

LAPORAN PRAKTIKUM
Asesmen
Praktikum CLO 1



Disusun Oleh :
Ade Fatkhul Anam/2211104051
SE 06 02

Asisten Praktikum : Aldi
Putra
Andini Nur Hidayah

Dosen Pengampu :
Wahyu Andi Saputra

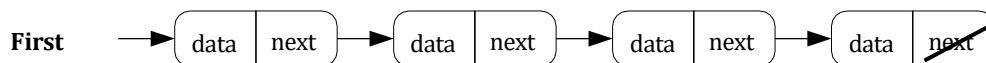
PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK FAKULTAS
INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO 2024

ATURAN ASESMEN:

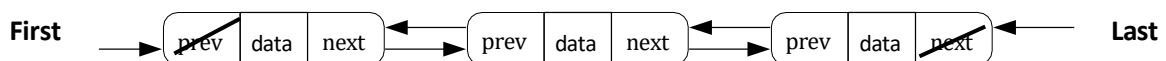
1. Individu, Closed Book (termasuk catatan kuliah, slide, rekaman atau referensi lainnya baik online ataupun offline).
2. Referensi yang boleh dibuka hanya buku modul praktikum Struktur Data yang diberikan oleh Lab IF.
3. Praktikan dilarang berdiskusi dengan sesama. Silahkan bertanya kepada asisten praktikum terkait kejelasan soal yang diberikan dan pesan error yang didapatkan.
4. Aturan lain mengikuti aturan ujian yang berlaku di Lab IF dan juga di Universitas Telkom.

FILE PRAKTIKUM (akan diberikan oleh asisten praktikum):

Diberikan sebuah file project C++ yang berisi ADT dan primitif dari **Single Linked List**



dan **Double Linked List**.



Daftar primitif dari linked list **Single/Double** adalah sebagai berikut:

function newElement(data : infotype) → address

{Mengembalikan alamat elemen list hasil alokasi, info berisi data dari masukan}

function createNewList() → List

{Mengembalikan sebuah list kosong}

function isEmpty(a: List) → **boolean**

{mengembalikan true apabila list a kosong, atau false apabila sebaliknya}

procedure insertFirst(**in/out** a:List, **in** p:address)

{I.S. Terdefinisi sebuah list a (mungkin kosong) dan sebuah pointer p yang berisi alamat elemen baru F.S. Elemen baru yang ditambahkan sebagai elemen pertama dari list a}

procedure insertAfter(**in/out** a:List, **in** x:infotype, **in** p:address)

{I.S. Terdefinisi sebuah list a (mungkin kosong), sebuah infotype x dan sebuah pointer p yang berisi alamat elemen baru F.S. Elemen baru yang ditambahkan setelah elemen list a dengan info adalah x (apabila x ditemukan di dalam list a)}

procedure insertLast(**in/out** a:List, **in** p:address)

{I.S. Terdefinisi sebuah list a (mungkin kosong) dan sebuah pointer p yang berisi alamat elemen baru F.S. Elemen baru yang ditambahkan sebagai elemen terakhir dari list a}

procedure deleteFirst(**in/out** a:List, p:address)

{I.S. Terdefinisi sebuah list a (mungkin kosong)}

F.S. Pointer p berisi alamat elemen pertama dari list a, dan elemen tersebut dihapus dari list a}

procedure deleteLast(**in/out** a:List, p:address)

{I.S. Terdefinisi sebuah list a (mungkin kosong)}

F.S. Pointer p berisi alamat elemen terakhir dari list a, dan elemen tersebut dihapus dari list a}

function length(a: List) → **integer**

{mengembalikan banyaknya elemen dari list a (list a mungkin kosong)}

function findElement(a: List, x: infotype) → address

{mengembalikan alamat dari elemen list a yang memiliki info adalah x, atau NIL apabila tidak ditemukan}

procedure printList(a: List)

{I.S. terdefinisi sebuah list a (mungkin kosong)}

F.S. menampilkan isi dari list a secara horizontal}

SOAL PRAKTIKUM:

1. **Lengkapi indentitas** Anda pada file main.cpp.

```
main.cpp > infotype > nilaiAsesmen
1 // 1. Nama: Ade Fatkhul Anam
2 // NIM: 211511051
3 // Kelas: SE07-02
```

2. **Pilih salah satu** dari 2 Linked List yang disediakan berdasarkan **NOMOR MEJA**:

- **Ganjil = Single List**
- **Genap = Double List**

```
// 2. Nomor Meja: Ganjil

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

3. **Lengkapi infotype pada ADT** sesuai soal cerita yang diberikan pada nomor 5 dan jenis list yang dipilih.

