

teori > tugasSingleLinkedList > C++ no1.cpp > main()

```
1  #include <iostream>
2  using namespace std;
3
4  //Deklarasi List
5  struct Node {
6      float info;
7      Node* next;
8  };
9
10 struct List {
11     Node* first;
12 };
13
14 //Membuat list kosong
15 void createList(List &L) {
16     L.first = NULL;
17 }
18
19 //Membuat elemen baru
20 Node* createElement(float X) {
21     Node* elm = new Node;
22     elm->info = X;
23     elm->next = NULL;
24     return elm;
25 }
26
27 //Menyisipkan elemen secara ascending
28 void insertAscending(List &L, Node* elm) {
29     if (L.first == NULL) {
30         L.first = elm;
31     }
32     else if (elm->info < L.first->info) {
33         elm->next = L.first;
34         L.first = elm;
35     }
36     else {
37         Node* p = L.first;
38         while (p->next != NULL && p->next->info < elm->info) {
39             p = p->next;
40         }
41         elm->next = p->next;
42         p->next = elm;
43     }
44 }
45
46 void printList(List L) {
47     Node* p = L.first;
48     while (p != NULL) {
49         cout << p->info << " ";
50         p = p->next;
51     }
52     cout << endl;
53 }
54
55 int main() {
56     createList(List);
57
58     insertAscending(List, createElement(10.5));
59     insertAscending(List, createElement(12.0));
60     insertAscending(List, createElement(20.9));
61     insertAscending(List, createElement(25.1));
62     insertAscending(List, createElement(15.5)); // insert tambahan untuk cek urutan
63
64     cout << "Isi Linked List (Ascending): ";
65     printList(List);
66
67     return 0;
68 }
69
```

teori > tugasSingleLinkedList > C++ no1.cpp > main()

```
1  #include <iostream>
2  using namespace std;
3
4  //Deklarasi List
5  struct Node {
6      float info;
7      Node* next;
8  };
9
10 struct List {
11     Node* first;
12 };
13
14 //Membuat list kosong
15 void createList(List &L) {
16     L.first = NULL;
17 }
18
19 //Membuat elemen baru
20 Node* createElement(float X) {
21     Node* elm = new Node;
22     elm->info = X;
23     elm->next = NULL;
24     return elm;
25 }
26
27 //Menyisipkan elemen secara ascending
28 void insertAscending(List &L, Node* elm) {
29     if (L.first == NULL) {
30         L.first = elm;
31     }
32     else if (elm->info < L.first->info) {
33         elm->next = L.first;
34         L.first = elm;
35     }
36     else {
37         Node* p = L.first;
38         while (p->next != NULL && p->next->info < elm->info) {
39             p = p->next;
40         }
41         elm->next = p->next;
42         p->next = elm;
43     }
44 }
45
46 void printList(List L) {
47     Node* p = L.first;
48     while (p != NULL) {
49         cout << p->info << " ";
50         p = p->next;
51     }
52     cout << endl;
53 }
54
55 int main() {
56     List L;
57     createList(L);
58
59     insertAscending(L, createElement(10.5));
60     insertAscending(L, createElement(12.0));
61     insertAscending(L, createElement(20.9));
62     insertAscending(L, createElement(25.1));
63     insertAscending(L, createElement(15.5)); // insert tambahan untuk cek urutan
64
65     cout << "Isi Linked List (Ascending): ";
66     printList(L);
67
68     return 0;
69 }
70
```

- PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no1.cpp -o no1 } ; if (\$?) { .\no1 }

Isi Linked List (Ascending): 10.5 12 15.5 20.9 25.1

- PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList>

teori > tugasSingleLinkedList >  no2.cpp > ...

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      float info;
6      Node* next;
7  };
8
9  struct List {
10     Node* first;
11 };
12
13 void createList(List &L) {
14     L.first = NULL;
15 }
16
17 Node* createElement(float X) {
18     Node* elm = new Node;
19     elm->info = X;
20     elm->next = NULL;
21     return elm;
22 }
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42 // Fungsi untuk mencari elemen bernilai X
43 Node* findElement(List L, float X) {
44     Node* p = L.first;
45     while (p != NULL && p->info != X) {
46         p = p->next;
47     }
48     return p;
49 }
50
```

```

50
51 // Menghapus elemen sesuai kondisi soal
52 void deleteByCondition(List &L, float X) {
53     Node* p = findElement(L, X);
54
55     if (p == NULL) {
56         cout << "Tidak ada elemen bernilai " << X << " dalam list." << endl;
57         return;
58     }
59
60     // Jika elemen pertama
61     if (p == L.first) {
62         L.first = p->next;
63         delete p;
64     }
65     // Jika elemen terakhir
66     else if (p->next == NULL) {
67         Node* q = L.first;
68         while (q->next != p) {
69             q = q->next;
70         }
71         q->next = NULL;
72         delete p;
73     }
74     // Jika elemen di tengah → hapus elemen setelah X
75     else {
76         Node* temp = p->next;
77         p->next = temp->next;
78         delete temp;
79     }
80 }

```

```

82 void printList(List L) {
83     Node* p = L.first;
84     while (p != NULL) {
85         cout << p->info << " ";
86         p = p->next;
87     }
88     cout << endl;
89 }
90
91 int main() {
92     List L;
93     createList(L);
94
95     insertAscending(L, createElement(10.5));
96     insertAscending(L, createElement(12.0));
97     insertAscending(L, createElement(20.9));
98     insertAscending(L, createElement(25.1));
99
100     cout << "Isi awal list: ";
101     printList(L);
102
103     float X;
104     cout << "Masukkan nilai X untuk dihapus: ";
105     cin >> X;
106
107     deleteByCondition(L, X);
108
109     cout << "Isi list setelah proses penghapusan: ";
110     printList(L);
111
112     return 0;
113 }

```

- PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no2.cpp -o no2 } ; if (\$?) { .\no2 }
 Isi awal list: 10.5 12 20.9 25.1
 Masukkan nilai X untuk dihapus: 10.5
 Isi list setelah proses penghapusan: 12 20.9 25.1
 PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no2.cpp -o no2 } ; if (\$?) { .\no2 }
- r3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no2.cpp -o no2 } ; if (\$?) { .\no2 }
 Isi awal list: 10.5 12 20.9 25.1
 Masukkan nilai X untuk dihapus: 12
 Isi list setelah proses penghapusan: 10.5 12 25.1
 PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no2.cpp -o no2 } ; if (\$?) { .\no2 }
- r3\strukturData\teori\tugasSingleLinkedList\" ; if (\$?) { g++ no2.cpp -o no2 } ; if (\$?) { .\no2 }
 Isi awal list: 10.5 12 20.9 25.1
 Masukkan nilai X untuk dihapus: 25.1
 Isi list setelah proses penghapusan: 10.5 12 20.9
- PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasSingleLinkedList> █