

# LAPORAN PRAKTIKUM

Matakuliah	Struktur Data
Pertemuan ke	9
Nama Praktikan	Wijayanto Agung Wibowo
NIM	22.11.4552
NILAI (diisi oleh dosen / asisten praktikum)	

## A. Tujuan

Setelah praktikum ini, praktikan diharapkan dapat:

1. Memahami tipe data dasar Double Linked List
2. Memahami tipe data bentukan
3. Stuktur Program menggunakan bahasa C++ Praktikum

## B. Hasil Percobaan

### 1. Percobaan 1

#### a) Tampilan Coding

```
1  /*program doubly linked list*/
2
3  #include <iostream>
4  #include <conio.h>
5  #include <stdlib.h>
6
7  using namespace std;
8
9  typedef struct node* simpul;
10 struct node
11 {
12     char Isi;
13     simpul kanan;
14     simpul kiri;
15 };
16
17 //Prototype fungsi
18 void Sisip_Depan(simpul& DL, char elemen);
19 void Sisip_Belakang(simpul& DL, char elemen);
20 void Sisip_Tengah1(simpul& DL, char elemen1, char elemen2);
21 void Sisip_Tengah2(simpul& DL, char elemen1, char elemen2);
22 void Hapus_Depan(simpul& DL);
23 void Hapus_Belakang(simpul& DL);
24 void Hapus_Tengah(simpul& DL, char elemen);
25 void Cetak(simpul DL);
26 void pencarianHuruf(simpul& DL, char elemen);
27 int jumlahNode(simpul& DL);
28 void sortingData(simpul& DL);
29
30
31 void menu() {
32     cout << "Menu: " << endl;
33     cout << "1. Menyisipkan Simpul Di Depan" << endl;
34     cout << "2. Menyisipkan Simpul Di Belakang" << endl;
35     cout << "3. Penyisipan Simpul Di Tengah Sebelum Node Tertentu " << endl;
36     cout << "4. Penyisipan Simpul Di Tengah Setelah Node Tertentu" << endl;
37     cout << "5. Hapus Simpul Depan" << endl;
38     cout << "6. Hapus Simpul Belakang" << endl;
39     cout << "7. Hapus Simpul Tengah" << endl;
40     cout << "8. Pencarian Data" << endl;
41     cout << "9. Pengurutan Data" << endl;
42     cout << "10. Cetak Data" << endl;
43     cout << "11. Exit" << endl;
44 }
```

```

46 //fungsi main
47
48 int main() {
49     char huruf, huruf2;
50     simpul DL = NULL;
51     int i, n, pilihan;
52     cout << "\t == OPERASI PADA DOUBLY LINKED LIST ==\n\n";
53     menu:
54     menu();
55     cout << "Masukan Pilihan Menu:"; cin >> pilihan;
56     switch (pilihan) {
57     case 1:
58         //sisip depan
59         cout << "==" Percobaan 1 == " << endl;
60         cout << "==" Penyisipan Simpul Di Depan == " << endl;
61         cout << "Masukan jumlah data : "; cin >> n;
62         cout << endl;
63         for (i = 1; i <= n; i++) {
64             cout << "Masukan huruf :"; cin >> huruf;
65             Sisip_Depan(DL, huruf);
66         }
67         Cetak(DL);
68         break;
69     case 2:
70         //Sisip belakang
71         cout << "\n\n == Percobaan 2 == " << endl;
72         cout << "==" Penyisipan Simpul Di Belakang == " << endl;
73         cout << "Masukan jumlah data : "; cin >> n;
74         cout << endl;
75         for (i = 1; i <= n; i++) {
76             cout << "Masukan huruf :"; cin >> huruf;
77             Sisip_Belakang(DL, huruf);
78         }
79         Cetak(DL);
80         break;
81     case 3:
82         //sisip simpul setelah simpul tertentu
83         cout << "\n\n == Percobaan 3 == " << endl;
84         cout << "==" Penyisipan Simpul Di Tengah Sebelum Node Tertentu == " << endl << endl;
85         cout << "Masukan huruf yang disisipkan :";
86         cin >> huruf;
87         cout << "Disisipkan Setelah Huruf : "; cin >> huruf2;
88         cout << huruf << " Disisipkan Setelah " << huruf2 << endl;
89         Sisip_Tengah1(DL, huruf, huruf2);
90         Cetak(DL);
91         break;
92     case 4:
93         //Sisip Simpul sebelum simpul tertentu
94         cout << "\n\n == Percobaan 4 == " << endl;
95         cout << "Penyisipan Simpul Di Tengah Setelah Node Tertentu == " << endl;
96         cout << "Masukan Huruf :";
97         cin >> huruf;
98         cout << "Disisip sebelum Huruf :";
99         cin >> huruf2;

