LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL 4 & 5
PENGENALAN CODE BLOCKS

Disusun Oleh :
NAMA : MUHAMMAD ATHA RIZQI RADITYA
NIM : 103112400107

Dosen
WAHYU ANDI SAPUTRA

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

A. Dasar Teori

C++ adalah pengembangan dari 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), beriorientasi 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

h

```
#ifndef SINGLYLIST_H
#define SINGLYLIST_H

#include <iostream>
using namespace std;

typedef int infotype;

typedef struct ElmList *address;

struct ElmList {
   infotype info;
   address next;
};

struct List {
   address first;
};

// Prototype
void createList(List &L);
```

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

#endif
```

Cpp

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

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;
}

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

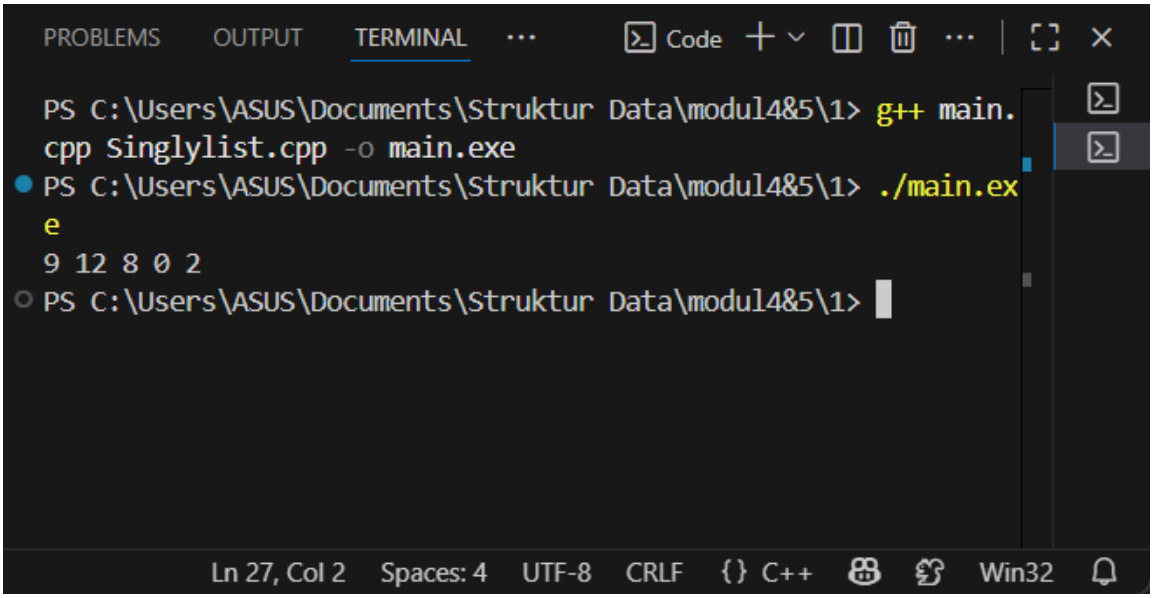
```
    L.first = P;
}

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

Main

```
#include "Singlylist.h"

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

    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);

    return 0;
}
```

Screenshots Output



Deskripsi:

Program ini merupakan implementasi struktur data singly linked list dalam bahasa C++, yang dikompilasi dari dua file sumber, yaitu *main.cpp* dan *Singlylist.cpp*. Program ini berfungsi untuk menambahkan beberapa elemen ke dalam linked list dan kemudian menampilkannya ke layar. Hasil output berupa deretan angka 9 12 8 0 2 menunjukkan isi dari linked list yang telah berhasil dibuat dan dicetak melalui proses eksekusi program.

