

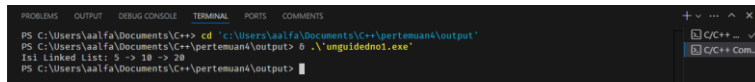
## UNGUIDED LAPORAN PRAKTIKUM STUKTUR DATA MODUL 4

### 1. Unguided

#### 4.1. Kode program :

```
1  #include <iostream>
2  using namespace std;
3
4  // Struktur node dari linked list
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 // Fungsi untuk menambah node di depan
11 void insertFront(Node** head, int newData) {
12     // Buat node baru
13     Node* newNode = new Node();
14     newNode->data = newData;
15
16     // Hubungkan node baru ke head lama
17     newNode->next = *head;
18
19     // Jadikan node baru sebagai head
20     *head = newNode;
21 }
22
23 // Fungsi untuk menambah node di belakang
24 void insertBack(Node** head, int newData) {
25     // Buat node baru
26     Node* newNode = new Node();
27     newNode->data = newData;
28     newNode->next = nullptr;
29
30     // Jika linked list kosong, jadikan node baru sebagai head
31     if (*head == nullptr) {
32         *head = newNode;
33         return;
34     }
35
36     // Jika tidak, cari node terakhir
37     Node* last = *head;
38     while (last->next != nullptr) {
39         last = last->next;
40     }
41
42     // Hubungkan node terakhir dengan node baru
43     last->next = newNode;
44 }
45
46 // Fungsi untuk mencetak seluruh isi linked list
47 void printList(Node* head) {
48     while (head != nullptr) {
49         cout << head->data;
50         if (head->next != nullptr) {
51             cout << " → ";
52         }
53         head = head->next;
54     }
55     cout << endl;
56 }
57
58 int main() {
59     Node* head = nullptr; // Inisialisasi linked list kosong
60
61     // Contoh input
62     insertFront(&head, 10); // Tambah node di depan (10)
63     insertBack(&head, 20);  // Tambah node di belakang (20)
64     insertFront(&head, 5);  // Tambah node di depan (5)
65
66     // Cetak linked list
67     cout << "Isi Linked List: ";
68     printList(head); // Output: 5 → 10 → 20
69
70     return 0;
71 }
72
```

Output dari kode program :



#### 4.2. Kode program :

```
1 #include <iostream>
2 using namespace std;
3
4 // Struktur node dari linked list
5 struct Node {
6     int data;
7     Node* next;
8 };
9
10 // Fungsi untuk menambah node di depan
11 void insertFront(Node** head, int newData) {
12     Node* newNode = new Node();
13     newNode->data = newData;
14     newNode->next = *head;
15     *head = newNode;
16 }
17
18 // Fungsi untuk menambah node di belakang
19 void insertBack(Node** head, int newData) {
20     Node* newNode = new Node();
21     newNode->data = newData;
22     newNode->next = nullptr;
23
24     if (*head == nullptr) {
25         *head = newNode;
26         return;
27     }
28
29     Node* last = *head;
30     while (last->next != nullptr) {
31         last = last->next;
32     }
33     last->next = newNode;
34 }
35
36 // Fungsi untuk menghapus node dengan nilai tertentu
37 void deleteNode(Node** head, int key) {
38     Node* temp = *head;
39     Node* prev = nullptr;
40
41     // Jika node head yang akan dihapus
42     if (temp != nullptr && temp->data == key) {
43         *head = temp->next; // Ubah head
44         delete temp; // Hapus node
45         return;
46     }
47
48     // Cari node yang akan dihapus, simpan node sebelumnya
49     while (temp != nullptr && temp->data != key) {
50         prev = temp;
51         temp = temp->next;
52     }
53
54     // Jika node dengan nilai tersebut tidak ditemukan
55     if (temp == nullptr) {
56         cout << "Nilai " << key << " tidak ditemukan dalam linked list.\n";
57         return;
58     }
59
60     // Hapus node
61     prev->next = temp->next;
62     delete temp;
63 }
64
65 // Fungsi untuk mencetak seluruh isi linked list
66 void printList(Node* head) {
67     while (head != nullptr) {
68         cout << head->data;
69         if (head->next != nullptr) {
70             cout << " -> ";
71         }
72         head = head->next;
73     }
74     cout << endl;
75 }
```

```

1  int main() {
2      Node* head = nullptr; // Inisialisasi linked list kosong
3
4      // Contoh input
5      insertFront(&head, 10); // Tambah node di depan (10)
6      insertBack(&head, 20); // Tambah node di belakang (20)
7      insertFront(&head, 5); // Tambah node di depan (5)
8
9      // Cetak linked list sebelum penghapusan
10     cout << "Linked List sebelum penghapusan: ";
11     printList(head); // Output: 5 → 10 → 20
12
13     // Hapus node dengan nilai 10
14     deleteNode(&head, 10);
15
16     // Cetak linked list setelah penghapusan
17     cout << "Linked List setelah penghapusan: ";
18     printList(head); // Output: 5 → 20
19
20     return 0;
21 }
22

