

**LAPORAN PRAKTIKUM
ALGORITMA & STRUKTUR DATA
MODUL 4**



DOUBLE LINK LIST

Oleh:

Rika Fauliana Rahmi NIM. 2410817120017

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM ALGORITMA & STRUKTUR DATA
MODUL 4

Laporan Praktikum Algoritma & Struktur Data Modul 4: Double Link List ini disusun sebagai syarat lulus mata kuliah Praktikum Algoritma & Struktur Data. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rika Fauliana Rahmi
NIM : 2410817120017

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Fauzan Ahsani
NIM. 2310817310009

Muti'a Maulida, S.Kom., M.TI.
NIP. 198810272019032013

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	4
DAFTAR GAMBAR	5
SOAL 1	7
I. Source Code	13
II. Output Program.....	26
III. Pembahasan.....	32
SOAL 2	34
I. Pembahasan.....	34
SOAL 3	37
I. Pembahasan.....	37
SOAL 4	39
I. Pembahasan.....	39
TAUTAN GIT.....	42

DAFTAR TABEL

Tabel 1 Source Code Soal 1	13
Tabel 2 Contoh Source Code Soal 2	35
Tabel 3 Contoh Source Code Soal 2	35
Tabel 4 Contoh Source Code Soal 2	35
Tabel 5 Contoh Source Code Soal 2	36
Tabel 6 Contoh Source Code Soal 3	37
Tabel 7 Contoh Source Code Soal 3	37
Tabel 8 Contoh Source Code Soal 3	38
Tabel 9 Contoh Source Code Soal 3	38
Tabel 10 Contoh Source Code Soal 3	38

DAFTAR GAMBAR

Gambar 1 screenshot tampilan pertama program di running	26
Gambar 2 screenshot tampilan program DLLNC (Head)	26
Gambar 3 screenshot tampilan DLLNC (Head) untuk pilihan 1	27
Gambar 4 screenshot tampilan DLLNC (Head) saat menampilkan data untuk pilihan 1	27
Gambar 5 screenshot tampilan DLLNC (Head) saat memilih pilihan 2	27
Gambar 6 screenshot tampilan DLLNC (Head) saat menampilkan data untuk pilihan 2	27
Gambar 7 screenshot tampilan DLLNC (Head) saat memilih pilihan 4	28
Gambar 8 screenshot tampilan DLLNC (Head) saat memilih pilihan 5	28
Gambar 9 screenshot tampilan DLLNC (Head) saat menampilkan data setelah dihapus depan dan belakang	28
Gambar 10 screenshot tampilan DLLNC (Head) saat memilih pilihan 6	28
Gambar 11 screenshot tampilan DLLNC (Head) saat menampilkan data setelah di reset	29
Gambar 12 screenshot tampilan awal DLLNC (Head dan Tail)	29
Gambar 13 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 1	29
Gambar 14 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah memilih pilihan 1	30
Gambar 15 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 2	30
Gambar 16 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah memilih pilihan 2	30
Gambar 17 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 4	31
Gambar 18 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 5	31
Gambar 19 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah dihapus depan dan belakang	31
Gambar 20 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 6	32

Gambar 21 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah di reset	32
Gambar 22 screenshot tampilan saat memasukkan data	39
Gambar 23 screenshot tampilan saat memasukkan data	39
Gambar 24 screenshot tampilan saat memasukkan data	40
Gambar 25 screenshot tampilan saat menampilkan data	40

