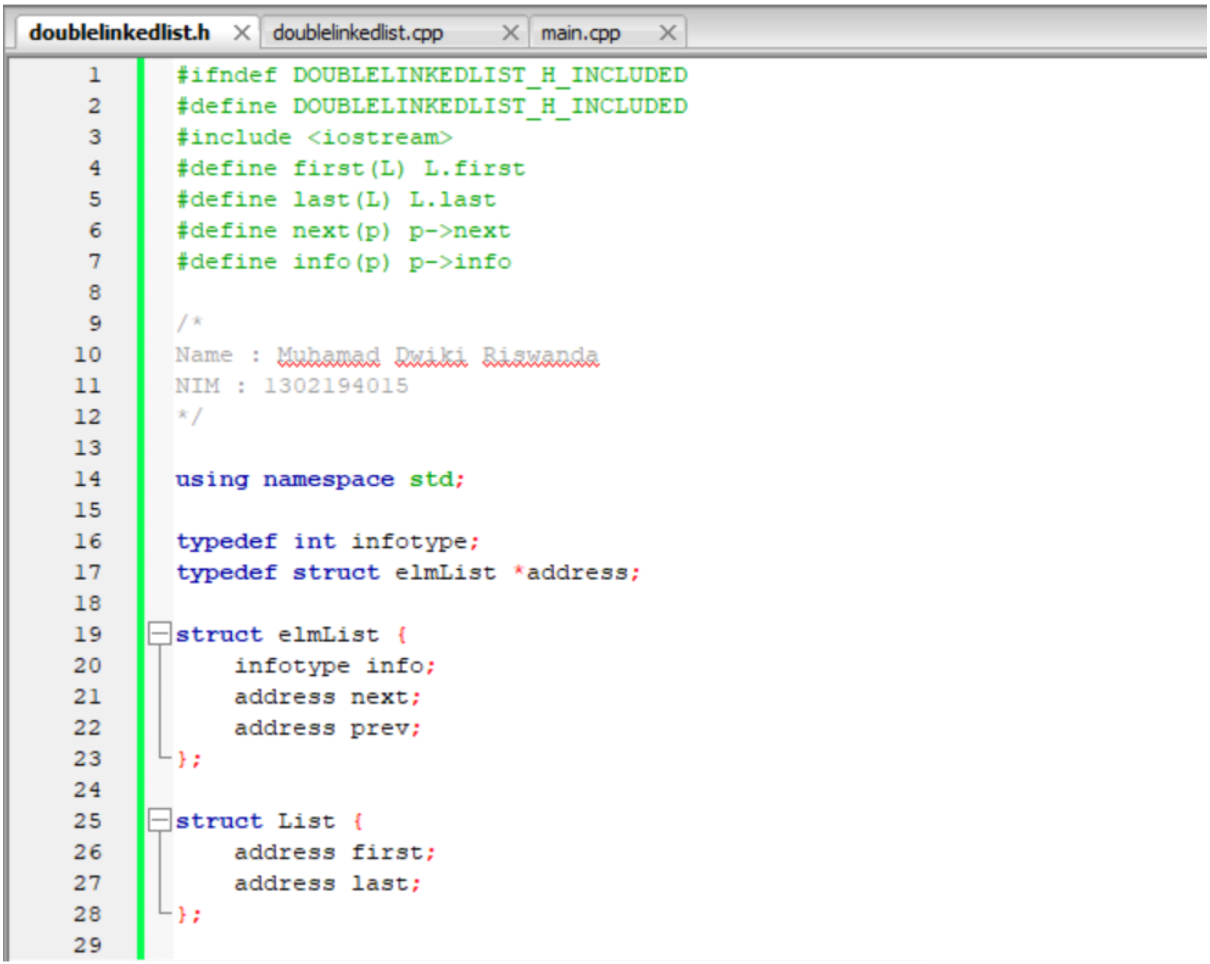
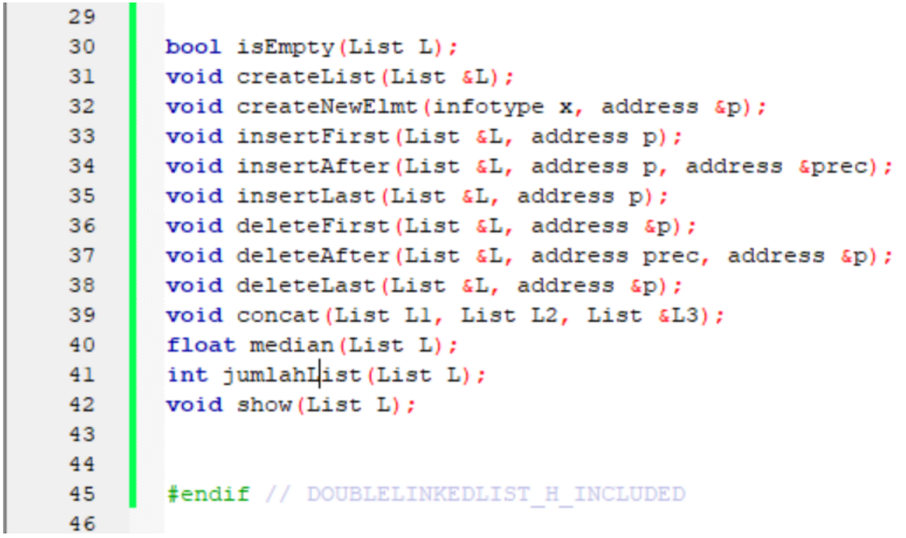
