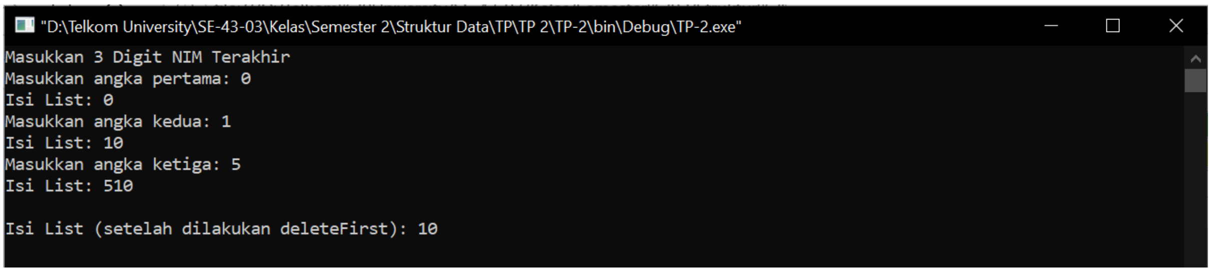
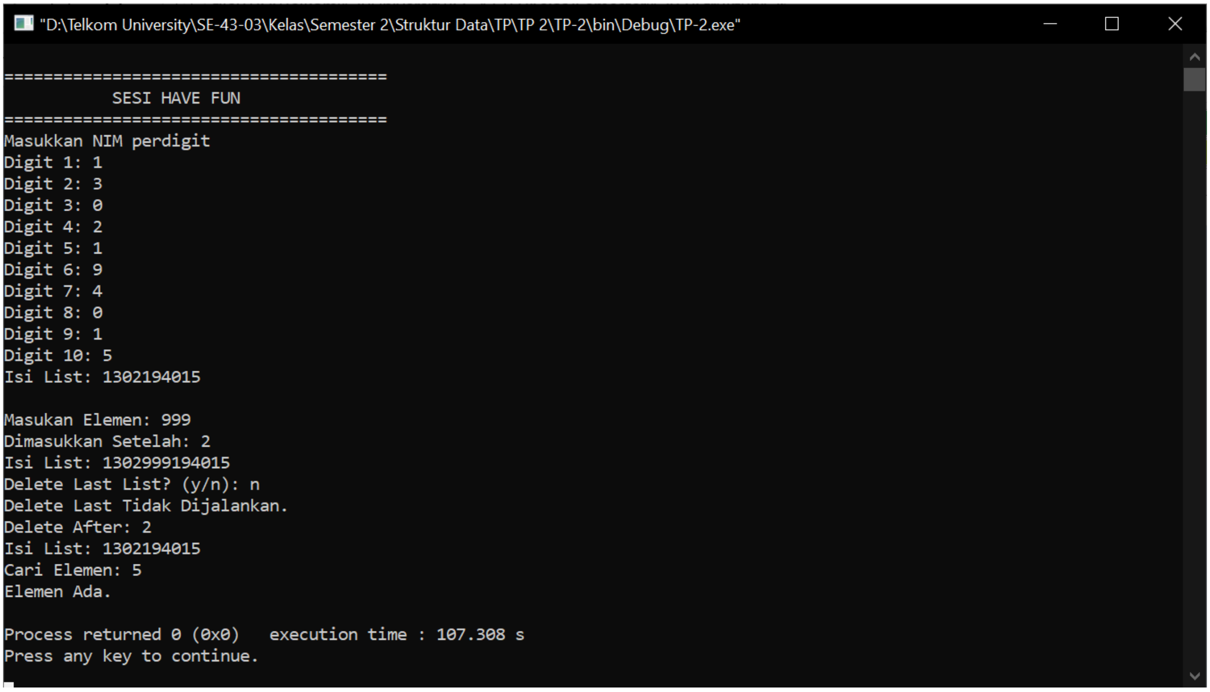
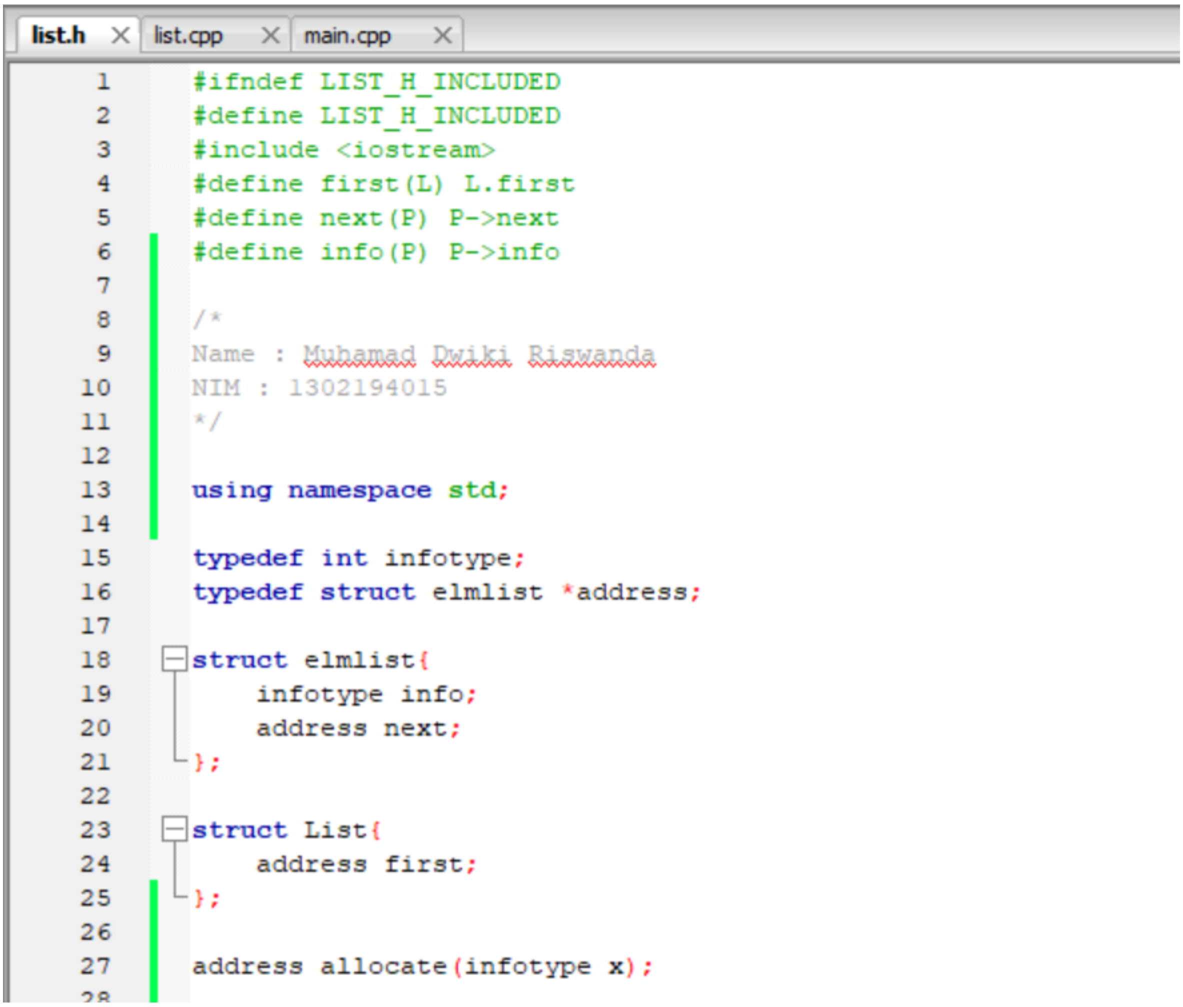