SOAL 1

1. Lengkapi coding pada function tambahDepanH() agar bisa berjalan dengan lancar. running, simpan program, dan screenshoot hasil running !

```
1  #include <conio.h>
2  #include <iostream>
3  #include <stdlib.h>
4
5  using namespace std;
6
7  typedef struct TNode {
8      string data;
9      TNode *next;
10     TNode *prev;
11 };
12
13 TNode *head, *tail;
14
15 int pil, menu;
16 char pilihan[1];
17 string dataBaru;
18
19 void initH();
20 void initHT();
21 int isEmptyH();
22 int isEmptyHT();
23
24 void tambahDepanH();
25 void tambahDepanHT();
26 void tambahBelakangH();
27 void tambahBelakangHT();
28 void hapusDepanH();
29 void hapusDepanHT();
30 void hapusBelakangH();
31 void hapusBelakangHT();
32 void tampilkanH();
33 void tampilkanHT();
34 void clearH();
35 void clearHT();
36
37 int main()
38 {
39     menu:
40     cout<<"Double Linked List Non Circular (DLLNC)"<<endl;
41     cout<<"===== "<<endl;
42     cout<<"Silahkan pilih program DLLNC yang ingin dijalankan!"<<endl;
43     cout<<"1. DLLNC dengan Head"<<endl;
44     cout<<"2. DLLNC dengan Head dan Tail"<<endl;
45     cout<<"3. Quit"<<endl;
46     cout<<"Pilihan : ";
47     cin>>menu;
48     system("cls");
49
50     if(menu==1){
51         do {
52             cout<<"Double Linked List Non Circular (DLLNC) (Head)"<<endl;
53             cout<<"===== "<<endl;
54             cout<<"1. Tambah Depan"<<endl;
55             cout<<"2. Tambah Belakang"<<endl;
56             cout<<"3. Tampilkan Data"<<endl;
57             cout<<"4. Hapus Depan"<<endl;
58             cout<<"5. Hapus Belakang"<<endl;
59             cout<<"6. Reset"<<endl;
60             cout<<"7. Kembali ke Menu"<<endl;
61             cout<<"Pilihan : ";
62             cin>>pilihan;
63             pil=atoi(pilihan);
```

```

63
64     switch(pil) {
65     case 1:
66         tambahDepanH();
67         break;
68     case 2:
69         tambahBelakangH();
70         break;
71     case 3:
72         tampilkanH();
73         break;
74     case 4:
75         hapusDepanH();
76         break;
77     case 5:
78         hapusBelakangH();
79         break;
80     case 6:
81         clearH();
82         break;
83     default:
84         system("cls");
85         goto menu;
86     }
87
88     cout<<"\npress any key to continue"<<endl;
89     getch();
90     system("cls");
91
92     } while (pil<7);
93 } else if(menu==2){
94     do {

```

```

95         cout<<"Double Linked List Non Circular (DLLNC) (Head dan Tail)"<<endl;
96         cout<<"-----"<<endl;
97         cout<<"1. Tambah Depan"<<endl;
98         cout<<"2. Tambah Belakang"<<endl;
99         cout<<"3. Tampilkan Data"<<endl;
100        cout<<"4. Hapus Depan"<<endl;
101        cout<<"5. Hapus Belakang"<<endl;
102        cout<<"6. Reset"<<endl;
103        cout<<"7. Kembali ke Menu"<<endl;
104        cout<<"Pilihan : ";
105        cin>>pilihan;
106        pil=atoi(pilihan);
107
108        switch(pil) {
109        case 1:
110            tambahDepanHT();
111            break;
112        case 2:
113            tambahBelakangHT();
114            break;
115        case 3:
116            tampilkanHT();
117            break;
118        case 4:
119            hapusDepanHT();
120            break;
121        case 5:
122            hapusBelakangHT();
123            break;
124        case 6:
125            clearHT();
126            break;
127        default:
128            system("cls");
129            goto menu;
130        }
131
132        cout<<"\npress any key to continue"<<endl;
133        getch();
134        system("cls");
135
136    } while (pil<7);
137 } else {
138     cout<<"\nTERIMA KASIH"<<endl;
139     cout<<"Program was made by Nama (NIM)."<<endl;
140 }
141 }
142

```



```

143 void initH(){
144     head = NULL;
145 }
146
147 void initHT(){
148     head = NULL;
149     tail = NULL;
150 }
151
152 int isEmptyH(){
153     if(head == NULL) return 1;
154     else return 0;
155 }
156
157 int isEmptyHT(){
158     if(tail == NULL) return 1;
159     else return 0;
160 }
161
162 void tambahDepanH() {
163     cout<<"Masukkan data : ";
164
165
166
167
168
169
170
171
172
173
174
175
176
177     cout << "Data \'"<<dataBaru<<"\' berhasil dimasukkan di bagian depan.";
178 }
179
180 void tambahDepanHT() {
181     cout<<"Masukkan data : ";
182     cin>>dataBaru;
183     TNode *baru;
184     baru = new TNode;
185     baru->data = dataBaru;
186     baru->next = NULL;
187     baru->prev = NULL;
188     if(isEmptyHT() == 1) {
189         head = baru;
190         tail = baru;

```

```

191     } else {
192         baru->next = head;
193         head->prev = baru;
194         head = baru;
195     }
196     cout << "Data \""<<dataBaru<<"\" berhasil dimasukkan di bagian depan.";
197 }
198
199 void tambahBelakangH() {
200     cout<<"Masukkan data : ";
201     cin>>dataBaru;
202     TNode *baru, *bantu;
203     baru = new TNode;
204     baru->data = dataBaru;
205     baru->next = NULL;
206     baru->prev = NULL;
207     if(isEmptyH() == 1) {
208         head = baru;
209     } else {
210         bantu = head;
211         while(bantu->next != NULL){
212             bantu = bantu->next;
213         }
214         bantu->next = baru;
215         baru->prev = bantu;
216     }
217     cout << "Data \""<<dataBaru<<"\" berhasil dimasukkan di bagian belakang.";
218 }
219

