LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL IV
SINGLY LINKED LIST

Disusun Oleh :
NAMA : Afief Amar Purnomo
NIM : 103112430067

Dosen
FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

A. Dasar Teori

Linked list adalah struktur data linear yang terdiri dari serangkaian elemen yang disebut “node”, di mana setiap node terhubung ke node berikutnya dalam urutan tertentu. Setiap node terdiri dari dua bagian utama: data yang disimpan dan referensi ke node berikutnya dalam urutan. Ini memungkinkan alokasi memori yang dinamis dan tidak perlu alokasi memori yang bersifat statis seperti yang terjadi pada array. Linked list memiliki beberapa jenis yang masing-masing memiliki karakteristik dan kegunaan yang berbeda antara lain :

Single linked list adalah jenis paling dasar dari linked list di mana setiap node terhubung ke node berikutnya dalam urutan linear. Setiap node memiliki dua bagian: data yang disimpan dan referensi ke node berikutnya dalam urutan. Single linked list tidak memiliki referensi kembali ke node sebelumnya, sehingga navigasi hanya bisa dilakukan dari depan ke belakang. Ini merupakan struktur data yang sederhana dan efisien untuk aplikasi yang memerlukan penyisipan atau penghapusan elemen di ujung linked list.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

(file Singlylist.h)

Singlylist.h | Singlylist.cpp | main.cpp

Singlylist.h > ...

```
1   //Nama : Afief Amar Purnomo
2   //NIM : 103112430067
3
4   #ifndef SINGLYLIST_H_INCLUDED
5   #define SINGLYLIST_H_INCLUDED
6
7   #include <iostream>
8
9   #define Nil NULL
10
11  using namespace std;
12
13  typedef int infotype;
14  typedef struct Elmlist *address;
15
16  struct Elmlist {
17      infotype info;
18      address next;
19  };
20
21  struct List {
22      address first;
23  };
24
25  // Deklarasi Prosedur dan Fungsi Primitif
    Tabnine | Edit | Test | Explain | Document
26  void CreateList(List &L);
    Tabnine | Edit | Test | Explain | Document
27  address alokasi (infotype x);
    Tabnine | Edit | Test | Explain | Document
28  void dealokasi(address &P);
    Tabnine | Edit | Test | Explain | Document
29  void insertFirst(List &L, address P);
    Tabnine | Edit | Test | Explain | Document
30  void insertLast(List &L, address P);
    Tabnine | Edit | Test | Explain | Document
31  void printInfo(List L);
32
33  #endif
```

(file Singlylist.cpp)

Singlylist.cpp > ...

```cpp
//Nama : Afief Amar Purnomo
//NIM : 103112430067

#include "singlylist.h"

Tabnine | Edit | Test | Explain | Document
void CreateList(List &L){
    L.first = Nil;
}

Tabnine | Edit | Test | Explain | Document
address alokasi(infotype x){
    address P = new Elmlist;
    P->info = x;
    P->next = Nil;
    return P;
}

Tabnine | Edit | Test | Explain | Document
void dealokasi(address &P){
    delete P;
    P = Nil;
}

Tabnine | Edit | Test | Explain | Document
void insertFirst(List &L, address P){
    P->next = L.first;
    L.first = P;
}

Tabnine | Edit | Test | Explain | Document
void insertLast(List &L, address P){
```

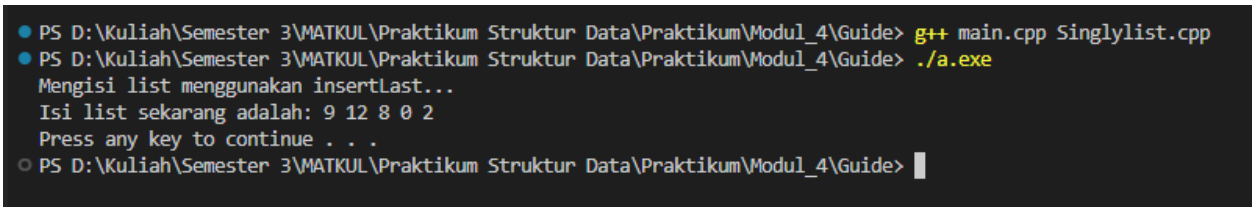
```
28      if(L.first == Nil){
29          insertFirst(L, P);
30      } else {
31          address Last = L.first;
32          while (Last->next != Nil){
33              Last = Last->next;
34          }
35          Last->next = P;
36      }
37  }
38
    Tabnine | Edit | Test | Explain | Document
39  void printInfo(List L){
40      address P = L.first;
41      if (P == NULL) {
42          cout << "List kosong" << endl;
43      } else {
44          while (P != NULL) {
45              cout << P->info << " ";
46              P = P->next;
47          }
48          cout << endl;
49      }
50  }
```

