

**LAPORAN PRAKTIKUM STRUKTUR
DATA**

**MODUL V
SINGLY LINKED LIST (BAGIAN KEDUA)**



Disusun Oleh :

NAMA : ADIKA AUNURFIKRI NOVIYANTO

NIM : 103112400195

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

Singlylist.h

```
#ifndef SINGLYLIST_H
#define SINGLYLIST_H
#include <iostream>
using namespace std;

typedef int infotype;

struct ElmList {
    infotype info;
    ElmList* next;
};

typedef ElmList* address;

struct List {
    address First;
};

void createList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void insertFirst(List &L, address P);
void printInfo(List L);

address findElm(List L, infotype x);
int totalInfo(List L);

#endif
```

Singlylist.cpp

```
#include "Singlylist.h"

void createList(List &L) {
    L.First = NULL;
}

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

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

void insertFirst(List &L, address P) {
    P->next = L.First;
    L.First = P;
}

void printInfo(List L) {
    address P = L.First;
    while (P != NULL) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}

address findElm(List L, infotype x) {
    address P = L.First;
```

```

while (P != NULL) {
    if (P->info == x) {
        cout << x << " ditemukan dalam list" << endl;
        return P;
    }
    P = P->next;
}
cout << x << " tidak ditemukan dalam list" << endl;
return NULL;
}

int totalInfo(List L) {
    int total = 0;
    address P = L.First;
    while (P != NULL) {
        total += P->info;
        P = P->next;
    }
    return total;
}

```

Main.cpp

```

#include "Singlylist.h"

int main() {
    List L;
    address P1, P2, P3, P4, P5;
    createList(L);

    P1 = alokasi(2);
    insertFirst(L, P1);

    P2 = alokasi(0);
    insertFirst(L, P2);
}

```

```

P3 = alokasi(8);
insertFirst(L, P3);

P4 = alokasi(12);
insertFirst(L, P4);

P5 = alokasi(9);
insertFirst(L, P5);

printInfo(L);

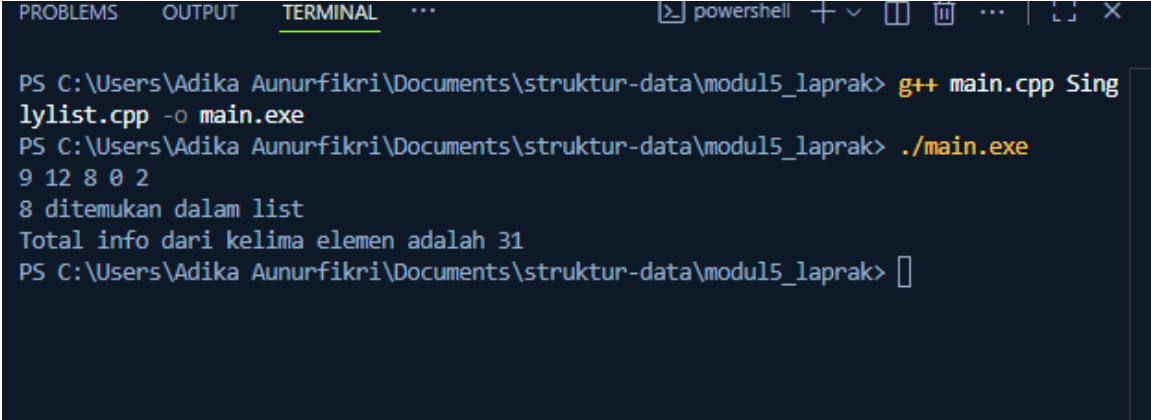
findElm(L, 8);

cout << "Total info dari kelima elemen adalah " << totalInfo(L) << endl;

return 0;
}

```

Screenshots Output:



```

PROBLEMS OUTPUT TERMINAL ...
powershell + v [ ] [ ] ... [ ] x
PS C:\Users\Adika Aunurfikri\Documents\struktur-data\modul5_laprak> g++ main.cpp Singlylist.cpp -o main.exe
PS C:\Users\Adika Aunurfikri\Documents\struktur-data\modul5_laprak> ./main.exe
9 12 8 0 2
8 ditemukan dalam list
Total info dari kelima elemen adalah 31
PS C:\Users\Adika Aunurfikri\Documents\struktur-data\modul5_laprak> 

```

Deskripsi:

Program ini merupakan implementasi dari Abstract Data Type (ADT) Singly Linked List menggunakan bahasa pemrograman C++. Tujuannya adalah untuk memahami cara kerja struktur data Linked List, yaitu kumpulan elemen yang saling terhubung melalui pointer, di mana setiap elemen (node) menyimpan nilai data dan alamat node berikutnya.

C. Kesimpulan

Program ini menunjukkan bagaimana Linked List bekerja secara dinamis, di mana elemen tidak disimpan secara berurutan di memori seperti array. Setiap node terhubung melalui pointer next, sehingga penambahan atau penghapusan elemen dapat dilakukan tanpa perlu memindahkan data lain. Melalui implementasi ini, pengguna dapat memahami prinsip dasar alokasi dinamis, traversal, dan manipulasi node dalam struktur data Linked List.

D. Referensi

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