"D:\Telkom University\SE-43-03\Kelas\Semester 2\Struktur Data\TP\TP 2\TP-2\bin\Debug\TP-2.exe"

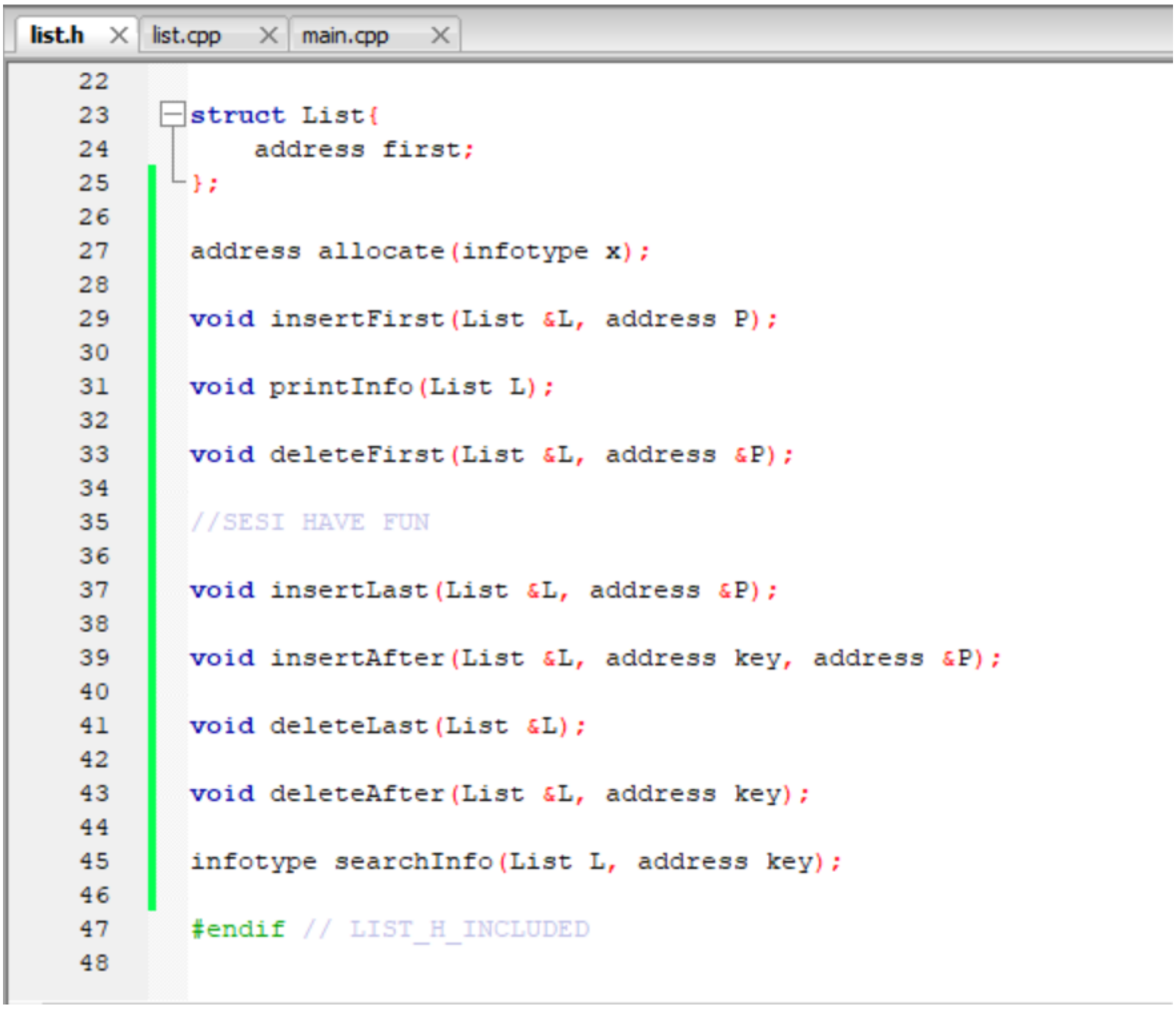
```
Masukkan 3 Digit NIM Terakhir
Masukkan angka pertama: 0
Isi List: 0
Masukkan angka kedua: 1
Isi List: 10
Masukkan angka ketiga: 5
Isi List: 510

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"

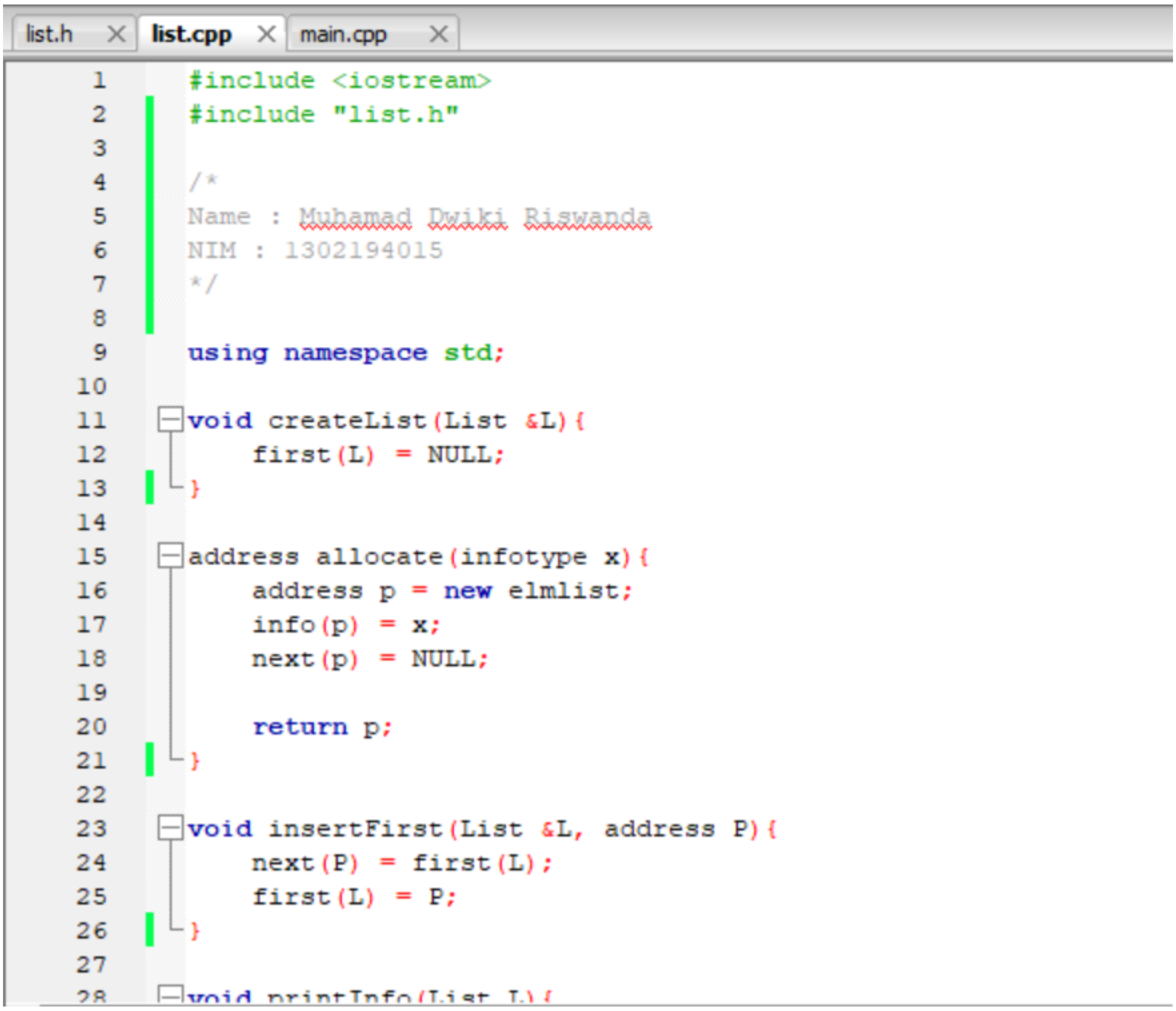
```

