LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL XIII

MULTI LINKED LIST

Disusun Oleh :
Nama : Besthian Guido Rafael Simbolon
NIM : 103112430258

Dosen
FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

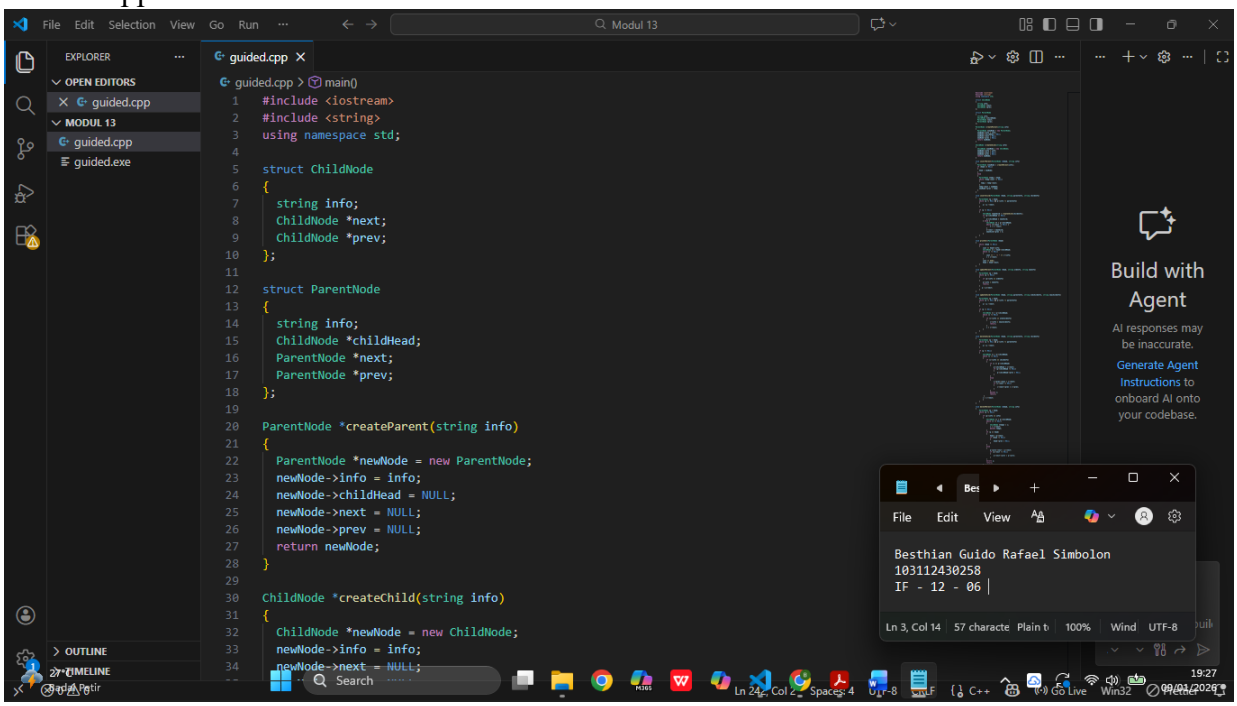
A. Dasar Teori

Linked List adalah struktur data dinamis yang terdiri dari kumpulan node yang saling terhubung menggunakan pointer, sehingga tidak memiliki ukuran tetap seperti array. Salah satu pengembangannya adalah Multi-Linked List, yaitu struktur data yang digunakan untuk menyimpan data dengan hubungan satu ke banyak (one-to-many). Dalam implementasinya, struktur ini biasanya menggunakan Doubly Linked List, di mana setiap node memiliki pointer next dan prev untuk memudahkan penelusuran dua arah. Keunikan Multi-Linked List terletak pada adanya pointer tambahan pada node induk (parent) yang menunjuk ke node pertama pada daftar anak (child), sehingga satu data parent dapat memiliki banyak data child. Struktur ini membentuk pola seperti pohon atau graf dan memungkinkan pengelolaan data secara fleksibel dan dinamis tanpa batasan ukuran seperti pada array statis.