```

```

89     Sisip_Tengah1(DL, huruf, huruf2);
90     Cetak(DL);
91     break;
92 case 4:
93     //Sisip Simpul sebelum simpul tertentu
94     cout << "\n\n == Percobaan 4 == " << endl;
95     cout << "Penyisipan Simpul Di Tengah Setelah Node Tertentu == " << endl;
96     cout << "Masukan Huruf :";
97     cin >> huruf;
98     cout << "Disisip sebelum Huruf :";
99     cin >> huruf2;
100    cout << huruf << " Disisip Sebelum " << huruf2 << endl;
101    Sisip_Tengah2(DL, huruf, huruf2);
102    Cetak(DL);
103    break;
104 case 5:
105     //Hapus Simpul Depan
106     cout << "\n\n == Percobaan 5 == " << endl;
107     cout << "==Hapus Simpul Depan == " << endl << endl;
108     cout << "\nSetelah Hapus Simpul Depan \n";
109     Hapus_Depan(DL);
110     Cetak(DL);
111     break;
112 case 6:
113     //Hapus simpul Belakang
114     cout << "\n\n == Percobaan 6 == " << endl;
115     cout << "== Hapus Simpul Belakang == " << endl << endl;
116     cout << "\nSetelah Hapus Simpul Belakang " << endl;
117     Hapus_Belakang(DL);
118     Cetak(DL);
119     break;
120 case 7:
121     //Hapus simpul tengah
122     cout << "\n\n == Percobaan 7 == " << endl;
123     cout << "== Hapus Simpul Tengah == " << endl << endl;
124     cout << "\n\nMasukan Huruf Tengah Yang Akan Dihapus :";
125     cin >> huruf;
126     Hapus_Tengah(DL, huruf);
127     Cetak(DL);
128     break;
129 case 8:
130     //Pencarian Data
131     cout << "\n\n== Pencarian Data == " << endl << endl;
132     cout << "Masukan Huruf Yang Ingin Dicari :"; cin >> huruf;
133     pencarianHuruf(DL, huruf);
134     break;
135 case 9:
136     //Sorting Data
137     cout << "\n\n==Sorting Data == " << endl;
138     sortingData(DL);
139     Cetak(DL);
140     break;
141 case 10:

```

```

141     case 10:
142         Cetak(DL);
143         break;
144     case 11:
145         cout << "Program berhenti" << endl;
146         exit;
147     default:
148         cout << "Masukan menu yang benar! " << endl;
149         system("pause");
150         system("cls");
151         goto menu;
152     }
153     system("pause");
154     system("cls");
155     goto menu;
156 }
157
158 //FUNGSI SISIP SIMPUL DEPAN
159 void Sisip_Depan(simpul& DL, char elemen)
160 {
161     simpul baru;
162     baru = (simpul)malloc(sizeof(simpul));
163     baru->Isi = elemen;
164     baru->kanan = NULL;
165     baru->kiri = NULL;
166     if (DL == NULL) {
167         DL = baru;
168     }
169     else
170     {
171         baru->kanan = DL;
172         DL->kiri = baru;
173         DL = baru;
174     }
175 }
176
177 //FUNGSI SISIP SIMPUL DI BELAKANG
178 void Sisip_Belakang(simpul& DL, char elemen)
179 {
180     simpul bantu, baru;
181     baru = (simpul)malloc(sizeof(simpul));
182     baru->Isi = elemen;
183     baru->kanan = NULL;
184     baru->kiri = NULL;
185     if (DL == NULL)
186         DL = baru;
187     else
188     {
189         bantu = DL;
190         while (bantu->kanan != NULL)
191             bantu = bantu->kanan;
192         bantu->kanan = baru;
193         baru->kiri = bantu;
194     }
195 }

