

LAPORAN PRAKTIKUM
Tugas Pendahuluan Modul 4
“Single Linked List”



Disusun Oleh:
Muhammad Atsal Rizandri - 21104057
Struktur Data SE07-01 / SE05-02

Dosen :
Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024

1. Langkah-langkah modul TP dari langkah 1 sampai 6 sebagai berikut

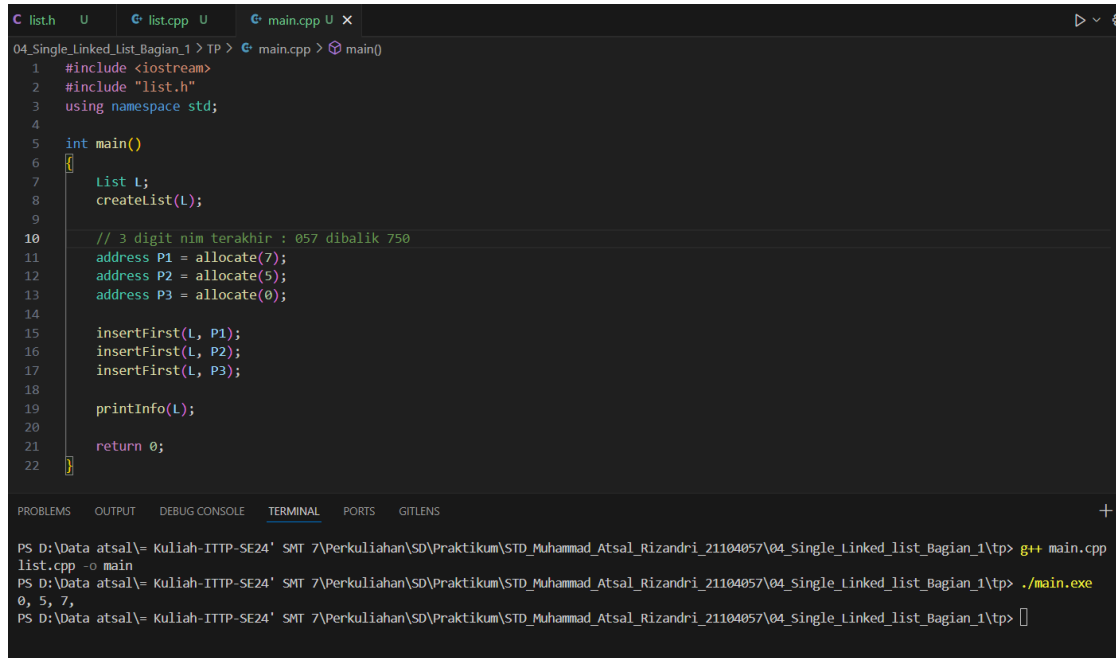
- File list.h

```
C list.h U x C list.cpp U C main.cpp U
04_Single_Linked_List_Bagian_1 > TP > C list.h > allocate(infotype)
1  #include <iostream>
2  #define first(L) L.first
3  #define next(P) P->next
4  #define info(P) P->info
5  using namespace std;
6
7  typedef int infotype;
8  typedef struct elmlist *address;
9
10 struct elmlist
11 {
12     infotype info;
13     address next;
14 };
15
16 struct List
17 {
18     address first;
19 };
20
21 void createList(List &L);
22
23 address allocate(infotype x);
24
25 void insertFirst(List &L, address P);
26
27 void printInfo(List L);
28
```

- File list.cpp.

```
C list.h U x C list.cpp U x C main.cpp U
04_Single_Linked_List_Bagian_1 > TP > C list.cpp > searchInfo(List, int)
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5  void createList(List &L)
6  {
7      // this procedure will initialize the list L
8      first(L) = NULL;
9  }
10
11 address allocate(infotype x)
12 {
13     address p = new elmlist;
14     info(p) = x;
15     next(p) = NULL;
16
17     return p;
18 }
19
20 void insertFirst(List &L, address P)
21 {
22     next(P) = first(L);
23     first(L) = P;
24 }
25
26 void printInfo(List L)
27 {
28     address p = first(L);
29     while (p != NULL)
30     {
31         cout << info(p) << ", ";
32         p = next(p);
33     }
34     cout << endl;
35 }
36
```