```

Output dari kode program :

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  focus follow in explorer (ctrl + click)
PS C:\Users\aa1fa\Documents\C++> cd "C:\Users\aa1fa\Documents\C++\pertemuan4\output"
PS C:\Users\aa1fa\Documents\C++\pertemuan4\output> g++ .\unguidedno2.exe
Linked List sebelum penghapusan: 5 -> 10 -> 20
Linked List setelah penghapusan: 5 -> 20
PS C:\Users\aa1fa\Documents\C++\pertemuan4\output>

```

### 4.3. Kode program :

```
1 #include <iostream>
2 using namespace std;
3
4 // Struktur node dari linked list
5 struct Node {
6     int data;
7     Node* next;
8 };
9
10 // Fungsi untuk menambah node di depan
11 void insertFront(Node** head, int newData) {
12     Node* newNode = new Node();
13     newNode->data = newData;
14     newNode->next = *head;
15     *head = newNode;
16 }
17
18 // Fungsi untuk menambah node di belakang
19 void insertBack(Node** head, int newData) {
20     Node* newNode = new Node();
21     newNode->data = newData;
22     newNode->next = nullptr;
23
24     if (*head == nullptr) {
25         *head = newNode;
26         return;
27     }
28
29     Node* last = *head;
30     while (last->next != nullptr) {
31         last = last->next;
32     }
33     last->next = newNode;
34 }
35
36 // Fungsi untuk mencari node dengan nilai tertentu
37 bool searchNode(Node* head, int key) {
38     Node* current = head;
39     while (current != nullptr) {
40         if (current->data == key) {
41             return true; // Node ditemukan
42         }
43         current = current->next;
44     }
45     return false; // Node tidak ditemukan
46 }
47
48 // Fungsi untuk menghitung panjang linked list
49 int countNodes(Node* head) {
50     int count = 0;
51     Node* current = head;
52     while (current != nullptr) {
53         count++;
54         current = current->next;
55     }
56     return count;
57 }
58
59 // Fungsi untuk mencetak seluruh isi linked list
60 void printList(Node* head) {
61     while (head != nullptr) {
62         cout << head->data;
63         if (head->next != nullptr) {
64             cout << " -> ";
65         }
66         head = head->next;
67     }
68     cout << endl;
69 }
70
71 int main() {
72     Node* head = nullptr; // Inisialisasi linked list kosong
73
74     // Contoh input
75     insertFront(&head, 10); // Tambah node di depan (10)
76     insertBack(&head, 20); // Tambah node di belakang (20)
77     insertFront(&head, 5); // Tambah node di depan (5)
78
79     // Cari node dengan nilai 20
80     int searchValue = 20;
81     if (searchNode(head, searchValue)) {
82         cout << "Node dengan nilai " << searchValue << " ditemukan." << endl;
83     } else {
84         cout << "Node dengan nilai " << searchValue << " tidak ditemukan." << endl;
85     }
86
87     // Hitung panjang linked list
88     int length = countNodes(head);
89     cout << "Panjang linked list: " << length << endl;
90
91     return 0;
92 }
93
```

### Output dari kode program :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\aa\fa\Documents\C++> cd 'c:\Users\aa\fa\Documents\C++\pertemuan4\output'
PS C:\Users\aa\fa\Documents\C++\pertemuan4\output> g++ .\unguidedno3.exe
Node dengan nilai 20 ditemukan.
Panjang linked list: 3
PS C:\Users\aa\fa\Documents\C++\pertemuan4\output>
```