```

```

196
197 //FUNGSI SISIP SIMPUL SETELAH SIMPUL TERTENTU
198 void Sisip_Tengah1(simpul& DL, char elemen1, char elemen2)
199 {
200     simpul bantu, baru;
201     baru = (simpul)malloc(sizeof(simpul));
202     baru->Isi = elemen1;
203     baru->kanan = NULL;
204     baru->kiri = NULL;
205     if (DL == NULL)
206         cout << "List Kosong....." << endl;
207     else
208     {
209         bantu = DL;
210         while (bantu->Isi != elemen2) bantu = bantu->kanan;
211         baru->kanan = bantu->kanan;
212         baru->kiri = baru;
213         bantu->kiri = bantu;
214         bantu->kanan->kiri = baru;
215         bantu->kanan = baru;
216     }
217 }
218
219 //FUNGSI SISIP SIMPUL SEBELUM SIMPUL TERTENTU
220 void Sisip_Tengah2(simpul& DL, char elemen1, char elemen2)
221 {
222     simpul bantu, baru;
223     baru = (simpul)malloc(sizeof(simpul));
224     baru->Isi = elemen1;
225     baru->kanan = NULL;
226     baru->kiri = NULL;
227     if (DL == NULL)
228         cout << "List Kosong ....." << endl;
229     else
230     {
231         bantu = DL;
232         while (bantu->kanan->Isi != elemen2) bantu = bantu->kanan;
233         baru->kanan = bantu->kanan;
234         baru->kiri = bantu;
235         bantu->kanan->kiri = baru;
236         bantu->kanan = baru;
237     }
238 }

```

```

240 //Fungsi mencetak isi linked list
241 void Cetak(simpul DL)
242 {
243     simpul bantu;
244     if (DL == NULL)
245         cout << "Linked list kosong ...." << endl;
246     else
247     {
248         bantu = DL;
249         cout << "Isi Linked List : ";
250         while (bantu->kanan != NULL)
251         {
252             cout << bantu->Isi << " <--> ";
253             bantu = bantu->kanan;
254         }
255         cout << bantu->Isi;
256     }
257 }
258
259 //Fungsi Hapus Simpul Depan
260 void Hapus_Depan(simpul& DL)
261 {
262     simpul Hapus;
263     if (DL == NULL)
264         cout << "Linked List kosong.....";
265     else
266     {
267         Hapus = DL;
268         DL = DL->kanan;
269         DL->kiri = NULL;
270         Hapus->kanan = NULL;
271         free(Hapus);
272     }
273 }
274
275 //Fungsi Hapus SIMPUL BELAKANG
276 void Hapus_Belakang(simpul& DL)
277 {
278     simpul bantu, hapus;
279     if (DL == NULL)
280         cout << "Linked List Kosong.....";
281     else
282     {
283         bantu = DL;
284         while (bantu->kanan->kanan != NULL)
285             bantu = bantu->kanan;
286         hapus = bantu->kanan;
287         bantu->kanan = NULL;
288         hapus->kiri = NULL;
289         free(hapus);
290     }
291 }
292

```

```

293 //FUNGSI HAPUS SIMPUL DI TENGAH
294 void Hapus_Tengah(simpul& DL, char elemen)
295 {
296     simpul bantu, hapus;
297     if (DL == NULL)
298         cout << "Linked List Kosong.....";
299     else
300     {
301         bantu = DL;
302         while (bantu->kanan->Isi != elemen)
303             bantu = bantu->kanan;
304         hapus = bantu->kanan;
305         bantu->kanan->kanan->kiri = bantu;
306         bantu->kanan = bantu->kanan->kanan;
307         hapus->kanan = NULL;
308         hapus->kiri = NULL;
309         free(hapus);
310     }
311 }

```

b) Hasil Running

1. Tampilan Menu

```

== OPERASI PADA DOUBLY LINKED LIST ==

Menu:
1. Menyisipkan Simpul Di Depan
2. Menyisipkan Simpul Di Belakang
3. Penyisipan Simpul Di Tengah Sebelum Node Tertentu
4. Penyisipan Simpul Di Tengah Setelah Node Tertentu
5. Hapus Simpul Depan
6. Hapus Simpul Belakang
7. Hapus Simpul Tengah
8. Pencarian Data
9. Pengurutan Data
10. Cetak Data
11. Exit
Masukan Pilihan Menu:

```

2. Menyisipkan Simpul Di Depan

```

Masukan Pilihan Menu:1
== Percobaan 1 ==
== Penyisipan Simpul Di Depan ==
Masukan jumlah data : 4

Masukan huruf :A
Masukan huruf :B
Masukan huruf :K
Masukan huruf :Z
Isi Linked List : Z <--> K <--> B <--> APress any key to continue . . .

