Laporan Praktikum

Assesmen

Disusun Oleh :

Fauzan Rofif Ardiyanto/2211104036

SE0702

Asisten Praktikum : Aldi Putra

Andini Nur Hidayah

Dosen Pengampu :

Wahyu Andi Saputra

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

Main.cpp

```cpp
// Nama :Fauzan Rofif Ardiyanto
// Kelas : S1SE0602
//  Nim : 2211104036 //

#include "list_mahasiswa.h"

int main()
{
    List L;
    createNewList(L);

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

    for (int i = 0; i < N; i++)
    {
        infotype data;

        cout << "Masukkan Nama: ";
        cin >> data.nama;
        cout << "Masukkan NIM: ";
        cin >> data.NIM;
        cout << "Masukkan Kelas: ";
        cin >> data.kelas;
        cout << "Masukkan Nilai Ujian: ";
        cin >> data.nilaiUjian;
        cout << "Masukkan Nilai Praktikum: ";
        cin >> data.nilaiPraktikum;

        address P = newElement(data);
        insertLast(L, P);
    }

    // Cetak isi list
    cout << "Data Mahasiswa:" << endl;
    printList(L);

    printHighestUjian(L);
    removeDuplicateNIM(L);
    cout << "Data Mahasiswa setelah menghapus duplikat:" << endl;
    printList(L);

    return 0;
}
```

List.cpp

```cpp
#include "list_mahasiswa.h"

void createNewList(List &L)
{
    L.first = nullptr;
    L.last = nullptr;
}

bool isEmpty(List L)
{
    return (L.first == nullptr && L.last == nullptr);
}

address newElement(infotype data)
{
    address P = new Elemen;
    P->info = data;
    P->next = nullptr;
    P->prev = nullptr;
    return P;
}

void insertLast(List &L, address P)
{
    if (isEmpty(L))
    {
        L.first = P;
        L.last = P;
    }
    else
    {
        P->prev = L.last;
        L.last->next = P;
        L.last = P;
    }
}

void printList(List L)
{
    if (isEmpty(L))
    {
        cout << "List kosong!" << endl;
    }
    else
    {
        address P = L.first;
        while (P != nullptr)
        {
            cout << "Nama: " << P->info.nama
                 << ", NIM: " << P->info.NIM
                 << ", Kelas: " << P->info.kelas
                 << ", Nilai Ujian: " << P->info.nilaiUjian
                 << ", Nilai Praktikum: " << P->info.nilaiPraktikum << endl;
            P = P->next;
        }
    }
}
```

```cpp
void printHighestUjian(List L)
{
    if (isEmpty(L))
    {
        cout << "List kosong!" << endl;
        return;
    }

    address P = L.first;
    address highest = P;

    while (P != nullptr)
    {
        if (P->info.nilaiUjian > highest->info.nilaiUjian)
        {
            highest = P;
        }
        P = P->next;
    }

    cout << "Mahasiswa dengan Nilai Ujian Tertinggi:" << endl;
    cout << "Nama: " << highest->info.nama
         << ", NIM: " << highest->info.NIM
         << ", Kelas: " << highest->info.kelas
         << ", Nilai Ujian: " << highest->info.nilaiUjian
         << ", Nilai Praktikum: " << highest->info.nilaiPraktikum << endl;
}

void removeDuplicateNIM(List & L)
{
    if (isEmpty(L))
    {
        return;
    }

    address P = L.first;
    while (P != nullptr)
    {
        address Q = P->next;
        while (Q != nullptr)
        {
            if (Q->info.NIM == P->info.NIM)
            {
                address duplicate = Q;
                if (Q->prev != nullptr)
                {
                    Q->prev->next = Q->next;
                }
                if (Q->next != nullptr)
                {
                    Q->next->prev = Q->prev;
                }
                if (Q == L.last)
                {
                    L.last = Q->prev;
                }
                Q = Q->next;
                delete duplicate;
            }
            else
            {
                Q = Q->next;
            }
        }
        P = P->next;
    }
}
```

list_mahasiswa.h

```h
#ifndef LIST_H
#define LIST_H

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

struct Mahasiswa
{
    string nama;
    int NIM;
    string kelas;
    float nilaiUjian;
    float nilaiPraktikum;
};

typedef Mahasiswa infotype;
typedef struct Elemen *address;

struct Elemen
{
    infotype info;
    address next;
    address prev;
};

struct List
{
    address first;
    address last;
};

void createNewList(List &L);
bool isEmpty(List L);
address newElement(infotype data);
void printList(List L);
void insertLast(List &L, address P);
void printHighestUjian(List L);
void removeDuplicateNIM(List &L);

#endif
```

Output :

```
 g++ main.cpp list.cpp -o oke
PS C:\Users\LENOVO\Documents\ITTP\TUGAS SEMESTER 5 ITTP\Praktikum STD\Prak_Git>
 ./oke
Masukkan jumlah mahasiswa: 2
Masukkan Nama: fauzan
Masukkan NIM: 36
Masukkan Kelas: 07
Masukkan Nilai Ujian: 100
Masukkan Nilai Praktikum: 90
Masukkan Nama: ziva
Masukkan NIM: 39
Masukkan Kelas: 07
Masukkan Nilai Ujian: 98
Masukkan Nilai Praktikum: 98
Data Mahasiswa:
Nama: fauzan, NIM: 36, Kelas: 07, Nilai Ujian: 100, Nilai Praktikum: 90
Nama: ziva, NIM: 39, Kelas: 07, Nilai Ujian: 98, Nilai Praktikum: 98
Mahasiswa dengan Nilai Ujian Tertinggi:
Nama: fauzan, NIM: 36, Kelas: 07, Nilai Ujian: 100, Nilai Praktikum: 90
Data Mahasiswa setelah menghapus duplikat:
Nama: fauzan, NIM: 36, Kelas: 07, Nilai Ujian: 100, Nilai Praktikum: 90
Nama: ziva, NIM: 39, Kelas: 07, Nilai Ujian: 98, Nilai Praktikum: 98
PS C:\Users\LENOVO\Documents\ITTP\TUGAS SEMESTER 5 ITTP\Praktikum STD\Prak_Git>
```