

**LAPORAN PRAKTIKUM  
ALGORITMA & STRUKTUR DATA  
MODUL 3**



**SINGLE 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 3**

Laporan Praktikum Algoritma & Struktur Data Modul 3: Single 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 GAMBAR .....	4
SOAL .....	5
PEMBAHASAN .....	11
Soal 1.....	11
Soal 2.....	11
Soal 3.....	12
Soal 4.....	12
Soal 5.....	13
Soal 6.....	13
Soal 7.....	14
Soal 8.....	15
Soal 9.....	16
Soal 10.....	17
TAUTAN GIT.....	18

## DAFTAR GAMBAR

Gambar 1 Screenshot jawaban soal 1.....	11
Gambar 2 Screenshot jawaban soal 2.....	11
Gambar 3 Screenshot jawaban soal 3.....	12
Gambar 4 Screenshot jawaban soal 4.....	12
Gambar 5 Screenshot jawaban soal 5.....	13
Gambar 6 Screenshot jawaban soal 6 hapus belakang.....	13
Gambar 7 Screenshot jawaban soal 6 menampilkan data .....	14
Gambar 8 Screenshot jawaban soal 7 menghapus angka 3 .....	14
Gambar 9 Screenshot jawaban soal 7 menampilkan data .....	15
Gambar 10 Screenshot jawaban soal 8.....	15

## SOAL

1. Cobalah program berikut, 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 };
11
12 TNode *head, *tail;
13
14 int pil;
15 char pilihan[2];
16 string dataBaru, data;
17
18 void init();
19 int isEmpty();
20
21 void tambahDepan();
22 void tambahBelakang();
23 void hapusDepan();
24 void hapusBelakang();
25 void tampilkan();
26 void reset();
27 void cariData();
28 void hapusData();
29 void sisipkanSebelum();
30 void sisipkanSetelah();
31
32 int main()
33 {
34     do {
35         cout<<"Single Linked List Circular (SLLC)"<<endl;
36         cout<<"-----"<<endl;
37         cout<<"1. Tambah Depan"<<endl;
38         cout<<"2. Tambah Belakang"<<endl;
39         cout<<"3. Hapus Depan"<<endl;
40         cout<<"4. Hapus Belakang"<<endl;
41         cout<<"5. Tampilkan Data"<<endl;
42         cout<<"6. Hapus Semua Elemen"<<endl;
43         cout<<"7. Cari Data"<<endl;
44         cout<<"8. Hapus Setiap Data Tertentu"<<endl;
45         cout<<"9. Sisipkan Node/Data Baru Sebelum Data Tertentu"<<endl;
46         cout<<"10. Sisipkan Node/Data Baru Setelah Data Tertentu"<<endl;
47         cout<<"11. Quit"<<endl;
48         cout<<"Pilihan : ";
49
50         cin>>pilihan;
51         pil=atoi(pilihan);
52
53         switch(pil) {
54             case 1:
55                 tambahDepan();
56                 break;
57             case 2:
58                 tambahBelakang();
59                 break;
60             case 3:
61                 hapusDepan();
62                 cout<<"Data \"<<data<<\" yang berada di depan telah berhasil dihapus."<<endl;
63                 break;
64             case 4:
65                 hapusBelakang();
66                 cout<<"Data \"<<data<<\" yang berada di belakang telah berhasil dihapus."<<endl;
67                 break;
68             case 5:
69                 tampilkan();
70                 break;
71             case 6:
72                 reset();
73                 break;
74             case 7:
75                 cariData();
76                 break;
77             case 8:
78                 hapusData();
79                 break;
80             case 9:
81                 sisipkanSebelum();
82                 break;
83             case 10:
84                 sisipkanSetelah();
85                 break;
86             default:
87                 cout<<"\nTERIMA KASIH"<<endl;
88                 cout<<"Program was made by Nama (NIM)."<<endl;
89
90                 cout<<"\nPress any key to continue!"<<endl;
91                 getch();
92                 system("cls");
93
94         } while (pil<11);
```

