

LAPORAN PRAKTIKUM

Modul 4

“SINGLE LINKED LIST (BAGIAN PERTAMA)”



Disusun Oleh:

Nama : Marsella Dwi Julianti

NIM : 2311104004

Kelas : SE07-01

Dosen :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. Membuat deklarasi tipe list

Buat file list.h dan ketik sintak sesuai gambar berikut.

```
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info
using namespace std;
typedef int infotype;
typedef struct elmList *address;

struct elmList {
    infotype info;
    address next;
};
struct List{
    address first;
};
```

Buat file list.cpp dan ketik sintak berikut

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

2. Membuat list kosong, yaitu procedure createList.

Tambahkan pada list.h primitif dari procedure createList

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

Tambahkan pada list.cpp implementasi dari procedure createList, sintak c++ sebagai berikut:

```
void createList(List &L){
    /** this procedure will initialize
        the list L*/
    first(L) = NULL;
}
```

3. Setelah list sudah ada, selanjutnya buatlah elemen dengan menggunakan fungsi allocate.

Tambahkan pada list.h primitif dari fungsi allocate

```
address allocate(infotype x);
```

Tambahkan pada list.cpp implementasi dari fungsi allocate, sintak c++ sebagai berikut:

```
address allocate(infotype x){
    address p = new elmList;
    info(p) = x;
    next(p) = NULL;
    return p;
}
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elemen list. Proses insert dapat menggunakan procedure Insert First, procedure Insert Last, atau procedure insert After. Pada Tugas Pendahuluan kali ini, akan dicontohkan menggunakan insert first. Tambahkan pada list.h primitif procedure insertFirst

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

Tambahkan pada list.cpp implementasi dari procedure insertFirst sesuai sintak berikut :

```
void insertFirst(List &L, address P){
    /** TODO: Insert the new element pointed
        by P to the first of list L*/
    // YOUR CODES HERE
    //-----
    next(P) = first(L);
    first(L) = P;
    //-----
}
```

5. Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan, maka kita perlu menampilkan isi list.

Tambahkan pada list.h primitif procedure printInfo

```
void printInfo (List L);
```

Tambahkan pada list.cpp implementasi dari proc printInfo, sintak C++ sebagai berikut :

```
void printInfo(List L) {  
    /** this procedure will output  
    the info of each element  
    in list L*/  
  
    address p = first(L);  
    while (p != NULL) {  
        cout << info(p) << " ";  
        p = next(p);  
    }  
    cout << endl;  
}
```

Jawab:

➤ File list.h

```
TP > C list.h > ...  
1  #ifndef LIST_H  
2  #define LIST_H  
3  
4  #include <iostream>  
5  using namespace std;  
6  
7  typedef int infotype;  
8  typedef struct elmist *address;  
9  
10 struct elmist {  
11     infotype info;  
12     address next;  
13 };  
14  
15 struct List {  
16     address first;  
17 };  
18  
19 void createlist(List &L);  
20 address allocate(infotype x);  
21 void insertFirst(List &L, address P);  
22 void printInfo(List L);  
23  
24 #endif // LIST_H
```

➤ File list.cpp

```
TP > G list.cpp > printInfo(List)  
1  #include "list.h"  
2  
3  void createlist(List &L) {  
4      /** this procedure will initialize the list L*/  
5      L.first = NULL; // Menggunakan akses langsung  
6  }  
7  
8  address allocate(infotype x) {  
9      address P = new elmist; // Pastikan menggunakan elmist  
10     P->info = x; // Menggunakan akses langsung  
11     P->next = NULL; // Menggunakan akses langsung  
12     return P;  
13 }  
14  
15 void insertFirst(List &L, address P) {  
16     P->next = L.first; // Menggunakan akses langsung  
17     L.first = P; // Menggunakan akses langsung  
18 }  
19  
20 void printInfo(List L) {  
21     address P = L.first; // Menggunakan akses langsung  
22     while (P != NULL) {  
23         cout << P->info << " "; // Menggunakan akses langsung  
24         P = P->next; // Menggunakan akses langsung  
25     }  
26     cout << endl;  
27 }
```

➤ Main.cpp

