

HANDS-ON LABS 4

STRUKTUR DATA



Oleh:

Martogi Jekson C. Siagian 2105541054

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS UDAYANA

2023

Hands-On Labs 4

Tugas

1. Berdasarkan contoh program sebelumnya, tambahkanlah 2 menu baru yakni:

- Menu untuk menghitung panjang linked list dan
- Menu untuk menghitung jumlah keseluruhan nilai yang sudah diinputkan pada linked list

Jawab:

Berikut Merupakan Kode yang telah saya buat dengan menambahkan 2 menu yaitu menghitung Panjang dan jumlah nya :

```
Tugas 4.cpp X
1  #include <iostream>
2  #include <conio.h>
3  #include <stdlib.h>
4
5  using namespace std;
6
7  struct TNode {
8      int data;
9      TNode *next;
10 };
11
12 TNode *head, *tail;
13
14 void createList() {
15     head = NULL;
16     tail = NULL;
17 }
18
19 int isEmpty() {
20     if(head == NULL) {
21         return 1;
22     } else {
23         return 0;
24     }
25 }
26
27 void insertDepan(int data) {
28     TNode *baru;
29     baru = new TNode;
30     baru->data = data;
31     baru->next = head;
32     head = baru;
33     if(head->next == NULL) {
34         tail = head;
35     }
36     cout<<"Data masuk\n";
37 }
38
```

Tugas 4.cpp X

```
39 void insertBelakang(int data) {
40     TNode *baru;
41     baru = new TNode;
42     baru->data = data;
43     if (isEmpty() == 0) {
44         tail->next = baru;
45         tail = baru;
46         tail->next = NULL;
47     } else {
48         head = baru;
49         tail = baru;
50         tail->next = NULL;
51     }
52     cout << "Data masuk\n";
53 }
54
55 void tampil() {
56     TNode *bantu;
57     bantu = head;
58     if (isEmpty() == 0) {
59         while (bantu != NULL) {
60             cout << bantu->data << "->";
61             bantu = bantu->next;
62         }
63         cout << "NULL";
64     } else {
65         cout << "Masih kosong\n";
66     }
67 }
68
69 void hapusDepan() {
70     TNode *hapus;
71     int d;
72     if (isEmpty() == 0) {
73         if (head != tail) {
74             hapus = head;
75             d = hapus->data;
76             head = head->next;
77             delete hapus;
78             cout << "Data yang dihapus = " << d << endl;
79         } else {
80             delete head;
81             head = NULL;
82             tail = NULL;
83             cout << "List telah kosong\n";
84         }
85     }
```

Tugas 4.cpp X

```
85     }else {
86         cout <<"Masih kosong\n";
87     }
88 }
89
90 void hapusBelakang() {
91     TNode *hapus;
92     int d;
93     if (isEmpty()==0) {
94         if(head!=tail) {
95             hapus=tail;
96             d=hapus->data;
97             tail=tail->next;
98             delete hapus;
99             tail->next=NULL;
100             cout<<"Data yang dihapus = "<<d<<endl;
101         } else {
102             delete head;
103             head = NULL;
104             tail = NULL;
105             cout<<"List telah kosong\n";
106         }
107     }else {
108         cout <<"Masih kosong\n";
109     }
110 }
111
112 void clear() {
113     TNode *bantu;
114     if (isEmpty()==0) {
115         bantu = head;
116         while(bantu!=NULL) {
117             bantu = head->next;
118             delete head;
119             head = bantu;
120         }
121         head = NULL;
122         tail = NULL;
123     } else {
124         cout <<"Masih kosong\n";
125     }
126 }
127
```

```
127
128 □ int countLength() {
129     int length = 0;
130     TNode *bantu = head;
131     □ while(bantu != NULL) {
132         length++;
133         bantu = bantu->next;
134     }
135     return length;
136 }
137
138 □ int countTotal() {
139     int total = 0;
140     TNode *bantu = head;
141     □ while(bantu != NULL) {
142         total += bantu->data;
143         bantu = bantu->next;
144     }
145     return total;
146 }
147
148 □ int main() {
149     int pil, data, length, total;
150     □ do {
151         system("cls");
152         cout<<" =====<<endl;
153         cout<<" = PROGRAM LINKED LIST  =<<endl;
154         cout<<" =====<<endl;
155         cout<<" = 1. Insert Depan      =<<endl;
156         cout<<" = 2. Insert Belakang   =<<endl;
157         cout<<" = 3. Delete Depan      =<<endl;
158         cout<<" = 4. Delete Belakang   =<<endl;
159         cout<<" = 5. Tampil Data       =<<endl;
160         cout<<" = 6. Clear              =<<endl;
161         cout<<" = 7. Count Length      =<<endl;
162         cout<<" = 8. Count Total       =<<endl;
163         cout<<" = 9. Exit              =<<endl;
164         cout<<" =====<<endl;
165         cout<<"Masukkan pilihan anda: ";
166         cin>>pil;
167
```

```

167
168     switch(pil) {
169         case 1:
170             cout<<"Masukkan data: ";
171             cin>>data;
172             insertDepan(data);
173             break;
174         case 2:
175             cout<<"Masukkan data: ";
176             cin>>data;
177             insertBelakang(data);
178             break;
179         case 3:
180             hapusDepan();
181             break;
182         case 4:
183             hapusBelakang();
184             break;
185         case 5:
186             tampil();
187             break;
188         case 6:
189             clear();
190             break;
191         case 7:
192             length = countLength();
193             cout<<"Panjang linked list = "<<length<<endl;
194             break;
195         case 8:
196             total = countTotal();
197             cout<<"Jumlah keseluruhan nilai = "<<total<<endl;
198             break;
199     }
200     getch();
201 } while(pil != 9);
202
203 return 0;
204 }
205