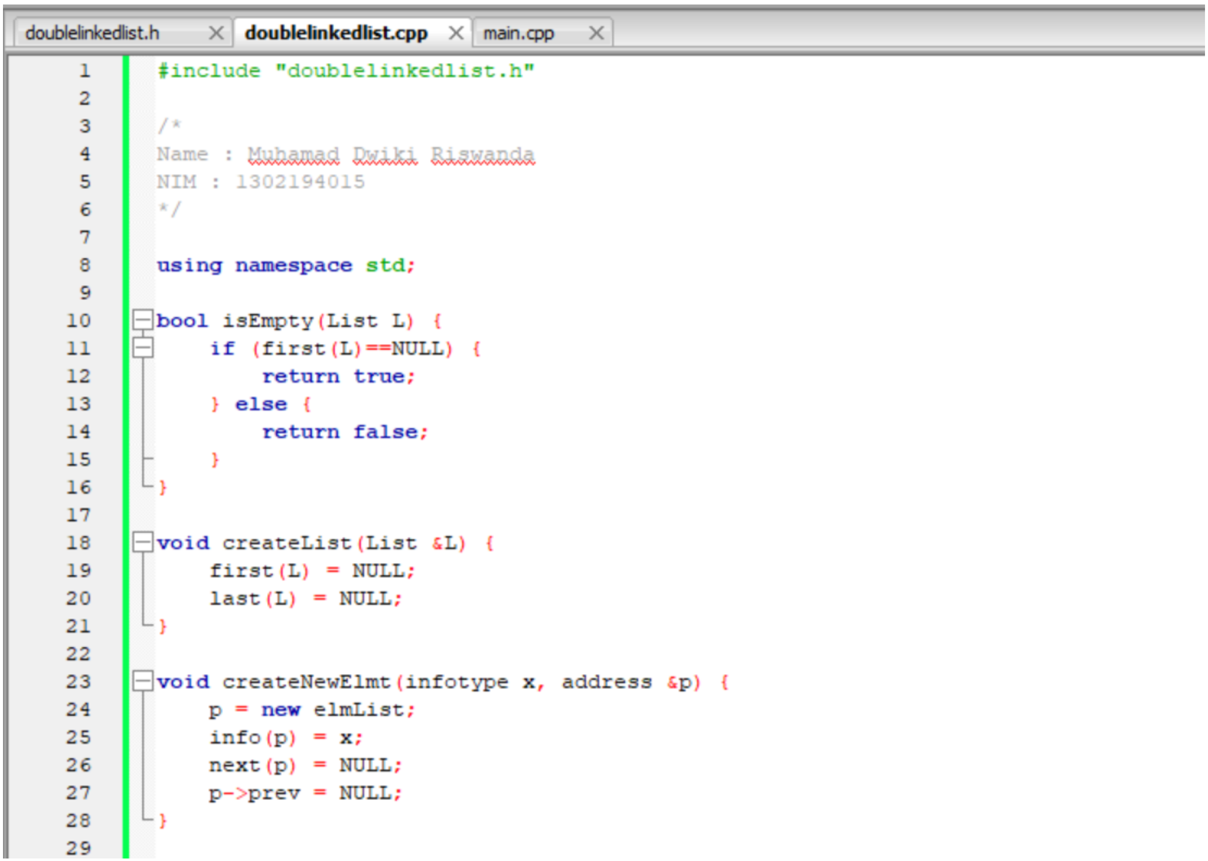