B. Guided

Guided 1

Guided.cpp



Screenshots Output

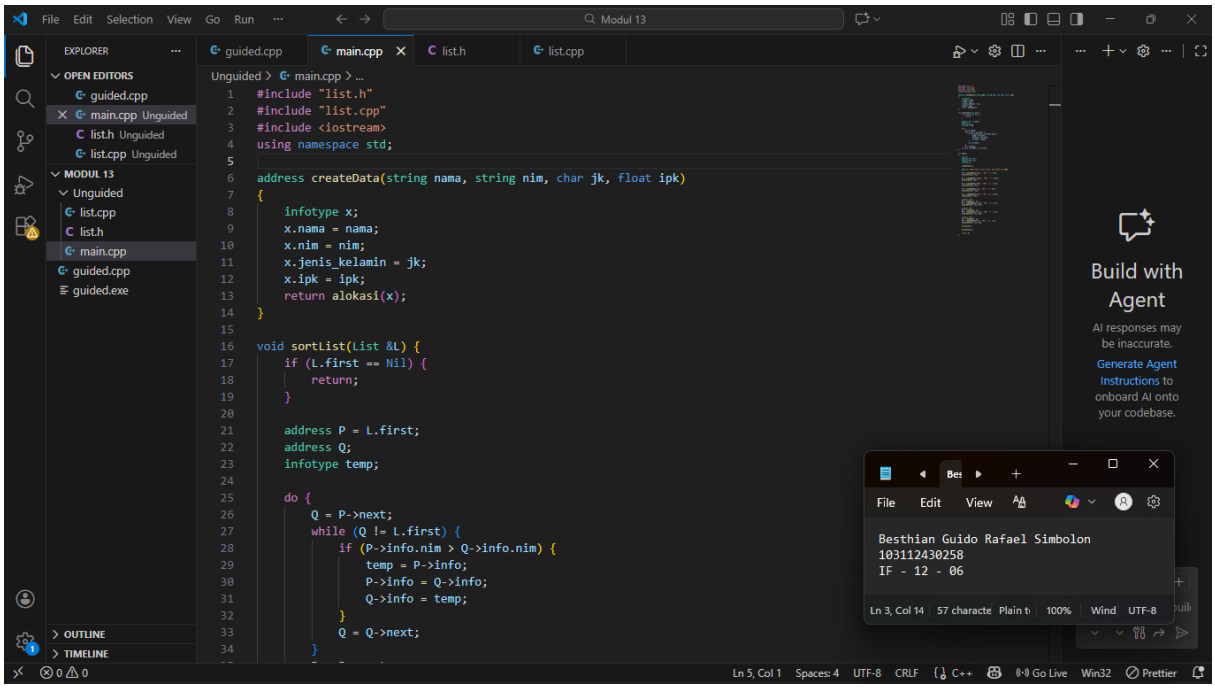
Deskripsi code :

Program ini merupakan implementasi sederhana struktur data Multi-Linked List (List of Lists) menggunakan bahasa C++, yang terdiri dari dua jenis node yaitu ParentNode sebagai induk dan ChildNode sebagai anak. Setiap node induk saling terhubung dalam bentuk doubly linked list, dan setiap node induk memiliki pointer khusus yang menunjuk ke daftar node anak yang juga tersusun sebagai doubly linked list. Program ini menyediakan operasi dasar CRUD (Create, Read, Update, Delete) yang memungkinkan pengguna menambah data parent dan child, menampilkan seluruh struktur data secara hierarkis, memperbarui data pada node tertentu, serta menghapus node parent atau child dengan tetap menjaga keterhubungan pointer agar struktur data tidak rusak. Pada fungsi main, program memperlihatkan alur penggunaan secara lengkap mulai dari penambahan data, pembaruan, hingga penghapusan untuk menunjukkan cara pengelolaan relasi antar node di dalam memori.