```

```

220 void tambahBelakangHT() {
221     cout<<"Masukkan data : ";
222     cin>>dataBaru;
223     TNode *baru;
224     baru = new TNode;
225     baru->data = dataBaru;
226     baru->next = NULL;
227     baru->prev = NULL;
228     if(isEmptyHT() == 1) {
229         head = baru;
230         tail = baru;
231     } else {
232         tail->next = baru;
233         baru->prev = tail;
234         tail = baru;
235     }
236     cout << "Data \""<<dataBaru<<"\" berhasil dimasukkan di bagian belakang.";
237 }

```

```

238
239 void tampilkanH() {
240     TNode *bantu;
241     bantu = head;
242     if(isEmptyH() == 0) {
243         while(bantu != NULL) {
244             cout<<bantu->data<<' ';
245             bantu = bantu->next;
246         }
247         cout<<endl;
248     } else cout<<"Tidak terdapat data pada Linked List";
249 }
250
251 void tampilkanHT() {
252     TNode *bantu;
253     bantu = head;
254     if(isEmptyHT() == 0) {
255         while(bantu != tail->next) {
256             cout<<bantu->data<<' ';
257             bantu = bantu->next;
258         }
259         cout<<endl;
260     } else cout<<"Tidak terdapat data pada Linked List";
261 }
262
263 void hapusDepanH() {
264     TNode *hapus;
265     string data;
266     if(isEmptyH() == 0) {
267         hapus = head;
268         data = hapus->data;
269         if(head->next != NULL) {
270             head = head->next;
271             head->prev = NULL;
272         } else {
273             initH();
274         }
275         delete hapus;
276         cout<<"Data \"<<data<<\" yang berada di depan telah berhasil dihapus.";
277     } else cout<<"Tidak terdapat data pada Linked List";
278 }
279
280 void hapusDepanHT() {
281     TNode *hapus;
282     string data;
283     if(isEmptyHT() == 0) {
284         hapus = head;
285         data = hapus->data;

```

```

286         if(head->next != NULL) {
287             head = head->next;
288             head->prev = NULL;
289         } else {
290             initHT();
291         }
292         delete hapus;
293         cout<<"Data \\"<<data<<"\" yang berada di depan telah berhasil dihapus.";
294     } else cout<<"Tidak terdapat data pada Linked List";
295 }
296
297 void hapusBelakangH() {
298     TNode *hapus;
299     string data;
300     if(isEmptyH() == 0) {
301         hapus = head;
302         while(hapus->next != NULL){
303             hapus = hapus->next;
304         }
305         data = hapus->data;
306         if(head->next != NULL) {
307             hapus->prev->next = NULL;
308         } else {
309             initH();
310         }
311         delete hapus;
312         cout<<"Data \\"<<data<<"\" yang berada di belakang telah berhasil dihapus.";
313     } else cout<<"Tidak terdapat data pada Linked List";
314 }
315
316 void hapusBelakangHT() {
317     TNode *hapus;
318     string data;
319     if(isEmptyHT() == 0) {
320         hapus = tail;
321         data = hapus->data;
322         if(head->next != NULL) {
323             tail = tail->prev;
324             tail->next = NULL;
325         } else {
326             initHT();
327         }
328         delete hapus;
329         cout<<"Data \\"<<data<<"\" yang berada di belakang telah berhasil dihapus.";
330     } else cout<<"Tidak terdapat data pada Linked List";
331 }
332
333 void clearH() {

```

```

334     TNode *bantu, *hapus;
335     bantu = head;
336     while(bantu != NULL) {
337         hapus = bantu;
338         bantu = bantu->next;
339         delete hapus;
340     }
341     inith();
342     cout<<"Seluruh data pada Linked List telah dibersihkan.";
343 }
344
345 void clearHT() {
346     TNode *bantu, *hapus;
347     bantu = head;
348     while(bantu != NULL) {
349         hapus = bantu;
350         bantu = bantu->next;
351         delete hapus;
352     }
353     inithHT();
354     cout<<"Seluruh data pada Linked List telah dibersihkan.";
355 }

