

# LAPORAN PRAKTIKUM

Matakuliah	Struktur Data
Pertemuan ke	8
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
2. Memahami tipe data bentukan
3. Stuktur Program menggunakan bahasa C++ Praktikum Linked list Non circular dan Circular

## B. Hasil Percobaan

### 1. Percobaan Praktikum 1

#### a) Tampilan Coding

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <conio.h>
4  #include <stdlib.h>
5  using namespace std;
6
7  typedef struct Data
8  {
9      int nilai;
10     Data* next;
11 };
12 Data* head;
13 Data* tail;
14
15 //void awal
16 void awal()
17 {
18     head = NULL;
19 }
20
21 //fungsi mengecek node kosong
22 bool isEmpty()
23 {
24     if (head == NULL)
25         return true;
26     return false;
27 }
28
```

```

29 //menambah data di depan
30 void tambahDataDepan(int DataBaru)
31 {
32     Data* baru;
33     baru = new Data;
34     baru->nilai = DataBaru;
35     baru->next = NULL;
36     if (isEmpty())
37     {
38         head = baru;
39         head->next = NULL;
40     }
41     else
42     {
43         baru->next = head;
44         head = baru;
45     }
46     cout << "Data Depan " << DataBaru << " Masuk" << endl;
47 }
48
49 //menambah data di belakang
50 void tambahDataBelakang(int DataBaru)
51 {
52     Data* baru, * bantu;
53     baru = new Data;
54     baru->nilai = DataBaru;
55     baru->next = NULL;
56     if (isEmpty())
57     {
58         head = baru;
59         head->next = NULL;
60     }
61     else
62     {
63         bantu = head;
64         while (bantu->next != NULL)
65         {
66             bantu = bantu->next;
67         }
68         bantu->next = baru;
69     }
70     cout << "Data Belakang " << DataBaru << " Masuk" << endl;
71 }
72
73 //menghapus node depan
74 void hapusDepan()
75 {
76     Data* hapus;
77     int d;
78     if (!isEmpty())
79     {
80         if (head->next != NULL)
81         {
82             hapus = head;
83             d = hapus->nilai;
84             head = hapus->next;
85             delete hapus;
86         }
87         else
88         {
89             d = head->nilai;
90             head = NULL;
91         }
92         cout << d << " Terhapus" << endl;
93     }
94     else cout << "Masih Kosong" << endl;
95 }

```

```

96
97 //hapus node belakang
98 void hapusBelakang()
99 {
100     Data* hapus, * bantu;
101     int tmp;
102     if (!isEmpty())
103     {
104
105         if (head->next != NULL)
106         {
107             bantu = head;
108             while (bantu->next->next != NULL)
109             {
110                 bantu = bantu->next;
111             }
112             hapus = bantu->next;
113             tmp = hapus->nilai;
114             bantu->next = NULL;
115             delete hapus;
116         }
117         else
118         {
119             tmp = head->nilai;
120             head = NULL;
121         }
122         cout << tmp << " Terhapus" << endl;
123     }
124     else cout << "Masih Kosong" << endl;
125 }
126
127 //mencetak linked list
128 void Cetak()
129 {
130     if (!isEmpty())
131     {
132         Data* bantu;
133         bantu = head;
134         do
135         {
136             cout << bantu->nilai << " ";
137             bantu = bantu->next;
138         } while (bantu != NULL);
139         cout << endl;
140     }
141 }

```

```

142
143 //fungsi mengukur panjang linked list
144 int panjang()
145 {
146     int count = 0;
147     if (!isEmpty())
148     {
149         count = 1;
150         Data* bantu;
151         bantu = head;
152         if (bantu->next == NULL)
153         {
154             count = 1;
155         }
156         else
157         {
158             do
159             {
160                 count++;
161                 bantu = bantu->next;
162             } while (bantu->next != NULL);
163         }
164     }
165     else
166     {
167         count = 0;
168     }
169     return count;
170 }
171
172

```

```

173 int main()
174 {
175     awal();
176     tambahDataBelakang(5);
177     tambahDataDepan(7);
178     tambahDataBelakang(17);
179     tambahDataBelakang(1);
180     tambahDataBelakang(27);
181     tambahDataBelakang(10);
182     cout << "Data pada single linked list non circular:" << endl;
183     Cetak();
184     cout << "Data paling depan dihapus:" << endl;
185     hapusDepan();
186     cout << "Data pada single linked list non circular:" << endl;
187     Cetak();
188     cout << "Data paling belakang dihapus:" << endl;
189     hapusBelakang();
190     cout << "Data pada single linked list non circular:" << endl;
191     Cetak();
192     cout << "Panjang linked list :" << endl;
193     cout << panjang();
194     system("pause");
195     return 0;
196 }

```

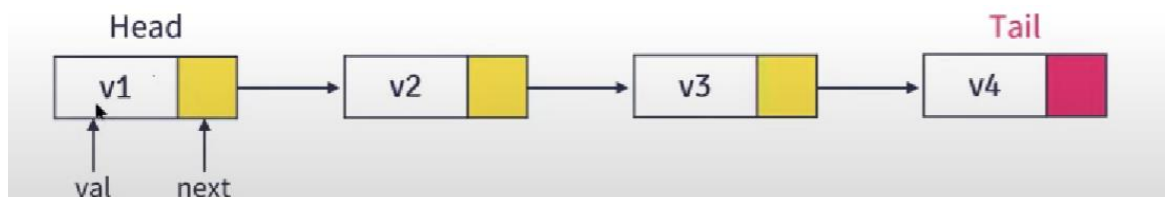
### b) Hasil Running