=======================================
           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.
```



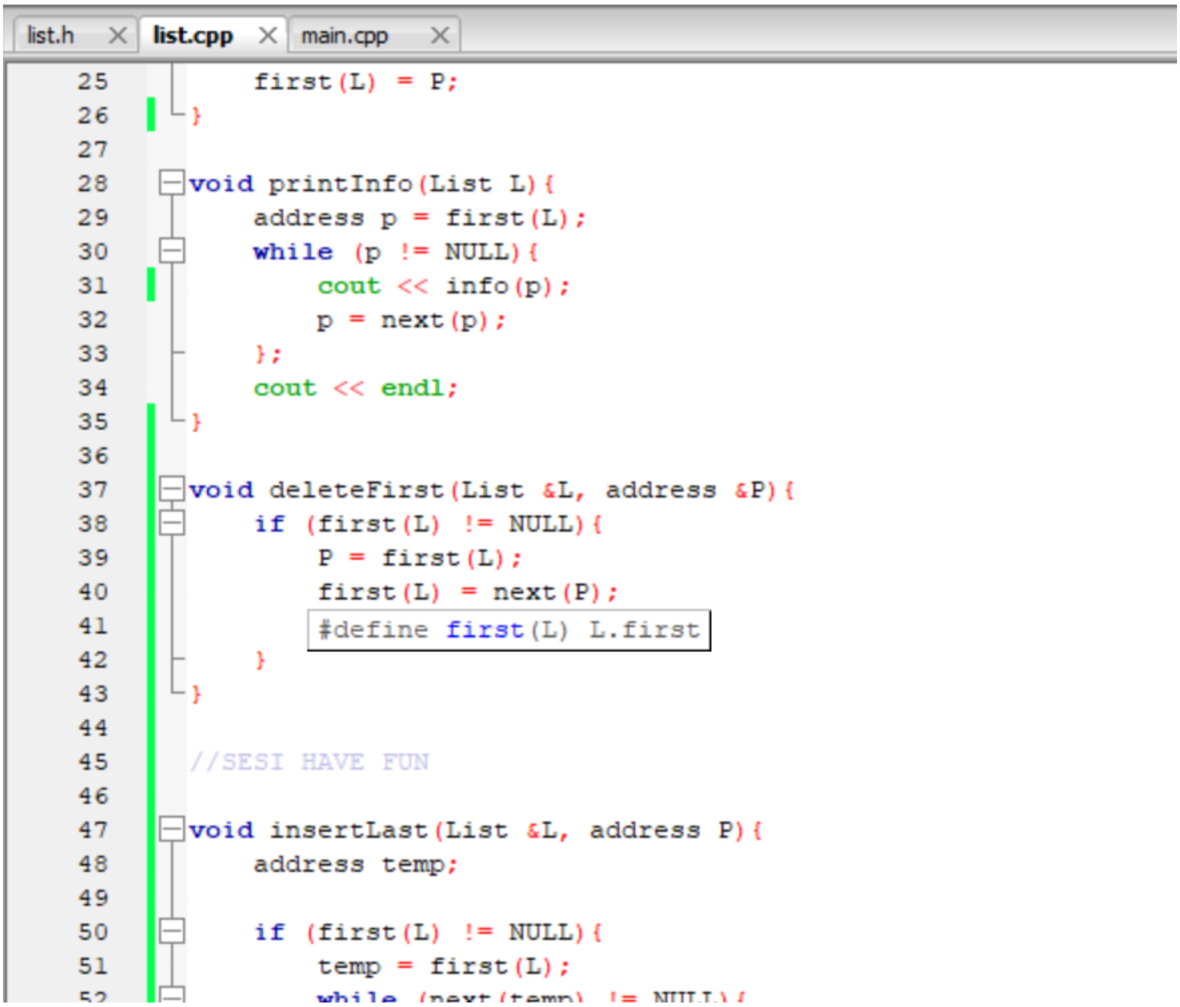
```
#ifndef LIST_H_INCLUDED
#define LIST_H_INCLUDED
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info