(file main.cpp)

main.cpp > main()

```cpp
//Nama : Afief Amar Purnomo
//NIM : 103112430067

#include <iostream>
#include <cstdlib>
#include "Singlylist.h"

using namespace std;

Tabnine | Edit | Test | Explain | Document
int main () {
    List L;
    address P;

    CreateList(L);

    cout << "Mengisi list menggunakan insertLast..." << endl;

    P = alokasi (9);
    insertLast(L, P);

    P = alokasi (12);
    insertLast(L, P);

    P = alokasi (8);
    insertLast(L, P);

    P = alokasi (0);
    insertLast(L, P);

     P = alokasi (2);
    insertLast(L, P);

    cout << "Isi list sekarang adalah: ";
    printInfo(L);

    system("pause");
    return 0;
}
```

Screenshots Output



Deskripsi:

Program di atas merupakan program dalam bahasa C++ yang menggunakan struktur data singly linked list untuk menyimpan dan menampilkan data secara berurutan. File singlylist.h berisi deklarasi struktur Elmlist yang menyimpan data (info) dan penunjuk ke elemen berikutnya (next), serta struktur List yang menyimpan alamat elemen pertama (first). File ini juga berisi deklarasi fungsi-fungsi seperti CreateList, insertFirst, insertLast, dan printInfo. Implementasi dari fungsi-fungsi tersebut terdapat di singlylist.cpp, di mana CreateList digunakan untuk membuat list kosong, alokasi untuk membuat node baru, insertLast untuk menambahkan data di akhir list, dan printInfo untuk menampilkan isi list. Pada file main.cpp, program utama membuat sebuah list, menambahkan beberapa data seperti 9, 12, 8, 0, dan 2 menggunakan fungsi insertLast, lalu menampilkan seluruh isi list. Program ini menunjukkan cara kerja penyimpanan data secara dinamis dengan konsep linked list di C++.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1 (soal 1)

(file playlist.h)

playlist.h

```
1   //Nama : Afief Amar Purnomo
2   //NIM : 103112430067
3
4   #ifndef PLAYLIST_H_INCLUDED
5   #define PLAYLIST_H_INCLUDED
6
7   #include <iostream>
8   #include <string>
9   using namespace std;
10
11  struct Lagu {
12      string judul;
13      string penyanyi;
14      float durasi;
15  };
16
17  struct Node {
18      Lagu info;
19      Node* next;
20  };
21
22  struct Playlist {
23      Node* first;
24  };
25
26  // Deklarasi fungsi
27  void createPlaylist(Playlist &L);
28  Node* alokasi(Lagu lagu);
29  void insertFirst(Playlist &L, Node* P);
30  void insertLast(Playlist &L, Node* P);
31  void insertAfterK(Playlist &L, Node* P, int k);
32  void deleteByJudul(Playlist &L, string judul);
33  void printPlaylist(Playlist L);
34
35  #endif
36
```

(file playlist.cpp)

playlist.cpp

```cpp
//Nama : Afief Amar Purnomo
//NIM : 103112430067

#include "Playlist.h"

void createPlaylist(Playlist &L) {
    L.first = nullptr;
}

Node* alokasi(Lagu lagu) {
    Node* P = new Node;
    P->info = lagu;
    P->next = nullptr;
    return P;
}

void insertFirst(Playlist &L, Node* P) {
    P->next = L.first;
    L.first = P;
}

void insertLast(Playlist &L, Node* P) {
    if (L.first == nullptr) {
        L.first = P;
    } else {
        Node* Q = L.first;
        while (Q->next != nullptr) {
            Q = Q->next;
        }
        Q->next = P;
    }
}
```

```
33
34  void insertAfterK(Playlist &L, Node* P, int k) {
35      if (L.first == nullptr) {
36          L.first = P;
37          return;
38      }
39
40      Node* Q = L.first;
41      int count = 1;
42      while (Q != nullptr && count < k) {
43          Q = Q->next;
44          count++;
45      }
46
47      if (Q == nullptr) {
48          cout << "Posisi ke-" << k << " tidak ditemukan. Lagu ditambahkan di akhir." << endl;
49          insertLast(L, P);
50      } else {
51          P->next = Q->next;
52          Q->next = P;
53      }
54  }
55
56  void deleteByJudul(Playlist &L, string judul) {
57      if (L.first == nullptr) {
58          cout << "Playlist kosong." << endl;
59          return;
60      }
61
62      Node *P = L.first, *prev = nullptr;
63      while (P != nullptr && P->info.judul != judul) {
64          prev = P;
65          P = P->next;
66      }
67
68      if (P == nullptr) {
69          cout << "Lagu dengan judul \"" << judul << "\" tidak ditemukan." << endl;
70          return;
71      }
72
73      if (prev == nullptr) {
74          L.first = P->next;
75      } else {
76          prev->next = P->next;
77      }
78
79      delete P;
80      cout << "Lagu \"" << judul << "\" berhasil dihapus." << endl;
81  }
82
83  void printPlaylist(Playlist L) {
84      if (L.first == nullptr) {
85          cout << "Playlist kosong." << endl;
86          return;
87      }
88
89      Node* P = L.first;
90      int i = 1;
91      cout << "\nDaftar Lagu dalam Playlist:\n";
92      while (P != nullptr) {
93          cout << i << ". Judul: " << P->info.judul
```

```
                << ", Penyanyi: " << P->info.penyanyi
                << ", Durasi: " << P->info.durasi << " menit" << endl;
            P = P->next;
            i++;
        }
        cout << endl;
    }