```
Data Belakang 5 Masuk
Data Depan 7 Masuk
Data Belakang 17 Masuk
Data Belakang 1 Masuk
Data Belakang 27 Masuk
Data Belakang 10 Masuk
Data pada single linked list non circular:
7 5 17 1 27 10
Data paling depan dihapus:
7 Terhapus
Data pada single linked list non circular:
5 17 1 27 10
Data paling belakang dihapus:
10 Terhapus
Data pada single linked list non circular:
5 17 1 27
Panjang linked list :
4
Press any key to continue . . . _
```

### c) Penjelasan

Linked List adalah salah satu bentuk struktur data, berisi kumpulan data (node) yang tersusun secara sekuensial, saling sambung - menyambung, dinamis dan terbatas. Masing-masing data dalam Linked List disebut dengan node (simpul) yang menempati alokasi memori secara dinamis dan biasanya berupa struct yang terdiri dari beberapa field.

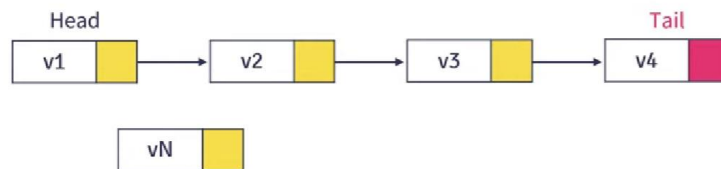
Single linked list terdiri dari sejumlah elemen(node) dimana setiap node memiliki penunjuk node berikutnya ke elemen (node) berikutnya. Penunjuk node terakhir adalah NULL, yang menunjukkan akhir dari single linked list.



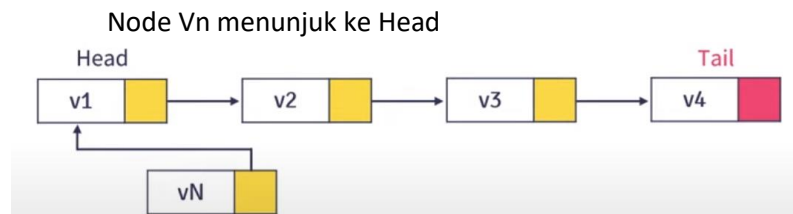
Penjelasan Fungsi coding diatas yaitu:

- 1) Void awal  
Head = null; Kita mendeklarasikan head = null karena awal dari pembuatan
- 2) Bool is Empty  
Fungsi untuk mengecek node berisi data kosong atau tidak. Algoritma nya akan return true bila data kosong dan false bila data sudah ada isinya.  
Apabila data kosong, maka akan membuat node baru ini akan menjadi head di setiap
- 3) Menambah data depan / Menyisipkan Node di awal.  
Langkah algoritmanya yaitu:

-Membuat value node baru (vn) dan memasukan data baru ke node baru.

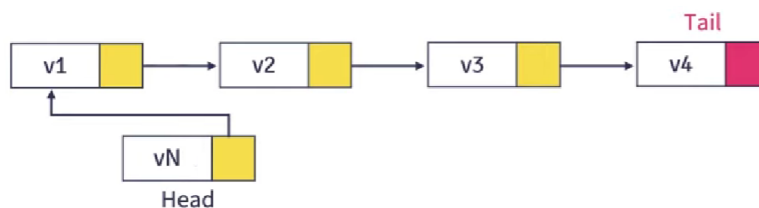


-Mengecek, Bila ada data maka:



Bila tidak ada node maka Vn ini akan langsung jadi Head.

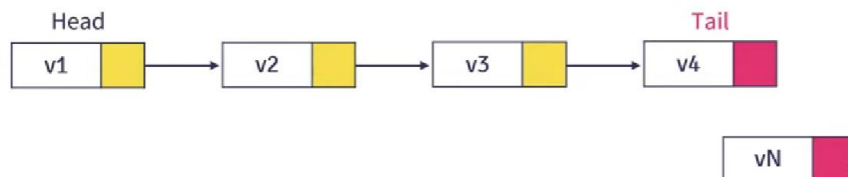
-Vn menjadi head



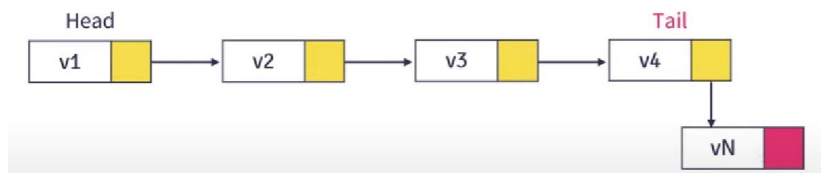
#### 4) Menambah data belakang / Menyisipkan Node di akhir

Langkah Algoritmanya yaitu:

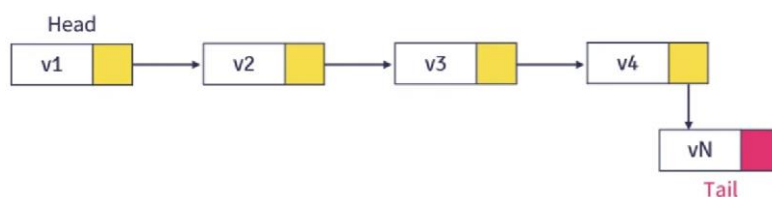
-Membuat Node baru dengan data baru (Vn)



- Tail node terakhir , set ke node baru

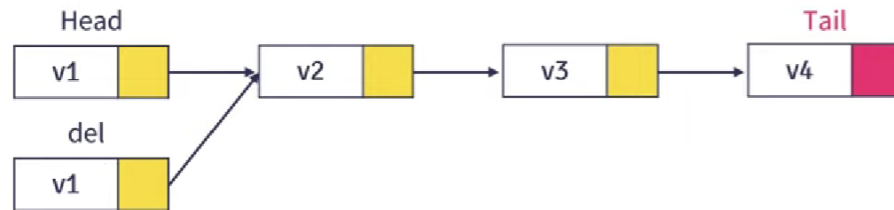


- Tail node baru diset menjadi NULL

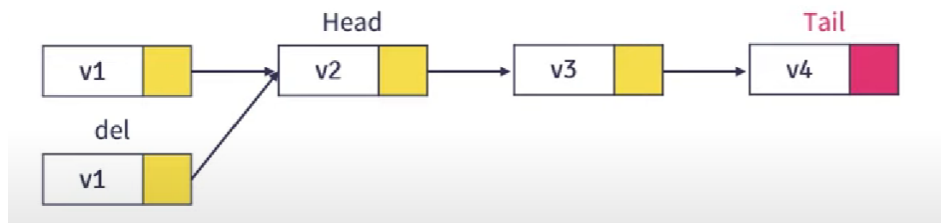


5) Menghapus node depan

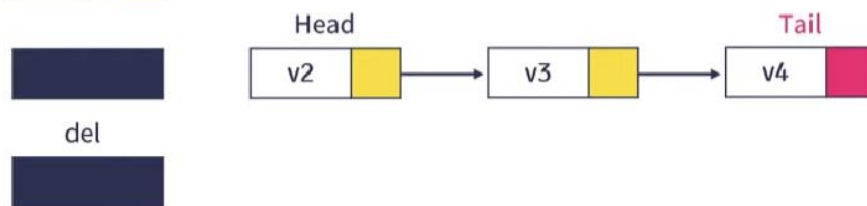
- Membuat node baru del/ hapus, dimana node baru ini berisi copyan nilai dan pointer dari head



- Memindahkan head ke posisi pointer del



- Menghapus node del/hapus



6) Menghapus data belakang

- Membuat node hapus dan bantu
- Node bantu dibuat berisi copyan nilai dan pointer dari head
- Rolling while terus dimana variable bantu ini akan sampai ke node terakhir
- Node hapus akan menampung nilai pointer next
- Tmp akan menampung nilai dari hapus nilai
- Menghapus Nilai node hapus

7) Mencetak linked list

- Membuat node bantu
- Head akan diubah ke node bantu
- Head ini akan berubah terus dan mencetak terus nilai dari node nya, dan akan berjalan berantai terus sampai bertemu kondisi dimana head -> tail = NULL.

8) Menghitung Panjang node

- Membuat node bantu
- Head akan diubah ke node bantu
- Head ini akan terus menambahkan nilai dari count sejalannjya node berantai terus sampai bertemu kondisi head->tail = NULL
- Return nilai dari count.

2. Studi kasus  
a) Tampilan Coding

