## doublelinkedlist.h [tp04] - Code::Blocks 17.12

**doublelinkedlist.h** | doublelinkedlist.cpp | main.cpp

```cpp
1    /*  NAMA ; MIKOGIZKA SATRIA KARTIKA
2        NIM : 1301194086 */
3    #ifndef DOUBLELINKEDLIST_H_INCLUDED
4    #define DOUBLELINKEDLIST_H_INCLUDED
5    #include <iostream>
6    #define first(L) L.first
7    #define last(L) L.last
8    #define info(P) P->info
9    #define next(P) P->next
10   #define prev(P) P->prev
11   using namespace std;
12   typedef int infotype;
13   typedef struct elmlist *address;
14
15   struct elmlist
16   {
17       infotype info;
18       address next, prev;
19   };
20
21   struct List
22   {
23       address first, last;
24   };
25
26   bool isEmpty(List L);
27   void createList(List &L);
28   void createNewElmt(infotype X, address &P);
29   void insertFirst(List &L, address P);
30   void insertAfter(List &L, address Prec, address P);
31   void insertLast(List &L, address P);
32   void deleteFirst(List &L, address &P);
33   void deleteAfter(List &L, address Prec, address &P);
34   void deleteLast(List &L, address &P);
35   void concat(List L1, List L2, List &L3);
36   double median(List L, infotype jumlah);
37   void showList(List L);
38   address cariElm(List L, address &P, infotype X);
39   #endif // DOUBLELINKEDLIST_H_INCLUDED
40
```

C/C++  |  Windows (CR+LF)  |  WINDOWS-1252  |  Line 11, Col 1, Pos 282

## doublelinkedlist.cpp [tp04] - Code::Blocks 17.12

doublelinkedlist.h | **doublelinkedlist.cpp** | main.cpp

```cpp
1    /*  NAMA ; MIKOGIZKA SATRIA KARTIKA
2        NIM : 1301194086 */
3    #include "doublelinkedlist.h"
4    using namespace std;
5
6    bool isEmpty(List L)
7    {
8        bool found;
9        found = false;
10       if(first(L) == NULL){
11           found = true;
12       }
13       return found;
14   }
15   void createList(List &L)
16   {
17       first(L) = NULL;
18       last(L) = NULL;
19   }
20   void createNewElmt(infotype X, address &P)
21   {
22       P = new elmlist;
23       info(P) = X;
24       next(P) = NULL;
25       prev(P) = NULL;
26   }
27   void insertFirst(List &L, address P)
28   {
29       if(first(L) == NULL){
30           first(L) = P;
31           last(L) = P;
32       }else{
33           next(P) = first(L);
34           prev(first(L)) = P;
35           first(L) = P;
36       }
37   }
38   void insertAfter(List &L, address Prec, address P)
39   {
40       next(P) = next(Prec);
```

C/C++  |  Windows (CR+LF)  |  WINDOWS-1252  |  Line 3, Col 30, Pos 91

```cpp
        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;
        }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);
```

**doublelinkedlist.cpp [tp04] - Code::Blocks 17.12**

doublelinkedlist.h | **doublelinkedlist.cpp** | main.cpp

```cpp
118         address q = first(L);
119         address p;
120         int a;
121         float b;
122         a = 0;
123         b = 0.0;
124         if (first(L)!=NULL) {
125             int jml = jumlah;
126             if (jml%2!=0) {
127                 for (int i =1; i<=(jml/2); i++) {
128                     q = next(q);
129                 }
130                 return info(q);
131             } else{
132                 while(a < (jml/2)){
133                     q = next(q);
134                     a++;
135                 }
136                 p = prev(q);
137                 b = (info(p)+info(q));
138                 return b/2;
139             }
140         } else {
141             return 0;
142         }
143     }
144     void showList(List L)
145     {
146         address P;
147         P = first(L);
148         cout<<"List: ";
149         while(P != NULL)
150         {
151             cout<<info(P)<<" ";
152             P = next(P);
153         }
154         cout<<endl;
155     }
156
```