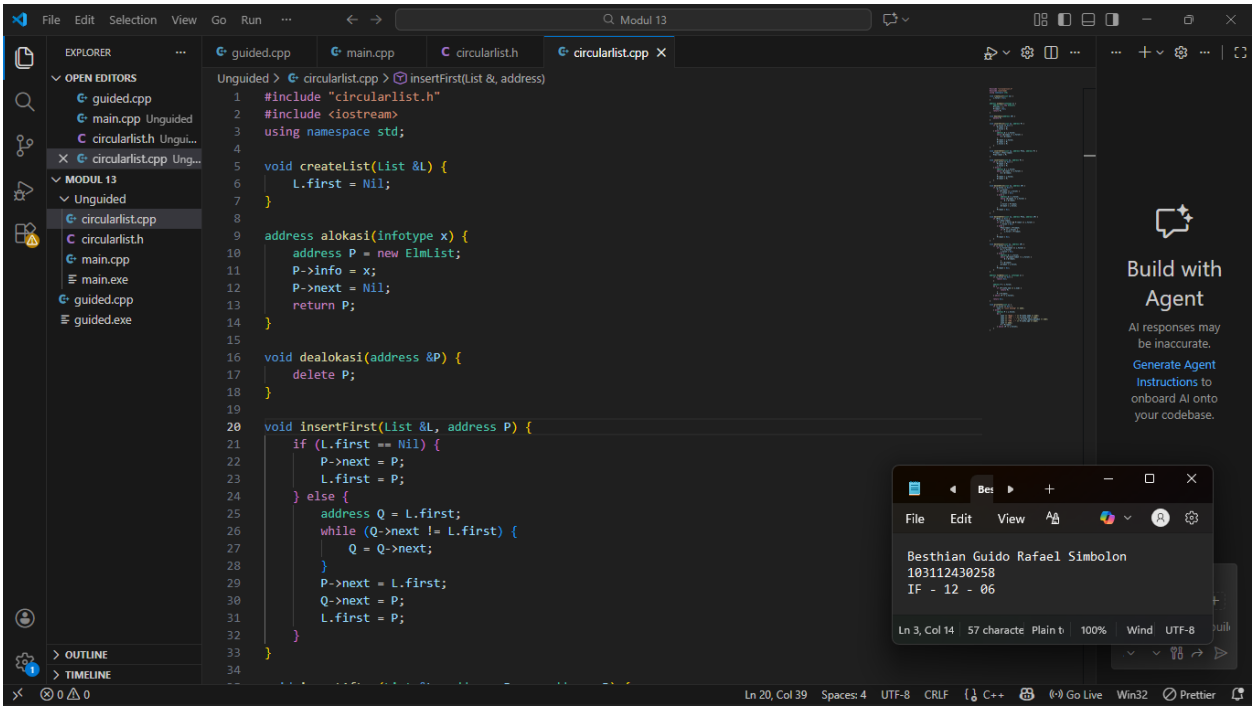
C. Unguided

Unguided 1

Main.cpp



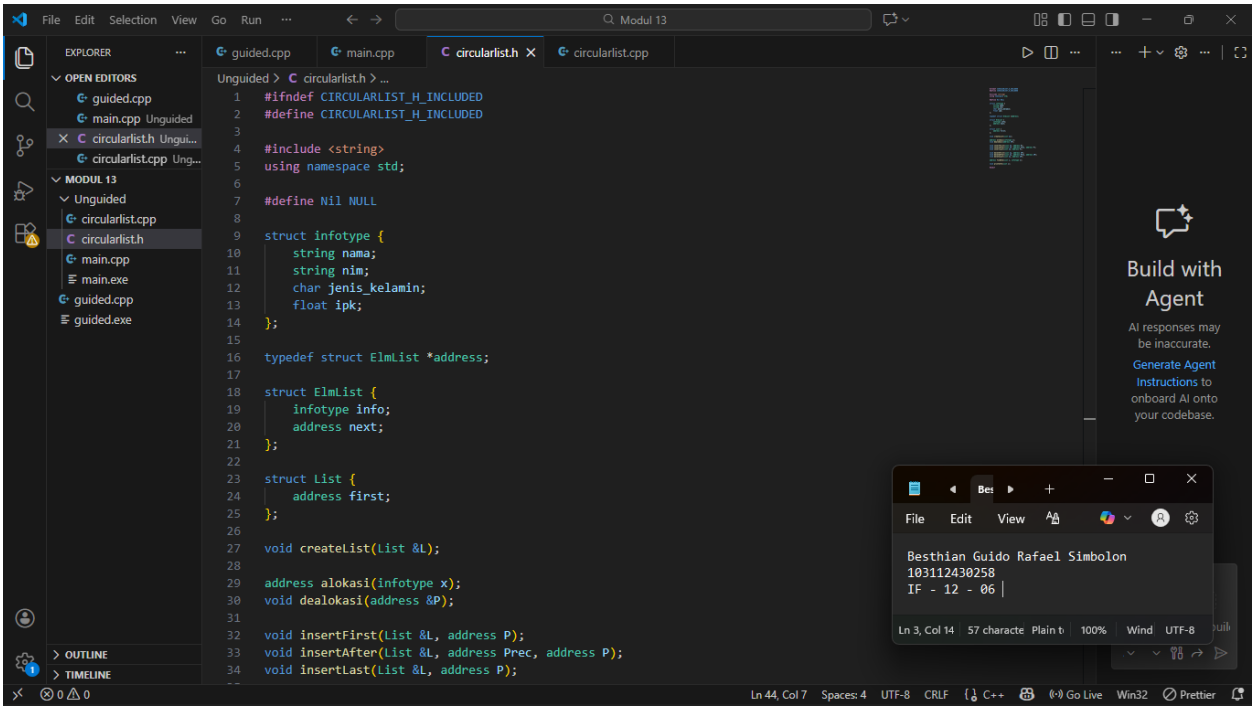
circularlist.cpp



```cpp
#include "circularlist.h"
#include <iostream>
using namespace std;

void createList(List &L) {
    L.first = Nil;
}

address alokasi(infotype x) {
    address P = new ElmList;
    P->info = x;
    P->next = Nil;
    return P;
}

void dealokasi(address &P) {
    delete P;
}

void insertFirst(List &L, address P) {
    if (L.first == Nil) {
        P->next = P;
        L.first = P;
    } else {
        address Q = L.first;
        while (Q->next != L.first) {
            Q = Q->next;
        }
        P->next = L.first;
        Q->next = P;
        L.first = P;
    }
}
```

```
Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06
```

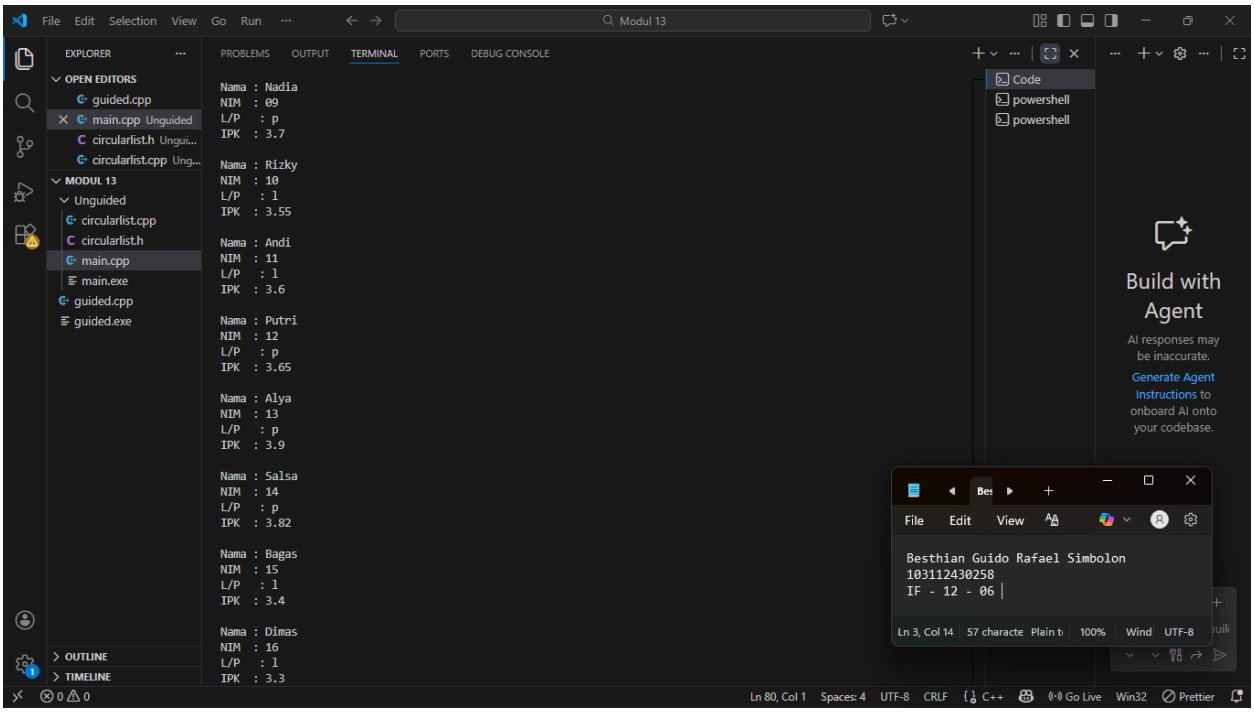
circularlist.h



```cpp
#ifndef CIRCULARLIST_H_INCLUDED
#define CIRCULARLIST_H_INCLUDED

#include <string>
using namespace std;

#define Nil NULL

struct infotype {
    string nama;
    string nim;
    char jenis_kelamin;
    float ipk;
};

typedef struct ElmList *address;

struct ElmList {
    infotype info;
    address next;
};

struct List {
    address first;
};

void createList(List &L);

address alokasi(infotype x);
void dealokasi(address &P);

void insertFirst(List &L, address P);
void insertAfter(List &L, address Prec, address P);
void insertLast(List &L, address P);
```

```
Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06
```