```
1  #include <iostream>
2
3  using namespace std;
4
5  //deklarasi single linked list
6  struct Buku {
7
8      //komponen / member
9      string judul, pengarang;
10     int tahunTerbit;
11
12     Buku* next;
13
14 };
15
16 Buku* head, * tail, * cur, * newNode, * del, * before;
17
18 //create single linked list
19 void createSingleLinkedList(string judul, string pengarang, int tB) {
20     head = new Buku();
21     head->judul = judul;
22     head->pengarang = pengarang;
23     head->tahunTerbit = tB;
24     head->next = NULL;
25     tail = head;
26 }
27
28 //print single linked list
29 int countSingleLinkedList() {
30     cur = head;
31     int jumlah = 0;
32     while (cur != NULL) {
33         jumlah++;
34         cur = cur->next;
35     }
36     return jumlah;
37 }
38
39 //tambahAwal Single linked list
40 void addFirst(string judul, string pengarang, int tB) {
41     newNode = new Buku();
42     newNode->judul = judul;
43     newNode->pengarang = pengarang;
44     newNode->tahunTerbit = tB;
45     newNode->next = head;
46     head = newNode;
47 }
```



```

48
49 //tambahAkhir Single Linked list
50 void addLast(string judul, string pengarang, int tB) {
51     newNode = new Buku();
52     newNode->judul = judul;
53     newNode->pengarang = pengarang;
54     newNode->tahunTerbit = tB;
55     newNode->next = NULL;
56     tail->next = newNode;
57     tail = newNode;
58 }
59
60 //tambah tengah single linked list
61 void addMiddle(string judul, string pengarang, int tB, int posisi) {
62     if (posisi < 1 || posisi > countSingleLinkedList()) {
63         cout << "Posisi diluar jangkauan" << endl;
64     }
65     else if (posisi == 1) {
66         cout << "Posisi bukan posisi tengah" << endl;
67     }
68     else {
69         newNode = new Buku();
70         newNode->judul = judul;
71         newNode->pengarang = pengarang;
72         newNode->tahunTerbit = tB;
73
74
75         //tranversing
76         cur = head;
77         int nomor = 1;
78         while (nomor < posisi - 1) {
79             cur = cur->next;
80             nomor++;
81         }
82         newNode->next = cur->next;
83         cur->next = newNode;
84     }
85 }
86
87 //remove First
88 void removeFirst() {
89     del = head;
90     head = head->next;
91     delete del;
92 }
93
94 //remove Last
95 void removeLast() {
96     del = tail;
97     cur = head;
98     while (cur->next != tail) {
99         cur = cur->next;
100     }
101     tail = cur;
102     tail->next = NULL;
103     delete del;
104 }

```

```

106 //remove middle
107 void removeMiddle(int posisi) {
108     if (posisi < 1 || posisi > countSingleLinkedList()) {
109         cout << "Posisi diluar jangkauan" << endl;
110     }
111     else if (posisi == 1) {
112         cout << "Posisi bukan posisi tengah" << endl;
113     }
114     else {
115         int nomor = 1;
116         cur = head;
117         while (nomor <= posisi) {
118             if (nomor == posisi - 1) {
119                 before = cur;
120             }
121             if (nomor == posisi) {
122                 del = cur;
123             }
124             cur = cur->next;
125             nomor++;
126         }
127         before->next = cur;
128         delete del;
129     }
130 }
131
132 //ubah awal single linked list
133 void changeFirst(string judul, string pengarang, int tB) {
134     head->judul = judul;
135     head->pengarang = pengarang;
136     head->tahunTerbit = tB;
137 }
138
139 //ubahAkhir single linked list
140 void changeLast(string judul, string pengarang, int tB) {
141     tail->judul = judul;
142     tail->pengarang = pengarang;
143     tail->tahunTerbit = tB;
144 }
145

```

```

132 //ubah awal single linked list
133 void changeFirst(string judul, string pengarang, int tB) {
134     head->judul = judul;
135     head->pengarang = pengarang;
136     head->tahunTerbit = tB;
137 }
138
139 //ubahAkhir single linked list
140 void changeLast(string judul, string pengarang, int tB) {
141     tail->judul = judul;
142     tail->pengarang = pengarang;
143     tail->tahunTerbit = tB;
144 }
145
146 //ubah Tengah Single linked list
147 void changeMiddle(string judul, string pengarang, int tB, int posisi) {
148     if (posisi < 1 || posisi == countSingleLinkedList()) {
149         cout << "Posisi diluar jangkauan" << "\n\n" << endl;;
150     }
151     else if (posisi == 1 || posisi == countSingleLinkedList()) {
152         cout << "Posisi bukan posisi tengah" << endl;
153     }
154     else {
155         cur = head;
156         int nomor = 1;
157         while (nomor < posisi) {
158             cur = cur->next;
159             nomor++;
160         }
161         cur->judul = judul;
162         cur->pengarang = pengarang;
163         cur->tahunTerbit = tB;
164     }
165 }
166
167 //print singleLinked List
168 void printSingleLinkedList() {
169     cout << "Jumlah data ada : " << countSingleLinkedList() << endl;
170     cur = head;
171     while (cur != NULL) {
172         cout << "Judul Buku : " << cur->judul << endl;
173         cout << "Pengarang Buku : " << cur->pengarang << endl;
174         cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
175         cur = cur->next;
176     }
177 }
178
179

```

