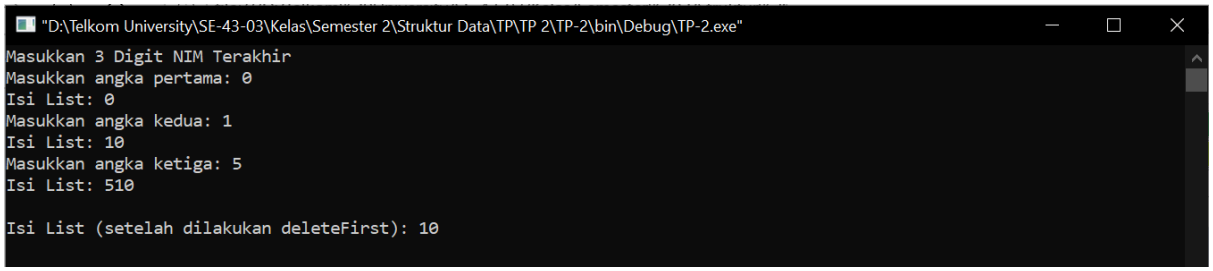
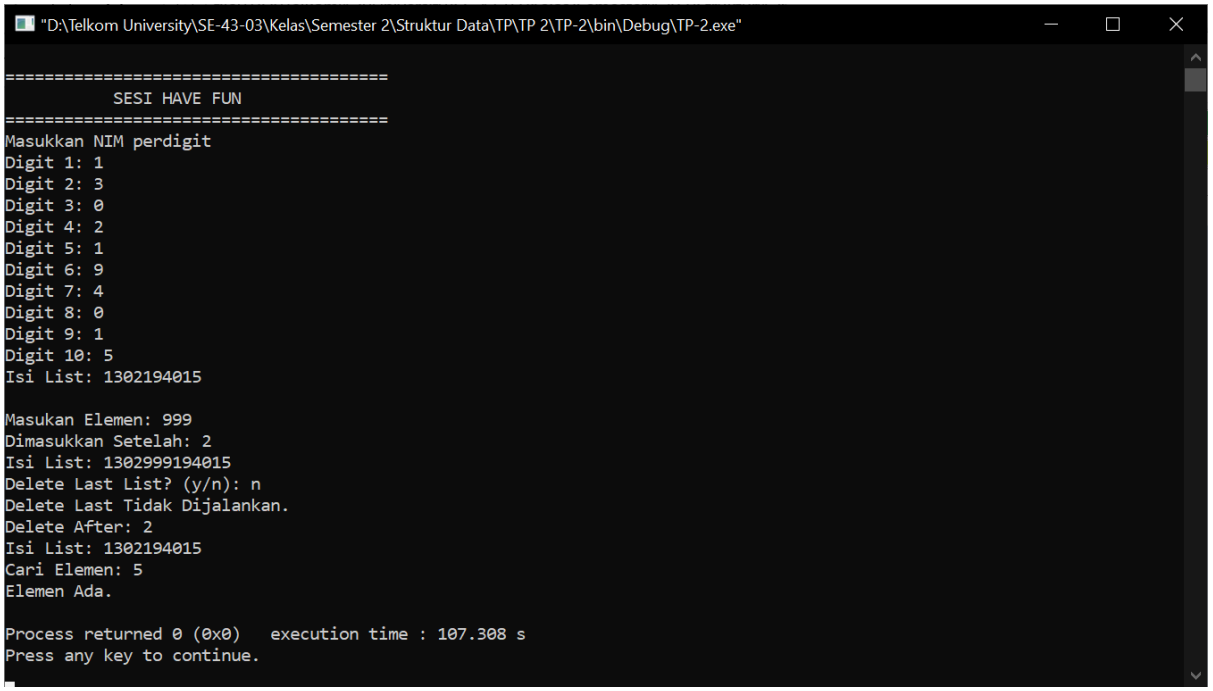

"D:\Telkom University\SE-43-03\Kelas\Semester 2\Struktur Data\TP\TP 2\TP-2\bin\Debug\TP-2.exe"
Masukkan 3 Digit NIM Terakhir
Masukkan angka pertama: 0
Isi List: 0
Masukkan angka kedua: 1
Isi List: 10
Masukkan angka ketiga: 5
Isi List: 510
Isi List (setelah dilakukan deleteFirst): 10