```

I. Source Code

Tabel 1 Source Code Soal 1

1	#include <conio.h>
2	#include <iostream>
3	#include <stdlib.h>
4	
5	using namespace std;
6	
7	typedef struct TNode {
8	string data;
9	TNode *next;
10	TNode *prev;
11	};
12	
13	TNode *head, *tail;
14	
15	int pil, menu;
16	char pilihan[1];

```

17  string dataBaru;
18
19  void initH();
20  void initHT();
21  int isEmptyH();
22  int isEmptyHT();
23
24  void tambahDepanH();
25  void tambahDepanHT();
26  void tambahBelakangH();
27  void tambahBelakangHT();
28  void hapusDepanH();
29  void hapusDepanHT();
30  void hapusBelakangH();
31  void hapusBelakangHT();
32  void tampilkanH();
33  void tampilkanHT();
34  void clearH();
35  void clearHT();
36
37  int main()
38  {
39      menu:
40      cout<<"Double    Linked    List    Non    Circular
(DLLNC) "<<endl;
41
42      cout<<"===== "<<endl;
43
44      cout<<"Silahkan pilih program DLLNC yang ingin
dijalankan!"<<endl;

```

```

43     cout<<"1. DLLNC dengan Head"<<endl;
44     cout<<"2. DLLNC dengan Head dan Tail"<<endl;
45     cout<<"3. Quit"<<endl;
46     cout<<"Pilihan : ";
47     cin>>menu;
48     system("cls");
49     if(menu==1){
50         do {
51             cout<<"Double Linked List Non Circular
(DLLNC) (Head) "<<endl;
52
53             cout<<"=====
===== "<<endl;
54             cout<<"1. Tambah Depan"<<endl;
55             cout<<"2. Tambah Belakang"<<endl;
56             cout<<"3. Tampilkan Data"<<endl;
57             cout<<"4. Hapus Depan"<<endl;
58             cout<<"5. Hapus Belakang"<<endl;
59             cout<<"6. Reset"<<endl;
60             cout<<"7. Kembali ke Menu"<<endl;
61             cout<<"Pilihan : ";
62             cin>>pilihan;
63             pil=atoi(pilihan);
64
65             switch(pil) {
66                 case 1:
67                     tambahDepanH();
68                     break;
69                 case 2:
69                     tambahBelakangH();

```

```

70         break;
71     case 3:
72         tampilkanH();
73         break;
74     case 4:
75         hapusDepanH();
76         break;
77     case 5:
78         hapusBelakangH();
79         break;
80     case 6:
81         clearH();
82         break;
83     default:
84         system("cls");
85         goto menu;
86     }
87
88     cout<<"<press any key to
continue>"<<endl;
89     getch();
90     system("cls");
91
92     } while (pil!=7);
93     } else if(menu==2){
94         do {
95             cout<<"Double Linked List Non Circular
(DLLNC) (Head dan Tail)"<<endl;

```



```

96      cout<<"=====
===="<<endl;
97          cout<<"1. Tambah Depan"<<endl;
98          cout<<"2. Tambah Belakang"<<endl;
99          cout<<"3. Tampilkan Data"<<endl;
100         cout<<"4. Hapus Depan"<<endl;
101         cout<<"5. Hapus Belakang"<<endl;
102         cout<<"6. Reset"<<endl;
103         cout<<"7. Kembali ke Menu"<<endl;
104         cout<<"Pilihan : ";
105         cin>>pilihan;
106         pil=atoi(pilihan);
107
108         switch(pil) {
109             case 1:
110                 tambahDepanHT();
111                 break;
112             case 2:
113                 tambahBelakangHT();
114                 break;
115             case 3:
116                 tampilkanHT();
117                 break;
118             case 4:
119                 hapusDepanHT();
120                 break;
121             case 5:
122                 hapusBelakangHT();
123                 break;

```

```

124         case 6:
125             clearHT();
126             break;
127         default:
128             system("cls");
129             goto menu;
130     }
131
132     cout<<"\npress any key to
continue"<<endl;
133     getch();
134     system("cls");
135
136     } while (pil<7);
137     } else {
138         cout<<"\nTERIMA KASIH"<<endl;
139         cout<<"Program was made by Rika Fauliana
Rahmi (2410817120017)."<<endl;
140     }
141 }
142
143 void initH() {
144     head = NULL;
145 }
146
147 void initHT() {
148     head = NULL;
149     tail = NULL;
150 }
151

```

```

152 int isEmptyH() {
153     if(head == NULL) return 1;
154     else return 0;
155 }
156
157 int isEmptyHT() {
158     if(tail == NULL) return 1;
159     else return 0;
160 }
161
162 void tambahDepanH() {
163     cout<<"Masukkan data : ";
164     cin >> dataBaru;
165     TNode *baru;
166     baru = new TNode;
167     baru->data = dataBaru;
168     baru->next = NULL;
169     baru->prev = NULL;
170     if (isEmptyH() == 1) {
171         head = baru;
172     } else {
173         baru->next = head;
174         head->prev = baru;
175         head = baru;
176     }
177     cout << "Data \""<<dataBaru<<"\" berhasil
dimasukkan di bagian depan.";
178 }
179
180 void tambahDepanHT() {