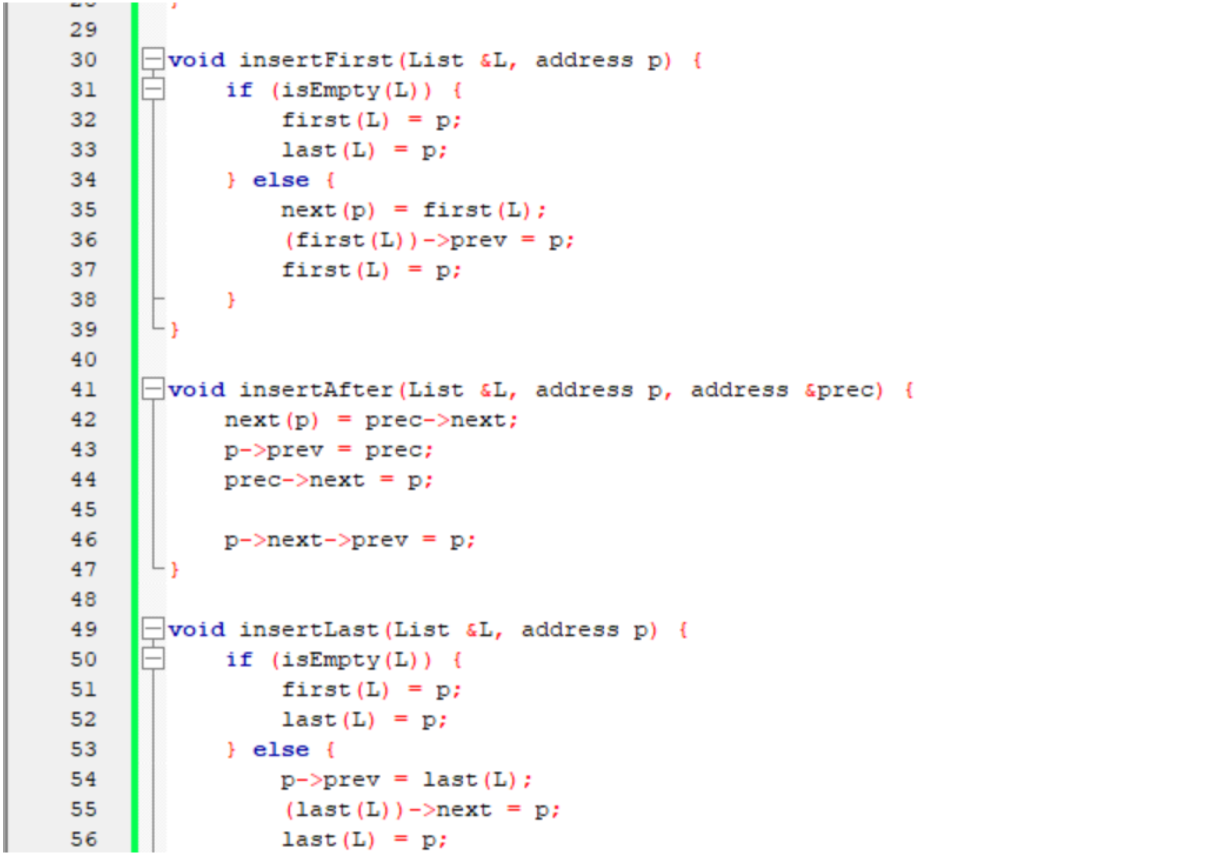
```
#ifndef DOUBLELINKEDLIST_H_INCLUDED
#define DOUBLELINKEDLIST_H_INCLUDED
#include <iostream>
#define first(L) L.first
#define last(L) L.last
#define next(p) p->next
#define info(p) p->info

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

using namespace std;

typedef int infotype;
typedef struct elmList *address;

struct elmList {
    infotype info;
    address next;
    address prev;
};

struct List {
    address first;
    address last;
};