"D:\Telkom University\SE-43-03\Kelas\Semester 2\Struktur Data\TP\TP 2\TP-2\bin\Debug\TP-2.exe"
==========
SESI HAVE FUN
==========
Masukkan NIM perdigit
Digit 1: 1
Digit 2: 3
Digit 3: 0
Digit 4: 2
Digit 5: 1
Digit 6: 9
Digit 7: 4
Digit 8: 0
Digit 9: 1
Digit 10: 5
Isi List: 1302194015
Masukan Elemen: 999
Dimasukkan Setelah: 2
Isi List: 1302999194015
Delete Last List? (y/n): n
Delete Last Tidak Dijalankan.
Delete After: 2
Isi List: 1302194015
Cari Elemen: 5
Elemen Ada.
Process returned 0 (0x0) execution time : 107.308 s
Press any key to continue.

list.h | list.cpp | main.cpp

```
#ifndef LIST_H_INCLUDED
#define LIST_H_INCLUDED
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info

/*
Name : Muhamad Dwiki Riswanda
NIM : 1302194015
*/

using namespace std;

typedef int infotype;
typedef struct elmlist *address;

struct elmlist{
    infotype info;
    address next;
};

struct List{
    address first;
};

address allocate(infotype x);
```

list.h | list.cpp | main.cpp

```
22
23  struct List{
24      address first;
25  };
26
27  address allocate(infotype x);
28
29  void insertFirst(List &L, address P);
30
31  void printInfo(List L);
32
33  void deleteFirst(List &L, address &P);
34
35  //SESI HAVE FUN
36
37  void insertLast(List &L, address &P);
38
39  void insertAfter(List &L, address key, address &P);
40
41  void deleteLast(List &L);
42
43  void deleteAfter(List &L, address key);
44
45  infotype searchInfo(List L, address key);
46
47  #endif // LIST_H_INCLUDED
48
```

list.h | list.cpp | main.cpp

```cpp
#include <iostream>
#include "list.h"

/*
Name : Muhamad Dwiki Riswanda
NIM : 1302194015
*/

using namespace std;

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

address allocate(infotype x){
    address p = new elmlist;
    info(p) = x;
    next(p) = NULL;

    return p;
}

void insertFirst(List &L, address P){
    next(P) = first(L);
    first(L) = P;
}

void printInfo(List L){
```

list.h | list.cpp | main.cpp

```
25      first(L) = P;
26  }
27
28  void printInfo(List L){
29      address p = first(L);
30      while (p != NULL){
31          cout << info(p);
32          p = next(p);
33      };
34      cout << endl;
35  }
36
37  void deleteFirst(List &L, address &P){
38      if (first(L) != NULL){
39          P = first(L);
40          first(L) = next(P);
41          #define first(L) L.first
42      }
43  }
44
45  //SESI HAVE FUN
46
47  void insertLast(List &L, address P){
48      address temp;
49
50      if (first(L) != NULL){
51          temp = first(L);
52          while (next(temp) != NULL){
```

list.h | list.cpp | main.cpp

```
52          while (next(temp) != NULL){
53              temp = next(temp);
54          }
55          next(temp) = P;
56      } else {
57          next(P) = first(L);
58          first(L) = P;
59      }
60  }
61
62  void insertAfter(List &L, address key, address P){
63      address temp;
64
65      if (first(L) != NULL){
66          temp = first(L);
67          while (info(temp) != info(key)){
68              temp = next(temp);
69          }
70      }
71      if (info(temp) == info(key)){
72          next(P) = next(temp);
73          next(temp) = P;
74
75      } else {
76          next(temp) = NULL;
77          cout << "Key Not Found" << endl;
78      }
```

list.h | list.cpp | main.cpp

```
76              next(temp) = NULL;
77              cout << "Key Not Found" << endl;
78          }
79      }
80
81      void deleteLast(List &L){
82          address P;
83          address Q;
84
85          if (first(L) != NULL){
86              P = first(L);
87              while (next(P) != NULL){
88                  Q = P;
89                  P = next(P);
90              }
91              next(Q) = NULL;
92          }
93      }
94
95      void deleteAfter(List &L, address key){
96          address P;
97          address Q;
98          if (first(L) != NULL){
99              P = first(L);
100             Q = next(P);
101             while ((info(P) != info(key)) && (next(P) != NULL)) {
102                 P = next(P);
103                 Q = next(P);
```