```

### 3. Menyisipkan Simpul Di Belakang

```
== Percobaan 2 ==  
== Penyisipan Simpul Di Belakang ==  
Masukan jumlah data : 1  
  
Masukan huruf :P  
Isi Linked List : Z <--> K <--> B <--> A <--> PPress any key to continue . . . █
```

### 4. Menyisipkan Simpul Di Tengah Sebelum Node Tertentu

```
Masukan Pilihan Menu:3  
  
== Percobaan 3 ==  
== Penyisipan Simpul Di Tengah Sebelum Node Tertentu ==  
Masukan huruf yang disisipkan :Q  
Disisipkan Setelah Huruf : B  
Q Disisipkan Setelah B  
Isi Linked List : Z <--> K <--> B <--> Q <--> A <--> PPress any key to continue . . . █
```

### 5. Menyisipkan Simpul Tengah Setelah Node Tertentu

```
Masukan Pilihan Menu:4  
  
== Percobaan 4 ==  
Penyisipan Simpul Di Tengah Setelah Node Tertentu ==  
Masukan Huruf :M  
Disisip sebelum Huruf :B  
M Disisip Sebelum B  
Isi Linked List : Z <--> K <--> M <--> B <--> Q <--> A <--> PPress any key to continue . . . █
```

### 6. Hapus Simpul Depan

```
Masukan Pilihan Menu:5  
  
== Percobaan 5 ==  
==Hapus Simpul Depan ==  
  
Setelah Hapus Simpul Depan  
Isi Linked List : K <--> M <--> B <--> Q <--> A <--> PPress any key to continue . . . █
```

### 7. Hapus Simpul Belakang

```
Masukan Pilihan Menu:6  
  
== Percobaan 6 ==  
== Hapus Simpul Belakang ==  
  
Setelah Hapus Simpul Belakang  
Isi Linked List : K <--> M <--> B <--> Q <--> APress any key to continue . . . █
```



## 8. Hapus Simpul Tengah

```
Masukan Pilihan Menu: /

== Percobaan 7 ==
== Hapus Simpul Tengah ==

Masukan Huruf Tengah Yang Akan Dihapus :M
Isi Linked List : K <--> B <--> Q <--> APress any key to continue . . .
```

### c) Penjelasan

Penjelasan dari langkahnya:

- 1) Menyisipkan Simpul depan  
Menyisipkan simpul depan yaitu dengan diberikannya Node baru. Sebelum itu, akan dilakukan pengkondisian, dimana node DL apabila NULL artinya Node kosong.  
Selain itu maka akan dilakukan proses penambahan node di depan:  
-baru->kanan = DL;  
  
-DL->kiri = baru;
- 2) Menyisipkan simpul di Belakang  
Menyusipkan simpul di belakang yaitu dengan di berikannya Node baru.
  - Setelah dilakukannya pengkondisian, lakukan pengisian data ke node baru.
  - Setelah itu node baru yang awalnya berada di DL akan di geser ke kanan terus sampai Node terakhir.
  - lakukan penambahan data disitu, dimana simpul kanan yang NULL akan di sambungkan ke node baru simpul kiri.
- 3) Menyisipkan Simpul Di Tengah Sebelum Node Tertentu
  - Membuat node baru yang berisi data yang ingin di inputkan
  - Lakukan pengkondisian data node yang berisi data yang ingin dicari dengan menggeser node bantu
  - Apabila node bantu sudah menemukan data yang sama dengan yang dicari, letakan penambahan data di node sebelumnya
- 4) Menyisipkan Simpul Di Tengah Setelah Node Tertentu
  - Membuat node baru yang berisi data yang ingin di inputkan
  - Lakukan pengkondisian data node yang berisi data yang ingin dicari dengan menggeser node bantu
  - Apabila node bantu sudah menemukan data yang sama dengan yang dicari, letakan penambahan data di node sebelumnya
- 5) Hapus Simpul depan
  - Membuat node bantu
  - Node bantu di arahkan ke DL
  - Bantu->kanan->kiri di buat NULL

- Hapus node bantu.

