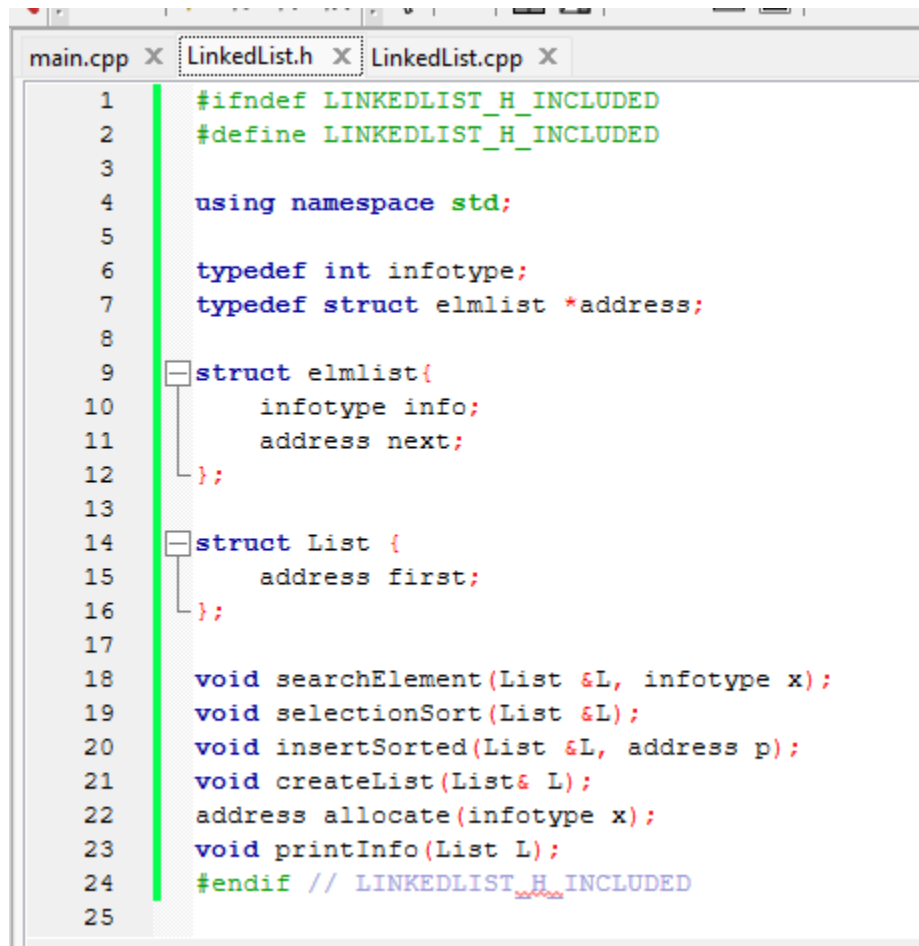


AHMAD RAFIANSYAH

103012400153

IF-48-11

NO1



The image shows a screenshot of a C++ code editor with three tabs: main.cpp, LinkedList.h (selected), and LinkedList.cpp. The code in LinkedList.h is as follows:

```
1  #ifndef LINKEDLIST_H_INCLUDED
2  #define LINKEDLIST_H_INCLUDED
3
4  using namespace std;
5
6  typedef int infotype;
7  typedef struct elmlist *address;
8
9  struct elmlist{
10     infotype info;
11     address next;
12 };
13
14 struct List {
15     address first;
16 };
17
18 void searchElement(List &L, infotype x);
19 void selectionSort(List &L);
20 void insertSorted(List &L, address p);
21 void createList(List& L);
22 address allocate(infotype x);
23 void printInfo(List L);
24 #endif // LINKEDLIST_H_INCLUDED
25
```

mp X LinkedList.h X LinkedList.cpp X

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

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

address allocate(infotype x) {
    address p = new elmList;
    p->info = x;
    p->next = nullptr;
    return p;
}

void searchElement(List &L, infotype x){
    address current;
    int position;

    current = L.first;
    position = 0;

    while (current != nullptr && current->info != x){
        position++;
        current = current->next;
    }

    if (current != nullptr){
        cout << "Elemen " << x << " ditemukan pada alamat: " << current << endl;
        cout << "Posisi: " << position << endl;
    } else {
        cout << "Elemen tidak ada didalam List" << endl;
    }
}
```

```

void selectionSort(List &L){
    address p, min, temp;
    infotype x;

    p = L.first;

    while (p != nullptr){
        min = p;
        temp = p->next;

        while (temp != nullptr){
            if (temp->info < min->info){
                min = temp;
            }
            temp = temp->next;
        }

        x = p->info;
        p->info = min->info;
        min->info = x;

        p = p->next;
    }
}

```

```

void insertSorted(List &L, address p){
    address q, prev;
    bool found;

    q = L.first;
    prev = nullptr;
    found = false;

    while (q != nullptr && found == false){
        if (q->info < p->info){
            prev = q;
            q = q->next;
        } else {
            found = true;
        }
    }

    p->next = q;

    if (prev == nullptr){
        p->next = L.first;
        L.first = p;
    } else {
        prev->next = p;
    }
}

```

```

void printInfo(List L){
    address p = L.first;

    while (p){
        cout << p->info << " ";
        p = p->next;
    }
    cout << endl;
}

```

main.cpp X

```

#include <iostream>
#include "LinkedList.h"

int main()
{
    List L;
    createList(L);
    address p1, p2, p3, p4, temp;

    p1 = allocate(39);
    p2 = allocate(13);
    p3 = allocate(87);
    p4 = allocate(24);

    L.first = p1;
    p1->next = p2;
    p2->next = p3;
    p3->next = p4;

    cout << "Isi list saat ini: ";
    temp = L.first;
    printInfo(L);
}

```

*main.cpp X

```
cout << endl;
cout << "Selection Sort" << endl;
selectionSort(L);

cout << "List sudah diurutkan: ";
temp = L.first;
printInfo(L);

cout << endl;
cout << "List setelah penyisipan: ";
temp = L.first;
printInfo(L);

cout << endl;
cout << "Mencari elemen 39" << endl;
searchElement(L, 39);

cout << endl;
cout << "Mencari elemen 88" << endl;
searchElement(L, 88);
return 0;
```

```
/** *
C:\Users\USER\Documents\C- X + v
nt
ts Sym
space
P5
Source
... Lin
... ma
... Heade

Isi list saat ini: 39 13 87 24

Selection Sort
List sudah diurutkan: 13 24 39 87

List setelah penyisipan: 13 24 39 87

Mencari elemen 39
Elemen 39 ditemukan pada alamat: 0x2587a5d19f0
Posisi: 2

Mencari elemen 88
Elemen tidak ada didalam List

Process returned 0 (0x0)   execution time : 0.068 s
Press any key to continue.
|
```