```
TP > main.cpp > main()
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5  int main() {
6      List L;
7      address P;
8      infotype x;
9
10     createList(L);
11
12     // Masukkan 3 digit NIM terakhir Anda di sini
13     x = 4; // Ganti dengan digit pertama
14     P = allocate(x);
15     insertFirst(L, P);
16
17     x = 0; // Ganti dengan digit kedua
18     P = allocate(x);
19     insertFirst(L, P);
20
21     x = 0; // Ganti dengan digit ketiga
22     P = allocate(x);
23     insertFirst(L, P);
24
25     cout << "Isi list: ";
26     printInfo(L);
27
28     return 0;
29 }
```

6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terakhir Anda di main.cpp Adapun gambaran isi dari main.cpp nya adalah sbb :

```
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5  int main()
6  {
7      // 1. Deklarasi create list
8
9
10     // 2. user input masukkan angka pertama yang akan dimasukkan user ke list
11
12     // 3. user input masukkan angka kedua yang akan dimasukkan user ke list
13
14     // 4. user input masukkan angka ketiga yang akan dimasukkan user ke list
15
16     // 5. user input masukkan angka keempat yang akan dimasukkan user ke list
17
18     // 6. user input masukkan angka kelima yang akan dimasukkan user ke list
19
20     // 7. user input masukkan angka keenam yang akan dimasukkan user ke list
21
22     return 0;
23 }
```

Tugas rekan-rekan adalah mengisi main.cpp di atas dengan sintak C++ sesuai dengan petunjuk. Buka kembali modul sebelumnya ya ^_^. Setelah selesai, compile dan run. Lalu **screen capture hasilnya**.

➤ Hasil Outputnya:

```

Mode                LastWriteTime         Length Name
-----
d----- 14/10/2024 13:52             .vscode
d----- 14/10/2024 16:08             GUIDED
d----- 14/10/2024 16:08             output
d----- 17/10/2024 22:54             TP
d----- 16/10/2024 14:20             UNGUIDED

PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1> cd TP
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP> g++ main.cpp list.cpp -o linkedlist
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP> ./linkedlist
Isi list: 0 0 4
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP>

```

Pada output ini, nantinya akan menampilkan sebuah output 3 digit NIM terakhir. Dimana ada fungsi 'createList' yang biasanya menginsialisasikan

pointer head list menjadi NULL untuk menandakan list kosong. Ada fungsi 'allocate' dimana ini mengalokasikan memori untuk sebuah node baru dan mengisi data node tersebut dengan nilai x. lalu ada fungsi 'insertFirst' dimana ini menambahkan node baru ke posisi awal list. Dan ada fungsi 'printInfo' yang akan melakukan traversal list dari awal hingga akhir dan mencetak data setiap node.

7. SESI HAVE FUN. Rekan-rekan dapat mencoba hal di bawah ini agar memudahkan saat praktikum:
 - i. Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
 - ii. Tambahkan Function searchInfo pada list.h dan list.cpp
 - iii. Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data (clue : gunakan looping). Dan NIM yang diinput, saat di show tidak boleh terurut terbalik (clue : gunakan insert Last) Tampilan

(underscore adalah inputan user):

```

Masukkan NIM perdigit
Digit 1 : 1
Digit 2 : 1
Digit 3 : 3
Digit 4 : 1
Digit 5 : 9
Digit 6 : 6
Digit 7 : 4
Digit 8 : 7
Digit 9 : 4
Digit 10 : 2
Isi list : 1131964742
  
```

Jawab:

➤ File list.h

```

TP > C listh > ...
26 #ifndef LIST_H
27 #define LIST_H
28
29 #include <iostream>
30 using namespace std;
31
32 typedef int infotype;
33 typedef struct elmist *address;
34
35 struct elmist {
36     infotype info;
37     address next;
38 };
39
40 struct List {
41     address first;
42 };
43
44 void createlist(List &L);
45 address allocate(infotype x);
46 void insertFirst(List &L, address P);
47 void insertLast(List &L, address P); // Prosedur insertLast
48 void insertAfter(List &L, address P, address prev); // Prosedur insertAfter
49 void deleteLast(List &L); // Prosedur deleteLast
50 void deleteAfter(List &L, address prev); // Prosedur deleteAfter
51 address searchInfo(List L, infotype x); // Fungsi searchInfo
52 void printInfo(List L);
53
54 #endif // LIST_H
  
```

➤ File list.cpp

