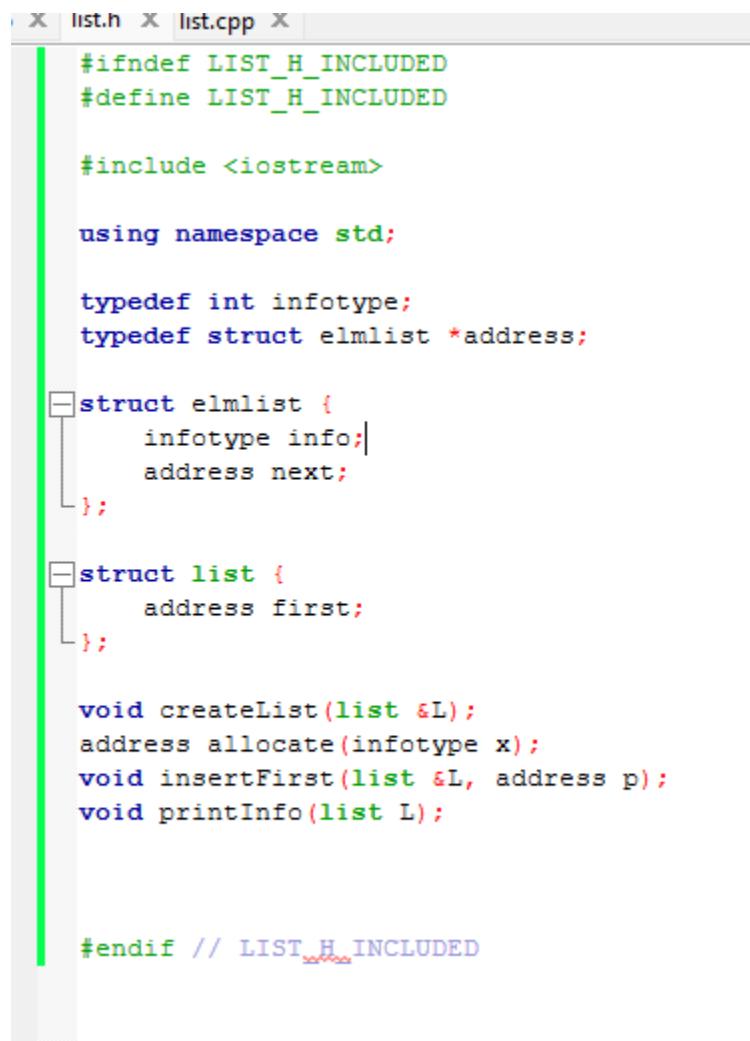


AHMAD RAFIANSYAH

103012400153

IF-48-11

List.h



The image shows a screenshot of a code editor with two tabs: "list.h" and "list.cpp". The "list.h" tab is active, displaying the following C++ code:

```
#ifndef LIST_H_INCLUDED
#define LIST_H_INCLUDED

#include <iostream>

using namespace std;

typedef int infotype;
typedef struct elmlist *address;

struct elmlist {
    infotype info;
    address next;
};

struct list {
    address first;
};

void createList(list &L);
address allocate(infotype x);
void insertFirst(list &L, address p);
void printInfo(list L);

#endif // LIST_H_INCLUDED
```

List.cpp

The screenshot shows a code editor window with two tabs at the top: "list.h" and "list.cpp". The "list.cpp" tab is active, indicating the current file being viewed. The code itself is a C++ program for managing a linked list. It includes headers for *iostream* and *list.h*, uses the *std* namespace, and defines four functions: *createList*, *allocate*, *insertFirst*, and *printInfo*. The *createList* function initializes the list's first pointer to *nullptr*. The *allocate* function creates a new list node (elmlist) and returns its address. The *insertFirst* function adds a new node to the beginning of the list by updating the first pointer and the next pointer of the new node. The *printInfo* function iterates through the list, printing each node's information separated by commas and ending with a new line.

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

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 insertFirst(list &L, address p) {
    p->next = L.first;
    L.first = p;
}

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

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

Main.cpp

```
list.h ^ list.cpp ^ |
#include "list.h"
#include <iostream>
using namespace std;

int main() {
    list L;
    createList(L);

    int x;
    address p;

    // Input angka pertama
    cout << "Masukkan angka pertama: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);
    printInfo(L);

    cout << "Masukkan angka kedua: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);
    printInfo(L);

    cout << "Masukkan angka ketiga: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);
    printInfo(L);

    cout << endl;
    return 0;
}
```

```
[File] [Edit] C:\Users\USER\Documents\C- X + | v
↙ Masukkan angka pertama: 1
1,
Masukkan angka kedua: 5
5, 1,
Masukkan angka ketiga: 3
3, 5, 1,

Process returned 0 (0x0) execution time : 5.202 s
Press any key to continue.
|
```