```

181 int main() {
182
183     cout << "Nama : Wijayanto Agung Wibowo" << endl;
184     cout << "Nim : 22.11.4552" << endl << endl;
185
186
187     cout << "Membuat single linked list" << endl;
188     createSingleLinkedList("Kata", "Geez & Aan", 2018);
189
190     printSingleLinkedList();
191
192     cout << "\n\n" << endl;
193
194     cout<<"Memasukan data ke posisi pertama" << endl;
195
196     addFirst("Dia adalah Kakakku", "Tere Liye", 2009);
197
198     printSingleLinkedList();
199
200     cout << "\n\n" << endl;
201
202     cout<<"Memasukan data ke posisi terakhir" << endl;
203
204     addLast("Aroma Karsa", "Dee Lestari", 2018);
205
206     printSingleLinkedList();
207
208     cout << "\n\n" << endl;
209
210     cout<<"Menghapus data pertama" << endl;
211
212     removeFirst();
213
214     printSingleLinkedList();
215
216     cout << "\n\n" << endl;
217
218     cout <<"Memasukan data terakhir" << endl;
219
220     addLast("11.11", "Fierza Besari", 2018);
221
222     printSingleLinkedList();
223
224     cout << "\n\n" << endl;
225
226     cout<<"Menghapus data terakhir" << endl;
227
228     removeLast();
229
230     printSingleLinkedList();
231
232     cout << "\n\n" << endl;
233
234     cout<<"Mengubah data pertama" << endl;

```

```

235
236 changeFirst("Berhenti di Kamu", "Gia Pratama", 2018);
237
238 printSingleLinkedList();
239
240 cout << "\n\n" << endl;
241
242 cout<<"Menambah data ke posisi tengah" << endl;
243
244 addMiddle("Bumi Manusia", "Framoedya Anata Toer", 2005, 2);
245
246 printSingleLinkedList();
247
248 cout << "\n\n" << endl;
249
250 cout<<"Menambah data ke posisi tengah" << endl;
251
252 addMiddle("Negeri 5 Menara", "Ahmad Fuadi", 2006, 2);
253
254 printSingleLinkedList();
255
256 cout << "\n\n" << endl;
257
258 cout << "Menghapus data ke - 5" << endl;
259
260 removeMiddle(5);
261
262 printSingleLinkedList();
263
264 cout << "\n\n" << endl;
265
266 cout<<"Mengubah data posisi tengah" << endl;
267
268 changeMiddle("Sang Pemimpi", "Andrea Hirata", 2006, 2);
269
270 printSingleLinkedList();
271
272 cout << "\n\n" << endl;
273
274 }
275

```

b) Hasil Running

Nama : Wijayanto Agung Wibowo  
Nim : 22.11.4552

Membuat single linked list  
Jumlah data ada : 1  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018

Memasukan data ke posisi pertama  
Jumlah data ada : 2  
Judul Buku : Dia adalah Kakakku  
Pengarang Buku : Tere Liye  
Tahun Terbit Buku : 2009  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018

Memasukan data ke posisi terakhir  
Jumlah data ada : 3  
Judul Buku : Dia adalah Kakakku  
Pengarang Buku : Tere Liye  
Tahun Terbit Buku : 2009  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Menghapus data pertama  
Jumlah data ada : 2  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Memasukan data terakhir  
Jumlah data ada : 3  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018  
Judul Buku : 11.11  
Pengarang Buku : Fierza Besari  
Tahun Terbit Buku : 2018

Menghapus data terakhir  
Jumlah data ada : 2  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Mengubah data pertama  
Jumlah data ada : 2  
Judul Buku : Berhenti di Kamu  
Pengarang Buku : Gia Pratama  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Menambah data ke posisi tengah  
Jumlah data ada : 3  
Judul Buku : Berhenti di Kamu  
Pengarang Buku : Gia Pratama  
Tahun Terbit Buku : 2018  
Judul Buku : Bumi Manusia  
Pengarang Buku : Framoedya Anata Toer  
Tahun Terbit Buku : 2005  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

```
Menambah data ke posisi tengah
Jumlah data ada : 4
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Negeri 5 Menara
Pengarang Buku : Ahmad Fuadi
Tahun Terbit Buku : 2006
Judul Buku : Bumi Manusia
Pengarang Buku : Framoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018
```

