

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 4 & 5
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Maulana Ananta Piliang
NIM : 103112400156

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

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
```

```
#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

```
PROBLEMS OUTPUT TERMINAL ... ☒ Code + ⌂ ⌂ ⌂ ... | ⌂ x

PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\1> g++ main.cpp Singlylist.cpp -o main.exe
● PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\1> ./main.exe
9 12 8 0 2
○ PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\1>
```

Deskripsi:

Program ini mengimplementasikan singly linked list dalam C++, menambahkan beberapa elemen, lalu menampilkannya. Output 9 12 8 0 2 menunjukkan bahwa data berhasil dimasukkan dan dicetak dengan benar.

```
#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
delokasi(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

The screenshot shows a terminal window in Visual Studio Code. The terminal tab is selected at the top. The output in the terminal is as follows:

```
PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\2> g++ main.cpp singlylist.cpp -o main.exe
>> .\main.exe
9 12 8 0 2
```

At the bottom of the terminal window, there are status indicators: Line 27, Col 2, Spaces: 4, UTF-8, CRLF, {}, C++, Win32, and a bell icon.

Deskripsi:

Program ini memperlihatkan proses kompilasi dan eksekusi program C++ di Visual Studio Code menggunakan perintah g++ main.cpp singlylist.cpp -o main.exe dan .\main.exe. Output berupa angka 9 12 8 0 2 menunjukkan bahwa program berhasil membuat singly linked list dan menampilkan elemen-elemennya.

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;  
}
```



```
#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);

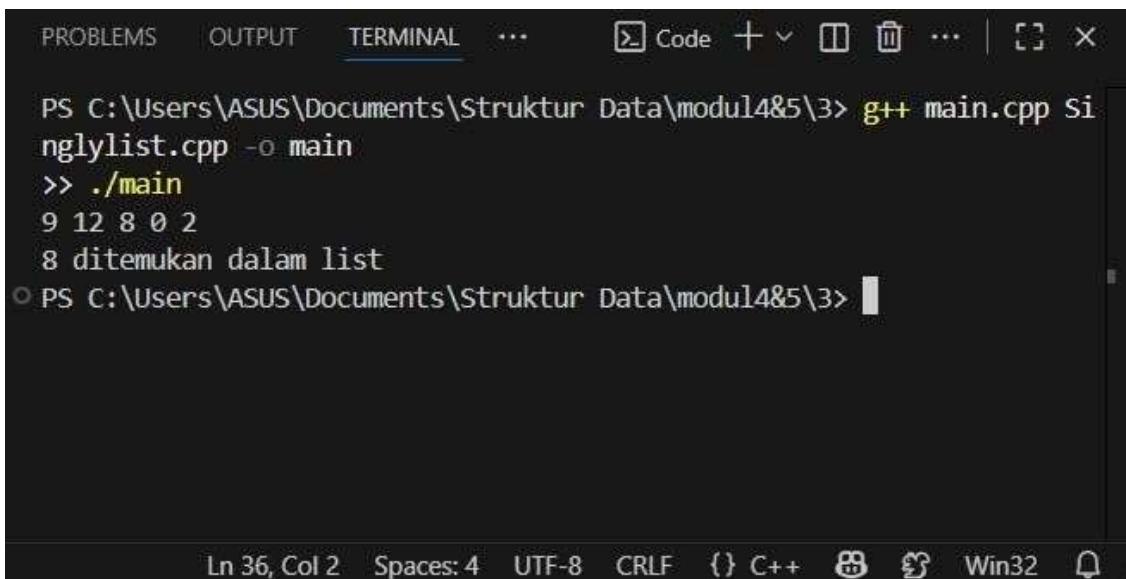
printInfo(L);

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



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab selected. The terminal window displays the following text:

```
PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\3> g++ main.cpp Singlylist.cpp -o main
>> ./main
9 12 8 0 2
8 ditemukan dalam list
PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\3>
```

At the bottom of the terminal window, status indicators show 'Ln 36, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', and file type icons for C++ and JSON.

Deskripsi:

Program ini menunjukkan proses kompilasi dan eksekusi program C++ di terminal Visual Studio Code menggunakan perintah `g++ main.cpp Singlylist.cpp -o main` dan `./main`. Output menampilkan angka `9 12 8 0 2` sebagai isi singly linked list, lalu pesan “`8 ditemukan dalam list`” yang menandakan bahwa program juga melakukan pencarian elemen. Dengan demikian, program ini merupakan implementasi singly linked list dengan fitur tambah data, tampil data, dan search..

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

```
PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\4> g++ main.cpp Singlylist.cpp -o main
● >> ./main
9 12 8 0 2
8 ditemukan dalam list
Total info dari kelima elemen adalah 31
○ PS C:\Users\ASUS\Documents\Struktur Data\modul4&5\4>
```

Ln 39, Col 2 Spaces: 4 UTF-8 CRLF {} C++ 🌐 ⚙️ Win32 📡

Deskripsi :

Program ini menampilkan hasil eksekusi program C++ di terminal Visual Studio Code. Setelah dikompilasi dengan g++ main.cpp Singlylist.cpp -o main dan dijalankan melalui ./main, program menghasilkan output 9 12 8 0 2 sebagai isi singly linked list. Program kemudian menampilkan pesan “8 ditemukan dalam list” sebagai hasil fungsi pencarian, serta “Total info dari kelima elemen adalah 31” yang menunjukkan proses penjumlahan seluruh elemen. Dengan demikian, program ini mengimplementasikan singly linked list dengan fitur tambah data, tampil data, pencarian elemen, dan perhitungan total nilai.

D. Kesimpulan

Dari praktikum Modul 4 dan 5 dapat disimpulkan bahwa mahasiswa berhasil mengimplementasikan *singly linked list* menggunakan C++. Program yang dibuat meliputi pembuatan list, penambahan elemen, pencarian data, hingga perhitungan total nilai. Hasil output menunjukkan bahwa seluruh fungsi berjalan dengan benar, seperti tampilan data “9 12 8 0 2”, temuan elemen “8 ditemukan dalam list”, serta total nilai “31”. Praktikum ini membantu memahami penggunaan pointer, alokasi memori dinamis, dan cara kerja linked list sebagai salah satu struktur data dasar.

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.