

**LAPORAN PRAKTIKUM STRUKTUR
DATA**

**MODUL IV
SINGLY LINKED LIST (BAGIAN PERTAMA)**



Disusun Oleh :

NAMA : IVAN RAMADHAN

NIM : 103112400186

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

C++ adalah pengembangan dari bahasa c yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

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

Guided 1

Playlist.h

```
#ifndef PLAYLIST_H
#define PLAYLIST_H

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

struct Lagu {
    string judul;
    string penyanyi;
    float durasi;
    Lagu* next;
};

class Playlist {
private:
    Lagu* head;

public:
    Playlist();
    void tambahAwal(string judul, string penyanyi, float durasi);
    void tambahAkhir(string judul, string penyanyi, float durasi);
    void tambahSetelahKe3(string judul, string penyanyi, float durasi);
    void hapusLagu(string judul);
    void tampilkan();
};

#endif
```

Playlist.cpp

```
#include "playlist.h"

Playlist::Playlist() {
    head = nullptr;
}

void Playlist::tambahAwal(string judul, string penyanyi, float durasi) {
    Lagu* baru = new Lagu{judul, penyanyi, durasi, head};
    head = baru;
}

void Playlist::tambahAkhir(string judul, string penyanyi, float durasi) {
    Lagu* baru = new Lagu{judul, penyanyi, durasi, nullptr};

    if (!head) {
        head = baru;
        return;
    }

    Lagu* temp = head;
    while (temp->next)
        temp = temp->next;

    temp->next = baru;
}

void Playlist::tambahSetelahKe3(string judul, string penyanyi, float durasi) {
    Lagu* baru = new Lagu{judul, penyanyi, durasi, nullptr};
    Lagu* temp = head;
    int count = 1;

    while (temp && count < 3) {
        temp = temp->next;
        count++;
    }

    if (!temp) {
        cout << "Playlist kurang dari 3 lagu.\n";
        delete baru;
        return;
    }

    baru->next = temp->next;
```

```

    temp->next = baru;
}

void Playlist::hapusLagu(string judul) {
    if (!head) {
        cout << "Playlist kosong.\n";
        return;
    }

    if (head->judul == judul) {
        Lagu* hapus = head;
        head = head->next;
        delete hapus;
        cout << "Lagu \"" << judul << "\" dihapus.\n";
        return;
    }

    Lagu* temp = head;
    while (temp->next && temp->next->judul != judul) {
        temp = temp->next;
    }

    if (!temp->next) {
        cout << "Lagu \"" << judul << "\" tidak ditemukan.\n";
        return;
    }

    Lagu* hapus = temp->next;
    temp->next = hapus->next;
    delete hapus;

    cout << "Lagu \"" << judul << "\" dihapus.\n";
}

void Playlist::tampilkan() {
    if (!head) {
        cout << "Playlist kosong.\n";
        return;
    }

    Lagu* temp = head;
    int i = 1;

    cout << "\n=== Daftar Lagu ===\n";
    while (temp) {
        cout << i++ << ". Judul: " << temp->judul

```

```

        << " | Penyanyi: " << temp->penyanyi
        << " | Durasi: " << temp->durasi << " menit\n";
    temp = temp->next;
}
}

```

Main.cpp

```

#include "playlist.h"

int main() {
    Playlist p;
    int pilih;
    string judul, penyanyi;
    float durasi;

    do {
        cout << "\n=== MENU PLAYLIST ===\n";
        cout << "1. Tambah lagu di awal\n";
        cout << "2. Tambah lagu di akhir\n";
        cout << "3. Tambah lagu setelah ke-3\n";
        cout << "4. Hapus lagu berdasarkan judul\n";
        cout << "5. Tampilkan playlist\n";
        cout << "0. Keluar\n";
        cout << "Pilih: ";
        cin >> pilih;
        cin.ignore();

        switch (pilih) {
            case 1:
                cout << "Judul lagu: ";
                getline(cin, judul);
                cout << "Penyanyi: ";
                getline(cin, penyanyi);
                cout << "Durasi (menit): ";
                cin >> durasi;
                p.tambahAwal(judul, penyanyi, durasi);
                break;

            case 2:
                cout << "Judul lagu: ";
                getline(cin, judul);
                cout << "Penyanyi: ";
                getline(cin, penyanyi);