```
Menghapus data ke - 5
Posisi diluar jangkauan
Jumlah data ada : 4
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Negeri 5 Menara
Pengarang Buku : Ahmad Fuadi
Tahun Terbit Buku : 2006
Judul Buku : Bumi Manusia
Pengarang Buku : Framoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018
```

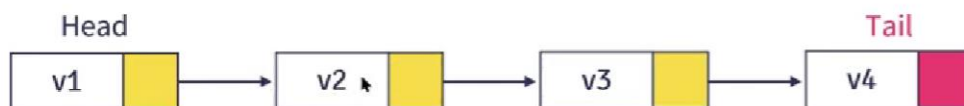
```
Mengubah data posisi tengah
Jumlah data ada : 4
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Sang Pemimpi
Pengarang Buku : Andrea Hirata
Tahun Terbit Buku : 2006
Judul Buku : Bumi Manusia
Pengarang Buku : Framoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018
```

### c) Penjelasan

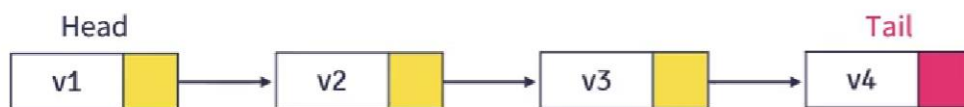
Single linked list non circular akan selalu tail node terakhir akan menghadap ke NULL.

Berikut penjelasan fungsinya.

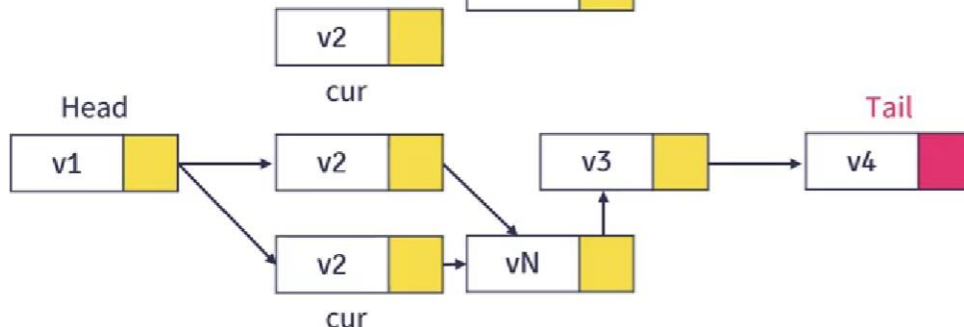
- 1) Create SingleLinkedList  
Membuat linked list sekalian meng set head ke node yang dibuat tadi
- 2) Print SingleLinkedList  
Cur di set ke head, lalu cur akan membaca Semua data di node dimana cur berada. Saat cur sampai di data pointer, maka akan berpindah ke Node selanjutnya. Selanjutnya akan selalu dibaca terus sampai posisi cur berada di akhir node (cur->tail=NULL)
- 3) Tambah Awal single Linked List
  - Membuat node baru
  - New node itu akan mengarahkan pointer ke head
  - Posisi head akan diset ke Node baru
- 4) Tambah akhir single linked list
  - Membuat node baru
  - tail next menunjuk node baru
  - node baru next menunjuk ke NULL
- 5) Tambah tengah single linked list
  - Mengecek dulu apakah Tambah posisi nodenya nya melebihi maksimal node atau tidak
 Jika bisa maka akan dilakukan operasi penambahan data Berikut:



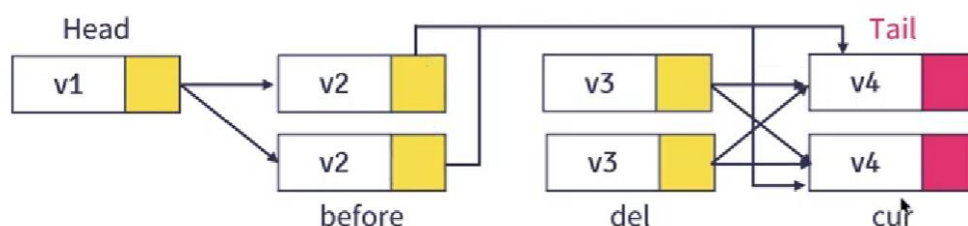
Tambah vN ke posisi 3



== loop (1 < posisi-1) ==>



Traversing:





6) Remove first

Menghapus node pertama

7) Remove last

Menghapus Node terakhir dan mengubah node terakhir -1 -> next menjadi NULL

8) Remove Middle

Menghapus Node tengah . Mengubah node next sebelum node yang dihapus ke menunjuk node yang dihapus selanjutnya

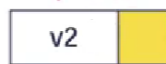


Hapus node ke-3



===== loop (1 <= posisi) =====>

== posisi - 1



before

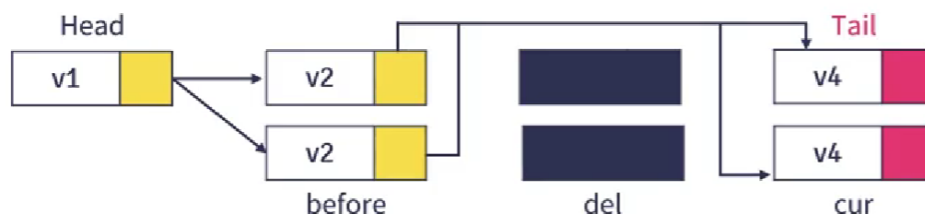
== posisi



del



cur



9) Ubah awal singleLinked List

Mengubah data single linked list urutan pertama

10) Ubah tengah single linked list

Mengubah data single linked list urutan tengah

3. Percobaan Latihan 1

a) Tampilan coding

```

1  #include <iostream>
2  using namespace std;
3
4  //deklarasi single linked list
5  struct Buku {
6
7      //komponen / member
8      string judul, pengarang;
9      int tahunTerbit;
10
11      Buku* next;
12
13  };
14
15  Buku* head, * tail, * cur, * newNode, * del, * before;
16
17  //create circular single linked list
18  void createCircularSingleLinkedList(string judul, string pengarang, int tB) {
19      head = new Buku();
20      head->judul = judul;
21      head->pengarang = pengarang;
22      head->tahunTerbit = tB;
23      tail = head;
24      tail->next = head;
25  }
26
27  //fungsi perhitungan node
28  int countSingleLinkedList() {
29      cur = head;
30      int jumlah = 0;
31      while (cur->next != head) {
32          jumlah++;
33          cur = cur->next;
34      }
35      jumlah++;
36      return jumlah;
37  }
38
39  //tambahAwal Circular Single linked list
40  void addFirst(string judul, string pengarang, int tB) {
41      newNode = new Buku();
42      newNode->judul = judul;
43      newNode->pengarang = pengarang;
44      newNode->tahunTerbit = tB;
45      newNode->next = head;
46      tail->next = newNode;
47      head = newNode;
48  }
49
50  //tambahAkhir Circular Single Linked list
51  void addLast(string judul, string pengarang, int tB) {
52      newNode = new Buku();
53      newNode->judul = judul;
54      newNode->pengarang = pengarang;
55      newNode->tahunTerbit = tB;
56      newNode->next = head;
57      tail->next = newNode;
58      tail = newNode;
59  }

```