```

Berikut Merupakan Penjelasan dari kode yang saya buat :

- **Include Header:** Program ini menggunakan tiga header yaitu `<iostream>`, `<conio.h>`, dan `<stdlib.h>`.
- **Namespace:** Program ini menggunakan namespace `std` untuk menggunakan kelas dan fungsi dari standar C++.
- **Struktur TNode:** Ini adalah definisi dari struktur node yang akan digunakan dalam linked list. Setiap node memiliki dua anggota: data yang menyimpan data integer, dan next yang merupakan pointer ke node berikutnya.
- **Deklarasi Variabel Global:** Variabel global head dan tail digunakan untuk menyimpan pointer ke kepala dan ekor dari linked list.

- **createList()** : Fungsi ini digunakan untuk menginisialisasi linked list dengan mengatur head dan tail menjadi NULL.
- **isEmpty()** : Fungsi ini memeriksa apakah linked list kosong atau tidak, dan mengembalikan 1 jika kosong dan 0 jika tidak.
- **insertDepan(int data)** : Fungsi ini digunakan untuk menyisipkan data ke depan linked list. Ini akan membuat node baru, mengatur head untuk menunjuk ke node baru tersebut, dan jika ini adalah satu-satunya node dalam linked list, maka tail juga akan menunjuk ke node baru tersebut.
- **insertBelakang(int data)** : Fungsi ini digunakan untuk menyisipkan data ke belakang linked list. Ini akan membuat node baru dan mengatur tail untuk menunjuk ke node baru tersebut. Jika linked list kosong, maka head juga akan menunjuk ke node baru tersebut.
- **tampil()** : Fungsi ini digunakan untuk menampilkan semua data dalam linked list, dari head hingga tail.
- **hapusDepan()** : Fungsi ini digunakan untuk menghapus node pertama dari linked list. Jika linked list masih memiliki lebih dari satu node, maka head akan diubah untuk menunjuk ke node berikutnya, dan node pertama akan dihapus.
- **hapusBelakang()** : Fungsi ini digunakan untuk menghapus node terakhir dari linked list. Jika linked list memiliki lebih dari satu node, maka tail akan diubah untuk menunjuk ke node sebelumnya, dan node terakhir akan dihapus.
- **clear()** : Fungsi ini digunakan untuk menghapus semua node dalam linked list dengan mengiterasi melalui linked list dan menghapus setiap node.
- **countLength()** : Fungsi ini digunakan untuk menghitung panjang (jumlah node) dari linked list dengan mengiterasi melalui linked list.

- **countTotal()** : Fungsi ini digunakan untuk menghitung total dari seluruh data dalam linked list dengan mengiterasi melalui linked list.
- **main()** : Ini adalah fungsi utama dari program. Program ini menampilkan menu operasi yang dapat digunakan pada linked list seperti menambah data, menghapus data, menampilkan data, dan lainnya. Loop do-while digunakan untuk memungkinkan pengguna terus menggunakan program hingga memilih untuk keluar (pilihan 9).

Seluruh operasi pada linked list diimplementasikan dalam fungsi-fungsi terpisah yang memanipulasi linked list dan menu utama memberikan akses pengguna ke operasi-operasi ini. Pada program ini saya menambahkan Fungsi **countLength()** & **countTotal()** Untuk Menghitung Panjang dan jumlah Linked list nya, Berikut Merupakan hasil dari program Ketika dijalankan.

MENU

```

=====
= PROGRAM LINKED LIST =
=====
= 1. Insert Depan      =
= 2. Insert Belakang  =
= 3. Delete Depan     =
= 4. Delete Belakang  =
= 5. Tampil Data      =
= 6. Clear            =
= 7. Count Length     =
= 8. Count Total      =
= 9. Exit             =
=====
Masukkan pilihan anda: |

```

Tampil Data

```

=====
= PROGRAM LINKED LIST =
=====
= 1. Insert Depan      =
= 2. Insert Belakang  =
= 3. Delete Depan     =
= 4. Delete Belakang  =
= 5. Tampil Data      =
= 6. Clear            =
= 7. Count Length     =
= 8. Count Total      =
= 9. Exit             =
=====
Masukkan pilihan anda: 5
9->5->7->NULL|

```


Count Length

```
"D:\Semester 5\Struktur Data" × +
=====
= PROGRAM LINKED LIST =
=====
= 1. Insert Depan      =
= 2. Insert Belakang  =
= 3. Delete Depan     =
= 4. Delete Belakang  =
= 5. Tampil Data      =
= 6. Clear            =
= 7. Count Length     =
= 8. Count Total      =
= 9. Exit             =
=====
Masukkan pilihan anda: 7
Panjang linked list = 3
```

Count Total

```
"D:\Semester 5\Struktur Data" × +
=====
= PROGRAM LINKED LIST =
=====
= 1. Insert Depan      =
= 2. Insert Belakang  =
= 3. Delete Depan     =
= 4. Delete Belakang  =
= 5. Tampil Data      =
= 6. Clear            =
= 7. Count Length     =
= 8. Count Total      =
= 9. Exit             =
=====
Masukkan pilihan anda: 8
Jumlah keseluruhan nilai = 21
```