```

95     }
96
97     void init(){
98         head = NULL;
99         tail = NULL;
100     }
101
102     int isEmpty() {
103         if(head == NULL) return 1;
104         else return 0;
105     }
106
107     void tambahDepan() {
108         cout<<"Masukkan data : ";
109         TNode *baru;
110         baru = new TNode;
111         cin>>dataBaru;
112         baru->data = dataBaru;
113         baru->next = baru;
114
115         if(isEmpty() == 1) {
116             head = baru;
117             tail = baru;
118         } else {
119             baru->next = head;
120             head = baru;
121             tail->next = head;
122         }
123         cout << "Data \""<<dataBaru<<"\" berhasil dimasukkan di bagian depan."<<endl;
124     }
125
126     void tambahBelakang() {
127         cout<<"Masukkan data : ";
128         TNode *baru;
129         baru = new TNode;
130         cin>>dataBaru;
131         baru->data = dataBaru;
132         baru->next = baru;
133
134         if(isEmpty() == 1) {
135             head = baru;
136             tail = baru;
137         } else {
138             tail->next = baru;
139             tail = baru;

```

```

140         tail->next = head;
141     }
142     cout << "Data \""<<dataBaru<<"\" berhasil dimasukkan di bagian belakang."<<endl;
143 }
144
145 void hapusDepan() {
146     if(isEmpty() == 0) {
147         TNode *hapus;
148         hapus = head;
149         data = hapus->data;
150
151         if(head != tail) {
152             head = head->next;
153             tail->next = head;
154         } else {
155             init();
156         }
157
158         delete hapus;
159     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
160 }
161
162 void hapusBelakang() {
163     if(isEmpty() == 0) {
164         TNode *hapus, *newTail;
165         hapus = tail;
166         data = hapus->data;
167
168         if(head != tail) {
169             newTail = head;
170             while(newTail->next != tail) {
171                 newTail = newTail->next;
172             }
173             tail = newTail;
174             tail->next = head;
175         } else {
176             init();
177         }
178
179         delete hapus;
180     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
181 }
182
183 void tampilkan() {
184     if(isEmpty() == 0) {
185         TNode *bantu;

```

```

186         bantu = head;
187
188         do {
189             cout<<bantu->data<<' ';
190             bantu = bantu->next;
191         } while(bantu != head);
192         cout<<endl;
193     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
194 }
195
196 void reset() {
197     if(isEmpty() == 0) {
198         TNode *bantu, *hapus;
199         bantu = head;
200
201         do {
202             hapus = bantu;
203             bantu = bantu->next;
204             delete hapus;
205         } while(bantu != head);
206
207         init();
208         cout<<"Seluruh elemen pada Linked List telah dibersihkan."<<endl;
209     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
210 }
211
212 void cariData() {
213     if(isEmpty() == 0) {
214         string cari;
215         cout<<"Masukkan data yang ingin dicari : ";
216         cin>>cari;
217
218         TNode *bantu, *hapus, *newTail, *bantuTampilkan;
219         bool apaDitemukan = false;
220
221         bantu = head;
222
223         do {
224             if(cari == bantu->data){
225                 cout<<"Setiap data yang berada di dalam tanda kurung siku ([...]) "
226                     <<"merupakan data yang anda cari."<<endl;
227                 cout<<"Linked List : ";
228                 bantuTampilkan = head;
229
230                 do {
231                     if(cari == bantuTampilkan->data)

```

```

232                     cout<<"["<<bantuTampilkan->data<<" ]";
233                     else
234                         cout<<bantuTampilkan->data<<' ';
235                     bantuTampilkan = bantuTampilkan->next;
236                 } while(bantuTampilkan != head);
237
238                 apaDitemukan = true;
239                 cout<<endl;
240                 break;
241             }
242             bantu = bantu->next;
243         } while(bantu != head);
244
245         if(apaDitemukan == false)
246             cout<<"Data \"<<cari<<"\" tidak ditemukan pada Linked List."<<endl;
247     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
248 }
249
250 void hapusData() {
251     if(isEmpty() == 0) {
252         string cari;
253         cout<<"Masukkan data yang ingin dihapus : ";
254         cin>>cari;
255
256         TNode *bantu, *sebelum, *hapus[255], *bantu2;
257         int hitung = 0;
258         bool apaDitemukan = false;
259
260         bantu = head;
261
262         do {
263             bantu2 = bantu;
264             if(cari == bantu->data){
265                 hapus[hitung++] = bantu;
266                 apaDitemukan = true;
267                 if(bantu != head && bantu != tail) {
268                     sebelum->next = bantu->next;
269                     bantu2 = sebelum;
270                 }
271             }
272             sebelum = bantu2;
273             bantu = bantu->next;
274         } while(bantu != head);
275
276         if(apaDitemukan == true) {
277             for(int i = 0; i < hitung; i++) {

```