- File main.cpp dan output



```

C list.h U list.cpp U main.cpp U X
04_Single_Linked_List_Bagian_1 > TP > main.cpp > main()
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5  int main()
6  {
7      List L;
8      createList(L);
9
10     // 3 digit nim terakhir : 057 dibalik 750
11     address P1 = allocate(7);
12     address P2 = allocate(5);
13     address P3 = allocate(0);
14
15     insertFirst(L, P1);
16     insertFirst(L, P2);
17     insertFirst(L, P3);
18
19     printInfo(L);
20
21     return 0;
22 }

```

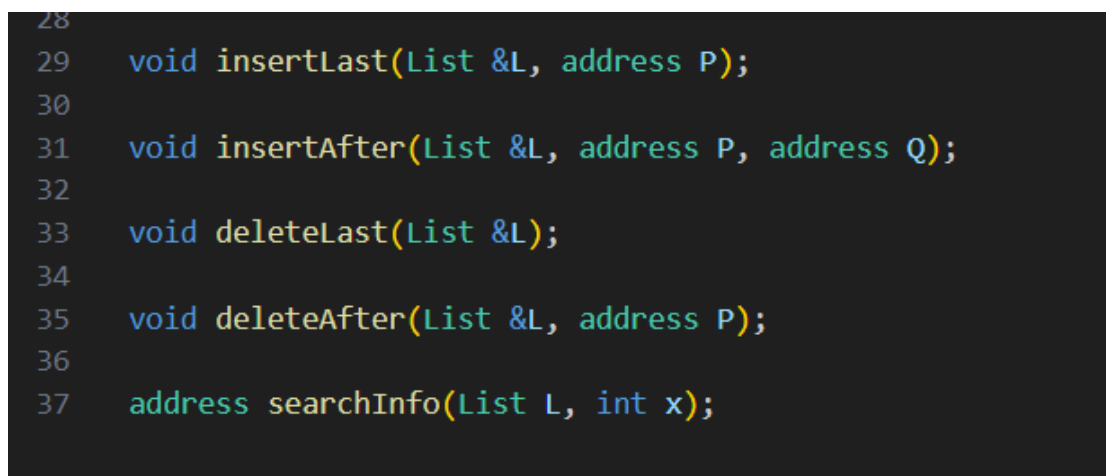
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS D:\Data atsal\ Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057\04_Single_Linked_list_Bagian_1\tp> g++ main.cpp
list.cpp -o main
PS D:\Data atsal\ Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057\04_Single_Linked_list_Bagian_1\tp> ./main.exe
0, 5, 7,
PS D:\Data atsal\ Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057\04_Single_Linked_list_Bagian_1\tp>

```

2. Sesi Have Fun. Menambahkan beberapa procedure di list.h dan menambahkan function di list.cpp, kemudian mengubah proses yang ada di main.cpp agar insert N data tidak satu persatu. Berikut file yang sudah diubah atau ditambahkan:

- File list.h



```

28
29 void insertLast(List &L, address P);
30
31 void insertAfter(List &L, address P, address Q);
32
33 void deleteLast(List &L);
34
35 void deleteAfter(List &L, address P);
36
37 address searchInfo(List L, int x);

```

- File list.cpp

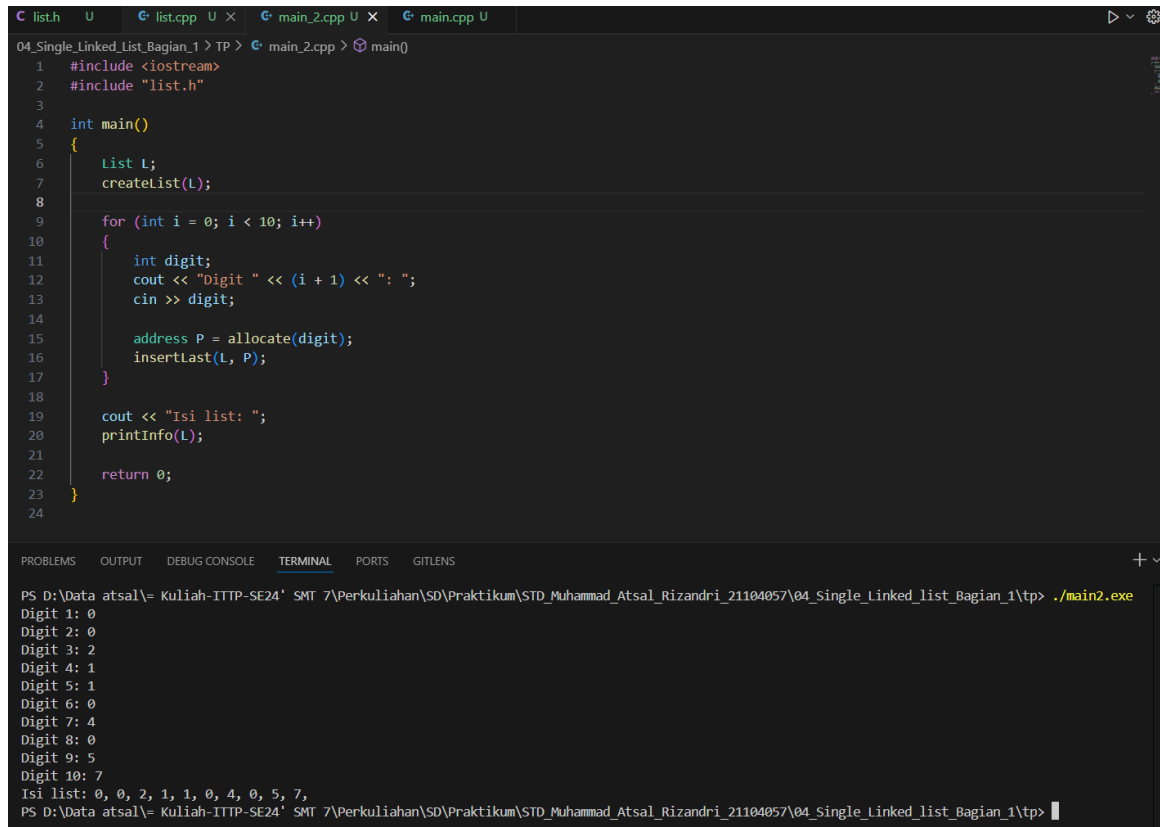
```