/*
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);
```



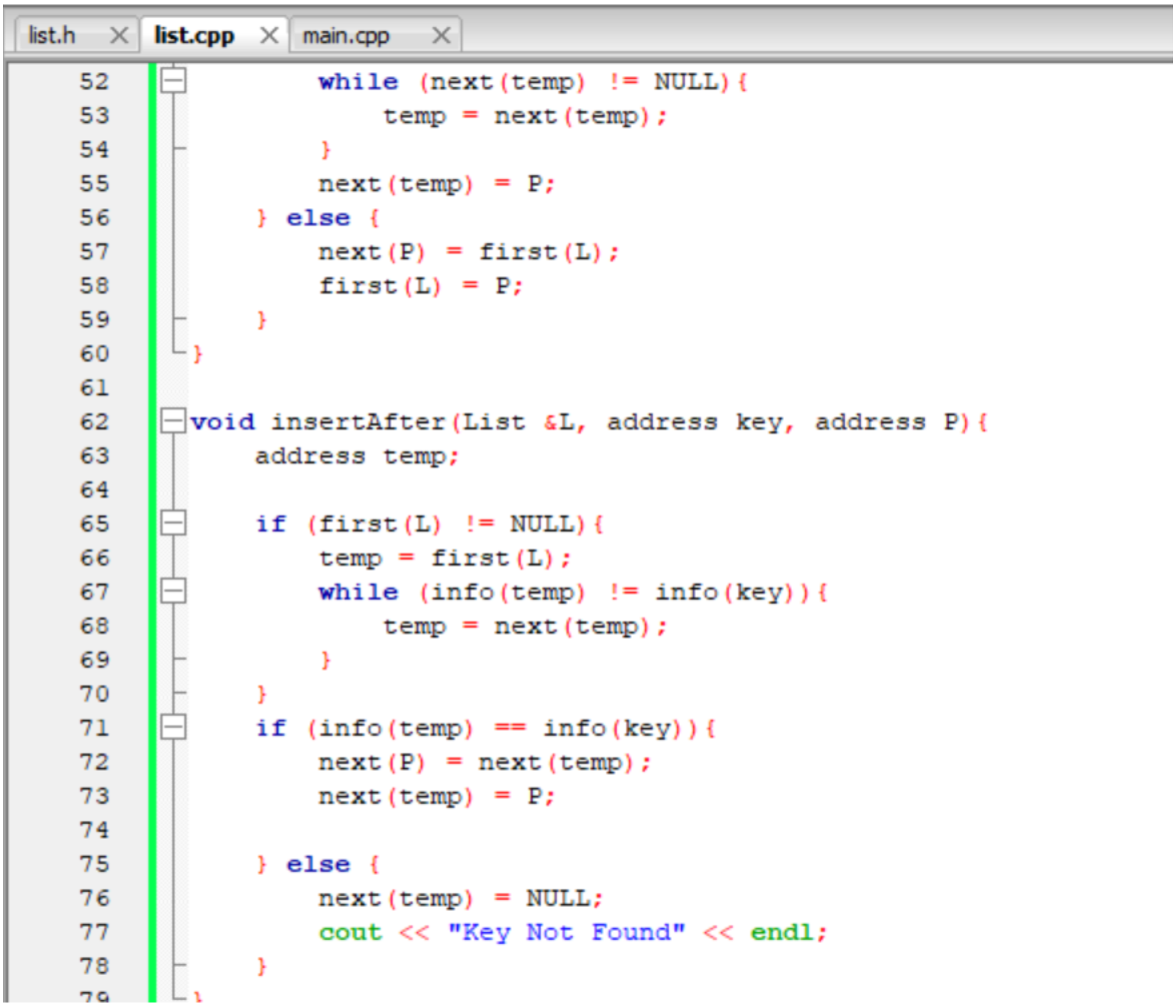
```

struct List{
    address first;
};

address allocate(infotype x);

void insertFirst(List &L, address P);

void printInfo(List L);

void deleteFirst(List &L, address &P);

//SESI HAVE FUN

void insertLast(List &L, address &P);

void insertAfter(List &L, address key, address &P);

void deleteLast(List &L);

void deleteAfter(List &L, address key);

infotype searchInfo(List L, address key);

#endif // LIST_H_INCLUDED

```



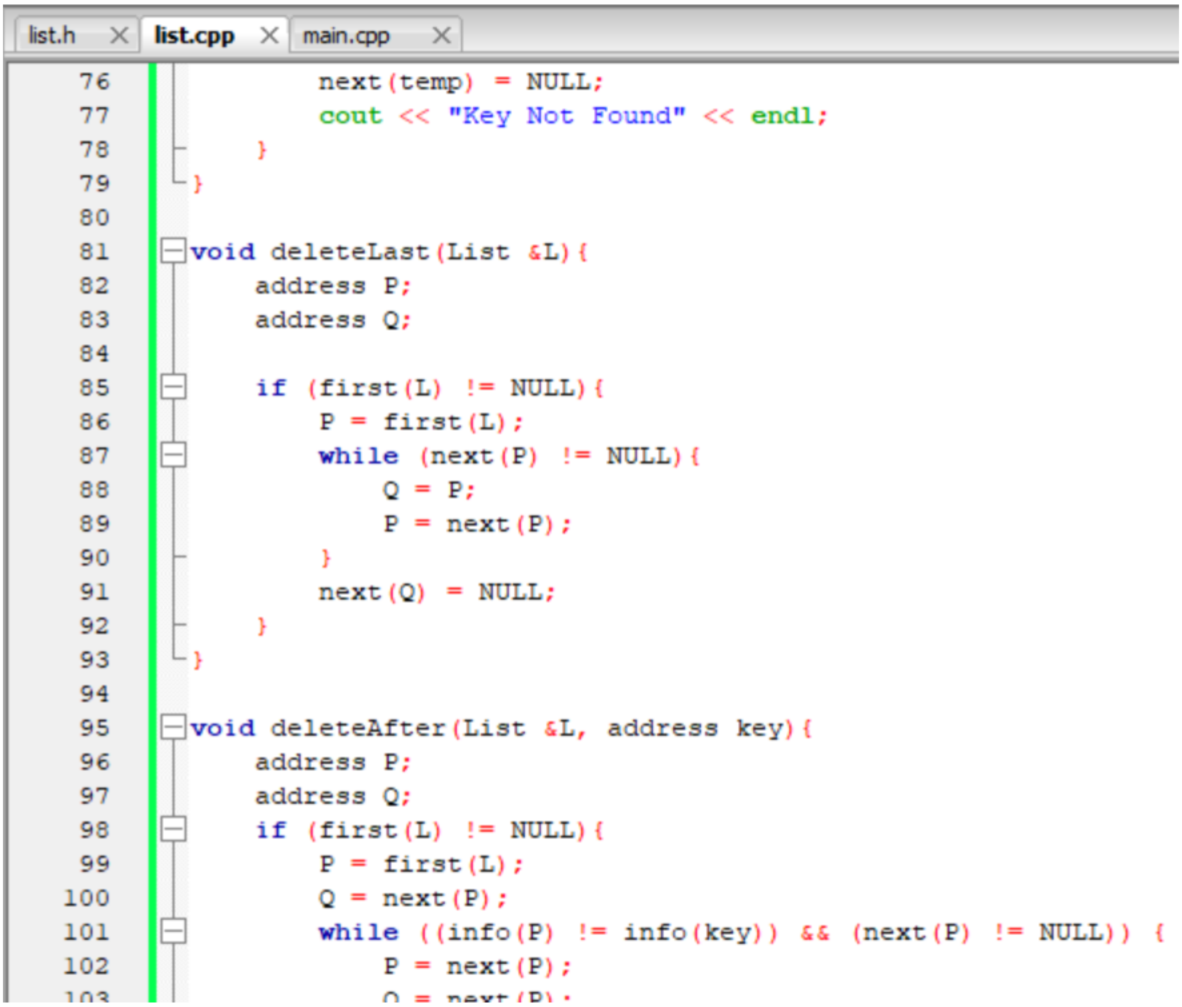
```
#include <iostream>
#include "list.h"

