

Main.cpp :

```
1  #include "SLL.h"
2
3  using namespace std;
4
5  int main()
6  {
7      List L;
8      address p,q;
9      CreateList_103032330054(L);
10     int NUM = 0, pilihan = 0, x;
11     infotype data;
12
13     pilihan = selectMenu_103032330054();
14     while (pilihan!=0){
15         switch(pilihan){
16             case 1:
17                 cout << "Jumlah data yang akan ditambahkan: ";
18                 cin >> x;
19                 for (int i = 0;i<x;i++){
20                     cout << endl << "Masukkan data baru: ";
21                     cin >> data;
22                     p = CreateNewElm_103032330054(data);
23                     InsertLast_103032330054(L,p);
24                 }
25                 break;
26
27             case 2:
28                 Show_103032330054(L);
29                 break;
30             case 3:
31                 p = findMin_103032330054(L);
32                 cout << endl << "Data dengan nilai minimal: "<< p->info << endl;
33                 break;
34             case 4:
35                 cout << "Masukkan data yang akan ditambahkan pada tengah list: ";
36                 cin >> data;
37                 insertMiddle_103032330054(L, data);
38                 break;
39         }
40         pilihan = selectMenu_103032330054();
41     }
42     cout << "BYE BYE" << endl;
43     return 0;
44 }
```

SLL.cpp:

```
1  #include "SLL.h"
2
3  void CreateList_103032330054(List &L){
4      L.first = NULL;
5  }
6  address CreateNewElm_103032330054(infotype data){
7      address p = new elmList;
8      p->next = NULL;
9      p->info=data;
10     return p;
11 }
12
13 void InsertLast_103032330054(List &L, address p){
14     if (L.first==NULL){
15         L.first = p;
16     }else{
17         address q = L.first;
18         while (q->next!=NULL){
19             q = q->next;
20         }
21         q->next = p;
22     }
23 }
24
25 void Show_103032330054(List L){
26     address p = L.first;
27     while (p!=NULL){
28         cout << p->info << ", ";
29         p=p->next;
30     }
31 }
32
33 address findMin_103032330054(List L){
34     address p = L.first;
35     address q = L.first;
36     while (p!=NULL){
37         if (q->info > p->info){
38             q = p;
39         }
40         p=p->next;
41     }
42     return q;
43 }
44
```

```
44
45 void insertMiddle_103032330054(List &L, int data){
46     address p,q;
47     q=CreateNewElm_103032330054(data);
48     int x = 0;
49     int y = 1;
50     p = L.first;
51     while(p!=NULL){
52         x++;
53         p=p->next;
54     }
55     if (x%2==0){
56         x = x/2;
57     }else{
58         x = (x/2) +1;
59     }
60
61     p = L.first;
62     while(y!=x){
63         p=p->next;
64         y++;
65     }
66     q->next = p->next;
67     p->next = q;
68 }
69
```

```
69
70 int selectMenu_103032330054() {
71     cout << endl << "===== MENU =====" << endl;
72     cout << "1. Menambah N data baru" << endl;
73     cout << "2. Menampilkan semua data" << endl;
74     cout << "3. Menampilkan data min" << endl;
75     cout << "4. Menambahkan data tengah" << endl;
76     cout << "0. Exit" << endl;
77     cout << "Pilihan menu: ";
78
79     int input = 0;
80     cin >> input;
81
82     return input;
83 }
84
```

SLL.h:

```
1  #ifndef SLL_H_INCLUDED
2  #define SLL_H_INCLUDED
3  #include <iostream>
4  using namespace std;
5  typedef int infotype;
6  typedef struct elmlist *address;
7  struct elmlist{
8      address next;
9      infotype info;
10 };
11 struct List{
12     address first;
13 };
14
15 void CreateList_103032330054(List &L);
16 address CreateNewElm_103032330054(infotype data);
17 void InsertLast_103032330054(List &L, address p);
18 void Show_103032330054(List L);
19 address findMin_103032330054(List L);
20 void insertMiddle_103032330054(List &L, int data);
21 int selectMenu_103032330054();
22 #endif // SLL_H_INCLUDED
23
```

Hasil:

```
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 1
Jumlah data yang akan ditambahkan: 5

Masukkan data baru: 40

Masukkan data baru: 90

Masukkan data baru: 30

Masukkan data baru: 70

Masukkan data baru: 50

===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 2
40, 90, 30, 70, 50,
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 3

Data dengan nilai minimal: 30
```

```
Data dengan nilai minimal: 30

===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 4
Masukkan data yang akan ditambahkan pada tengah list: 99

===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 2
40, 90, 30, 99, 70, 50,
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 0
BYE BYE

Process returned 0 (0x0)   execution time : 51.599 s
Press any key to continue.
```