```



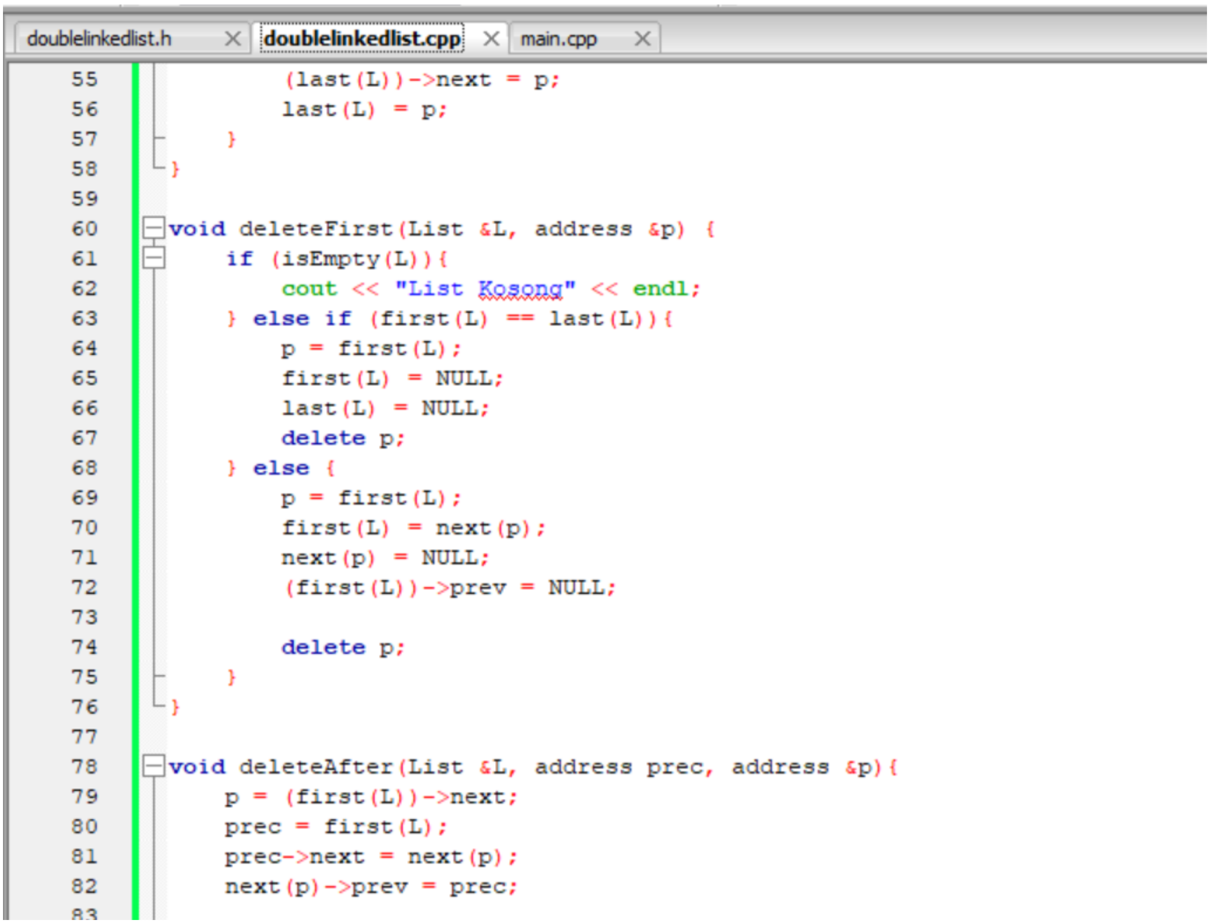
```

bool isEmpty(List L);
void createList(List &L);
void createNewElmt(infotype x, address &p);
void insertFirst(List &L, address p);
void insertAfter(List &L, address p, address &prec);
void insertLast(List &L, address p);
void deleteFirst(List &L, address &p);
void deleteAfter(List &L, address prec, address &p);
void deleteLast(List &L, address &p);
void concat(List L1, List L2, List &L3);
float median(List L);
int jumlahList(List L);
void show(List L);


#endif // DOUBLELINKEDLIST_H_INCLUDED

