
    Node* temp = l.first;
    while (temp != nullptr) {
        cout << "Nama: " << temp->data.nama
            << ", NIM: " << temp->data.nim
            << ", Kelas: " << temp->data.kelas
            << ", Nilai Assesment: " << temp->data.nilaiAsesmen
            << ", Nilai Praktikum: " << temp->data.nilaiPraktikum << endl;
        temp = temp->next;
    }
}

void addMahasiswa(List& l, int N) {
    for (int i = 0; i < N; ++i) {
        Mahasiswa m;
        cout << "Masukkan Nama: "; cin >> m.nama;
        cout << "Masukkan NIM: "; cin >> m.nim;
        cout << "Masukkan Kelas: "; cin >> m.kelas;
        cout << "Masukkan Nilai Assesment: "; cin >> m.nilaiAsesmen;
        cout << "Masukkan Nilai Praktikum: "; cin >> m.nilaiPraktikum;

        Node* newNode = newElement(m);
        if (stoi(m.nim) % 2 == 0) {
            insertLast(l, newNode);
        } else {
            insertFirst(l, newNode);
        }
    }
}

void findHighestAsesmen(List l) {
    if (isEmpty(l)) {

```

```

        cout << "List tidak ada!" << endl;
        return;
    }

    Node* temp = l.first;
    Node* highest = temp;

    while (temp != nullptr) {
        if (temp->data.nilaiAsesmen > highest->data.nilaiAsesmen) {
            highest = temp;
        }
        temp = temp->next;
    }

    cout << "Mahasiswa dengan nilai Assesment Tinggi:" << endl;
    cout << "Nama: " << highest->data.nama
        << ", NIM: " << highest->data.nim
        << ", Nilai Assesment: " << highest->data.nilaiAsesmen << endl;
}

void deleteDuplicates(List& l) {
    Node *current = l.first, *prev = nullptr;

    while (current != nullptr) {
        Node* runner = current->next;
        prev = current;

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

int main() {
    List l = newList();
    int N;

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

    addMahasiswa(l, N);
    cout << "\nData Mahasiswa:" << endl;
}

```

```

    printList(1);

    cout << "\nMencari mahasiswa dengan Nilai Assesment Tinggi:" << endl;
    findHighestAsesmen(1);

    cout << "\nMenghapus data dengan NIM yang sama:" << endl;
    deleteDuplicates(1);
    printList(1);

    return 0;
}

```

### Output :

Main.cpp

```

Masukkan jumlah mahasiswa: 3
Masukkan Nama: budi
Masukkan NIM: 12345
Masukkan Kelas: se56
Masukkan Nilai Asesmen: 65
Masukkan Nilai Praktikum: 67
Masukkan Nama: waluyo
Masukkan NIM: 12345
Masukkan Kelas: se76
Masukkan Nilai Asesmen: 56
Masukkan Nilai Praktikum: 77
Masukkan Nama: darmoyo
Masukkan NIM: 11229
Masukkan Kelas: se45
Masukkan Nilai Asesmen: 87
Masukkan Nilai Praktikum: 78

Data Mahasiswa:
Nama: darmoyo, NIM: 11229, Kelas: se45, Nilai Asesmen: 87, Nilai Praktikum: 78
Nama: waluyo, NIM: 12345, Kelas: se76, Nilai Asesmen: 56, Nilai Praktikum: 77
Nama: budi, NIM: 12345, Kelas: se56, Nilai Asesmen: 65, Nilai Praktikum: 67

Mencari mahasiswa dengan nilai asesmen tertinggi:
Mahasiswa dengan nilai asesmen tertinggi:
Nama: darmoyo, NIM: 11229, Nilai Asesmen: 87

Menghapus data dengan NIM duplikat:
Nama: darmoyo, NIM: 11229, Kelas: se45, Nilai Asesmen: 87, Nilai Praktikum: 78
Nama: waluyo, NIM: 12345, Kelas: se76, Nilai Asesmen: 56, Nilai Praktikum: 77

```



## **B. Penjelasan Program**

Program ini adalah struktur data Single Linked List untuk menyimpan data mahasiswa yang terdiri dari nama, nim, kelas, nilai asesment, dan nilai praktikum. Program untuk menambahkan data mahasiswa ke dalam list menggunakan metode *Insert First*. Selain itu, program dapat mencetak seluruh data mahasiswa, mencari dan menampilkan mahasiswa dengan nilai asesmen tertinggi, serta menghapus data duplikat berdasarkan NIM. Program ini memastikan hanya angka yang dapat diterima sebagai NIM. Setelah data dimasukkan, pengguna dapat melihat data mahasiswa dalam list dan hasil setelah penghapusan duplikat.