C listh U  list.cpp U X  main_2.cpp U  main.cpp U
04_Single_Linked_List_Bagian_1 > TP > list.cpp > ...

37 void insertLast(List &L, address P)
38 {
39     if (L.first == NULL)
40     {
41         L.first = P;
42     }
43     else
44     {
45         address Q = L.first;
46         while (Q->next != NULL)
47         {
48             Q = Q->next;
49         }
50         Q->next = P;
51     }
52 }
53
54 void insertAfter(List &L, address P, address Q)
55 {
56     P->next = Q->next;
57     Q->next = P;
58 }
59
60 void deleteLast(List &L)
61 {
62     if (L.first == NULL)
63     {
64         cout << "List kosong, tidak ada yang dihapus." << endl;
65     }
66     else if (L.first->next == NULL)
67     {
68         delete L.first;
69         L.first = NULL;
70     }
71     else
72     {
73         address Q = L.first;
74         while (Q->next->next != NULL)
75         {
76             Q = Q->next;
77         }
78         delete Q->next;
79         Q->next = NULL;
80     }
81 }
82
83 void deleteAfter(List &L, address P)
84 {
85     if (P != NULL && P->next != NULL)
86     {
87         address Q = P->next;
88         P->next = Q->next;
89         delete Q;
90     }
91 }
92
93 address searchInfo(List L, int x)
94 {
95     address P = L.first;
96     while (P != NULL)
97     {
98         if (P->info == x)
99         {
100             return P;
101         }
102         P = P->next;
103     }
104     return NULL;
105 }
106

```

- File main_2.cpp beserta outputnya



The screenshot displays a C++ IDE with two tabs: `list.h` and `main_2.cpp`. The `main_2.cpp` file is open, showing the following code:

```
1 #include <iostream>
2 #include "list.h"
3
4 int main()
5 {
6     List L;
7     createList(L);
8
9     for (int i = 0; i < 10; i++)
10    {
11        int digit;
12        cout << "Digit " << (i + 1) << ": ";
13        cin >> digit;
14
15        address P = allocate(digit);
16        insertLast(L, P);
17    }
18
19    cout << "Isi list: ";
20    printInfo(L);
21
22    return 0;
23 }
24
```

The terminal window at the bottom shows the execution of the program. The prompt is `PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057\04_Single_Linked_list_Bagian_1\tp>`. The user has run `./main2.exe`, and the output is as follows:

```
Digit 1: 0
Digit 2: 0
Digit 3: 2
Digit 4: 1
Digit 5: 1
Digit 6: 0
Digit 7: 4
Digit 8: 0
Digit 9: 5
Digit 10: 7
Isi list: 0, 0, 2, 1, 1, 0, 4, 0, 5, 7,
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057\04_Single_Linked_list_Bagian_1\tp>
```