```

(file main.cpp)

main.cpp

```
//Nama : Afief Amar Purnomo
//NIM : 103112430067

#include <iostream>
#include "Playlist.h"
using namespace std;

int main() {
    Playlist L;
    createPlaylist(L);

    Lagu lagu1 = {"Hati-Hati di Jalan", "Tulus", 4.2};
    Lagu lagu2 = {"Sial", "Mahalini", 3.8};

    insertLast(L, alokasi(lagu1));
    insertLast(L, alokasi(lagu2));

    int pilihan;
    do {
        cout << "============================" << endl;
        cout << "     MENU PLAYLIST LAGU" << endl;
        cout << "============================" << endl;
        cout << "1. Tambah lagu di awal playlist" << endl;
        cout << "2. Tambah lagu di akhir playlist" << endl;
        cout << "3. Tambah lagu setelah playlist ke-3" << endl;
        cout << "4. Hapus lagu berdasarkan judul" << endl;
        cout << "5. Tampilkan seluruh lagu" << endl;
        cout << "0. Keluar" << endl;
        cout << "============================" << endl;
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();
```

```
33
34          if (pilihan == 1 || pilihan == 2 || pilihan == 3) {
35              Lagu laguBaru;
36              cout << "Masukkan judul lagu: ";
37              getline(cin, laguBaru.judul);
38              cout << "Masukkan nama penyanyi: ";
39              getline(cin, laguBaru.penyanyi);
40              cout << "Masukkan durasi lagu (menit): ";
41              cin >> laguBaru.durasi;
42              cin.ignore();
43
44              Node* P = alokasi(laguBaru);
45
46              if (pilihan == 1) {
47                  insertFirst(L, P);
48              } else if (pilihan == 2) {
49                  insertLast(L, P);
50              } else if (pilihan == 3) {
51                  insertAfterK(L, P, 3);
52              }
53
54              cout << "Lagu berhasil ditambahkan!" << endl << endl;
55
56          } else if (pilihan == 4) {
57              string judul;
58              cout << "Masukkan judul lagu yang ingin dihapus: ";
59              getline(cin, judul);
60              deleteByJudul(L, judul);
61
62          } else if (pilihan == 5) {
63              printPlaylist(L);
64
65          } else if (pilihan == 0) {
66              cout << "Keluar dari program..." << endl;
67          } else {
68              cout << "Pilihan tidak valid!" << endl;
69          }
70
71          cout << endl;
72      } while (pilihan != 0);
73
74      return 0;
75  }
76
```

Screenshots Output

1) Tampilkan List Lagu

```
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_4\Unguide> g++ main.cpp playlist.cpp
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_4\Unguide> ./a.exe
============================
    MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 5

Daftar Lagu dalam Playlist:
1. Judul: Hati-Hati di Jalan, Penyanyi: Tulus, Durasi: 4.2 menit
2. Judul: Sial, Penyanyi: Mahalini, Durasi: 3.8 menit
```

2) Tambah lagu di awal playlist

```
============================
    MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 1
Masukkan judul lagu: Melukis Senja
Masukkan nama penyanyi: Budi Doremi
Masukkan durasi lagu (menit): 4.0
Lagu berhasil ditambahkan!
```

```
============================
    MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 5

Daftar Lagu dalam Playlist:
1. Judul: Melukis Senja, Penyanyi: Budi Doremi, Durasi: 4 menit
2. Judul: Hati-Hati di Jalan, Penyanyi: Tulus, Durasi: 4.2 menit
3. Judul: Sial, Penyanyi: Mahalini, Durasi: 3.8 menit
```

3) Tambah lagu di akhir playlist

```
============================
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 2
Masukkan judul lagu: Aku Yang Salah
Masukkan nama penyanyi: Mahalini & Nuca
Masukkan durasi lagu (menit): 3.5
Lagu berhasil ditambahkan!
```

```
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 5

Daftar Lagu dalam Playlist:
1. Judul: Melukis Senja, Penyanyi: Budi Doremi, Durasi: 4 menit
2. Judul: Hati-Hati di Jalan, Penyanyi: Tulus, Durasi: 4.2 menit
3. Judul: Sial, Penyanyi: Mahalini, Durasi: 3.8 menit
4. Judul: Aku Yang Salah, Penyanyi: Mahalini & Nuca, Durasi: 3.5 menit
```