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 3, Col 30, Pos 91

---

**main.cpp [tp04] - Code::Blocks 17.12**

doublelinkedlist.h | doublelinkedlist.cpp | **main.cpp**

```cpp
1   /*  NAMA ; MIKOGIZKA SATRIA KARTIKA
2       NIM : 1301194086 */
3   #include "doublelinkedlist.h"
4   int main()
5   {
6       infotype X, Y, jumlah, jumlah2;
7       float i;
8       address P, Prec;
9       bool cek;
10      List L, L1, L2, L3;
11      cek = false;
12      X = 0;
13      jumlah = 1;
14      Y = 0;
15
16      createList(L);
17      createList(L1);
18      createList(L2);
19      createList(L3);
20      cout<<"cek list, apakah list kosong?: ";
21      cek = isEmpty(L);
22      if(cek == true)
23      {
24          cout<<"ya\n";
25      }else{
26          cout<<"tidak\n";
27      }
28
29      cout<<"Insert first:\n";
30      cin>>X;
31      while(X != 0){
32          createNewElmt(X, P);
33          insertFirst(L, P);
34          jumlah++;
35          cin>>X;
36      }
37      showList(L);
38      cout<<endl;
39
40      cout<<"Insert after:\n";
41      cout<<"Input: ";
42      cin>>X;
```

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 20, Col 41, Pos 405

```
main.cpp [tp04] - Code::Blocks 17.12

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

doublelinkedlist.h  ×  doublelinkedlist.cpp  ×  main.cpp  ×

42          cin>>X;
43          createNewElmt(X, P);
44          cout<<"Setelah: ";
45          cin>>Y;
46          Prec = cariElm(L, Prec, Y);
47          insertAfter(L, Prec, P);
48          showList(L);
49          cout<<endl;
50
51          cout<<"Insert last:\n";
52          cin>>X;
53          while(X != 0){
54              createNewElmt(X, P);
55              insertLast(L, P);
56              jumlah++;
57              cin>>X;
58          }
59          showList(L);
60          cout<<endl;
61
62          cout<<"Delete first:\n";
63          deleteFirst(L, P);
64          showList(L);
65          jumlah = jumlah - 1;
66          cout<<endl;
67
68          cout<<"Delete after:\n";
69          cout<<"Hapus: ";
70          Y = 0;
71          cin>>X;
72          P = cariElm(L, P, X);
73          cout<<"Setelah: ";
74          cin>>Y;
75          Prec = cariElm(L, Prec, Y);
76          deleteAfter(L, Prec, P);
77          showList(L);
78          jumlah = jumlah - 1;
79          cout<<endl;
80
81          cout<<"Delete last:\n";
82          deleteLast(L, P);
83          showList(L);
```

```
83          showList(L);
84          jumlah = jumlah - 1;
85          cout<<endl;
86
87          cout<<"Jumlah elemen: "<<jumlah<<endl;
88
89          i = median(L, jumlah);
90          cout<<"Median: "<<i<<endl;
91
92          cout<<"\nconcat:\n";
93          jumlah = 0;
94          cout<<"L1:\n";
95          cin>>X;
96          while(X != 0){
97              createNewElmt(X, P);
98              insertLast(L1, P);
99              jumlah++;
100             cin>>X;
101         }
102         showList(L1);
103         cout<<"L2:\n";
104         jumlah2 = 0;
105         cin>>X;
106         while(X != 0){
107             createNewElmt(X, P);
108             insertLast(L2, P);
109             jumlah2++;
110             cin>>X;
111         }
112         showList(L2);
113         cout<<endl;
114         concat(L1, L2, L3);
115         cout<<"L3:\n";
116         showList(L3);
117         return 0;
118     }
119
```

C/C++        Windows (CR+LF)    WINDOWS-1252    Line 20, Col 41, Pos 405

C/C++        Windows (CR+LF)    WINDOWS-1252    Line 20, Col 41, Pos 405

```
cek list, apakah list kosong?: ya
Insert first:
4 2 0
List: 2 4

Insert after:
Input: 3
Setelah: 2
List: 2 3 4

Insert last:
5 6 7 8 9 0
List: 2 3 4 5 6 7 8 9

Delete first:
List: 3 4 5 6 7 8 9

Delete after:
Hapus: 6
Setelah: 5
List: 3 4 5 7 8 9

Delete last:
List: 3 4 5 7 8

Jumlah elemen: 5
Median: 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


Process returned 0 (0x0)   execution time : 113.679 s
Press any key to continue.
```