doublelinkedlist.cpp | main.cpp | doublelinkedlist.h

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

/* NAMA: FAHMI RAZAN RAMDANI
   NIM: 1301194054 */
using namespace std;
typedef int infotype;
typedef struct elmlist *address;

struct elmlist
{
    infotype info;
    address next, prev;
};

struct List
{
    address first, 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 Prec, address P);
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);
double median(List L, infotype jumlah);
void showList(List L);
address cariElm(List L, address &P, infotype X);
#endif // DOUBLELINKEDLIST_H_INCLUDED
```

main.cpp [TP4] - Code::Blocks 17.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

doublelinkedlist.cpp | main.cpp | doublelinkedlist.h

```cpp
#include "doublelinkedlist.h"
/* NAMA: FAHMI RAZAN RAMDANI
   NIM: 13011|94054 */

int main()
{
    infotype X, Y, jumlah, jumlah2;
    float i;
    address P, Prec;
    bool cek;
    List L, L1, L2, L3;
    cek = false;
    X = 0;
    jumlah = 1;
    Y = 0;

    createList(L);
    createList(L1);
    createList(L2);
    createList(L3);
    cout<<"cek list, list kosong: ";
    cek = isEmpty(L);
    if(cek == true)
    {
        cout<<"ya\n";
    }else{
        cout<<"tidak\n";
    }

    cout<<"Insert first:\n";
    cin>>X;
    while(X != 0){
        createNewElmt(X, P);
        insertFirst(L, P);
        jumlah++;
        cin>>X;
    }
    showList(L);
    cout<<endl;

    cout<<"Insert after:\n";
    cout<<"Input: ";
    cin>>X;
    createNewElmt(X, P);
    cout<<"Setelah: ";
    cin>>Y;
    Prec = cariElm(L, Prec, Y);
    insertAfter(L, Prec, P);
    showList(L);
    cout<<endl;

    cout<<"Insert last:\n";
    cin>>X;
    while(X != 0){
        createNewElmt(X, P);
        insertLast(L, P);
        jumlah++;
        cin>>X;
    }
    showList(L);
    cout<<endl;

    cout<<"Delete first:\n";
    deleteFirst(L, P);
    showList(L);
    jumlah = jumlah - 1;
    cout<<endl;

    cout<<"Delete after:\n";
    cout<<"Hapus: ";
```

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 3, Col 14, Pos 74 | Insert | Read/W

21:08

main.cpp [TP4] - Code::Blocks 17.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

doublelinkedlist.cpp | main.cpp | doublelinkedlist.h

```
    cin>>X;
    while(X != 0){
        createNewElmt(X, P);
        insertLast(L, P);
        jumlah++;
        cin>>X;
    }
    showList(L);
    cout<<endl;

    cout<<"Delete first:\n";
    deleteFirst(L, P);
    showList(L);
    jumlah = jumlah - 1;
    cout<<endl;

    cout<<"Delete after:\n";
    cout<<"Hapus: ";
    Y = 0;
    cin>>X;
    P = cariElm(L, P, X);
    cout<<"Setelah: ";
    cin>>Y;
    Prec = cariElm(L, Prec, Y);
    deleteAfter(L, Prec, P);
    showList(L);
    jumlah = jumlah - 1;
    cout<<endl;

    cout<<"Delete last:\n";
    deleteLast(L, P);
    showList(L);
    jumlah = jumlah - 1;
    cout<<endl;

    cout<<"Jumlah elemen: "<<jumlah<<endl;

    i = median(L, jumlah);
    cout<<"Median: "<<i<<endl;

    cout<<"\nconcat:\n";
    jumlah = 0;
    cout<<"L1:\n";
    cin>>X;
    while(X != 0){
        createNewElmt(X, P);
        insertLast(L1, P);
        jumlah++;
        cin>>X;
    }
    showList(L1);
    cout<<"L2:\n";
    jumlah2 = 0;
    cin>>X;
    while(X != 0){
        createNewElmt(X, P);
        insertLast(L2, P);
        jumlah2++;
        cin>>X;
    }
    showList(L2);
    cout<<endl;
    concat(L1, L2, L3);
    cout<<"L3:\n";
    showList(L3);
    cout<<"\nNAMA: FAHMI RAZAN RAMDANI\n";
    cout<<"NIM: 1301194054\n";
    return 0;
}
```

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 3, Col 14, Pos 74 | Insert | Read/W

21:08

doublelinkedlist.cpp | main.cpp | doublelinkedlist.h

