

LAPORAN PRAKTIKUM
PERTEMUAN 4
SINGLE LINKED LIST (BAGIAN PERTAMA)



Nama :

Salman Alfarisi (2311104036)

Dosen :

Yudha Islami Sulistya, S.Kom.,
M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. UNGUIDED

1. SOAL 1

```
1  #include <iostream>
2  using namespace std;
3
4
5  struct Node{
6      int info;
7      Node* next;
8  };
9
10 Node* alokasi(int x){
11     Node* newNode = new Node();
12     newNode->info = x;
13     newNode->next = nullptr;
14     return newNode;
15 }
16 void insertFirst(Node** first, int X) {
17     Node* P = alokasi(X);
18     P->next = *first;
19     *first = P;
20 }
21
22
23 void insertLast(Node** first, int X) {
24     Node* P = alokasi(X);
25     if (*first == nullptr) {
26         *first = P;
27     } else {
28         Node* Q = *first;
29         while (Q->next != nullptr) {
30             Q = Q->next;
31         }
32         Q->next = P;
33     }
34 }
35
36
37 int main(){
38     Node * first = nullptr;
39     insertFirst(&first,10 );
40     insertLast(&first,20 );
41     insertFirst(&first, 5);
42
43     Node* temp = first;
44     while (temp != nullptr) {
45         std::cout << temp->info;
46         if (temp->next != nullptr) {
47             std::cout << "->";
48         }
49         temp = temp->next;
50     }
51
52     return 0;
53 }
```

insertFirst dan insertLast adalah dua cara untuk menambahkan node dalam linked list, masing-masing di depan dan di belakang. insertFirst lebih efisien dalam hal waktu, sementara insertLast memberikan cara untuk menambahkan elemen di akhir list, menjaga urutan elemen yang ditambahkan.

Keduanya penting dalam pengelolaan linked list dan menunjukkan fleksibilitas struktur data ini dalam menyimpan dan mengorganisir data. Contoh Output:

```
PROBLEMS  OUTPUT  DEBUG CONS
PS C:\ITTP\TUGAS\SEMESTER 3\S
NGUIDED_01.cpp -o UNGUIDED_01
5->10->20
PS C:\ITTP\TUGAS\SEMESTER 3\S
```

2. SOAL 2

```
1  #include <iostream>
2  using namespace std;
3
4
5  struct Node{
6      int info;
7      Node* next;
8  };
9
10 Node* alokasi(int x){
11     Node* newNode = new Node();
12     newNode->info = x;
13     newNode->next = nullptr;
14     return newNode;
15 }
16 void insertFirst(Node** first, int X) {
17     Node* P = alokasi(X);
18     P->next = *first;
19     *first = P;
20 }
21
22
23 void insertLast(Node** first, int X) {
24     Node* P = alokasi(X);
25     if (*first == nullptr) {
26         *first = P;
27     } else {
28         Node* Q = *first;
29         while (Q->next != nullptr) {
30             Q = Q->next;
31         }
32         Q->next = P;
33     }
34 }
35
36
37 bool deleteNode(Node** first, int value) {
38     if (*first == nullptr) {
39         return false;
40     }
41     Node* current = *first;
42     Node* previous = nullptr;
43
44     while (current != nullptr && current->info != value) {
45         previous = current;
46         current = current->next;
47     }
48     if (current == nullptr) {
49         return false;
50     }
51     if (previous == nullptr) {
52         *first = current->next;
53     } else {
54         previous->next = current->next;
55     }
56
57     delete current;
58     return true;
59 }
60
61
62
63 int main(){
64     Node * first = nullptr;
65     insertFirst(&first,10 );
66     insertLast(&first,20 );
67     insertFirst(&first, 5);
68     deleteNode(&first, 10);
69
70     Node* temp = first;
71     while (temp != nullptr) {
72         std::cout << temp->info;
73         if (temp->next != nullptr) {
74             std::cout << "->";
75         }
76         temp = temp->next;
77     }
78
79     return 0;
80 }
```

Dalam kode ini kita menghapus nilai yang telah kita tentukan contohnya adalah nilai 10.

contoh output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\UNGUIDED_02> g++ UNGUIDED_02.cpp -o UNGUIDED_02 } ;
5->20
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\UNGUIDED_02>
```

3. SOAL 3

```
1  #include <iostream>
2  using namespace std;
3
4
5  struct Node{
6      int info;
7      Node* next;
8  };
9
10 Node* alokasi(int x){
11     Node* newNode = new Node();
12     newNode->info = x;
13     newNode->next = nullptr;
14     return newNode;
15 }
16 void insertFirst(Node** first, int X) {
17     Node* P = alokasi(X);
18     P->next = *first;
19     *first = P;
20 }
21
22
23 void insertLast(Node** first, int X) {
24     Node* P = alokasi(X);
25     if (*first == nullptr) {
26         *first = P;
27     } else {
28         Node* Q = *first;
29         while (Q->next != nullptr) {
30             Q = Q->next;
31         }
32         Q->next = P;
33     }
34 }
35
36 bool searchNode(Node* first, int value) {
37     Node* current = first;
38     while (current != nullptr) {
39         if (current->info == value) {
40             return true;
41         }
42         current = current->next;
43     }
44     return false;
45 }
46
47 int countLength(Node* first) {
48     int count = 0;
49     Node* current = first;
50     while (current != nullptr) {
51         count++;
52         current = current->next;
53     }
54     return count;
55 }
56
57 int main(){
58     Node * first = nullptr;
59     insertFirst(&first,10 );
60     insertLast(&first,20);
61     insertFirst(&first, 5);
62
63
64
65     int nilaiDicari = 20;
66     if (searchNode(first, nilaiDicari)) {
67         std::cout << "Node dengan nilai " << nilaiDicari << " ditemukan" << endl;
68     } else {
69         std::cout << "Node dengan nilai " << nilaiDicari << " tidak ditemukan." << std::endl;
70     }
71     int length = countLength(first);
72     cout<<"Panjang Linked List = "<<countLength(first);
73
74     return 0;
75 }
```

Dalam kode ini kita mencari angka 20 apakah ada di linked list di atas, dan kita mencetak panjang dari list kita, contoh output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\UNGUIDED_03.cpp -o UNGUIDED_03 } ; if ($?)
Node dengan nilai 20 ditemukan
Panjang Linked List = 3
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA
```