/*
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){
```



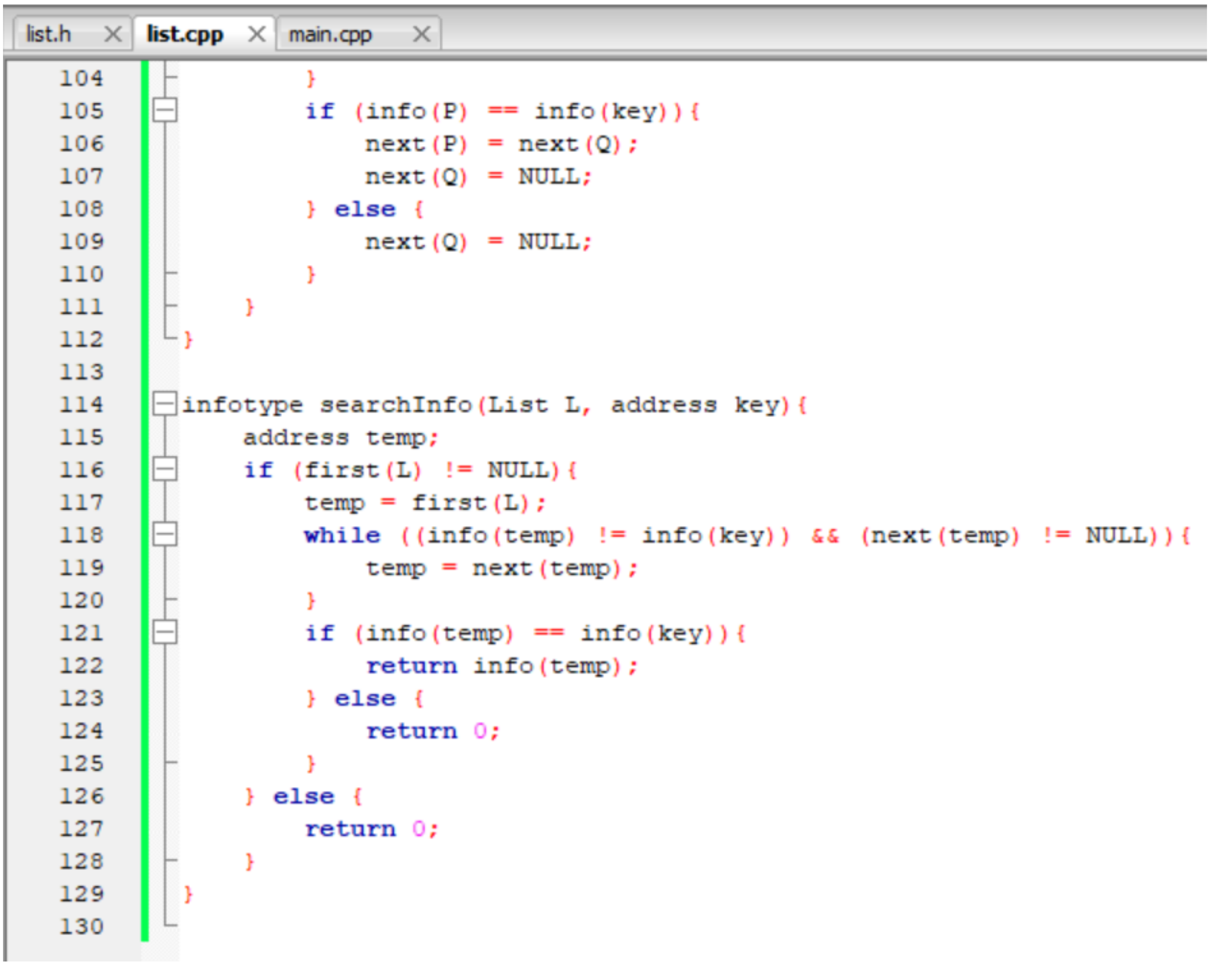
```
        first(L) = P;
}

void printInfo(List L){
    address p = first(L);
    while (p != NULL){
        cout << info(p);
        p = next(p);
    };
    cout << endl;
}

void deleteFirst(List &L, address &P){
    if (first(L) != NULL){
        P = first(L);
        first(L) = next(P);
        #define first(L) L.first
    }
}

//SESI HAVE FUN

void insertLast(List &L, address P){
    address temp;

    if (first(L) != NULL){
        temp = first(L);
        while (next(temp) != NULL){
```



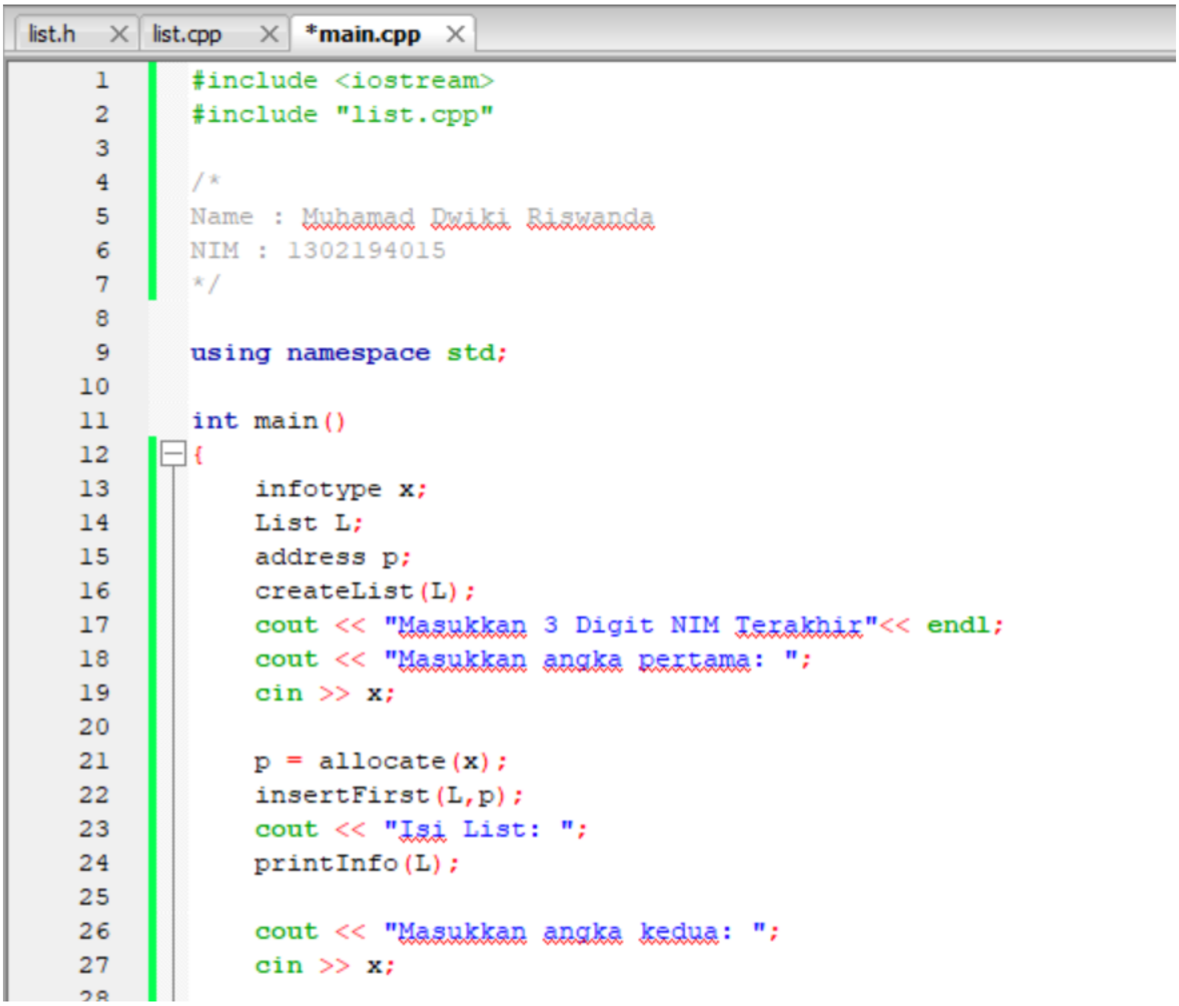
```
        while (next(temp) != NULL){
            temp = next(temp);
        }
        next(temp) = P;
    } else {
        next(P) = first(L);
        first(L) = P;
    }
}