```

60
61 //tambah node tengah single linked list
62 void addMiddle(string judul, string pengarang, int tB, int posisi) {
63     if (posisi < 1 || posisi > countSingleLinkedList()) {
64         cout << "Posisi diluar jangkauan" << endl;
65     }
66     else if (posisi == 1) {
67         cout << "Posisi bukan posisi tengah" << endl;
68     }
69     else {
70         newNode = new Buku();
71         newNode->judul = judul;
72         newNode->pengarang = pengarang;
73         newNode->tahunTerbit = tB;
74
75
76         //tranversing
77         cur = head;
78         int nomor = 1;
79         while (nomor < posisi - 1) {
80             cur = cur->next;
81             nomor++;
82         }
83         newNode->next = cur->next;
84         cur->next = newNode;
85     }
86 }
87
88 //remove First
89 void removeFirst() {
90     del = head;
91     head = head->next;
92     tail->next = head;
93     delete del;
94 }
95
96 //remove Last
97 void removeLast() {
98     del = tail;
99     cur = head;
100     while (cur->next != tail) {
101         cur = cur->next;
102     }
103     tail = cur;
104     tail->next = head;
105     delete del;
106 }
107
108 //remove middle
109 void removeMiddle(int posisi) {
110     if (posisi < 1 || posisi > countSingleLinkedList()) {
111         cout << "Posisi diluar jangkauan" << endl;
112     }
113     else if (posisi == 1) {
114         cout << "Posisi bukan posisi tengah" << endl;
115     }
116     else {
117         int nomor = 1;
118         cur = head;
119         while (nomor <= posisi) {
120             if (nomor == posisi - 1) {
121                 before = cur;
122             }
123             if (nomor == posisi) {
124                 del = cur;
125             }
126             cur = cur->next;
127             nomor++;
128         }
129         before->next = cur;
130         delete del;
131     }
132 }
133

```

```

134 //ubah awalan node circular single linked list
135 void changeFirst(string judul, string pengarang, int tB) {
136     head->judul = judul;
137     head->pengarang = pengarang;
138     head->tahunTerbit = tB;
139 }
140
141
142 //ubahAkhir circular single linked list
143 void changeLast(string judul, string pengarang, int tB) {
144     tail->judul = judul;
145     tail->pengarang = pengarang;
146     tail->tahunTerbit = tB;
147 }
148
149 //ubah Tengah Circular Single linked list
150 void changeMiddle(string judul, string pengarang, int tB, int posisi) {
151     if (posisi < 1 || posisi == countSingleLinkedList()) {
152         cout << "Posisi diluar jangkauan" << "\n\n" << endl;;
153     }
154     else if (posisi == 1 || posisi == countSingleLinkedList()) {
155         cout << "Posisi bukan posisi tengah" << endl;
156     }
157     else {
158         cur = head;
159         int nomor = 1;
160         while (nomor < posisi) {
161             cur = cur->next;
162             nomor++;
163         }
164         cur->judul = judul;
165         cur->pengarang = pengarang;
166         cur->tahunTerbit = tB;
167     }
168 }
169
170 //print Circular single Linkedlist
171 void printCircularSingleLinkedList() {
172     cout << "Jumlah data ada : " << countSingleLinkedList() << endl;
173     cur = head;
174     while (cur->next != head) {
175         cout << "Judul Buku : " << cur->judul << endl;
176         cout << "Pengarang Buku : " << cur->pengarang << endl;
177         cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
178
179         cur = cur->next;
180     }
181     cout << "Judul Buku : " << cur->judul << endl;
182     cout << "Pengarang Buku : " << cur->pengarang << endl;
183     cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
184 }
185
186

```

```

187 int main() {
188
189     cout << "Nama : Wijayanto Agung Wibowo" << endl;
190     cout << "Nim : 22.11.4552" << endl << endl;
191
192
193     cout << "Membuat Circular single linkedlist" << endl;
194     createCircularSingleLinkedList("Kata", "Geez & Aan", 2018);
195
196     printCircularSingleLinkedList();
197
198     cout << "\n\n" << endl;
199
200     cout << "Memasukan data ke posisi pertama" << endl;
201
202     addFirst("Dia adalah Kakakku", "Tere Liye", 2009);
203
204     printCircularSingleLinkedList();
205
206     cout << "\n\n" << endl;
207
208     cout << "Memasukan data ke posisi terakhir" << endl;
209
210     addLast("Aroma Karsa", "Dee Lestari", 2018);
211
212     printCircularSingleLinkedList();
213
214     cout << "\n\n" << endl;
215
216     cout << "Menghapus data pertama" << endl;
217
218     removeFirst();
219
220     printCircularSingleLinkedList();
221
222     cout << "\n\n" << endl;
223
224     cout << "Memasukan data terakhir" << endl;
225
226     addLast("11.11", "Fierza Besari", 2018);
227
228     printCircularSingleLinkedList();
229
230     cout << "\n\n" << endl;

```

```

231
232     cout << "Menghapus data terakhir" << endl;
233
234     removeLast();
235
236     printCircularSingleLinkedList();
237
238     cout << "\n\n" << endl;
239
240     cout << "Mengubah data pertama" << endl;
241
242     changeFirst("Berhenti di Kamu", "Gia Pratama", 2018);
243
244     printCircularSingleLinkedList();
245
246     cout << "\n\n" << endl;
247
248     cout << "Menambah data ke posisi tengah" << endl;
249
250     addMiddle("Bumi Manusia", "Framoedya Anata Toer", 2005, 2);
251
252     printCircularSingleLinkedList();
253
254     cout << "\n\n" << endl;
255
256     cout << "Menambah data ke posisi tengah" << endl;
257
258     addMiddle("Negeri 5 Menara", "Ahmad Fuadi", 2006, 2);
259
260     printCircularSingleLinkedList();
261
262     cout << "\n\n" << endl;
263
264     cout << "Menghapus data ke - 5" << endl;
265
266     removeMiddle(5);
267
268     printCircularSingleLinkedList();
269
270     cout << "\n\n" << endl;
271
272     cout << "Mengubah data posisi tengah" << endl;
273
274     changeMiddle("Sang Pemimpi", "Andrea Hirata", 2006, 2);
275
276     printCircularSingleLinkedList();
277
278     cout << "\n\n" << endl;
279
280 }
281

```

b) Tampilan Hasil Runing

Nama : Wijayanto Agung Wibowo  
Nim : 22.11.4552

Membuat Circular single linkedlist  
Jumlah data ada : 1  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018

Memasukan data ke posisi pertama  
Jumlah data ada : 2  
Judul Buku : Dia adalah Kakakku  
Pengarang Buku : Tere Live  
Tahun Terbit Buku : 2009  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018

Memasukan data ke posisi terakhir  
Jumlah data ada : 3  
Judul Buku : Dia adalah Kakakku  
Pengarang Buku : Tere Live  
Tahun Terbit Buku : 2009  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Menghapus data pertama  
Jumlah data ada : 2  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Memasukan data terakhir  
Jumlah data ada : 3  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018  
Judul Buku : 11.11  
Pengarang Buku : Fierza Besari  
Tahun Terbit Buku : 2018

Menghapus data terakhir  
Jumlah data ada : 2  
Judul Buku : Kata  
Pengarang Buku : Geez & Aan  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Mengubah data pertama  
Jumlah data ada : 2  
Judul Buku : Berhenti di Kamu  
Pengarang Buku : Gia Pratama  
Tahun Terbit Buku : 2018  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Menambah data ke posisi tengah  
Jumlah data ada : 3  
Judul Buku : Berhenti di Kamu  
Pengarang Buku : Gia Pratama  
Tahun Terbit Buku : 2018  
Judul Buku : Bumi Manusia  
Pengarang Buku : Framoedya Anata Toer  
Tahun Terbit Buku : 2005  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

Menambah data ke posisi tengah  
Jumlah data ada : 4  
Judul Buku : Berhenti di Kamu  
Pengarang Buku : Gia Pratama  
Tahun Terbit Buku : 2018  
Judul Buku : Negeri 5 Menara  
Pengarang Buku : Ahmad Fuadi  
Tahun Terbit Buku : 2006  
Judul Buku : Bumi Manusia  
Pengarang Buku : Framoedya Anata Toer  
Tahun Terbit Buku : 2005  
Judul Buku : Aroma Karsa  
Pengarang Buku : Dee Lestari  
Tahun Terbit Buku : 2018