```cpp
#include "doublelinkedlist.h"
/* NAMA: FAHMI RAZAN RAMDANI
   NIM: 1301194054*/
using namespace std;

bool isEmpty(List L)
{
    bool found;
    found = false;
    if(first(L) == NULL){
        found = true;
    }
    return found;
}
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;
    prev(P) = NULL;
}
void insertFirst(List &L, address P)
{
    if(first(L) == NULL){
        first(L) = P;
        last(L) = P;
    }else{
        next(P) = first(L);
        prev(first(L)) = P;
        first(L) = P;
    }
}
void insertAfter(List &L, address Prec, address P)
{
    next(P) = next(Prec);
    next(Prec) = P;
    prev(P) = Prec;
    prev(next(P)) = P;
}
address cariElm(List L, address &P, infotype X)
{
    bool found;
    found = false;
    P = first(L);
    while(P != NULL && !found){
        if(info(P) == X){
            return P;
            found = true;
        }
        P = next(P);
    }
    if(info(P) == X){
        return P;
    }
}
void insertLast(List &L, address P)
{
    address Q;
    if(last(L) == NULL){
        first(L) = P;
        last(L) = P;
    }else{
        Q = last(L);
        next(Q) = P;
        prev(P) = Q;
        last(L) = P;
    }
}
void deleteFirst(List &L, address &P)
{
    if(first(L) == last(L)){
        P = first(L);
        first(L) = NULL;
        last(L) = NULL;
```


```
            last(L) = NULL;
        }else{
            P = first(L);
            first(L) = next(P);
            prev(first(L)) = NULL;
            next(P) = NULL;
        }
    }
void deleteAfter(List &L, address Prec, address &P)
{
    P = next(Prec);
    next(Prec) = next(P);
    prev(next(P)) = Prec;
    prev(P) = NULL;
    next(P) = NULL;
}
void deleteLast(List &L, address &P)
{
    address Q;
    if(first(L) == last(L)){
        P = first(L);
        first(L) = NULL;
        last(L) = NULL;
    }else{
        P = last(L);
        Q = prev(last(L));
        last(L) = Q;
        next(Q) = NULL;
        prev(P) = NULL;
    }
}
void concat(List L1, List L2, List &L3)
{
    L3 = L1;
    next(last(L3)) = first(L2);
    prev(first(L2)) = last(L3);
}
double median(List L, infotype jumlah)
{
    address q = first(L);
    address p;
    int a;
    float b;
    a = 0;
    b = 0.0;
    if (first(L)!=NULL) {
        int jml = jumlah;
        if (jml%2!=0) {
            for (int i =1; i<=(jml/2); i++) {
                q = next(q);
            }
            return info(q);
        } else{
            while(a < (jml/2)){
                q = next(q);
                a++;
            }
            p = prev(q);
            b = (info(p)+info(q));
            return b/2;
        }
    } else {
        return 0;
    }
}
void showList(List L)
{
    address P;
    P = first(L);
    cout<<"List: ";
    while(P != NULL)
    {
        cout<<info(P)<<" ";
        P = next(P);
    }
    cout<<endl;
}
```

"D:\Fahmi\File Kuliah\Tingkat satu\Struktur Data...

```
cek list, list kosong: ya
Insert first:
3 1 0
List: 1 3

Insert after:
Input: 2
Setelah: 1
List: 1 2 3

Insert last:
4 5 6 7 0
List: 1 2 3 4 5 6 7

Delete first:
List: 2 3 4 5 6 7

Delete after:
Hapus: 5
Setelah: 4
List: 2 3 4 6 7

Delete last:
List: 2 3 4 6

Jumlah elemen: 4
Median: 3.5

concat:
L1:
1 2 3 0
List: 1 2 3
L2:
4 5 6 0
List: 4 5 6

L3:
List: 1 2 3 4 5 6

NAMA: FAHMI RAZAN RAMDANI
NIM: 1301194054

Process returned 0 (0x0)   execution time : 28.933 s
```