LAPORAN PRAKTIKUM
MODUL 7
“UJIAN PRAKTIKUM 1”

Disusun Oleh:

Tiurma Grace Angelina 2311104042

SE-07-02

Dosen :

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

Code:

ujianpraktikum1.cpp > main()

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

using namespace std;


struct Mahasiswa {
    string nama;
    int NIM;
    string kelas;
    int nilaiAsesmen;
    int nilaiPraktikum;
};

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

struct DoubleLinkedList {
    Node* head;
    Node* tail;
};

DoubleLinkedList newList() {
    DoubleLinkedList list;
    list.head = nullptr;
    list.tail = nullptr;
```

```
26  DoubleLinkedList newList() {
29      list.tail = nullptr;
30      return list;
31  }
32
33  Node* newElement(Mahasiswa data) {
34      Node* newNode = new Node();
35      newNode->data = data;
36      newNode->next = nullptr;
37      newNode->prev = nullptr;
38      return newNode;
39  }
40
41  bool isEmpty(DoubleLinkedList list) {
42      return list.head == nullptr;
43  }
44
45  void insertLast(DoubleLinkedList &list, Node* newNode) {
46      if (isEmpty(list)) {
47          list.head = list.tail = newNode;
48      } else {
49          newNode->prev = list.tail;
50          list.tail->next = newNode;
51          list.tail = newNode;
52      }
53  }
54
55  void printList(DoubleLinkedList list) {
56      Node* current = list.head;
```

```cpp
void printList(DoubleLinkedList list) {
    Node* current = list.head;
    while (current != nullptr) {
        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;
    }
}

Mahasiswa findMaxAsesmen(DoubleLinkedList list) {
    Node* current = list.head;
    Mahasiswa maxData = current->data;
    while (current != nullptr) {
        if (current->data.nilaiAsesmen > maxData.nilaiAsesmen) {
            maxData = current->data;
        }
        current = current->next;
    }
    return maxData;
}

void removeDuplicate(DoubleLinkedList &list) {
    Node* current = list.head;
    while (current != nullptr) {
        Node* checker = current->next;
        while (checker != nullptr) {
```

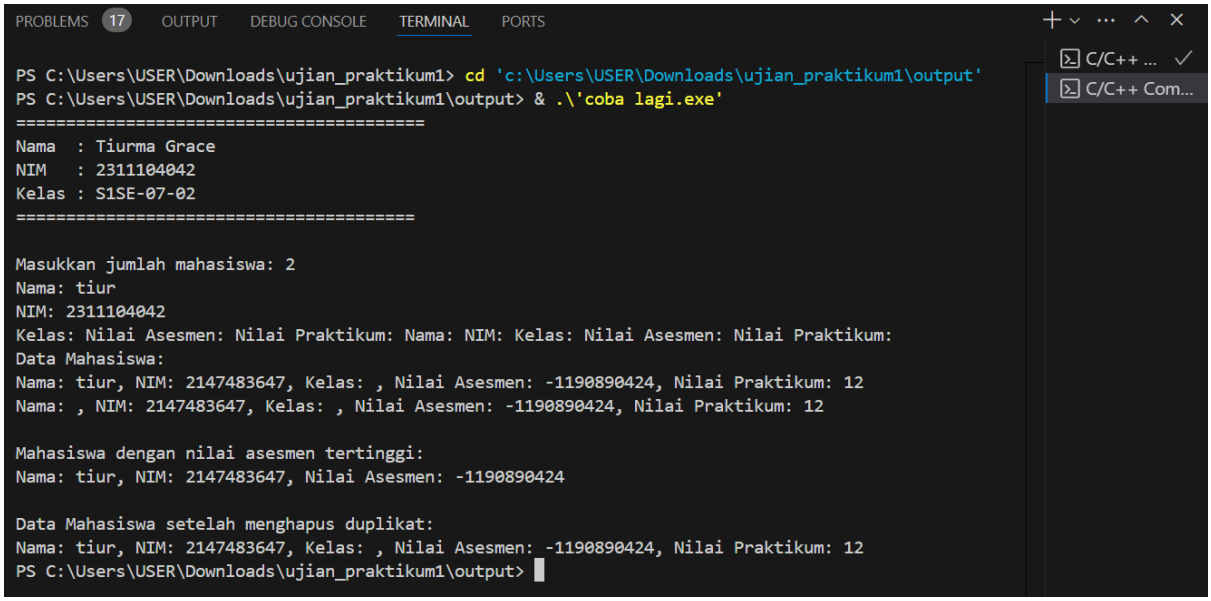
```
    while (current != nullptr) {
        Node* checker = current->next;
        while (checker != nullptr) {
            if (checker->data.NIM == current->data.NIM) {
                Node* duplicate = checker;
                if (checker->next != nullptr) {
                    checker->next->prev = checker->prev;
                }
                if (checker->prev != nullptr) {
                    checker->prev->next = checker->next;
                }
                if (checker == list.tail) {
                    list.tail = checker->prev;
                }
                checker = checker->next;
                delete duplicate;
            } else {
                checker = checker->next;
            }
        }
        current = current->next;
    }
}