void insertAfter(List &L, address key, address P){
    address temp;

    if (first(L) != NULL){
        temp = first(L);
        while (info(temp) != info(key)){
            temp = next(temp);
        }
    }
    if (info(temp) == info(key)){
        next(P) = next(temp);
        next(temp) = P;

    } else {
        next(temp) = NULL;
        cout << "Key Not Found" << endl;
    }
}
```



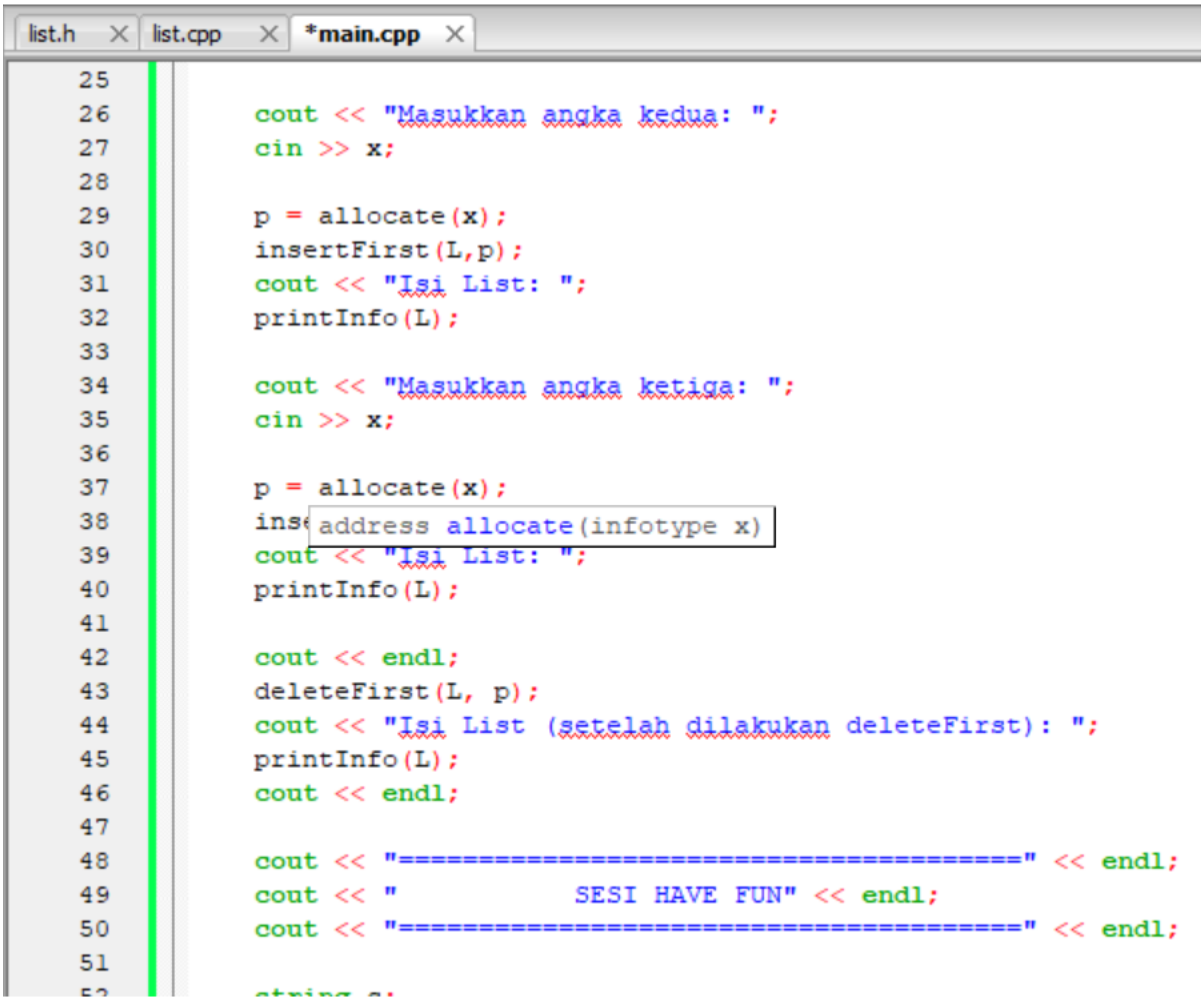
```cpp
        next(temp) = NULL;
        cout << "Key Not Found" << endl;
    }
}

void deleteLast(List &L){
    address P;
    address Q;

    if (first(L) != NULL){
        P = first(L);
        while (next(P) != NULL){
            Q = P;
            P = next(P);
        }
        next(Q) = NULL;
    }
}