```

278         if(hapus[i] == head){
279             hapusDepan();
280         } else if(hapus[i] == tail) {
281             hapusBelakang();
282         } else {
283             delete hapus[i];
284         }
285     }
286
287     cout<<"Setiap data \""<<cari<<" yang terdapat pada Linked List telah dihapus";
288     } else cout<<"Data \""<<cari<<" tidak ditemukan pada Linked List."<<endl;
289
290     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
291 }
292
293 void sisipkanSebelum() {
294     if(isEmpty() == 0) {
295         TNode *bantu, *sebelum;
296         string nextData;
297         bool apaAda;
298
299         bantu = head;
300         sebelum = tail;
301
302         cout<<"Sisipkan data baru sebelum data : ";
303         cin>>nextData;
304
305         do {
306             if(nextData == bantu->data){
307                 apaAda = true;
308                 break;
309             } else {
310                 sebelum = bantu;
311                 bantu = bantu->next;
312             }
313         } while(bantu != head);
314
315         if(apaAda==true) {
316             cout<<"Masukkan data yang ingin ditambahkan : ";
317             cin>>dataBaru;
318
319             TNode *baru;
320             baru = new TNode;
321
322             baru->data = dataBaru;

```



```

323     baru->next = bantu;
324
325     sebelum->next = baru;
326
327     if(bantu == head){
328         head = baru;
329     }
330
331     cout << "Data \'"<<dataBaru<<\'\" berhasil disisipkan sebelum data \'"<<nextData<<\'\"."<<endl;
332 } else {
333     cout<<"Tidak terdapat data \'"<<nextData<<\'\" pada Linked List."<<endl;
334 }
335 } else cout<<"Tidak terdapat data pada Linked List."<<endl;
336 }
337
338 void sisipkanSetelah() {
339     if(isEmpty() == 0) {
340         TNode *bantu;
341         string prevData;
342         bool apaAda;
343
344         bantu = head;
345
346         cout<<"Sisipkan data baru setelah data : ";
347         cin>>prevData;
348
349         do {
350             if(prevData == bantu->data){
351                 apaAda = true;
352                 break;
353             } else {
354                 bantu = bantu->next;
355             }
356         } while(bantu != head);
357
358         if(apaAda==true) {
359             cout<<"Masukkan data yang ingin ditambahkan : ";
360             cin>>dataBaru;
361
362             TNode *baru;
363             baru = new TNode;
364
365             baru->data = dataBaru;
366             baru->next = bantu->next;
367
368             bantu->next = baru;
369
370             if(bantu == tail){
371                 tail = baru;
372             }
373
374             cout << "Data \'"<<dataBaru<<\'\" berhasil disisipkan setelah data \'"<<prevData<<\'\"."<<endl;
375         } else {
376             cout<<"Tidak terdapat data \'"<<prevData<<\'\" pada Linked List."<<endl;
377         }
378     } else cout<<"Tidak terdapat data pada Linked List."<<endl;
379 }
380

```

2. Lakukan tambah data depan 3, 4, 7, 9, 10, 12 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !
3. Lakukan tambah data belakang 3, 7, 1, 4, 3 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !
4. Apa yang terjadi jika mencari angka 2 pada Single Linked List Circular (SLLC) pada data yang telah ditambahkan/dimasukkan sebelumnya dan screenshoot hasilnya
5. Coba cari angka 7 dan screenshoot hasilnya !

6. Lakukan hapus belakang dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !
7. Lakukan hapus setiap angka 3 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !
8. Tampilkan data lalu jelaskan yang mana head dan yang mana tail.
9. Jika baris ke 103 dan 104 dihapus maka apa yang akan terjadi pada saat memasukkan data, dan jelaskan mengapa?
10. Jelaskan apa itu variabel head dan tail pada slhc!

## PEMBAHASAN

### Soal 1

Cobalah program berikut, running, simpan program, dan screenshoot hasil running !

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan =
```

*Gambar 1 Screenshot jawaban soal 1*

### Soal 2

Lakukan tambah data depan 3, 4, 7, 9, 10, 12 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
ta Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 5
<12 <10 <9 <7 <4 <3
```

*Gambar 2 Screenshot jawaban soal 2*

### Soal 3

Lakukan tambah data belakang 3, 7, 1, 4, 3 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 5
<12 <10 <9 <7 <4 <3 <3 <7 <1 <4 <3
```

Gambar 3 Screenshot jawaban soal 3

### Soal 4

Apa yang terjadi jika mencari angka 2 pada Single Linked List Circular (SLLC) pada data yang telah ditambahkan/dimasukkan sebelumnya dan screenshoot hasilnya