```



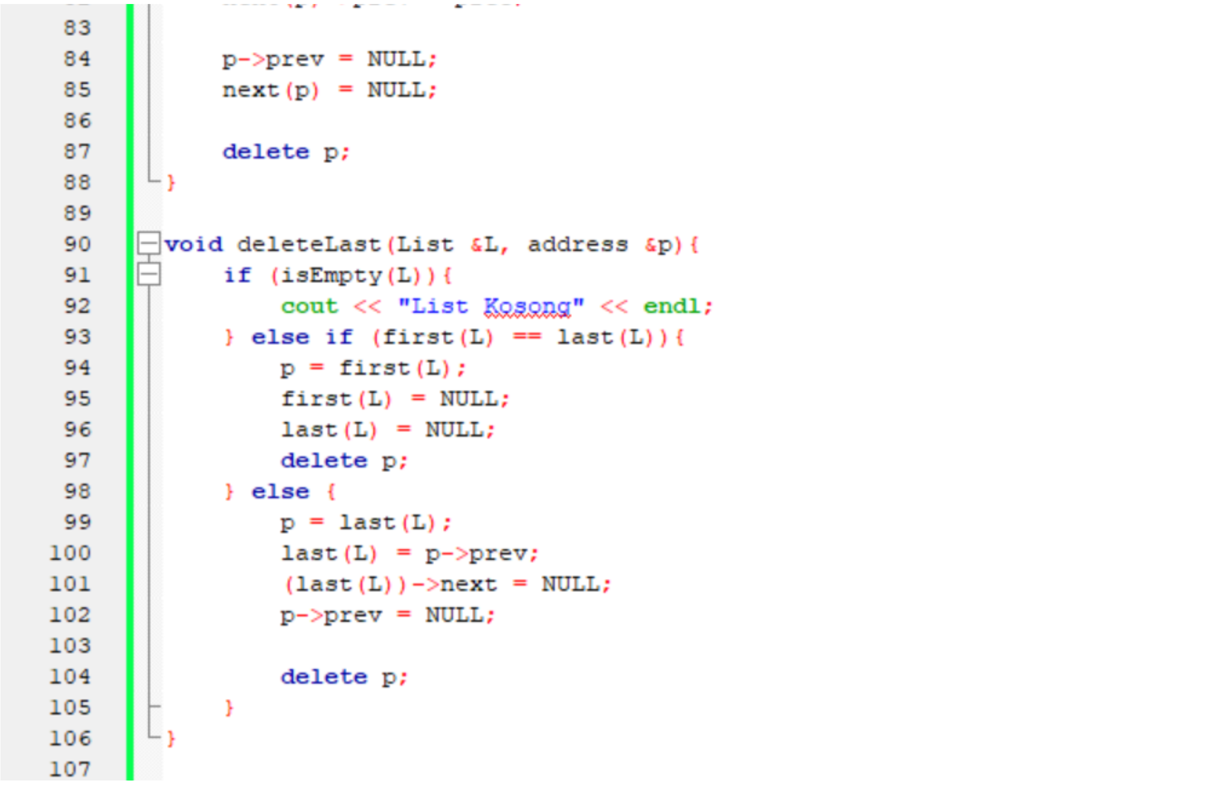
```cpp
#include "doublelinkedlist.h"

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

using namespace std;

bool isEmpty(List L) {
    if (first(L)==NULL) {
        return true;
    } else {
        return false;
    }
}

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

void createNewElmt(infotype x, address &p) {
    p = new elmList;
    info(p) = x;
    next(p) = NULL;
    p->prev = NULL;
}

```



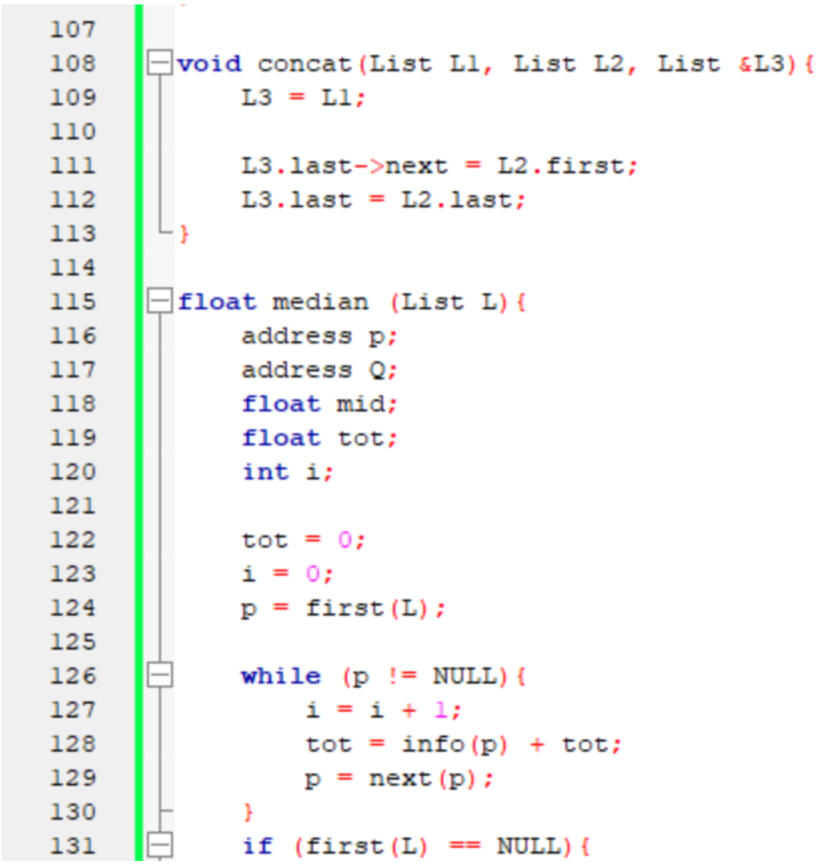
```cpp

void insertFirst(List &L, address p) {
    if (isEmpty(L)) {
        first(L) = p;
        last(L) = p;
    } else {
        next(p) = first(L);
        (first(L))->prev = p;
        first(L) = p;
    }
}

void insertAfter(List &L, address p, address &prec) {
    next(p) = prec->next;
    p->prev = prec;
    prec->next = p;

    p->next->prev = p;
}

void insertLast(List &L, address p) {
    if (isEmpty(L)) {
        first(L) = p;
        last(L) = p;
    } else {
        p->prev = last(L);
        (last(L))->next = p;
        last(L) = p;
```