```

```

181     cout<<"Masukkan data : ";
182     cin>>dataBaru;
183     TNode *baru;
184     baru = new TNode;
185     baru->data = dataBaru;
186     baru->next = NULL;
187     baru->prev = NULL;
188     if(isEmptyHT() == 1) {
189         head = baru;
190         tail = baru;
191     } else {
192         baru->next = head;
193         head->prev = baru;
194         head = baru;
195     }
196     cout << "Data \""<<dataBaru<<"\" berhasil
dimasukkan di bagian depan.";
197 }
198
199 void tambahBelakangH() {
200     cout<<"Masukkan data : ";
201     cin>>dataBaru;
202     TNode *baru, *bantu;
203     baru = new TNode;
204     baru->data = dataBaru;
205     baru->next = NULL;
206     baru->prev = NULL;
207     if(isEmptyH() == 1) {
208         head = baru;
209     } else {

```

```

210         bantu = head;
211         while(bantu->next != NULL){
212             bantu = bantu->next;
213         }
214         bantu->next = baru;
215         baru->prev = bantu;
216     }
217     cout << "Data \\"<<dataBaru<<\"\\\" berhasil
dimasukkan di bagian belakang.";
218 }
219
220 void tambahBelakangHT() {
221     cout<<"Masukkan data : ";
222     cin>>dataBaru;
223     TNode *baru;
224     baru = new TNode;
225     baru->data = dataBaru;
226     baru->next = NULL;
227     baru->prev = NULL;
228     if(isEmptyHT() == 1) {
229         head = baru;
230         tail = baru;
231     } else {
232         tail->next = baru;
233         baru->prev = tail;
234         tail = baru;
235     }
236     cout << "Data \\"<<dataBaru<<\"\\\" berhasil
dimasukkan di bagian belakang.";
237 }

```

```

238
239 void tampilkanH() {
240     TNode *bantu;
241     bantu = head;
242     if(isEmptyH() == 0) {
243         while(bantu != NULL) {
244             cout<<bantu->data<<' ';
245             bantu = bantu->next;
246         }
247         cout<<endl;
248     } else cout<<"Tidak terdapat data pada Linked
List";
249 }
250
251 void tampilkanHT() {
252     TNode *bantu;
253     bantu = head;
254     if(isEmptyHT() == 0) {
255         while(bantu != tail->next) {
256             cout<<bantu->data<<' ';
257             bantu = bantu->next;
258         }
259         cout<<endl;
260     } else cout<<"Tidak terdapat data pada Linked
List";
261 }
262
263 void hapusDepanH() {
264     TNode *hapus;
265     string data;

```

```

266     if(isEmptyH() == 0) {
267         hapus = head;
268         data = hapus->data;
269         if(head->next != NULL) {
270             head = head->next;
271             head->prev = NULL;
272         } else {
273             initH();
274         }
275         delete hapus;
276         cout<<"Data \""<<data<<"\" yang berada di
depan telah berhasil dihapus.";
277     } else cout<<"Tidak terdapat data pada Linked
List";
278 }
279
280 void hapusDepanHT() {
281     TNode *hapus;
282     string data;
283     if(isEmptyHT() == 0) {
284         hapus = head;
285         data = hapus->data;
286         if(head->next != NULL) {
287             head = head->next;
288             head->prev = NULL;
289         } else {
290             initHT();
291         }
292         delete hapus;

```

```

293         cout<<"Data \""<<data<<"\" yang berada di
        depan telah berhasil dihapus.";
294     } else cout<<"Tidak terdapat data pada Linked
        List";
295 }
296
297 void hapusBelakangH() {
298     TNode *hapus;
299     string data;
300     if(isEmptyH() == 0) {
301         hapus = head;
302         while(hapus->next != NULL){
303             hapus = hapus->next;
304         }
305         data = hapus->data;
306         if(head->next != NULL) {
307             hapus->prev->next = NULL;
308         } else {
309             initH();
310         }
311         delete hapus;
312         cout<<"Data \""<<data<<"\" yang berada di
        belakang telah berhasil dihapus.";
313     } else cout<<"Tidak terdapat data pada Linked
        List";
314 }
315
316 void hapusBelakangHT() {
317     TNode *hapus;
318     string data;

```



```

319     if(isEmptyHT() == 0) {
320         hapus = tail;
321         data = hapus->data;
322         if(head->next != NULL) {
323             tail = tail->prev;
324             tail->next = NULL;
325         } else {
326             initHT();
327         }
328         delete hapus;
329         cout<<"Data \""<<data<<"\" yang berada di
belakang telah berhasil dihapus.";
330     } else cout<<"Tidak terdapat data pada Linked
List";
331 }
332
333 void clearH() {
334     TNode *bantu, *hapus;
335     bantu = head;
336     while(bantu != NULL) {
337         hapus = bantu;
338         bantu = bantu->next;
339         delete hapus;
340     }
341     initH();
342     cout << "Seluruh data pada Linked List telah
dibersihkan.";
343 }
344
345 void clearHT() {

