

**TP Modul 5 - Single Linked List (Bagian Kedua)**

tp5.h

```
main.cpp X tp5.h X tp5.cpp X
1      #include <iostream>
2
3      using namespace std;
4
5      typedef int infotype;
6      typedef struct elmList *address;
7
8      struct elmList {
9          infotype info;
10         address next;
11     };
12
13     struct List{
14         address first;
15     };
16
17     void createList_103032330095(List &L);
18
19     address createNewElm_103032330095(infotype data);
20
21     void insertLast_103032330095(List &L, address P);
22
23     address findMin_103032330095(List L);
24
25     void insertMiddle_103032330095(List &L, int value);
26
27     void showAllData_103032330095(List L);
28
29     void tambahData_103032330095(List &L, int data);
30
31     int selectMenu_103032330095();
```

## tp5.cpp

```

main.cpp x tp5.h x tp5.cpp x
1  #include "tp5.h"
2  void createList_103032330095(List &L){
3      L.first = NULL;
4  }
5  address createNewElm_103032330095(infotype data) {
6      address P = new elmList;
7      P->info = data;
8      P->next = NULL;
9      return P;
10 }
11
12 void insertLast_103032330095(List &L, address P){
13     address Q;
14     if (L.first == NULL){
15         L.first = P;
16     } else {
17         Q = L.first;
18         while (Q->next != NULL){
19             Q = Q->next;
20         }
21         Q->next = P;
22     }
23 }
24
25 address findMin_103032330095(List L){
26     address P, minData;
27     minData = L.first;
28     P = L.first->next;
29     while (P != NULL){
30         if (P->info < minData->info){
31             minData = P;
32         }
33         P = P->next;
34     }
35     return minData;
36 }

```

```

main.cpp x tp5.h x tp5.cpp x
37 void insertMiddle_103032330095(List &L, int value){
38     address P, prec;
39     int i, n;
40     P = createNewElm_103032330095(value);
41     prec = L.first;
42     n = 1;
43     while (prec->next != NULL) {
44         prec = prec->next;
45         n++;
46     }
47     if (n % 2 == 1){
48         n++;
49     }
50     n = n / 2;
51     prec = L.first;
52     for (i = 1; i < n; i++){
53         prec = prec->next;
54     }
55     P->next = prec->next;
56     prec->next = P;
57 }
58

```

```

main.cpp x tp5.h x tp5.cpp x
59 void showAllData_103032330095(List L){
60     address P;
61     if (L.first == NULL){
62         cout << "List Kosong." << endl;
63     } else {
64         P = L.first;
65         cout << "Data dalam list : ";
66         while (P != NULL){
67             cout << P->info;
68             if (P->next != NULL){
69                 cout << ", ";
70             }
71             P = P->next;
72         }
73         cout << endl;
74     }
75 }
76
77 void tambahData_103032330095(List &L, int data){
78     address P;
79     P = createNewElm_103032330095(data);
80     insertLast_103032330095(L, P);
81 }
82
83 int selectMenu_103032330095(){
84     int input = 0;
85     cout << "=====MENU===== << endl;
86     cout << "1. Menambah N data baru" << endl;
87     cout << "2. Menampilkan semua data" << endl;
88     cout << "3. Menampilkan nilai terkecil" << endl;
89     cout << "4. Menambah data ke tengah list" << endl;
90     cout << "0. Exit" << endl;
91     cin >> input;
92     return input;
93 }

```

## main.cpp

```

main.cpp x tp5.h x *tp5.cpp x
1 #include "tp5.h"
2
3 int main(){
4     List L;
5     address P, minData;
6     int pilihan = 0, i, n, data;
7     createList_103032330095(L);
8     pilihan = selectMenu_103032330095();
9     while (pilihan != 0){
10         switch(pilihan){
11             case 1:
12                 cout << "Jumlah data yang akan ditambahkan : ";
13                 cin >> n;
14                 for (i = 1; i <= n; i++){
15                     cout << "Masukkan data baru : ";
16                     cin >> data;
17                     tambahData_103032330095(L, data);
18                 }
19                 break;
20
21             case 2:
22                 showAllData_103032330095(L);
23                 break;
24
25             case 3:
26                 cout << "Data terkecil dalam list : ";
27                 cout << findMin_103032330095(L) ->info << endl;
28                 break;

```

```

29
30         case 4:
31             cout << "Masukkan data yang akan ditambahkan ke tengah list : ";
32             cin >> data;
33             insertMiddle_103032330095(L, data);
34             break;
35         }
36         pilihan = selectMenu_103032330095();
37     }
38     cout << "ANDA TELAH KELUAR DARI PROGRAM" << endl;
39     return 0;
40 }

```

output

```

D:\KuliahGena\SMT 3\Strukt  × + ▾
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit
1
Jumlah data yang akan ditambahkan : 5
Masukkan data baru : 50
Masukkan data baru : 10
Masukkan data baru : 20
Masukkan data baru : 15
Masukkan data baru : 40
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit
2
Data dalam list : 50, 10, 20, 15, 40
=====MENU=====

```

```

D:\KuliahGena\SMT 3\Strukt  × + ▾
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit
3
Data terkecil dalam list : 10
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit
4
Masukkan data yang akan ditambahkan ke tengah list : 30
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit
2
Data dalam list : 50, 10, 20, 30, 15, 40
=====MENU=====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nilai terkecil
4. Menambah data ke tengah list
0. Exit

```