void deleteAfter(List &L, address key){
    address P;
    address Q;
    if (first(L) != NULL){
        P = first(L);
        Q = next(P);
        while ((info(P) != info(key)) && (next(P) != NULL)) {
            P = next(P);
            Q = next(P);
```



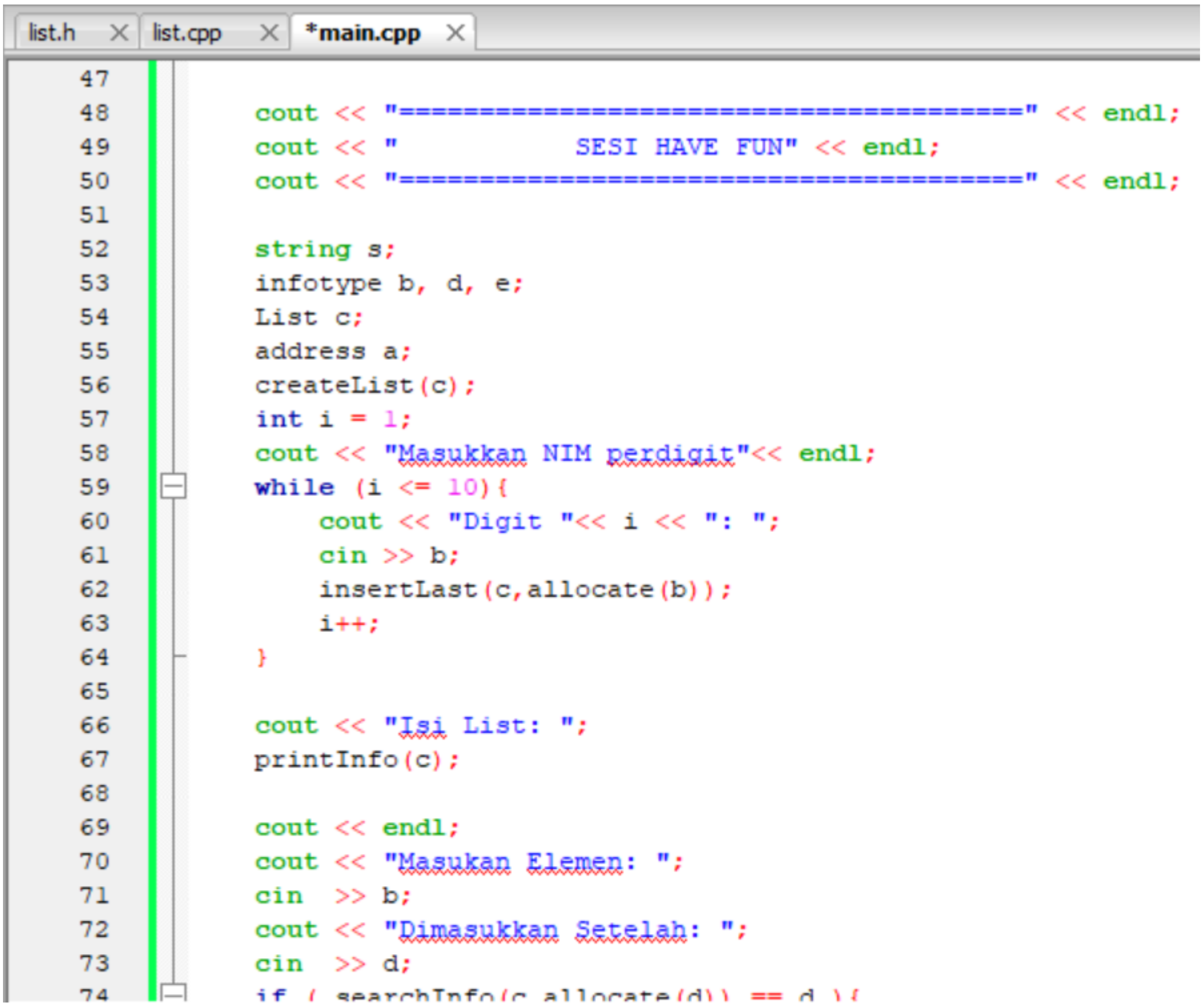
```
        }
        if (info(P) == info(key)){
            next(P) = next(Q);
            next(Q) = NULL;
        } else {
            next(Q) = NULL;
        }
    }
}

infotype searchInfo(List L, address key){
    address temp;
    if (first(L) != NULL){
        temp = first(L);
        while ((info(temp) != info(key)) && (next(temp) != NULL)){
            temp = next(temp);
        }
        if (info(temp) == info(key)){
            return info(temp);
        } else {
            return 0;
        }
    } else {
        return 0;
    }
}
```



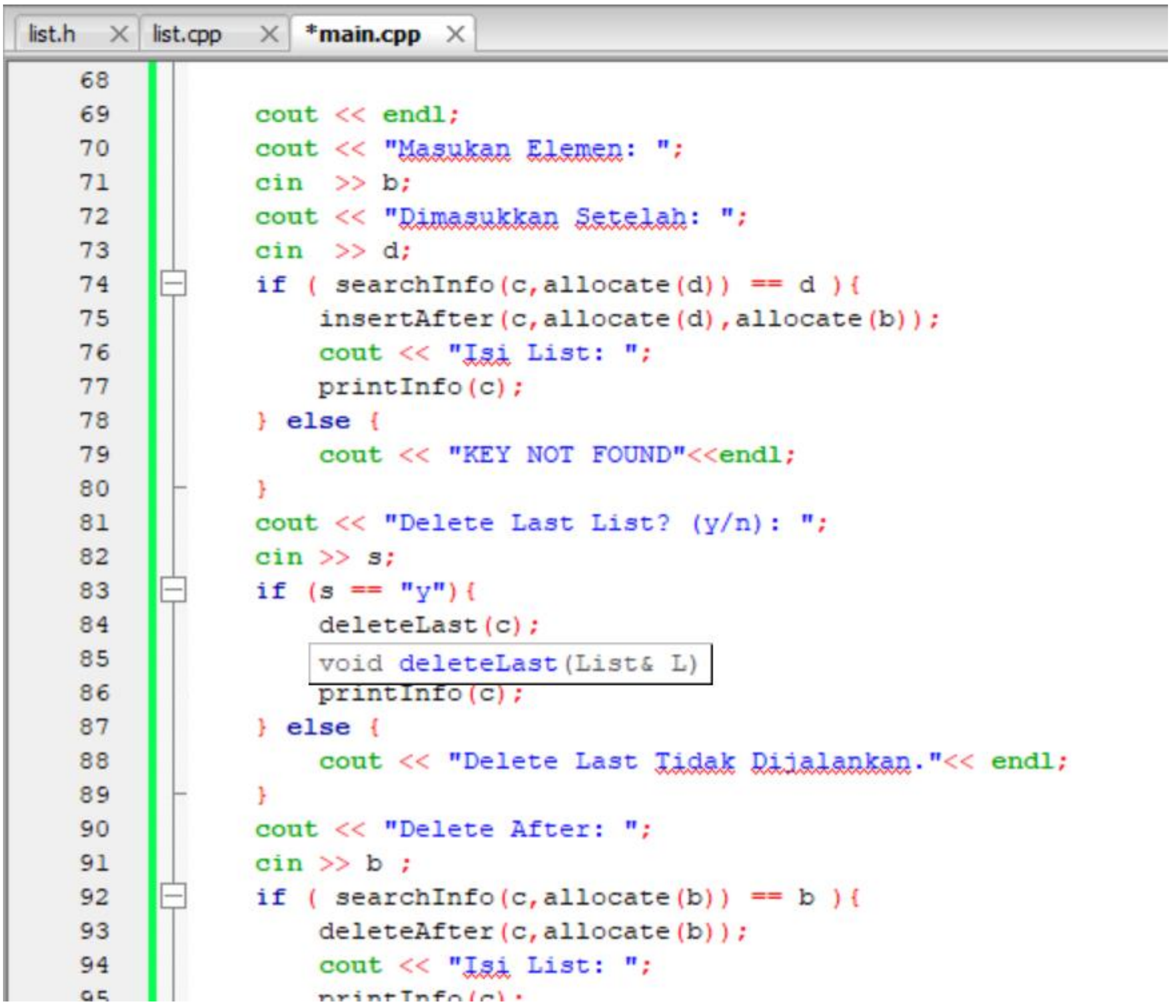
```
#include <iostream>
#include "list.cpp"