```

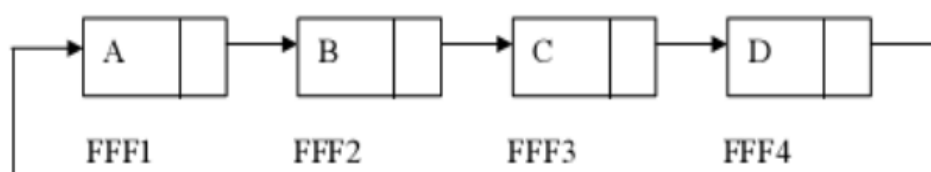
Menghapus data ke - 5
Posisi diluar jangkauan
Jumlah data ada : 4
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Negeri 5 Menara
Pengarang Buku : Ahmad Fuadi
Tahun Terbit Buku : 2006
Judul Buku : Bumi Manusia
Pengarang Buku : Franoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018

Mengubah data posisi tengah
Jumlah data ada : 4
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Sang Pemimpi
Pengarang Buku : Andrea Hirata
Tahun Terbit Buku : 2006
Judul Buku : Bumi Manusia
Pengarang Buku : Franoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018

```

### c) Penjelasan

Single Linked List Circular adalah Single Linked List yang pointer nextnya menunjuk pada dirinya sendiri. Ilustrasinya sebagai berikut:



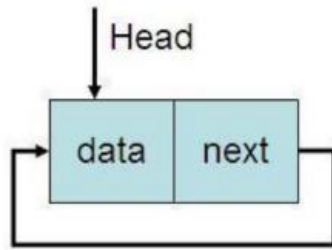
- Setiap node pada linked list mempunyai field yang berisi pointer ke node berikutnya, dan juga memiliki field yang berisi data.
- Pada akhir linked list, node terakhir akan menunjuk ke node terdepan sehingga linked list tersebut berputar.
- Node terakhir akan menunjuk lagi ke head

Penjelasan fungsi dari coding, Sebagai Berikut:

#### 1) Create circular Single Linkedlist

Membuat node baru dimana node awal ini akan menjadi head dan tail, tailnya akan menunjuk head





2) Count Single linked list

Fungsi yang akan mereturn jumlah dari keseluruhan Node dari node awal(head) ke node akhir(tail)

3) Menambah data depan / Menyisipkan Node di awal.

Langkah algoritmanya yaitu:

- Membuat newNode berisi data baru
- Newnode->next mengarah ke head
- tail->next mengarah ke newNode
- head di set ke newNode

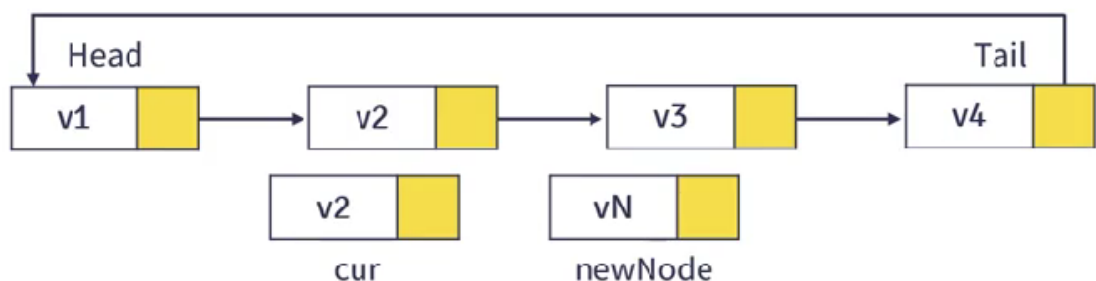
4) Menambah data belakang

- Membuat newNode berisi data baru
- newNode->next nya mengarah ke head
- tail->next mengarah ke newNode
- tail di set ke newNode

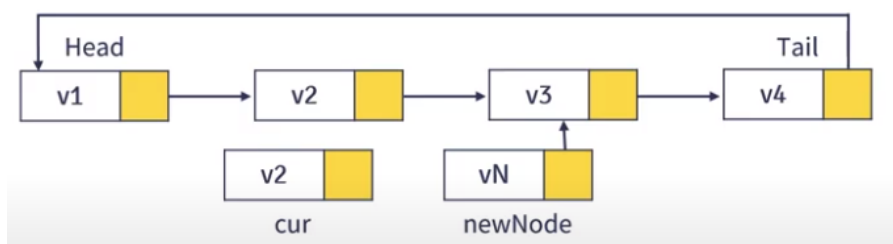
5) Menambah Node tengah Circular Single Linked list

-Mengecek dahulu apakah posisi data yang dimasukan melebihi atau kurang dari jumlah node atau tidak. Jika posisi nilai data melebihi jumlah node maka tidak bisa

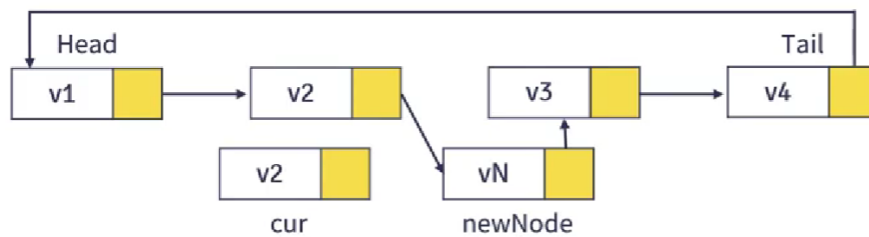
- Membuat newNode berisi data baru



-newNode-next akan mengarah ke posisi cur->next



-cur->next diubah ke newNode



#### 6) Remove first

Menghapus node pertama

- Del sama dengan head
- Head dipindah ke posisi head-> next
- Tail->next di set ke head
- Delete del

#### 7) Remove last

Menghapus Node terakhir dan mengubah node terakhir -1 -> next menjadi menunjuk head

- Del sama dengan tail
- Set cur ke node sebelum tail
- Tail di set ke cur
- Tail->next mengarah ke head
- Delet del

#### 8) Remove Middle

Menghapus node yang berada di antara node head dan node tail

- Pengkondisian bahwa posisi yang akan di hapus itu lebih dari posisi head(node pertama) dan kurang dari tail (node terakhir)
- Node before diarahkan ke posisi-1
- Node del diarahkan ke posisi yang ingin dihapus
- Before->next diarahkan ke cur->next( urutan node setelah posisi)
- Delete node del

#### 9) Ubah awalan node

Mengubah node awal

#### 10) Change last node

Mengubah node terakhir

#### 11) Change middle

Mengubah node di antara node head dan node tail

#### 12) Print circular node

Mencetak Semua node dari head sampai tail

#### 4. Percobaan Latihan 2

Menambahkan fungsi pencarian untuk judul buku dan nama pengarang di single linked list circular

##### a) Tampilan coding