4) Tambah lagu setelah playlist ke 3

```
============================
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 3
Masukkan judul lagu: Los Dol
Masukkan nama penyanyi: Denny Caknan
Masukkan durasi lagu (menit): 4.0
Lagu berhasil ditambahkan!
```

```
============================
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 5

Daftar Lagu dalam Playlist:
1. Judul: Melukis Senja, Penyanyi: Budi Doremi, Durasi: 4 menit
2. Judul: Hati-Hati di Jalan, Penyanyi: Tulus, Durasi: 4.2 menit
3. Judul: Sial, Penyanyi: Mahalini, Durasi: 3.8 menit
4. Judul: Los Dol, Penyanyi: Denny Caknan, Durasi: 4 menit
5. Judul: Aku Yang Salah, Penyanyi: Mahalini & Nuca, Durasi: 3.5 menit
```

5) Hapus lagu berdasarkan judul

```
============================
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 4
Masukkan judul lagu yang ingin dihapus: Aku Yang Salah
Lagu "Aku Yang Salah" berhasil dihapus.
```

```
     MENU PLAYLIST LAGU
============================
1. Tambah lagu di awal playlist
2. Tambah lagu di akhir playlist
3. Tambah lagu setelah playlist ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan seluruh lagu
0. Keluar
============================
Pilih menu: 5

Daftar Lagu dalam Playlist:
1. Judul: Melukis Senja, Penyanyi: Budi Doremi, Durasi: 4 menit
2. Judul: Hati-Hati di Jalan, Penyanyi: Tulus, Durasi: 4.2 menit
3. Judul: Sial, Penyanyi: Mahalini, Durasi: 3.8 menit
4. Judul: Los Dol, Penyanyi: Denny Caknan, Durasi: 4 menit
```

Deskripsi:

Program di atas merupakan program dalam bahasa C++ yang berfungsi untuk mengelola daftar lagu menggunakan struktur data singly linked list. File playlist.h berisi deklarasi struktur Lagu yang menyimpan informasi judul, penyanyi, dan durasi lagu, serta struktur Node yang menghubungkan setiap lagu dengan node berikutnya melalui pointer next, dan struktur Playlist yang menyimpan alamat node pertama. Di dalam file tersebut juga dideklarasikan fungsi-fungsi dasar seperti createPlaylist, alokasi, insertFirst, insertLast, insertAfterK, deleteByJudul, dan printPlaylist. Implementasi fungsi-fungsi tersebut terdapat di playlist.cpp, di mana setiap fungsi memiliki peran masing-masing, seperti membuat playlist baru, menambahkan lagu di awal, akhir, atau setelah posisi tertentu, menghapus lagu berdasarkan judul, dan menampilkan seluruh isi playlist. File main.cpp berfungsi sebagai program utama yang menampilkan menu interaktif bagi pengguna untuk memilih operasi yang ingin dilakukan, seperti menambah atau menghapus lagu, serta menampilkan daftar lagu dalam playlist. Program ini memanfaatkan konsep linked list untuk mengatur data secara dinamis, sehingga memungkinkan penambahan dan penghapusan lagu tanpa batasan ukuran array.

D. Kesimpulan

Berdasarkan hasil praktikum pada modul Singly Linked List, dapat disimpulkan bahwa penggunaan struktur data linked list dalam bahasa C++ memungkinkan pengelolaan data secara dinamis tanpa batasan ukuran seperti pada array. Melalui penerapan pointer dan node, setiap elemen dapat ditambahkan atau dihapus dengan efisien tanpa perlu menggeser elemen lain. Pada bagian guided, mahasiswa memahami konsep dasar pembuatan linked list, mulai dari pembuatan node, penyisipan data di awal maupun akhir list, hingga penelusuran dan penampilan isi list.

Sementara pada bagian unguided, konsep tersebut diterapkan dalam bentuk program pengelolaan playlist lagu yang mendemonstrasikan operasi penambahan, penghapusan, dan penampilan data secara interaktif. Secara keseluruhan, praktikum ini memperkuat pemahaman tentang manipulasi pointer, alokasi memori dinamis, serta pentingnya struktur data linked list dalam membangun program yang efisien dan terstruktur.

E. Referensi

AKCoding.com. (2024, March 9). Singly Linked List: Explained, Examples, and Applications. Medium. https://akcoding.medium.com/singly-linked-list-explained-examples-and-applications-378209429a2a

Fazry. (2024, May 25). Linked List: Pengertian dan Implementasi Dasar. Rumah Coding. https://rumahcoding.co.id/linked-list-pengertian-dan-implementasi-dasar/