```
// 3. Lengkapi infotype pada ADT
Codeium: Refactor | Explain
struct infotype {
    string namaMahasiswa;
    string NIM;
    string kelas;
    int nilaiAsesmen;
    int nilaiPraktikum;
};

Codeium: Refactor | Explain
struct Node {
    infotype info;
    Node* next;
};

typedef Node* address;
Codeium: Refactor | Explain
struct List {
    address first;
};
```

4. **Lengkapi subprogram** berikut ini sesuai dengan soal cerita yang diberikan pada nomor 5 dan jenis list yang dipilih.

- a. Subprogram yang **wajib** diisi:

- Function newElement
- Function newList
- Procedure printList

- b. Lengkapi procedure lain yang diminta untuk menyelesaikan soal cerita pada nomor 5.
- c. Praktikan **diperbolehkan menambahkan subprogram lain** apabila diperlukan untuk menyelesaikan soal cerita pada nomor 5.
- d. Buatlah main program (main.cpp) untuk memanggil subprogram yang sudah dibuat sesuai dengan soal cerita pada nomor 5.

```
// 4. Melengkapi Subprogram
// Function newElement
Codeium: Refactor | Explain | X
address newElement(infotype data) {
    address p = new Node;
    p->info = data;
    p->next = nullptr;
    return p;
}

// Function createNewList
Codeium: Refactor | Explain | X
List createNewList() {
    List L;
    L.first = nullptr;
    return L;
}

// Function printList
Codeium: Refactor | Explain | X
void printList(List L) {
    address p = L.first;
    while (p != nullptr) {
        cout << "Nama: " << p->info.namaMahasiswa
            << ", NIM: " << p->info.NIM
            << ", Kelas: " << p->info.kelas
            << ", Nilai Asesmen: " << p->info.nilaiAsesmen
            << ", Nilai Praktikum: " << p->info.nilaiPraktikum << endl;
        p = p->next;
    }
}
```

5. SOAL

Seorang dosen MK STD ingin mengimplementasikan struktur data linked list untuk menyimpan nilai mahasiswa. Berikut adalah data nilai mahasiswa yang akan disimpan : Nama mahasiswa, NIM, Kelas, Nilai Asesmen, Nilai Praktikum.

- a. Buatlah subprogram untuk menambahkan sebanyak N data baru kedalam list. List mungkin kosong. **Catatan : Mekanisme Insert First untuk NIM ganjil. Insert last untuk NIM genap.**

```
// 5. Soal
// a. Subprogram Utama
Codeium: Refactor | Explain | X
void insertFirst(List &L, address p) {
    p->next = L.first;
    L.first = p;
}

Codeium: Refactor | Explain | Generate Function Comment | X
void insertLast(List &L, address p) {
    if (L.first == nullptr) {
        L.first = p;
    } else {
        address q = L.first;
        while (q->next != nullptr) {
            q = q->next;
        }
        q->next = p;
    }
}

// Validasi NIM sebelum di-convert menggunakan stoi
Codeium: Refactor | Explain | X
bool isNumeric(const string &str) {
    for (char c : str) {
        if (!isdigit(c)) return false;
    }
    return true;
}
}
```

```
Codeium: Refactor | Explain | Generate Function Comment | X
void tambahData(List &L, int N) {
    for (int i = 0; i < N; i++) {
        infotype data;
        cout << "Masukkan Nama: "; cin >> data.namaMahasiswa;
        cout << "Masukkan NIM: "; cin >> data.NIM;

        // Validasi input NIM
        if (!isNumeric(data.NIM)) {
            cout << "NIM tidak valid. Harus berupa angka." << endl;
            i--; // Ulangi iterasi untuk data ini
            continue;
        }

        cout << "Masukkan Kelas: "; cin >> data.kelas;
        cout << "Masukkan Nilai Asesmen: "; cin >> data.nilaiAsesmen;
        cout << "Masukkan Nilai Praktikum: "; cin >> data.nilaiPraktikum;

        address p = newElement(data);

        if (stoi(data.NIM) % 20 == 0) {
            insertLast(L, p);
        } else {
            insertFirst(L, p);
        }
    }
}
}
```

- b. Tampilkan data mahasiswa yang memiliki nilai asesmen paling tinggi.