```

```

346         TNode *bantu, *hapus;
347         bantu = head;
348         while(bantu != NULL) {
349             hapus = bantu;
350             bantu = bantu->next;
351             delete hapus;
352         }
353         initHT();
354         cout << "Seluruh data pada Linked List telah
dibersihkan.";
355     }

```

II. Output Program

```

Double Linked List Non Circular (DLLNC)
=====
Silahkan pilih program DLLNC yang ingin dijalankan!
1. DLLNC dengan Head
2. DLLNC dengan Head dan Tail
3. Quit
Pilihan :

```

Gambar 1 screenshot tampilan pertama program di running

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : █

```

Gambar 2 screenshot tampilan program DLLNC (Head)

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 1
Masukkan data : 1
Data "1" berhasil dimasukkan di bagian depan.<press any key to continue>
```

Gambar 3 screenshot tampilan DLLNC (Head) untuk pilihan 1

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
3 2 1
<press any key to continue>
```

Gambar 4 screenshot tampilan DLLNC (Head) saat menampilkan data untuk pilihan 1

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 2
Masukkan data : 6
Data "6" berhasil dimasukkan di bagian belakang.<press any key to continue>
```

Gambar 5 screenshot tampilan DLLNC (Head) saat memilih pilihan 2

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
3 2 1 6 7 8
<press any key to continue>
```

Gambar 6 screenshot tampilan DLLNC (Head) saat menampilkan data untuk pilihan 2

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 4
Data "3" yang berada di depan telah berhasil dihapus.<press any key to continue>

```

Gambar 7 screenshot tampilan DLLNC (Head) saat memilih pilihan 4

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 5
Data "8" yang berada di belakang telah berhasil dihapus.<press any key to continue>

```

Gambar 8 screenshot tampilan DLLNC (Head) saat memilih pilihan 5

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
2 1 6 7
<press any key to continue>

```

Gambar 9 screenshot tampilan DLLNC (Head) saat menampilkan data setelah dihapus depan dan belakang

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 6
Seluruh data pada Linked List telah dibersihkan.<press any key to continue>

```

Gambar 10 screenshot tampilan DLLNC (Head) saat memilih pilihan 6

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
Tidak terdapat data pada Linked List<press any key to continue>

```

Gambar 11 screenshot tampilan DLLNC (Head) saat menampilkan data setelah di reset

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 

```

Gambar 12 screenshot tampilan awal DLLNC (Head dan Tail)

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 1
Masukkan data : 1
Data "1" berhasil dimasukkan di bagian depan.
press any key to continue

```

Gambar 13 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 1

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
3 2 1

press any key to continue

```

Gambar 14 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah memilih pilihan 1

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 2
Masukkan data : 5
Data "5" berhasil dimasukkan di bagian belakang.
press any key to continue

```

Gambar 15 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 2

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
3 2 1 5 6 7

press any key to continue

```

Gambar 16 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah memilih pilihan 2

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 4
Data "3" yang berada di depan telah berhasil dihapus.
press any key to continue

```

Gambar 17 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 4

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 5
Data "7" yang berada di belakang telah berhasil dihapus.
press any key to continue

```

Gambar 18 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 5

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
2 1 5 6

```

Gambar 19 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah dihapus depan dan belakang

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 6
Seluruh data pada Linked List telah dibersihkan.
press any key to continue

```

Gambar 20 screenshot tampilan DLLNC (Head dan Tail) saat memilih pilihan 6