Screenshot Output



Deskripsi:

Program ini merupakan implementasi struktur data **Circular Single Linked List** menggunakan bahasa C++ yang digunakan untuk mengelola data mahasiswa, meliputi nama, NIM, jenis kelamin, dan IPK. Pada struktur ini, node terakhir tidak menunjuk ke `NULL`, melainkan kembali ke node pertama (first), sehingga membentuk sebuah lingkaran. Program disusun secara modular dengan pemisahan file header, file implementasi, dan file `main` agar kode lebih rapi dan mudah dipahami. Operasi dasar yang disediakan meliputi pembuatan node, penyisipan data di awal, akhir, atau setelah node tertentu, penghapusan, serta pencarian data. Pada fungsi `main`, program menampilkan contoh penggunaan dengan memasukkan beberapa data mahasiswa secara acak, kemudian mengurutkannya berdasarkan NIM menggunakan algoritma sederhana, dan akhirnya menampilkan seluruh data ke layar.

D. Kesimpulan

Multi Linked List yang telah dilakukan, dapat disimpulkan bahwa struktur data ini sangat efektif untuk mengelola data yang memiliki hubungan **satu ke banyak (one-to-many)**. Dengan adanya node induk (parent) yang terhubung ke beberapa node anak (child) melalui pointer, data dapat disusun secara hierarkis dan lebih terorganisir. Praktikum ini juga menunjukkan pentingnya pemahaman terhadap penggunaan pointer dan alokasi memori dinamis agar proses penambahan, penghapusan, dan pencarian data dapat berjalan dengan baik tanpa merusak struktur list. Secara keseluruhan, Multi Linked List memberikan fleksibilitas tinggi dalam pengelolaan data yang kompleks dan menjadi solusi yang tepat untuk kasus data bertingkat.

.

E. Referensi

Kadir, A. (2012). *Algoritma dan struktur data*. Yogyakarta: Andi Offset.

Kurniawan, D. (2016). *Struktur data menggunakan C++*. Jakarta: Elex Media Komputindo.

Rosa, A. S., & Shalahuddin, M. (2018). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Bandung: Informatika.

Sutabri, T. (2012). *Konsep sistem informasi*. Yogyakarta: Andi Offset.

Program Studi Informatika. (Tahun). *Modul praktikum struktur data: Linked list dan pengembangannya*. Indonesia: Perguruan Tinggi terkait.