```
// b. Data mahasiswa yang memiliki asesmen paling tinggi

Codeium: Refactor | Explain | Generate Function Comment | X
infotype cariNilaiTertinggi(List L) {
    address p = L.first;
    infotype maxInfo = p->info;

    while (p != nullptr) {
        if (p->info.nilaiAsesmen > maxInfo.nilaiAsesmen) {
            maxInfo = p->info;
        }
        p = p->next;
    }

    return maxInfo;
}
```

- c. Buatlah subprogram untuk menghapus data mahasiswa yang duplikat (Misal hapus data mahasiswa jika NIMnya duplikat).

```
// c. Buatlah subprogram untuk menghapus data mahasiswa yang duplikat

Codeium: Refactor | Explain | X
void hapusDuplikat(List &L) {
    address p = L.first;
    while (p != nullptr && p->next != nullptr) {
        address q = p;
        while (q->next != nullptr) {
            if (q->next->info.NIM == p->info.NIM) {
                address temp = q->next;
                q->next = q->next->next;
                delete temp;
            } else {
                q = q->next;
            }
        }
        p = p->next;
    }
}
```

Main program:

```
// 6. Main Program
Codeium: Refactor | Explain | X
int main() {
    List L = createNewList();

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

    tambahData(L, N);
    cout << "\nData Mahasiswa:\n";
    printList(L);

    infotype max = cariNilaiTertinggi(L);
    cout << "\nMahasiswa dengan nilai asesmen tertinggi:\n";
    cout << "Nama: " << max.namaMahasiswa
        << ", NIM: " << max.NIM
        << ", Nilai Asesmen: " << max.nilaiAsesmen << endl;

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

    return 0;
}
```

Output:

```
PS D:\PRAKTIKUM DATA STRUCTURE\LAPRAK\10_Assessment Praktikum CLO 1\output> & .\'asesmen.exe'
Masukkan jumlah data: 3
Masukkan Nama: ade
Masukkan NIM: 22111
Masukkan Kelas: se07
Masukkan Nilai Asesmen: 90
Masukkan Nilai Praktikum: 90
Masukkan Nama: candra
Masukkan NIM: 22222
Masukkan Kelas: se06
Masukkan Nilai Asesmen: 90
Masukkan Nilai Praktikum: 90
Masukkan Nama: ade
Masukkan NIM: 22111
Masukkan Kelas: se07
Masukkan Nilai Asesmen: 90
Masukkan Nilai Praktikum: 90

Data Mahasiswa:
Nama: ade, NIM: 22111, Kelas: se07, Nilai Asesmen: 90, Nilai Praktikum: 90
Nama: candra, NIM: 22222, Kelas: se06, Nilai Asesmen: 90, Nilai Praktikum: 90
Nama: ade, NIM: 22111, Kelas: se07, Nilai Asesmen: 90, Nilai Praktikum: 90

Mahasiswa dengan nilai asesmen tertinggi:
Nama: ade, NIM: 22111, Nilai Asesmen: 90

Data setelah menghapus duplikat:
Nama: ade, NIM: 22111, Kelas: se07, Nilai Asesmen: 90, Nilai Praktikum: 90
Nama: candra, NIM: 22222, Kelas: se06, Nilai Asesmen: 90, Nilai Praktikum: 90
PS D:\PRAKTIKUM DATA STRUCTURE\LAPRAK\10_Assessment Praktikum CLO 1\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\LAPRAK\10_Assessment Praktikum CLO 1\output'
PS D:\PRAKTIKUM DATA STRUCTURE\LAPRAK\10_Assessment Praktikum CLO 1\output> & .\'asesmen.exe'
Masukkan jumlah data: cd 'd:\PRAKTIKUM DATA STRUCTURE\LAPRAK\10_Assessment Praktikum CLO 1\output'
```