```

Double Linked List Non Circular (DLLNC) (Head dan Tail)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
Tidak terdapat data pada Linked List
press any key to continue

```

Gambar 21 screenshot tampilan DLLNC (Head dan Tail) saat menampilkan data setelah di reset

III. Pembahasan

#include <conio.h>, #include <iostream>, #include <stdlib.h>,
Mengimpor library yang dibutuhkan :

- conio.h untuk fungsi getch()
- iostream untuk input/output standar
- stdlib.h untuk system("cls") dan atoi()

using namespace std; , agar tidak perlu menulis std:: di depan fungsi-fungsi seperti cout, cin, dll.

typedef struct TNode {, Struktur TNode untuk node dari double linked list, menyimpan data, pointer ke node berikut (next) dan sebelumnya (prev).

`initH()`, mengatur pointer head menjadi NULL, membuat linked list kosong.

`initHT()`, sama seperti `initH()`, ditambah `tail` juga diset ke NULL.

`isEmptyH()`, mengecek apakah linked list (dengan head) kosong.

`isEmptyHT()`, mengecek apakah linked list (dengan head-tail) kosong.

`tambahDepanH()`, menambah simpul di bagian depan list (head-only).

`tambahDepanHT()`, menambah simpul di bagian belakang list (head & tail).

`hapusDepanH()`, menghapus simpul dari depan list (head-only).

`hapusDepanHT()`, menghapus simpul dari depan list (head & tail).

`hapusBelakangH()`, menghapus simpul dari belakang list (head-only).

`hapusBelakangHT()`, menghapus simpul dari belakang list (head & tail).

`tampilkanH()`, menampilkan seluruh data dari head ke belakang (head-only).

`tampilkanHT()`, menampilkan seluruh data dari head ke tail (head & tail).

`clearH()`, menghapus seluruh simpul (reset list) untuk head-only.

`clearHT()`, menghapus seluruh simpul (reset list) untuk head & tail.

SOAL 2

2. Apa fungsi `next` pada coding?

I. Pembahasan

Dalam implementasi struktur Double Linked List (DLL) pada kode program ini, pointer `next` memainkan peran penting dalam mengelola hubungan antar simpul (node) dalam arah maju. Setiap simpul (yang didefinisikan dalam `struct TNode`) memiliki dua pointer: `next` yang menunjuk ke simpul berikutnya, dan `prev` yang menunjuk ke simpul sebelumnya. Fokus utama pada pointer `next` adalah untuk memungkinkan traversal atau penelusuran dari simpul pertama (`head`) menuju simpul terakhir (`tail`), serta mendukung berbagai operasi seperti penambahan, penghapusan, dan penampilan data dalam urutan yang benar.

Pada fungsi seperti `tambahBelakangH`, `tampilkanH`, dan `hapusBelakangH`, pointer `next` digunakan untuk melakukan iterasi dari satu node ke node berikutnya hingga mencapai ujung list. Ini memungkinkan program untuk menemukan posisi akhir list ketika ingin menambahkan data di belakang, mencetak seluruh isi list, atau menghapus simpul terakhir. Tanpa adanya `next`, program tidak akan mampu mengetahui simpul mana yang datang setelah simpul tertentu, sehingga operasi-operasi tersebut tidak dapat dilaksanakan dengan benar.

Selain itu, dalam variasi implementasi dengan head dan tail (HT), pointer `next` juga digunakan untuk menyambungkan simpul baru ke ujung list dengan efisien. Dengan adanya pointer `tail`, traversal penuh menggunakan `next` bisa dihindari saat menambahkan simpul di belakang, tetapi `next` tetap digunakan untuk memperbarui hubungan antar simpul agar struktur list tetap konsisten dan dapat ditelusuri dari depan.

Dengan demikian, pointer `next` bukan hanya penunjuk sederhana ke simpul berikutnya, melainkan elemen vital dalam membangun struktur list yang dinamis dan

fleksibel. Ia mendukung pengelolaan data yang terstruktur, efisien, dan memungkinkan berbagai operasi manipulasi data dilakukan dengan cara yang teratur dan logis. Tanpa `next`, list tidak akan memiliki "rantai penghubung" ke depan, dan fungsionalitas utama dari linked list akan hilang.

Contoh :

Tabel 2 Contoh Source Code Soal 2

```
bantu = head;
while(bantu->next != NULL) {
    bantu = bantu->next;
}
bantu->next = baru;
baru->prev = bantu;
```

`next` digunakan untuk melintasi list sampai akhir, lalu menyambungkan simpul baru di bagian belakang.

Tabel 3 Contoh Source Code Soal 2

```
while(bantu != NULL) {
    cout<<bantu->data<<' ';
    bantu = bantu->next;
}
```

`next` digunakan untuk bergerak dari simpul ke simpul hingga akhir list saat menampilkan semua data.

Tabel 4 Contoh Source Code Soal 2

```
while(hapus->next != NULL) {
    hapus = hapus->next;
}
hapus->prev->next = NULL;
```

`next` membantu menemukan simpul terakhir yang akan dihapus.

Tabel 5 Contoh Source Code Soal 2