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 7
Masukkan data yang ingin dicari : 2
Data "2" tidak ditemukan pada Linked List.
```

Gambar 4 Screenshot jawaban soal 4

### Soal 5

Coba cari angka 7 dan screenshoot hasilnya !

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 7
Masukkan data yang ingin dicari : 7
Setiap data yang berada di dalam tanda kurung siku ([...]) merupakan data yang anda cari.
Linked List : 12 10 9 [7] 4 3 3 [7] 1 4 3
```

*Gambar 5 Screenshot jawaban soal 5*

### Soal 6

Lakukan hapus belakang dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

```
Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 4
Data "3" yang berada di belakang telah berhasil dihapus.
```

*Gambar 6 Screenshot jawaban soal 6 hapus belakang*

```

Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 5
<12 <10 <9 <7 <4 <3 <3 <7 <1 <4

```

*Gambar 7 Screenshot jawaban soal 6 menampilkan data*

### Soal 7

Lakukan hapus setiap angka 3 dan kemudian lakukan tampilkan data lalu screenshot hasilnya !

```

Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 8
Masukkan data yang ingin dihapus : 3
Setiap data "3" yang terdapat pada Linked List telah dihapus

```

*Gambar 8 Screenshot jawaban soal 7 menghapus angka 3*

```

Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 5
<12 <10 <9 <7 <4 <7 <1 <4

```

*Gambar 9 Screenshot jawaban soal 7 menampilkan data*

### Soal 8

Tampilkan data lalu jelaskan yang mana head dan yang mana tail.

```

Single Linked List Circular (SLLC)
=====
1. Tambah Depan
2. Tambah Belakang
3. Hapus Depan
4. Hapus Belakang
5. Tampilkan Data
6. Hapus Semua Elemen
7. Cari Data
8. Hapus Setelah Data Tertentu
9. Sisipkan Node/Data Baru Sebelum Data Tertentu
10. Sisipkan Node/Data Baru Setelah Data Tertentu
11. Quit
Pilihan = 5
<12 <10 <9 <7 <4 <7 <1 <4

```

*Gambar 10 Screenshot jawaban soal 8*

Head ialah angka 12

Tail ialah angka 4

### Soal 9

Jika baris ke 103 dan 104 dihapus maka apa yang akan terjadi pada saat memasukkan data, dan jelaskan mengapa?

program berhenti setelah memasukkan data pada opsi. Hal ini terjadi karena fungsi `isEmpty()` tidak lagi berfungsi dengan benar setelah baris 103 dan 104 dihapus.

#### 1. Baris yang dihapus

```
if(head == NULL) return 1;  
else return 0;
```

- Baris ini adalah inti dari fungsi `isEmpty()`, yang memeriksa apakah linked list kosong.
- Setelah baris ini dihapus, fungsi `isEmpty()` tidak mengembalikan nilai apa pun, sehingga menyebabkan undefined behavior.

#### 2. Dampak pada fungsi

- Fungsi `tambahDepan()` menggunakan `isEmpty()` untuk memeriksa apakah linked list kosong.
- Karena `isEmpty()` tidak mengembalikan nilai, kondisi `if(isEmpty() == 1)` tidak dapat dievaluasi dengan benar. Hal ini menyebabkan program tidak tahu apakah harus menginisialisasi `head` dan `tail` atau menambahkan node baru ke linked list.

#### 3. Hasilnya

Program berhenti atau crash setelah Anda memasukkan data karena pointer `head` dan `tail` tidak diatur dengan benar. Pointer `next` dari node baru juga tidak diatur, sehingga iterasi atau operasi lain pada linked list menjadi tidak valid.



**Soal 10**

Jelaskan apa itu variabel head dan tail pada slhc!

Dalam struktur slhc, terdapat dua variabel penting, yaitu head dan tail.

Head = variabel yang menunjuk ke node pertama dalam linked list. Node inilah yang menjadi titik awal untuk menelusuri seluruh elemen dalam list. Head digunakan saat ingin menelusuri atau menampilkan semua elemen dari awal.

Tail = variabel yang menunjuk node terakhir dalam linked list. Dalam slhc, node `tail` memiliki pointer `next` yang mengarah kembali ke `head`, sehingga membentuk struktur melingkar (circular). `tail` memudahkan penambahan data di akhir dan memastikan struktur tetap sirkular.

## **TAUTAN GIT**

<https://github.com/DSA25-ULM/task-3-single-circular-linked-list-rikafaulianarahmi>