NAMA: Kadek Rizky Fransisca Putra / 1301194035

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

```cpp
#include <iostream>
#include "doublelinkedlist.h"
using namespace std;
/*Nama: Kadek Rizky Fransisca Putra
  Nim : 1301194035
*/
int main()
{
    float inimedian;
    List L1, L2,L3;
    address P , Prec;
    createList(L1);
    createNewElmt(2, P);
    insertFirst(L1, P);


    createNewElmt(3, P);
    insertLast(L1, P);


    createNewElmt(1, P);
    insertFirst(L1, P);


    createNewElmt(5, P);
    insertLast(L1, P);

    cout<<"Ini list ke-1: ";lihatlist(L1);

    deleteLast(L1, P);
    cout<<"Ini list setelah delete last: ";lihatlist(L1);
    deleteFirst(L1, P);
    cout<<"Ini list setelah delete first: ";lihatlist(L1);
    cout<<endl;

    createList(L2);
    createNewElmt(7, P);
```

```
38      insertFirst(L2, P);
39
40      Prec = P;
41      createNewElmt(8, P);
42      insertAfter(L2,Prec,P);
43
44      createNewElmt(9, P);
45      insertLast(L2, P);
46
47      Prec = P;
48      createNewElmt(10, P);
49      insertAfter(L2, Prec,P);
50
51      createNewElmt(11, P);
52      insertLast(L2, P);
53
54      cout<<"Ini list ke-2: ";lihatlist(L2);
55
56      cout<<endl;
57
58      createList(L3);
59      concat(L1,L2,L3);
60      cout<<"Ini List concat: ";
61      lihatlist(L3);
62
63      inimedian = median(L3);
64      cout<<"Ini median dari list concat: "<<inimedian<<endl;
65
66
67  }
```

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

```
1  #ifndef DOUBLELINKEDLIST_H_INCLUDED
2  #define DOUBLELINKEDLIST_H_INCLUDED
3  /*Nama: Kadek Rizky Fransisca Putra
4    Nim : 1301194035
5  */
6  using namespace std;
7  typedef int infotype;
8  typedef struct elmtList *address;
9  struct elmtList{
10     infotype info;
11     address prev;
12     address next;
13 };
14 struct List{
15     address first;
16     address last;
17 };
18 bool isEmpty(List L);
19 void createList(List &L);
20 void createNewElmt(infotype X, address &P);
21 void insertFirst(List &L, address P);
22 void insertAfter(List &L, address &Prec, address &P);
23 void insertLast(List &L, address P);
24 void deleteFirst(List &L, address &P);
25 void deleteAfter(List &L, address &Prec, address &P);
26 void deleteLast(List &L, address &P);
27 void concat(List L1, List L2, List &L3);
28 void lihatlist(List L);
29 float median(List L);
30
31 #endif // DOUBLELINKEDLIST_H_INCLUDED
32
```

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

```cpp
#include <iostream>
#include "doublelinkedlist.h"
using namespace std;
/*Nama: Kadek Rizky Fransisca Putra
  Nim : 1301194035
*/
bool isEmpty(List L){
    bool cek;
    if(L.first == NULL){
        cek = true;
    }else{
        cek = false;
    }
    return cek;
}
void createList(List &L){
    L.first = NULL;
    L.last = NULL;
}
void createNewElmt(infotype X, address &P){
    P = new elmtList;
    P->info = X;
    P->next = NULL;
    P->prev = NULL;
}
void insertFirst(List &L, address P){
    if(isEmpty(L)==true){
        L.first = P;
        L.last = P;
    }
    else{
        P->next = L.first;
        L.first->prev = P;
        L.first = P;
    }
}
void insertAfter(List &L, address &Prec, address &P){
```

```
37  void insertAfter(List &L, address &Prec, address &P){
38      if(isEmpty(L)==true){
39          cout<<"List masih kosong !";
40      }
41      if(Prec->next == NULL){
42          P->prev = Prec;
43          Prec->next = P;
44          L.last = P;
45      }
46      else{
47          P->next = Prec->next;
48          Prec->next = P;
49          P->next->prev = P;
50          P->prev = Prec;
51      }
52
53  }
54  void insertLast(List &L, address P){
55      if(L.first == NULL){
56          L.first = P;
57          L.last = P;
58      }
59      else{
60          L.last->next =P;
61          P->prev = L.last;
62          L.last = P;
63      }
64  }
65  void deleteFirst(List &L, address &P){
66      if(L.first == NULL){
67          L.first = NULL;
68          L.last = NULL;
69      }
70      else{
71          P = L.first;
72          L.first = P->next;
73          P->next = NULL;
```

```
74          L.first->prev = NULL;
75      }
76  }
77  void deleteAfter(List &L, address &Prec, address &P){
78      P = Prec->next;
79      Prec->next = P->next;
80      P->next->prev = Prec;
81      P->next = NULL;
82      P->prev = NULL;
83  }
84  void deleteLast(List &L, address &P){
85      P = L.last;
86      L.last = P->prev;
87      P->prev = NULL;
88      L.last->next = NULL;
89  }
90  void lihatlist(List L){
91      address bantuan = L.first;
92      while(bantuan != NULL){
93          cout<< bantuan->info << ", ";
94          bantuan = bantuan->next;
95      }
96      cout<<endl;
97  }
98  void concat(List L1, List L2, List &L3){
99      L3.first = L1.first;
100     L3.last = L2.last;
101     L2.first->prev = L1.last;
102     L1.last->next = L2.first;
103
104 }
105 float median(List L){
106     address bantuan , bantuan1;
107     int i, j,k;
108     float inimedian;
109     bantuan = L.first;
```

```
109         bantuan = L.first;
110         i = 1;
111         while(bantuan->next != NULL){
112             bantuan = bantuan->next;
113             i++;
114         }
115         k =  i /2;
116         bantuanl = L.first;
117         if(i%2==0){
118             j=1;
119             while(j < k){
120                 bantuanl = bantuanl->next;
121             j++;
122             }
123             inimedian = (double)(bantuanl->info + bantuanl->next->info)/2;
124         }else if(i%2!=0) {
125             j=1;
126             while(j < k){
127             bantuanl = bantuanl->next;
128             j++;
129             inimedian = (double)bantuanl->next->info;
130
131         }
132         }
133         return inimedian;
134
135
136 }
```

"D:\Kumpulan Tugas Rizky\struktur data\STRUKTUR DATA MOD 4\bin\Debug\STRUKTUR

```
Ini list ke-1: 1, 2, 3, 5,
Ini list setelah delete last: 1, 2, 3,
Ini list setelah delete first: 2, 3,

Ini list ke-2: 7, 8, 9, 10, 11,

Ini List concat: 2, 3, 7, 8, 9, 10, 11,
Ini median dari list concat: 8

Process returned 0 (0x0)   execution time : 0.217 s
Press any key to continue.
```