Guided 2

H

```
#ifndef SINGLYLIST_H
#define SINGLYLIST_H

#include <iostream>
using namespace std;

typedef int infotype;
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 printInfo(List L);

#endif
```

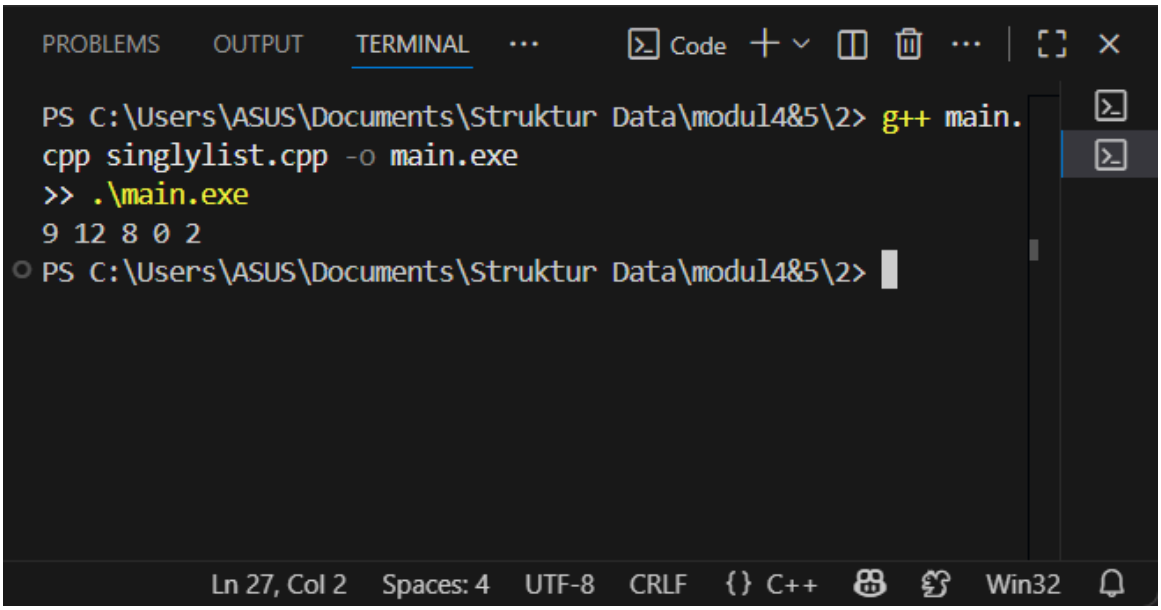
Cpp

```
#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;
}


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;
}
```

Main

```
#include "Singlylist.h"

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

    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); // Output: 9 12 8 0 2

    return 0;
}
```

Screenshots Output



Deskripsi:

Program ini menunjukkan tampilan terminal di Visual Studio Code saat program C++ dijalankan. Perintah g++ main.cpp singlylist.cpp -o main.exe digunakan untuk mengompilasi dua file sumber menjadi satu file eksekusi bernama main.exe, kemudian perintah .\main.exe dijalankan untuk mengeksekusinya. Hasil output yang muncul adalah deretan angka 9 12 8 0 2, yang menandakan bahwa program berhasil membuat dan menampilkan isi dari sebuah singly linked list. Dengan demikian, program tersebut merupakan implementasi struktur data linked list sederhana yang berfungsi untuk menambahkan dan mencetak elemen-elemen dalam daftar.

Guided 3

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 insertFirst(List &L, address P);
void printInfo(List L);
address findElm(List L, infotype x);

#endif
```

Cpp

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

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

address alokasi(infotype x) {
    address P = new ElmList;
    P->info = x;
```