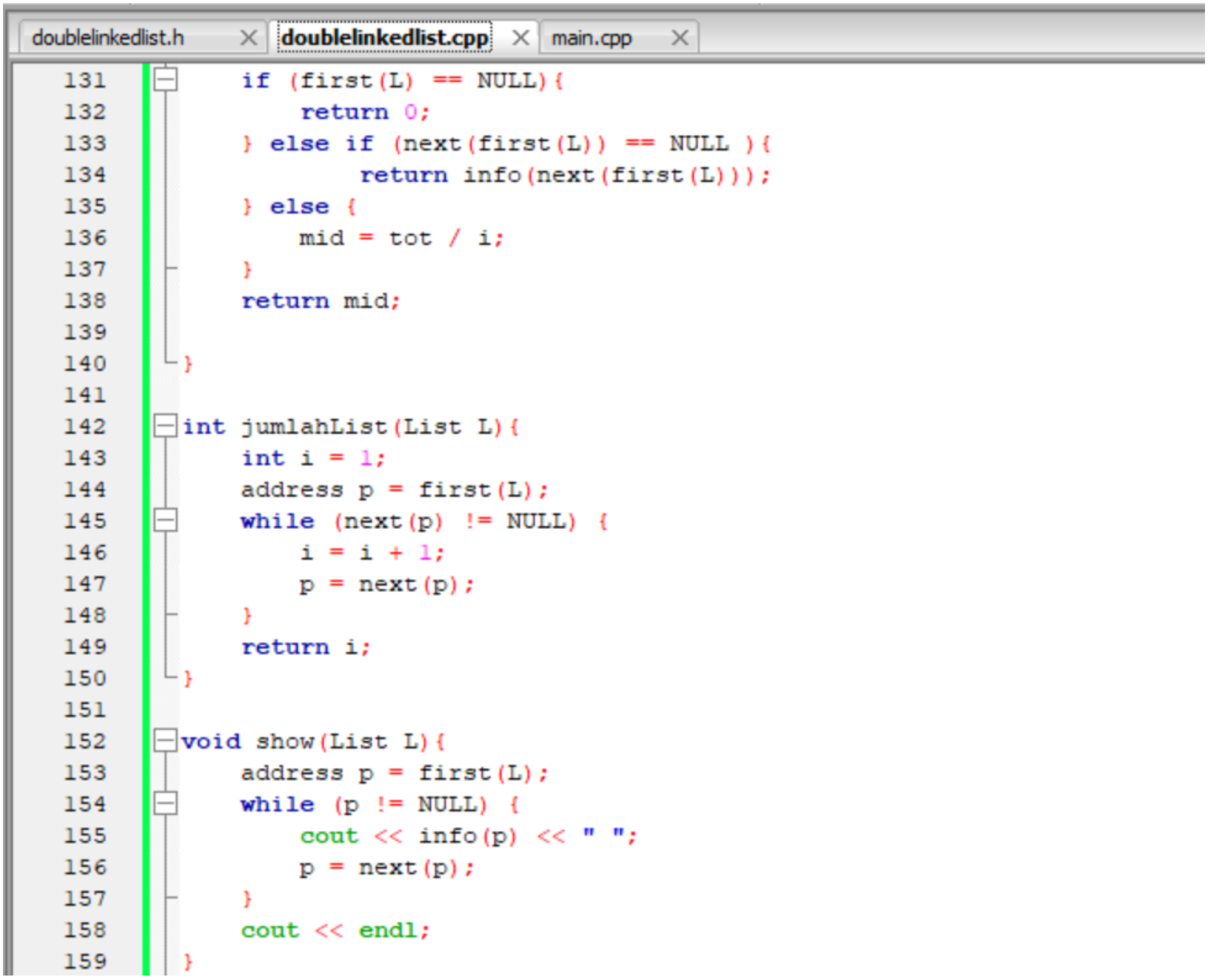
```
        (last(L))->next = p;
        last(L) = p;
    }
}

void deleteFirst(List &L, address &p) {
    if (isEmpty(L)){
        cout << "List Kosong" << endl;
    } else if (first(L) == last(L)){
        p = first(L);
        first(L) = NULL;
        last(L) = NULL;
        delete p;
    } else {
        p = first(L);
        first(L) = next(p);
        next(p) = NULL;
        (first(L))->prev = NULL;

        delete p;
    }
}

void deleteAfter(List &L, address prec, address &p){
    p = (first(L))->next;
    prec = first(L);
    prec->next = next(p);
    next(p)->prev = prec;
```