#### 6) Hapus Simpul Belakang

- Buat node bantu yang mengarah ke DL dahulu
- Geser node bantu sampai akhir di node data
- Apabila sudah di akhir node(bantu->kanan=NULL), maka Node bantu->kiri->kanan dibuat NULL
- Hapus Node bantu.

#### a. Percobaan 2

Membuat Pencarian data

##### a) Tampilan Coding

```

313 void pencarianHuruf(simpul& DL, char elemen) {
314     simpul bantu;
315     int posisi = 1;
316     if (DL == NULL)
317         cout << "Linked List Kosong.....";
318     else
319     {
320         bantu = DL;
321         while (bantu->kanan != NULL) {
322             if (bantu->Isi == elemen) {
323                 cout << "Data Ada Di Node Ke:" << posisi;
324                 break;
325             }
326             posisi++;
327             bantu = bantu->kanan;
328             if (bantu->kanan == NULL ) {
329                 cout << "Data Yang Dicari Tidak Ada!" << endl;
330             }
331         }
332     }
333 }

```

##### b) Hasil Running

```

Masukan Pilihan Menu: 7

== Percobaan 7 ==
== Hapus Simpul Tengah ==

Masukan Huruf Tengah Yang Akan Dihapus :M
Isi Linked List : K <--> B <--> Q <--> A Press any key to continue . . .

```

##### c) Penjelasan

Untuk melakukan pencarian data, diberikan node bantu yang mengarah ke DL. Bila kondisi Isi dari node bantu ini tidak sama, maka posisi DL akan

berpindah ke kanan mengikuti node nya sampai simpul kanan = NULL ,  
yang berarti node bantu sudah berada di akhir node data.

b. Percobaan 3

Membuat Fungsi Pengurutan Pada Data

a) Tampilan Coding

```
335 int jumlahNode(simpul &DL) {  
336     simpul bantu;  
337     int count=1;  
338     if (DL == NULL) {  
339         return count;  
340     }  
341     else {  
342         bantu = DL;  
343         while (bantu->kanan != NULL) {  
344             count++;  
345             bantu = bantu->kanan;  
346         }  
347         return count;  
348     }  
349 }  
350  
351 void sortingData(simpul& DL) {  
352     simpul bantu;  
353     int ukuran = jumlahNode(DL);  
354     int hitung = 1;  
355     if (DL == NULL) {  
356         cout << "Linked List Kosong.....";  
357     }  
358     else {  
359         //langkah sorting  
360         for (int i = 1; i <= ukuran; i++) {  
361             bantu = DL;  
362             for (int j = 1; j < ukuran; j++) {  
363                 if (bantu->Isi > bantu->kanan->Isi) {  
364                     char temp = bantu->Isi;  
365                     bantu->Isi = bantu->kanan->Isi;  
366                     bantu->kanan->Isi = temp;  
367                 }  
368                 bantu = bantu->kanan;  
369             }  
370         }  
371     }  
372 }  
373 }
```

b) Hasil Running

```
Masukan Pilihan Menu:9  
==Sorting Data ==  
Isi Linked List : A <--> B <--> K <--> QPress any key to continue . . .
```

c) Penjelasan

Untuk membuat Sorting data, di butuhkan fungsi lagi yaitu jumlah Node, Dimana ini akan menghitung jumlah node yang ada.

Setelah itu di lakukan sorting data sesuai dengan node yang ada.

Metode yang dilakukan diatas adalah sorting menggunakan bubble sort.

### C. Kesimpulan

Setelah melakukan percobaan pada Latihan 1 dst saya dapat memahami bahwa Double Linked List Non Circular atau sering disingkat DLLNC adalah sebuah Linked List yang terdiri dari dua arah pointer, dengan node yang saling terhubung, namun kedua pointernya menunjuk ke NULL. Setiap node pada linked list mempunyai field yang berisi data dan pointer yang saling berhubungan dengan node yang lainnya.

Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya

Untuk pembentukan node baru , mulanya pointer next dan prev akan menunjuk ke nilai NULL

Selanjutnya pointer prev akan menunjuk ke node sebelumnya , dan pointer next akan menunjuk ke node selanjutnya pada list.