list.h | list.cpp | main.cpp

```
104             }
105             if (info(P) == info(key)){
106                 next(P) = next(Q);
107                 next(Q) = NULL;
108             } else {
109                 next(Q) = NULL;
110             }
111         }
112     }
113
114 infotype searchInfo(List L, address key){
115     address temp;
116     if (first(L) != NULL){
117         temp = first(L);
118         while ((info(temp) != info(key)) && (next(temp) != NULL)){
119             temp = next(temp);
120         }
121         if (info(temp) == info(key)){
122             return info(temp);
123         } else {
124             return 0;
125         }
126     } else {
127         return 0;
128     }
129 }
130
```

list.h | list.cpp | *main.cpp

```cpp
#include <iostream>
#include "list.cpp"

/*
Name : Muhamad Dwiki Riswanda
NIM : 1302194015
*/

using namespace std;

int main()
{
    infotype x;
    List L;
    address p;
    createList(L);
    cout << "Masukkan 3 Digit NIM Terakhir"<< endl;
    cout << "Masukkan angka pertama: ";
    cin >> x;

    p = allocate(x);
    insertFirst(L,p);
    cout << "Isi List: ";
    printInfo(L);

    cout << "Masukkan angka kedua: ";
    cin >> x;
```

list.h | list.cpp | *main.cpp

```
25
26      cout << "Masukkan angka kedua: ";
27      cin >> x;
28
29      p = allocate(x);
30      insertFirst(L,p);
31      cout << "Isi List: ";
32      printInfo(L);
33
34      cout << "Masukkan angka ketiga: ";
35      cin >> x;
36
37      p = allocate(x);
38      ins address allocate(infotype x)
39      cout << "Isi List: ";
40      printInfo(L);
41
42      cout << endl;
43      deleteFirst(L, p);
44      cout << "Isi List (setelah dilakukan deleteFirst): ";
45      printInfo(L);
46      cout << endl;
47
48      cout << "========================================" << endl;
49      cout << "            SESI HAVE FUN" << endl;
50      cout << "========================================" << endl;
51
```

list.h | list.cpp | *main.cpp

```
47
48      cout << "=======================================" << endl;
49      cout << "            SESI HAVE FUN" << endl;
50      cout << "=======================================" << endl;
51
52      string s;
53      infotype b, d, e;
54      List c;
55      address a;
56      createList(c);
57      int i = 1;
58      cout << "Masukkan NIM perdigit"<< endl;
59      while (i <= 10){
60          cout << "Digit "<< i << ": ";
61          cin >> b;
62          insertLast(c,allocate(b));
63          i++;
64      }
65
66      cout << "Isi List: ";
67      printInfo(c);
68
69      cout << endl;
70      cout << "Masukan Elemen: ";
71      cin  >> b;
72      cout << "Dimasukkan Setelah: ";
73      cin  >> d;
74      if ( searchInfo(c,allocate(d)) == d ){
```

list.h | list.cpp | *main.cpp

```
68
69      cout << endl;
70      cout << "Masukan Elemen: ";
71      cin  >> b;
72      cout << "Dimasukkan Setelah: ";
73      cin  >> d;
74      if ( searchInfo(c,allocate(d)) == d ){
75          insertAfter(c,allocate(d),allocate(b));
76          cout << "Isi List: ";
77          printInfo(c);
78      } else {
79          cout << "KEY NOT FOUND"<<endl;
80      }
81      cout << "Delete Last List? (y/n): ";
82      cin >> s;
83      if (s == "y"){
84          deleteLast(c);
85          void deleteLast(List& L)
86          printInfo(c);
87      } else {
88          cout << "Delete Last Tidak Dijalankan."<< endl;
89      }
90      cout << "Delete After: ";
91      cin >> b ;
92      if ( searchInfo(c,allocate(b)) == b ){
93          deleteAfter(c,allocate(b));
94          cout << "Isi List: ";
95          printInfo(c);
```

list.h | list.cpp | *main.cpp

```
83      if (s == "y"){
84          deleteLast(c);
85          cout << "Isi List: ";
86          printInfo(c);
87      } else {
88          cout << "Delete Last Tidak Dijalankan."<< endl;
89      }
90      cout << "Delete After: ";
91      cin >> b ;
92      if ( searchInfo(c,allocate(b)) == b ){
93          deleteAfter(c,allocate(b));
94          void deleteAfter(List& L, address key)
95          printInfo(c);
96      } else {
97          cout << "KEY NOT FOUND"<<endl;
98      }
99      cout << "Cari Elemen: ";
100     cin >> d;
101     b = searchInfo(c,allocate(d));
102     if (b == d){
103         cout << "Elemen Ada."<< endl;
104     }  else {
105         cout << "Maaf, Elemen Tidak Ada."<<endl;
106     }
107
108     return 0;
109 }
```