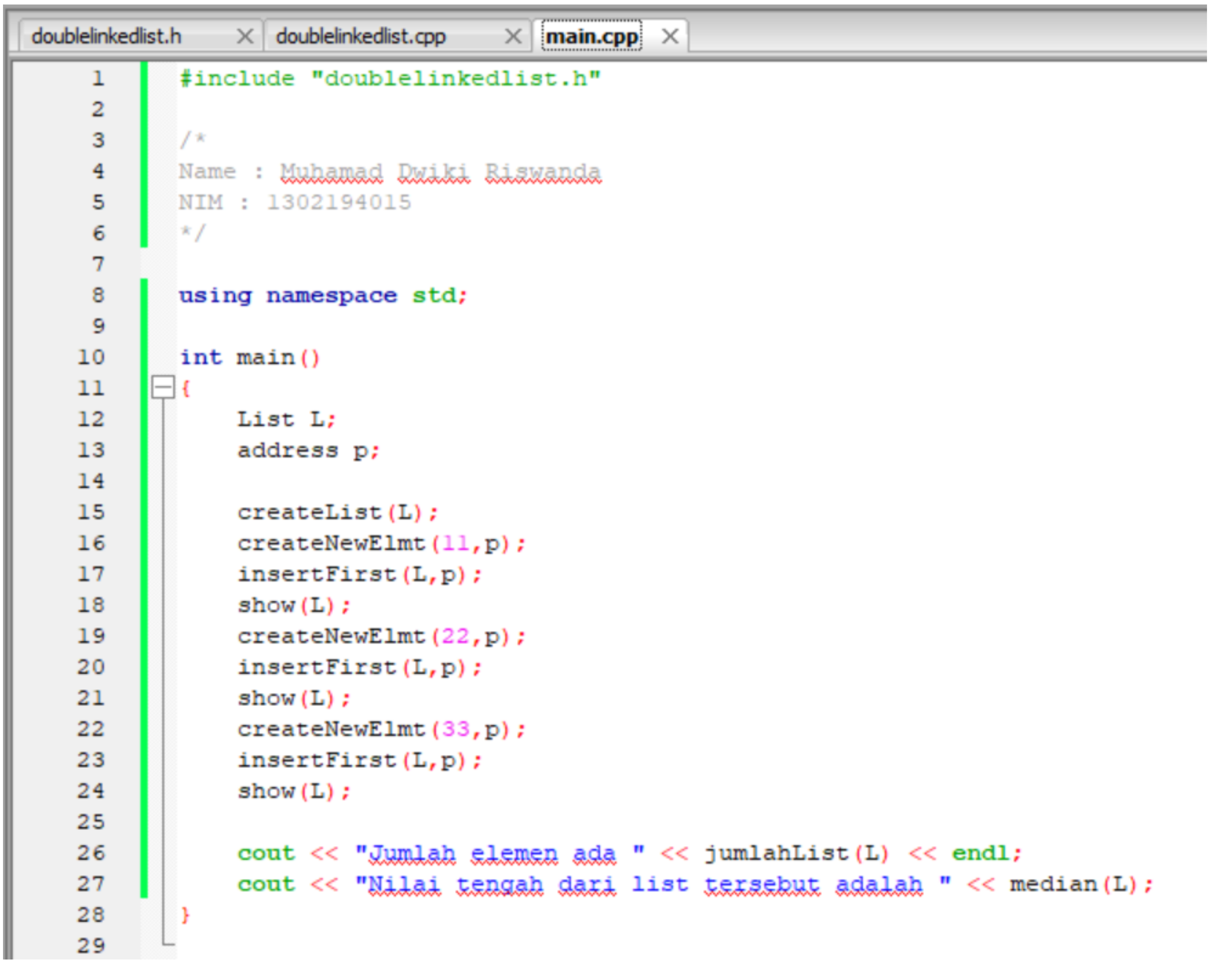
```

    p->prev = NULL;
    next(p) = NULL;

    delete p;
}

void deleteLast(List &L, address &p){
    if (isEmpty(L)){
        cout << "List Kosong" << endl;
    } else if (first(L) == last(L)){
        p = first(L);
        first(L) = NULL;
        last(L) = NULL;
        delete p;
    } else {
        p = last(L);
        last(L) = p->prev;
        (last(L))->next = NULL;
        p->prev = NULL;

        delete p;
    }
}