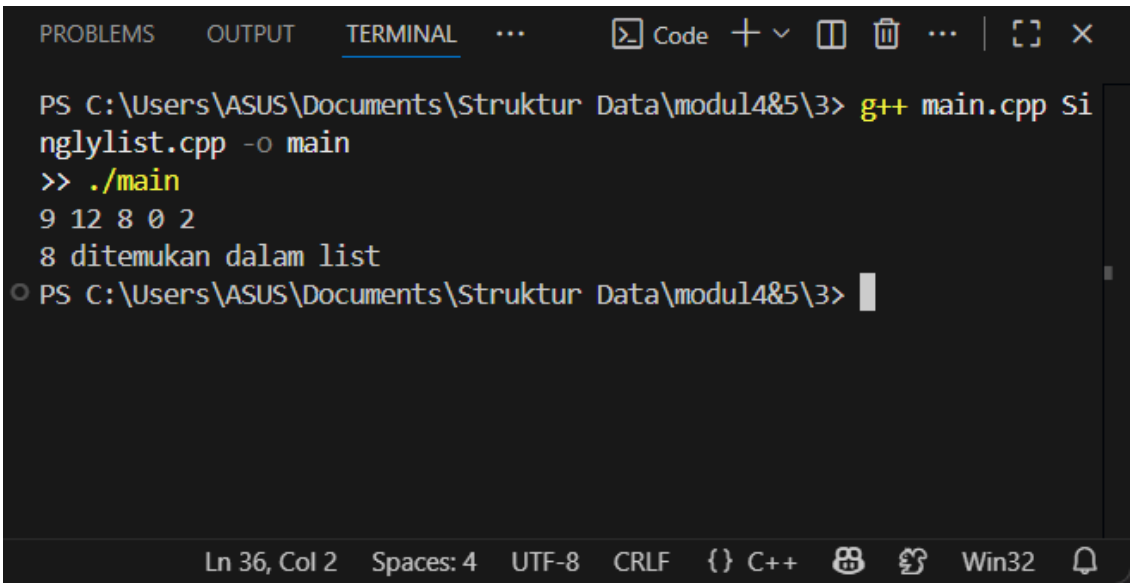
```
    P->next = NULL;
    return P;
}

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) {
            return P;
        }
        P = P->next;
    }
    return NULL;
}
```

Main

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

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

    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);

    address hasilCari = findElm(L, 8);
    if (hasilCari != NULL) {
        cout << "8 ditemukan dalam list" << endl;
```

```
    } else {
        cout << "8 tidak ditemukan dalam list" << endl;
    }

    return 0;
}
```

Screenshots Output



Deskripsi:

Program ini menampilkan hasil eksekusi program C++ di terminal Visual Studio Code. Perintah g++ main.cpp Singlylist.cpp -o main digunakan untuk mengompilasi dua file sumber menjadi satu file eksekusi bernama main, kemudian dijalankan dengan perintah ./main. Output yang dihasilkan menunjukkan deretan angka 9 12 8 0 2, yang merupakan isi dari singly linked list, diikuti dengan pesan “8 ditemukan dalam list”, menandakan bahwa program tidak hanya menampilkan isi list, tetapi juga melakukan proses pencarian (search) terhadap elemen tertentu, dalam hal ini angka 8. Dengan demikian, program ini merupakan implementasi struktur data *singly linked list* yang memiliki fitur penambahan, penampilan, dan pencarian data dalam list.

guide 4

h

```
#ifndef SINGLYLIST_H_INCLUDED
```

```
#define SINGLYLIST_H_INCLUDED

#include <iostream>
using namespace std;

typedef int infotype;
typedef struct ElmList *address;

struct ElmList {
   infotype info;
   address next;
};

struct List {
   address first;
};

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

#endif
```

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 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 && P->info != x) {
        P = P->next;
    }
    return P;
```