```

```
    cout << "Durasi (menit): ";
    cin >> durasi;
    p.tambahAkhir(judul, penyanyi, durasi);
    break;
```

case 3:

```
    cout << "Judul lagu: ";
    getline(cin, judul);
    cout << "Penyanyi: ";
    getline(cin, penyanyi);
    cout << "Durasi (menit): ";
    cin >> durasi;
    p.tambahSetelahKe3(judul, penyanyi, durasi);
    break;
```

case 4:

```
    cout << "Masukkan judul lagu yang ingin dihapus: ";
    getline(cin, judul);
    p.hapusLagu(judul);
    break;
```

case 5:

```
    p.tampilkan();
    break;
```

case 0:

```
    cout << "Keluar dari program.\n";
    break;
```

default:

```
    cout << "Pilihan tidak valid.\n";
```

```
    }
```

```
} while (pilih != 0);
```

```
return 0;
```

```
}
```

```
PS C:\xampp\htdocs\Modul Alpro\C++ SMT 3> g++ main.cpp playlist.cpp -o program
>> ./program
```

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

Tambah Lagu di Awal

```
=== MENU PLAYLIST ===
1. Tambah lagu di awal
2. Tambah lagu di akhir
3. Tambah lagu setelah ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan playlist
0. Keluar
Pilih: 1
Judul lagu: sedia aku sebelum hujan
Penyanyi: indigaf
Durasi (menit): 3.3
```

Tambah Lagu di Akhir

```
=== MENU PLAYLIST ===
1. Tambah lagu di awal
2. Tambah lagu di akhir
3. Tambah lagu setelah ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan playlist
0. Keluar
Pilih: 2
Judul lagu: Separuh aku
Penyanyi: Noah
Durasi (menit): 3.3
```

Tambah Lagu Setelah ke-3

```
=== MENU PLAYLIST ===
1. Tambah lagu di awal
2. Tambah lagu di akhir
3. Tambah lagu setelah ke-3
4. Hapus lagu berdasarkan judul
5. Tampilkan playlist
0. Keluar
Pilih: 3
Judul lagu: Hapus aku
Penyanyi: Nidji
Durasi (menit): 3.3
Playlist kurang dari 3 lagu.
```

Tampilkan Playlist

```
5. Tampilkan playlist
0. Keluar
Pilih: 5

=== Daftar Lagu ===
1. Judul: sedia aku sebelum hujan | Penyanyi: indigaf | Durasi: 3.3 menit
2. Judul: Separuh aku | Penyanyi: Noah | Durasi: 3.3 menit
```

Hapus Lagu Berdasarkan Judul

```
4. Hapus lagu berdasarkan judul
5. Tampilkan playlist
0. Keluar
Pilih: 4
Masukkan judul lagu yang ingin dihapus: Separuh aku
Lagu "Separuh aku" dihapus.
```

Deskripsi:

Program ini merupakan aplikasi manajemen playlist lagu yang menggunakan struktur data linked list. Setiap lagu disimpan dalam node yang berisi judul, penyanyi, durasi, dan pointer ke lagu berikutnya. Program menyediakan menu interaktif untuk menambah lagu di awal, menambah lagu di akhir, menambah lagu setelah lagu ke-3, menghapus lagu berdasarkan judul, serta menampilkan seluruh playlist. Dengan pendekatan berbasis kelas (Playlist), program ini menunjukkan penerapan konsep OOP, pointer, dan manajemen memori dinamis dalam C++.

C. Kesimpulan

Program Playlist Lagu ini menerapkan konsep struktur data linked list tunggal dalam bahasa C++. Data lagu disimpan secara dinamis menggunakan pointer, sehingga memungkinkan penambahan dan penghapusan lagu tanpa batasan ukuran array. Program menyediakan menu interaktif untuk menambah lagu di awal, di akhir, setelah lagu ke-3, menghapus lagu berdasarkan judul, serta menampilkan seluruh playlist. Dengan pemisahan kode ke dalam header (.h) dan source (.cpp), program menjadi lebih terstruktur, modular, dan mudah dikembangkan, sekaligus melatih pemahaman konsep class, struct, pointer, dan manajemen memori dinamis dalam pemrograman C++.

D. Referensi

- W3Resource. (2020). *C++ String Exercises: Convert digit/number to words*.
- GeeksforGeeks. (2020). *Loops in C++ (for, while, do-while)*.