```



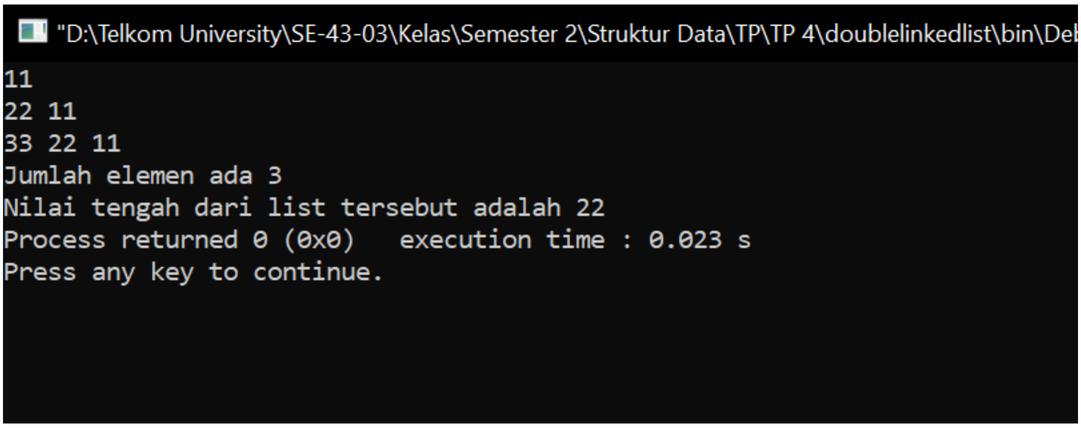
```
void concat(List L1, List L2, List &L3){
    L3 = L1;

    L3.last->next = L2.first;
    L3.last = L2.last;
}

float median (List L){
    address p;
    address Q;
    float mid;
    float tot;
    int i;

    tot = 0;
    i = 0;
    p = first(L);

    while (p != NULL){
        i = i + 1;
        tot = info(p) + tot;
        p = next(p);
    }
    if (first(L) == NULL){
```



```
    if (first(L) == NULL){
        return 0;
    } else if (next(first(L)) == NULL ){
            return info(next(first(L)));
    } else {
        mid = tot / i;
    }
    return mid;

}

int jumlahList(List L){
    int i = 1;
    address p = first(L);
    while (next(p) != NULL) {
        i = i + 1;
        p = next(p);
    }
    return i;
}

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



```cpp
#include "doublelinkedlist.h"

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

using namespace std;

int main()
{
    List L;
    address p;

    createList(L);
    createNewElmt(11,p);
    insertFirst(L,p);
    show(L);
    createNewElmt(22,p);
    insertFirst(L,p);
    show(L);
    createNewElmt(33,p);
    insertFirst(L,p);
    show(L);

    cout << "Jumlah elemen ada " << jumlahList(L) << endl;
    cout << "Nilai tengah dari list tersebut adalah " << median(L);
}

```


"D:\Telkom University\SE-43-03\Kelas\Semester 2\Struktur Data\TP\TP 4\doublelinkedlist\bin\De

```
11
22 11
33 22 11
Jumlah elemen ada 3
Nilai tengah dari list tersebut adalah 22
Process returned 0 (0x0)   execution time : 0.023 s
Press any key to continue.
```