```
//Mencari judul buku
void searchTitle(string title) {
    cout << "Mencari judul buku: " << title << endl;
    cur = head;
    bool hasil = 0;
    while (cur->next != head) {
        if (cur->judul == title) {
            cout << "Hasil >> " << endl;
            cout << "Judul Buku : " << cur->judul << endl;
            cout << "Pengarang Buku : " << cur->pengarang << endl;
            cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
            hasil = 1;
        }
        cur = cur->next;
    }
    if (cur->judul == title) {
        cout << "Judul Buku : " << cur->judul << endl;
        cout << "Pengarang Buku : " << cur->pengarang << endl;
        cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
    }
    else if (hasil == 0) {
        cout << "Hasil : Judul buku tidak ditemukan";
    }
}
```

```
//mencari judul buku berdasarkan pengarang
void searchPengarang(string namaPengarang) {
    cout << "Mencari buku nama pengarang: " << namaPengarang << endl;
    cur = head;
    bool hasil = 0;
    while (cur->next != head) {
        if (cur->pengarang == namaPengarang) {
            cout << "Hasil >> " << endl;
            cout << "Judul Buku : " << cur->judul << endl;
            cout << "Pengarang Buku : " << cur->pengarang << endl;
            cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
            hasil = 1;
        }
        cur = cur->next;
    }
    if (cur->pengarang == namaPengarang) {
        cout << "Judul Buku : " << cur->judul << endl;
        cout << "Pengarang Buku : " << cur->pengarang << endl;
        cout << "Tahun Terbit Buku : " << cur->tahunTerbit << endl;
    }
    else if (hasil == 0) {
        cout << "Hasil : Judul buku tidak ditemukan";
    }
}
```

b) Hasil Running

```
Mencari judul buku: Sang Pemimpi
Hasil >>
Judul Buku : Sang Pemimpi
Pengarang Buku : Andrea Hirata
Tahun Terbit Buku : 2006

Mencari buku nama pengarang: Framoedya Anata Toer
Hasil >>
Judul Buku : Bumi Manusia
Pengarang Buku : Framoedya Anata Toer
Tahun Terbit Buku : 2005
```

c) Penjelasan

Diatas adalah code algoritma untuk mencari nilai data dalam node.

-Mulai node pertama sebagai head

- melakukan perulangan while terus menerus sampai bertemu head

- Jika data pada node saat ini cocok dengan data yang diinput, maka data di node tersebut akan ditampilkan.

-Jika data pada node saat ini tidak cocok dengan data yang diinput, pindahkan pointer ke node berikutnya dalam linked list dan lanjutkan perulangan.

5. Latihan 3

Membuat fungsi untuk menyortir buku berdasarkan tahun terbit

a) Tampilan coding

```

//menyortir tahun terbit buku
void sorting() {
    int n = countSingleLinkedList();

    for (int i = 0; i < n-1; i++) {
        cur = head;
        for (int j = i+1; j < n; j++) {
            if (cur->tahunTerbit > cur->next->tahunTerbit) {
                string tempJudul = cur->judul;
                string tempPengarang = cur->pengarang;
                int tempTahunTerbit = cur->tahunTerbit;

                cur->judul = cur->next->judul;
                cur->pengarang = cur->next->pengarang;
                cur->tahunTerbit = cur->next->tahunTerbit;

                cur->next->judul = tempJudul;
                cur->next->pengarang = tempPengarang;
                cur->next->tahunTerbit = tempTahunTerbit;
            }
            cur = cur->next;
        }
    }
}

```

Dalam fungsi main();

```

sorting();

cout << "Data setelah di urutkan ke tahun terbit buku >> " <<endl;

printCircularSingleLinkedList();

```

b) Tampilan hasil running

```

Data setelah di urutkan ke tahun terbit buku >>
Jumlah data ada : 4
Judul Buku : Bumi Manusia
Pengarang Buku : Franoedya Anata Toer
Tahun Terbit Buku : 2005
Judul Buku : Sang Pemimpi
Pengarang Buku : Andrea Hirata
Tahun Terbit Buku : 2006
Judul Buku : Berhenti di Kamu
Pengarang Buku : Gia Pratama
Tahun Terbit Buku : 2018
Judul Buku : Aroma Karsa
Pengarang Buku : Dee Lestari
Tahun Terbit Buku : 2018

```

### c) Penjelasan

Untuk mencari buku berdasarkan judul dan pengarang pada linked list, di tambahkan fungsi sorting.

Fungsi sorting berguna untuk menyortir buku berdasarkan tahun terbit buku.

### C. Kesimpulan

Setelah melakukan percobaan pada Latihan 1 dst saya dapat memahami Bahwa linked list itu ada dua.

Penggunaan linked list sendiri mempunyai berbagai macam kelebihan melebihi array.