```
tail->next = baru;  
baru->prev = tail;  
tail = baru;
```

next disetel untuk menghubungkan tail saat ini ke simpul baru.

SOAL 3

3. Apa fungsi prev pada coding?

I. Pembahasan

Pointer `prev` dalam Double Linked List berfungsi sebagai penghubung ke simpul sebelumnya. Perannya sangat penting untuk:

- Memastikan traversal atau penelusuran dua arah: dari awal ke akhir (`next`), dan dari akhir ke awal (`prev`).
- Mempermudah operasi penghapusan dan penyisipan simpul di posisi mana pun.
- Menjaga integritas hubungan antar simpul sehingga setiap node bisa diakses dari dua arah.

Tanpa `prev`, linked list hanya dapat ditelusuri satu arah (maju), dan menjadi kurang fleksibel, terutama jika kita ingin menghapus simpul dari belakang atau dari tengah tanpa traversal panjang. Dengan kata lain, `prev` menjadikan Double Linked List jauh lebih dinamis dan efisien dibanding struktur satu arah.

Contoh :

Tabel 6 Contoh Source Code Soal 3

```
typedef struct TNode {  
    string data;  
    TNode *next;  
    TNode *prev;  
};
```

Pointer `prev` adalah penunjuk ke simpul sebelumnya dalam urutan list. Peran utama `prev` adalah memungkinkan navigasi dua arah, yaitu dari simpul saat ini ke simpul sebelumnya (mundur), melengkapi pointer `next` yang berjalan ke depan.

Tabel 7 Contoh Source Code Soal 3

```
baru->next = head;  
head->prev = baru;  
head = baru;
```

Di sini, `prev` menghubungkan simpul lama (`head`) ke simpul baru di depannya, sehingga struktur dua arah tetap utuh.

Tabel 8 Contoh Source Code Soal 3

```
tail->next = baru;  
baru->prev = tail;  
tail = baru;
```

`prev` digunakan untuk menyambungkan simpul baru ke simpul sebelumnya (yang tadinya `tail`), agar dapat ditelusuri mundur dari simpul baru ke simpul lama.

Tabel 9 Contoh Source Code Soal 3

```
tail = tail->prev;  
tail->next = NULL;
```

Pointer `prev` memungkinkan program melangkah mundur satu simpul ke belakang, untuk mengatur simpul baru sebagai `tail` dan memutus simpul terakhir.

Tabel 10 Contoh Source Code Soal 3

```
hapus->prev->next = NULL;
```

`prev` digunakan untuk melacak simpul sebelumnya agar dapat memutus sambungan ke simpul yang ingin dihapus.

SOAL 4

4. Gantilah baris 244 dan 256 dari `cout<<bantu->data<<' '`; menjadi `cout<<head->data<<' '`; lalu jawab pertanyaan berikut :
- A. Apa yang terjadi jika anda menambahkan beberapa data pada program lalu tampilkan datanya, dan screenshoot hasilnya.
- B. Jelaskan mengapa hal tersebut bisa terjadi dan data apa yang ditampilkan oleh program?

I. Pembahasan

- A. Jika menambahkan beberapa data ke dalam program dan mencoba menampilkan datanya, program akan mencetak nilai dari `head->data` berulang kali, sebanyak jumlah node yang ada dalam linked list. Semua elemen yang ditampilkan akan sama dengan data yang ada di node pertama (`head`).

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 1
Masukkan data : 1
Data "1" berhasil dimasukkan di bagian depan.<press any key to continue>
```

Gambar 22 screenshot tampilan saat memasukkan data

```
Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 1
Masukkan data : 2
Data "2" berhasil dimasukkan di bagian depan.<press any key to continue>
```

Gambar 23 screenshot tampilan saat memasukkan data

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 1
Masukkan data : 3
Data "3" berhasil dimasukkan di bagian depan.<press any key to continue>

```

Gambar 24 screenshot tampilan saat memasukkan data

```

Double Linked List Non Circular (DLLNC) (Head)
=====
1. Tambah Depan
2. Tambah Belakang
3. Tampilkan Data
4. Hapus Depan
5. Hapus Belakang
6. Reset
7. Kembali ke Menu
Pilihan : 3
3 3 3
<press any key to continue>

```

Gambar 25 screenshot tampilan saat menampilkan data

B. Hal ini terjadi karena pada baris yang diubah,

mengganti `cout<<bantu->data<<'`

`';` dengan `cout<<head->data<<' '`;

- Sebelumnya :

Variabel `bantu` digunakan untuk iterasi melalui setiap node dalam linked list, sehingga setiap node ditampilkan satu per satu.

- Setelah perubahan :

Program tidak lagi menggunakan `bantu` untuk menampilkan data dari setiap node. Sebaliknya, program hanya mencetak data dari node pertama (`head`) berulang kali, karena `head->data` tidak berubah selama iterasi.

Data yang ditampilkan oleh program adalah nilai dari `head->data`, yaitu data dari node pertama dalam linked list.

TAUTAN GIT

<https://github.com/DSA25-ULM/task-4-non-circular-doubly-linked-list-rikafaulianarahmi>