```

C- Guided01.cpp  C- Guided02.cpp  C- soal01.cpp  C- soal02.cpp
TP > C- list.cpp > printInfo(List)

29 #include "list.h"
30
31 void createList(List &L) {
32     L.first = NULL;
33 }
34
35 address allocate(intofype x) {
36     address P = new elmist;
37     P->info = x;
38     P->next = NULL;
39     return P;
40 }
41
42 void insertFirst(List &L, address P) {
43     P->next = L.first;
44     L.first = P;
45 }
46
47 void insertLast(List &L, address P) {
48     if (L.first == NULL) {
49         L.first = P; // Jika list kosong, elemen pertama
50     } else {
51         address last = L.first;
52         while (last->next != NULL) {
53             last = last->next; // Mencari elemen terakhir
54         }
55         last->next = P; // Menambahkan elemen baru di akhir
56     }
57 }
58
59 void insertAfter(List &L, address P, address prev) {

```

```

60     if (prev != NULL) {
61         P->next = prev->next;
62         prev->next = P;
63     }
64 }
65
66 void deleteLast(List &L) {
67     if (L.first != NULL) {
68         if (L.first->next == NULL) {
69             delete L.first;
70             L.first = NULL; // List menjadi kosong
71         } else {
72             address last = L.first;
73             address secondLast = NULL;
74             while (last->next != NULL) {
75                 secondLast = last;
76                 last = last->next; // Mencari elemen terakhir
77             }
78             delete last; // Menghapus elemen terakhir
79             if (secondLast != NULL) {
80                 secondLast->next = NULL; // Mengubah next dari elemen kedua terakhir menjadi NULL
81             }
82         }
83     }
84 }
85
86 void deleteAfter(List &L, address prev) {
87     if (prev != NULL && prev->next != NULL) {
88         address toDelete = prev->next;
89         prev->next = toDelete->next;

```

```

90         delete toDelete;
91     }
92 }
93
94 address searchInfo(List L, intofype x) {
95     address P = L.first;
96     while (P != NULL) {
97         if (P->info == x) {
98             return P; // Mengembalikan alamat elemen yang ditemukan
99         }
100         P = P->next;
101     }
102     return NULL; // Jika tidak ditemukan
103 }
104
105 void printInfo(List L) {
106     address P = L.first;
107     while (P != NULL) {
108         cout << P->info; // Tidak ada koma di akhir
109         P = P->next; // Melanjutkan ke elemen berikutnya
110     }
111     cout << endl;
112 }

```

➤ File main.cpp

```
TP > main.cpp > main()
30
31 #include <iostream>
32 #include "list.h"
33 using namespace std;
34
35 int main() {
36     List L;
37     address P;
38     infotype x;
39
40     createList(L);
41
42     // Masukkan 10 digit NIM
43     for (int i = 1; i <= 10; i++) {
44         cout << "Digit " << i << " : ";
45         cin >> x; // Input digit NIM
46         P = allocate(x);
47         insertLast(L, P); // Menggunakan insertLast untuk memasukkan di akhir
48     }
49
50     cout << "Isi list: ";
51     printInfo(L);
52
53     return 0;
54 }
```

➤ Hasil outputnya:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
d----- 14/10/2024 16.08 GUIDED
d----- 14/10/2024 16.08 GUIDED
d----- 14/10/2024 16.08 output
d----- 14/10/2024 16.08 output
d----- 17/10/2024 22.58 TP
d----- 16/10/2024 14.20 UNGUIDED
d----- 16/10/2024 14.20 UNGUIDED

PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1> cd TP
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP> g++ main.cpp list.cpp -o program
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP> ./program
Digit 1 : 2
Digit 2 : 3
Digit 3 : 1
Digit 4 : 1
Digit 5 : 1
Digit 6 : 0
Digit 7 : 4
Digit 8 : 0
Digit 9 : 0
Digit 10 : 4
Isi list: 2311104004
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\04_Single_Linked_List_Bagian_1\TP> 
```

Pada output ini nantinya akan menampilkan output NIM dengan 10 digit. Dimana ini memiliki fungsi 'insertLast' yang menambahkan node baru ke posisi terakhir dalam list. Dan ada loop 'for' yang digunakan untuk mengulang proses input dan penambahan node sebanyak 10 kali.