

LAPORAN PRAKTIKUM
Modul 04
“Single Linked List Bagian Pertama”



Disusun Oleh:
Yoga Eka Pratama - 2311104023
Kelas
SE-07-1
Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tugas Pendahuluan

1. Membuat deklarasi tipe list

list.h

```
h list.h x
h list.h > List > first
1  #ifndef LIST_H
2  #define LIST_H
3
4  #include <iostream>
5  #define first(L) L.first
6  #define next(P) P->next
7  #define info(P) P->info
8
9  using namespace std;
10
11 typedef int infotype;
12 typedef struct elmList *address;
13
14 struct elmList {
15     infotype info;
16     address next;
17 };
18
19 struct List {
20     address first;
21 };
22
```

list.cpp

```
C++ list.cpp x
C++ list.cpp > ...
1  #include "list.h"
2  #include <iostream>
3
4  using namespace std;
5
```

2. Membuat List kosong, yaitu procedure createList

```
5
6  void createList(List &L) {
7      first(L) = NULL;
8  }
9
```

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

```
9
10 address allocate(infotype x) {
11     address p = new elmList;
12     info(p) = x;
13     next(p) = NULL;
14     return p;
15 }
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elem list

```

16
17 void insertFirst(List &L, address P) {
18     next(P) = first(L);
19     first(L) = P;
20 }
21

```

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

```

22 void printInfo(List L) {
23     address p = first(L);
24     while (p != NULL) {
25         cout << info(p) << ", ";
26         p = next(p);
27     }
28     cout << endl;
29 }
30

```

6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedure di atas, maka mari buat sebuah list berisi 3 elemen yang berisi 3 digit nim terakhir nim anda di main.cpp

Main.cpp

```

C++ main.cpp x
C++ main.cpp > main()
1  #include <iostream>
2  #include "list.h"
3  #include "list.cpp"
4
5  using namespace std;
6
7  int main() {
8      List L;
9      createList(L);
10
11      int data;
12      cout << "Masukkan angka pertama yang ingin diinput ke List: ";
13      cin >> data;
14      address P = allocate(data);
15      insertFirst(L, P);
16      cout << "Isi list setelah input pertama: ";
17      printInfo(L);
18
19      cout << "Masukkan angka kedua yang ingin diinput ke List: ";
20      cin >> data;
21      P = allocate(data);
22      insertFirst(L, P);
23      cout << "Isi list setelah input kedua: ";
24      printInfo(L);
25
26      cout << "Masukkan angka ketiga yang ingin diinput ke List: ";
27      cin >> data;
28      P = allocate(data);
29      insertFirst(L, P);
30      cout << "Isi list setelah input ketiga: ";
31      printInfo(L);
32
33      return 0;
34
35

```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
cd "/Users/macairml/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/output"
./"main"
macairml@MacBook-Air-MacAirM1 TP % cd "/Users/macairml/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/output"
macairml@MacBook-Air-MacAirM1 output % ./"main"
Masukkan angka pertama yang ingin diinput ke List: 0
Isi list setelah input pertama: 0,
Masukkan angka kedua yang ingin diinput ke List: 2
Isi list setelah input kedua: 2, 0,
Masukkan angka ketiga yang ingin diinput ke List: 3
Isi list setelah input ketiga: 3, 2, 0,
macairml@MacBook-Air-MacAirM1 output %
```

7. Sesi Have Fun

List.cpp

```
C++ list.cpp X
sesi have fun > C++ list.cpp > ...
1 #include "list.h"
2 #include <iostream>
3 using namespace std;
4
5 // menggunakan :: untuk inialisasi method di luar list.h
6 List::List() {
7     head = nullptr;
8 }
9 List::~List() {
10     while (head != nullptr) {
11         Node* temp = head;
12         head = head->next;
13         delete temp;
14     }
15 }
16 void List::insertLast(int data) {
17     Node* newNode = new Node;
18     newNode->data = data;
19     newNode->next = nullptr;
20
21     if (head == nullptr) {
22         head = newNode;
23     } else {
24         Node* temp = head;
25         while (temp->next != nullptr) {
26             temp = temp->next;
27         }
28         temp->next = newNode;
29     }
30 }
31
32 void List::insertAfter(int key, int data) {
33     Node* temp = head;
34     while (temp != nullptr && temp->data != key) {
35         temp = temp->next;
36     }
37     if (temp != nullptr) {
38         Node* newNode = new Node;
39         newNode->data = data;
40         newNode->next = temp->next;
41         temp->next = newNode;
42     }
43 }
```

```
C++ list.cpp X
sesi have fun > C++ list.cpp > searchInfo(int)

45 void List::deleteLast() {
46     if (head == nullptr) return;
47
48     if (head->next == nullptr) {
49         delete head;
50         head = nullptr;
51     } else {
52         Node* temp = head;
53         while (temp->next->next != nullptr) {
54             temp = temp->next;
55         }
56         delete temp->next;
57         temp->next = nullptr;
58     }
59 }
60 void List::deleteAfter(int key) {
61     Node* temp = head;
62     while (temp != nullptr && temp->data != key) {
63         temp = temp->next;
64     }
65     if (temp != nullptr && temp->next != nullptr) {
66         Node* delNode = temp->next;
67         temp->next = temp->next->next;
68         delete delNode;
69     }
70 }
71 Node* List::searchInfo(int key) {
72     Node* temp = head;
73     while (temp != nullptr) {
74         if (temp->data == key)
75             return temp;
76         temp = temp->next;
77     }
78     return nullptr;
79 }
80 void List::display() {
81     Node* temp = head;
82     while (temp != nullptr) {
83         cout << temp->data;
84         temp = temp->next;
85     }
86     cout << endl;
87 }
```

List.h

```
C++ list.cpp list.h X
sesi have fun > h list.h > ...

1 #ifndef LIST_H
2 #define LIST_H
3
4 struct Node {
5     int data;
6     Node* next;
7 };
8
9 class List {
10 private:
11     Node* head;
12 public:
13     List();
14     ~List();
15     void insertLast(int data);
16     void insertAfter(int key, int data);
17     void deleteLast();
18     void deleteAfter(int key);
19     Node* searchInfo(int key);
20     void display();
21 };
22
23 #endif
24
```

Main.cpp

```
C++ list.cpp C++ main.cpp X
sesi have fun > C++ main.cpp > ...
1 #include <iostream>
2 #include "list.h"
3 #include "list.cpp"
4
5 using namespace std;
6
7 int main() {
8     List myList;
9     int NIM[10];
10
11     cout << "Masukkan NIM perdigit\n";
12     for (int i = 0; i < 10; ++i) {
13         cout << "Digit " << i + 1 << " : ";
14         cin >> NIM[i];
15         myList.insertLast(NIM[i]);
16     }
17     cout << "Isi list : ";
18     myList.display();
19
20     return 0;
21 }
22
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● macairm1@MacBook-Air-MacAirM1 output % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/sesi have fun/output"
"
./main"
● macairm1@MacBook-Air-MacAirM1 output % ./main"
Masukkan NIM perdigit
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 : 2
Digit 10 : 3
Isi list : 2311104023
○ macairm1@MacBook-Air-MacAirM1 output % █
```