/*
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;
```



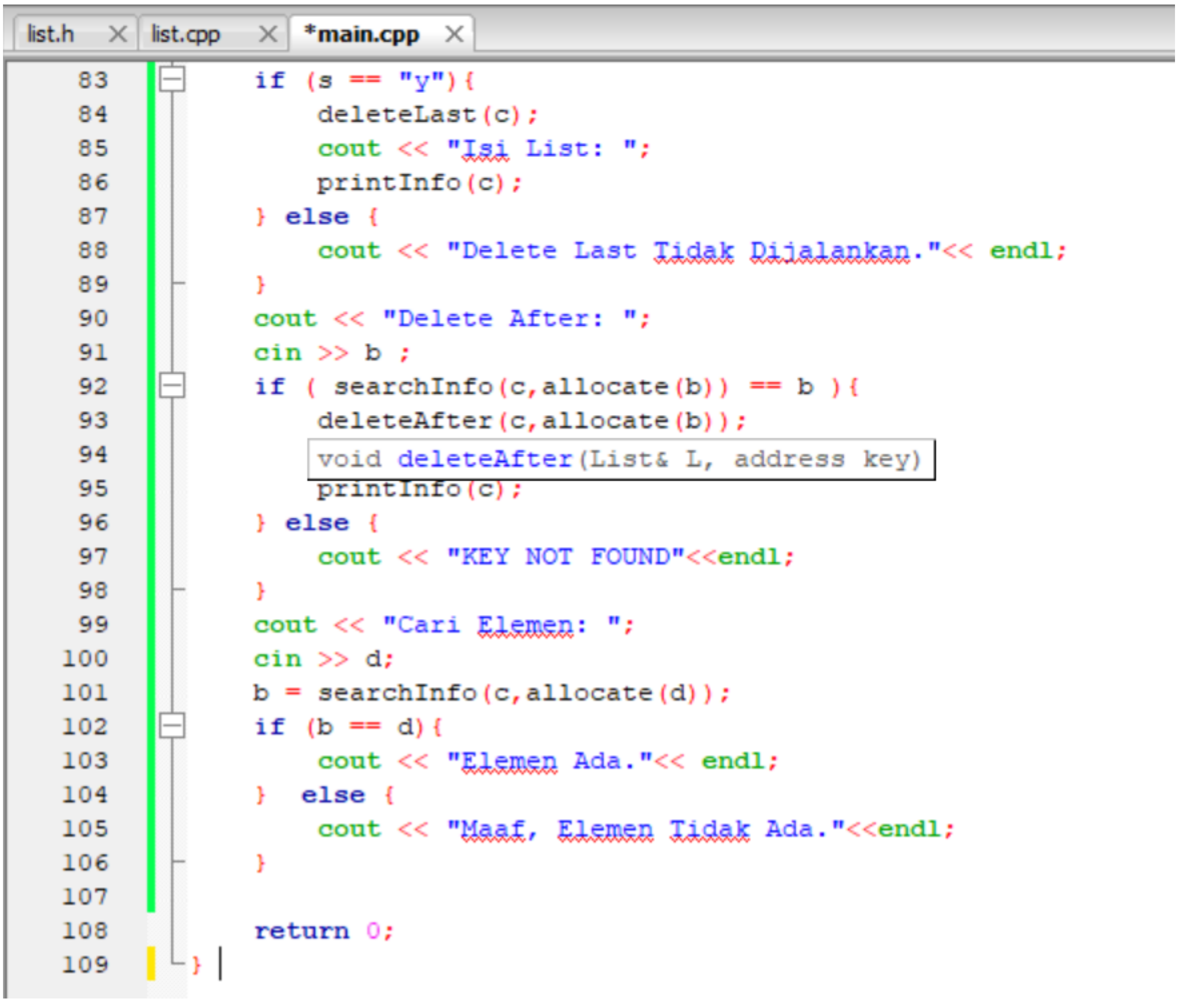
```

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

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

    cout << "Masukkan angka ketiga: ";
    cin >> x;

    p = allocate(x);
    ins address allocate(infotype x)
    cout << "Isi List: ";
    printInfo(L);

    cout << endl;
    deleteFirst(L, p);
    cout << "Isi List (setelah dilakukan deleteFirst): ";
    printInfo(L);
    cout << endl;

    cout << "========================================" << endl;
    cout << "           SESI HAVE FUN" << endl;
    cout << "========================================" << endl;

    string s;
```



```

    cout << "=======================================" << endl;
    cout << "           SESI HAVE FUN" << endl;
    cout << "=======================================" << endl;

    string s;
    infotype b, d, e;
    List c;
    address a;
    createList(c);
    int i = 1;
    cout << "Masukkan NIM perdigit"<< endl;
    while (i <= 10){
        cout << "Digit "<< i << ": ";
        cin >> b;
        insertLast(c,allocate(b));
        i++;
    }

    cout << "Isi List: ";
    printInfo(c);

    cout << endl;
    cout << "Masukan Elemen: ";
    cin  >> b;
    cout << "Dimasukkan Setelah: ";
    cin  >> d;
    if ( searchInfo(c,allocate(d)) == d ){
```



```

    cout << endl;
    cout << "Masukan Elemen: ";
    cin  >> b;
    cout << "Dimasukkan Setelah: ";
    cin  >> d;
    if ( searchInfo(c,allocate(d)) == d ){
        insertAfter(c,allocate(d),allocate(b));
        cout << "Isi List: ";
        printInfo(c);
    } else {
        cout << "KEY NOT FOUND"<<endl;
    }
    cout << "Delete Last List? (y/n): ";
    cin >> s;
    if (s == "y"){
        deleteLast(c);

        printInfo(c);
    } else {
        cout << "Delete Last Tidak Dijalankan."<< endl;
    }
    cout << "Delete After: ";
    cin >> b ;
    if ( searchInfo(c,allocate(b)) == b ){
        deleteAfter(c,allocate(b));
        cout << "Isi List: ";
        printInfo(c);
```



```cpp
    if (s == "y"){
        deleteLast(c);
        cout << "Isi List: ";
        printInfo(c);
    } else {
        cout << "Delete Last Tidak Dijalankan."<< endl;
    }
    cout << "Delete After: ";
    cin >> b ;
    if ( searchInfo(c,allocate(b)) == b ){
        deleteAfter(c,allocate(b));

        printInfo(c);
    } else {
        cout << "KEY NOT FOUND"<<endl;
    }
    cout << "Cari Elemen: ";
    cin >> d;
    b = searchInfo(c,allocate(d));
    if (b == d){
        cout << "Elemen Ada."<< endl;
    }  else {
        cout << "Maaf, Elemen Tidak Ada."<<endl;
    }

    return 0;
}
```