

Main.cpp:

```
SLL.cpp X main.cpp X SLL.h X
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     char c;
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             case 2:
27                 Show_103032330054(L);
28                 break;
29             case 3:
30                 p = shortestName_103032330054(L);
31                 cout << endl << "Data terpendek: " << p->info << endl;
32                 break;
33             case 4:
34                 cout << "Masukkan banyaknya data yang ingin ditampilkan: ";
35                 cin >> NUM;
36                 cout << endl << "Masukkan karakter pertama dari data tersebut: ";
37                 cin >> c;
38                 showFirstKNameC_103032330054(L,NUM,c);
39                 break;
40         }
41
42         pilihan = selectMenu_103032330054();
43     }
44     cout << "BYE BYE" << endl;
45     return 0;
46 }
47
```

SLL.cpp:

```
SLL.cpp X main.cpp X SLL.h X
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 shortestName_103032330054(List L){
34     address p = L.first;
35     address min = L.first;
36     while (p!=NULL){
37         if (min->info.size() > p->info.size()){
38             min = p;
39         }
40         p=p->next;
41     }
42     return min;
43 }
44
45 void showFirstKNameC_103032330054(List L, int k, char c){
46     address p = L.first;
47     int l = 0;
48     while (p!=NULL && k > l){
49         if (p->info[0] == toupper(c) || p->info[0] == tolower(c)){
50             cout << p->info << ", ";
51             l++;
52         }
53         p = p->next;
54     }
55 }
```

```
56
57 int selectMenu_103032330054() {
58     cout << endl << "===== MENU =====" << endl;
59     cout << "1. Menambah N data baru" << endl;
60     cout << "2. Menampilkan semua data" << endl;
61     cout << "3. Menampilkan nama terpendek" << endl;
62     cout << "4. Menampilkan k pengunjung yang berawalan c" << endl;
63     cout << "0. Exit" << endl;
64     cout << "Pilihan menu: ";
65
66     int input = 0;
67     cin >> input;
68
69     return input;
70 }
71
```

SLL.h:

```
1  #ifndef SLL_H_INCLUDED
2  #define SLL_H_INCLUDED
3  #include <iostream>
4  #include <string>
5  using namespace std;
6  typedef string infotype;
7  typedef struct elmlist *address;
8  struct elmlist{
9      address next;
10     infotype info;
11 };
12 struct List{
13     address first;
14 };
15
16 void CreateList_103032330054(List &L);
17 address CreateNewElm_103032330054(infotype data);
18 void InsertLast_103032330054(List &L, address p);
19 void Show_103032330054(List L);
20 address shortestName_103032330054(List L);
21 void showFirstKNameC_103032330054(List L, int k, char c);
22 int selectMenu_103032330054();
23 #endif // SLL_H_INCLUDED
24
```

Hasil:

```
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 1
Jumlah data yang akan ditambahkan: 8

Masukkan data baru: Ihab

Masukkan data baru: Hasanain

Masukkan data baru: Akmal

Masukkan data baru: Susilo

Masukkan data baru: Bambang

Masukkan data baru: Yudoyono

Masukkan data baru: Ihub

Masukkan data baru: ihabb

===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 2
Ihab, Hasanain, Akmal, Susilo, Bambang, Yudoyono, Ihub, ihabb,
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 3
```

```
Pilihan menu: 3
Data terpendek: Ihab
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 4
Masukkan banyaknya data yang ingin ditampilkan: 1
Masukkan karakter pertama dari data tersebut: i
Ihab,
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 4
Masukkan banyaknya data yang ingin ditampilkan: 99
Masukkan karakter pertama dari data tersebut: i
Ihab, Ihub, ihabb,
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan nama terpendek
4. Menampilkan k pengunjung yang berawalan c
0. Exit
Pilihan menu: 0
BYE BYE
```