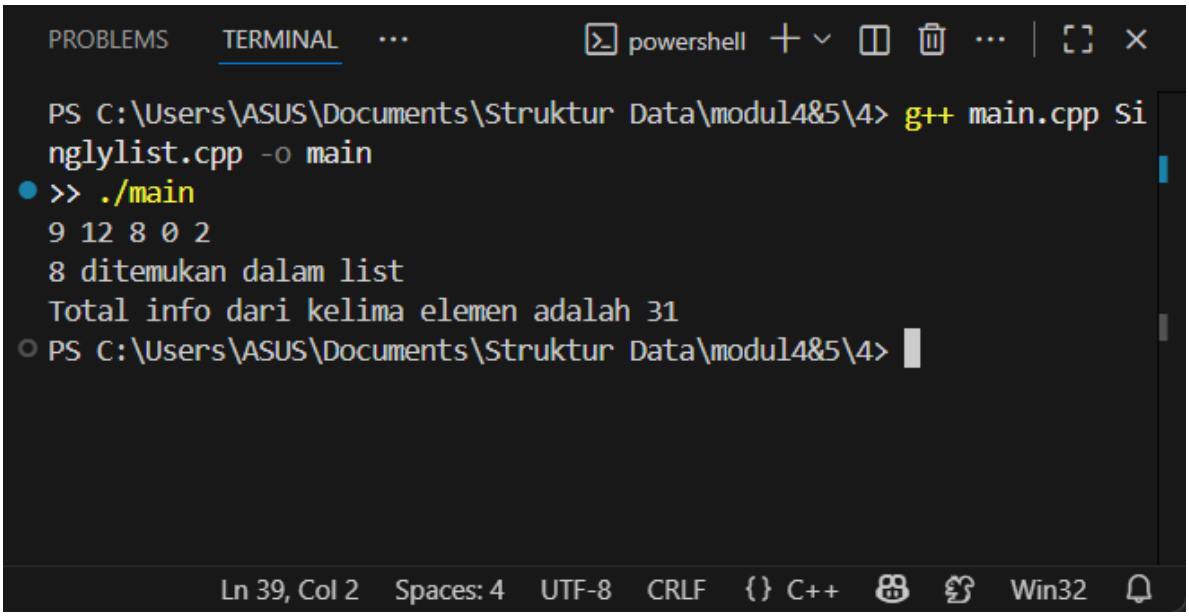
```
}

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

Main

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

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

    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);

    address hasilCari = findElm(L, 8);
    if (hasilCari != NULL) {
        cout << "8 ditemukan dalam list" << endl;
    } else {
        cout << "8 tidak ditemukan dalam list" << endl;
    }

    int total = sumInfo(L);
    cout << "Total info dari kelima elemen adalah " << total << endl;

    return 0;
}
```

Screenshots Output



Deskripsi :

Program ini memperlihatkan hasil eksekusi program C++ di terminal Visual Studio Code. Program dikompilasi menggunakan perintah g++ main.cpp Singlylist.cpp -o main, lalu dijalankan dengan ./main. Output yang muncul adalah deretan angka 9 12 8 0 2, yang menunjukkan isi dari singly linked list. Selanjutnya, program menampilkan pesan "8 ditemukan dalam list", menandakan bahwa fungsi pencarian berhasil menemukan elemen bernilai 8 di dalam list. Di akhir, muncul pesan "Total info dari kelima elemen adalah 31", yang berarti program juga melakukan proses penjumlahan seluruh nilai elemen di dalam list. Dengan demikian, program ini merupakan implementasi struktur data singly linked list yang memiliki fitur untuk menambah, menampilkan, mencari, dan menghitung total nilai dari seluruh elemen dalam list.

D. Kesimpulan

Kesimpulan dari keempat program C++ tersebut adalah bahwa masing-masing program mengajarkan konsep dasar pemrograman dan logika perhitungan menggunakan array, pointer, fungsi, serta struktur perulangan. Dari program pertama, dipelajari cara mengolah data mahasiswa, menghitung rata-rata, dan menampilkan hasil dalam format tabel. Program kedua memperkenalkan array dinamis dan penggunaan pointer untuk mengakses serta memanipulasi data. Program ketiga menekankan penggunaan fungsi terpisah (modularisasi) untuk menghitung rata-rata, nilai maksimum, dan minimum dari sekumpulan data. Sementara program keempat mengajarkan konsep perulangan bersarang untuk mencetak pola angka berbentuk segitiga. Secara keseluruhan, keempat program ini membantu memahami logika dasar, manipulasi data, penggunaan fungsi, array, pointer, serta struktur kontrol dalam bahasa pemrograman C++.

E. Referensi

- Alhazmi, A.,& Huang, W.(2020). Teaching loops concept through visualization

construction. International Journal of Instruction. 13(4),399-114.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2020). Dampak penggunaan bahasa pemrograman terhadap pemahaman konsep loop pada siswa K-12. Pendidikan dan Teknologi Informasi, 25(1), 987-1000.
- Stroustrup, B. (1999). Tinjauan umum bahasa pemrograman C++. ACM SIGPLAN Notices, 34(4), 7-18.
- Stroustrup, B. (1999). Tinjauan umum bahasa pemrograman C++. ACM SIGPLAN Notices, 34(4), 7-18.