int main() {
    // Menampilkan identitas
    cout << "=========================================" << endl;
```

```cpp
void removeDuplicate(DoubleLinkedList &list) {

int main() {
    // Menampilkan identitas
    cout << "=======================================" << endl;
    cout << "Nama  : Tiurma Grace" << endl;
    cout << "NIM   : 2311104042" << endl;
    cout << "Kelas : S1SE-07-02" << endl;
    cout << "=======================================" << endl;

    DoubleLinkedList list = newList();

    // Menambahkan data mahasiswa
    int N;
    cout << "\nMasukkan jumlah mahasiswa: ";
    cin >> N;

    for (int i = 0; i < N; i++) {
        Mahasiswa mhs;
        cout << "Nama: "; cin >> mhs.nama;
        cout << "NIM: "; cin >> mhs.NIM;
        cout << "Kelas: "; cin >> mhs.kelas;
        cout << "Nilai Asesmen: "; cin >> mhs.nilaiAsesmen;
        cout << "Nilai Praktikum: "; cin >> mhs.nilaiPraktikum;

        Node* newNode = newElement(mhs);
        insertLast(list, newNode); // Menggunakan Insert Last
    }
```


```cpp
int main() {
    for (int i = 0; i < N; i++) {
        insertLast(list, newNode); // Menggunakan Insert Last
    }
```


```cpp

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

    // Menampilkan mahasiswa dengan nilai asesmen tertinggi
    Mahasiswa maxAsesmen = findMaxAsesmen(list);
    cout << "\nMahasiswa dengan nilai asesmen tertinggi:\n";
    cout << "Nama: " << maxAsesmen.nama << ", NIM: " << maxAsesmen.NIM
         << ", Nilai Asesmen: " << maxAsesmen.nilaiAsesmen << endl;

    // Menghapus data duplikat
    removeDuplicate(list);
    cout << "\nData Mahasiswa setelah menghapus duplikat:\n";
    printList(list);

    return 0;
}
```

Output:


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
C/C++ ...
C/C++ Com...
PS C:\Users\USER\Downloads\ujian_praktikum1> cd 'c:\Users\USER\Downloads\ujian_praktikum1\output'
PS C:\Users\USER\Downloads\ujian_praktikum1\output> & .\'coba lagi.exe'
==
Nama : Tiurma Grace
NIM : 2311104042
Kelas : S1SE-07-02
==
Masukkan jumlah mahasiswa: 2
Nama: tiur
NIM: 2311104042
Kelas: Nilai Asesmen: Nilai Praktikum: Nama: NIM: Kelas: Nilai Asesmen: Nilai Praktikum:
Data Mahasiswa:
Nama: tiur, NIM: 2147483647, Kelas: , Nilai Asesmen: -1190890424, Nilai Praktikum: 12
Nama: , NIM: 2147483647, Kelas: , Nilai Asesmen: -1190890424, Nilai Praktikum: 12
Mahasiswa dengan nilai asesmen tertinggi:
Nama: tiur, NIM: 2147483647, Nilai Asesmen: -1190890424
Data Mahasiswa setelah menghapus duplikat:
Nama: tiur, NIM: 2147483647, Kelas: , Nilai Asesmen: -1190890424, Nilai Praktikum: 12
PS C:\Users\USER\Downloads\ujian_praktikum1\output>