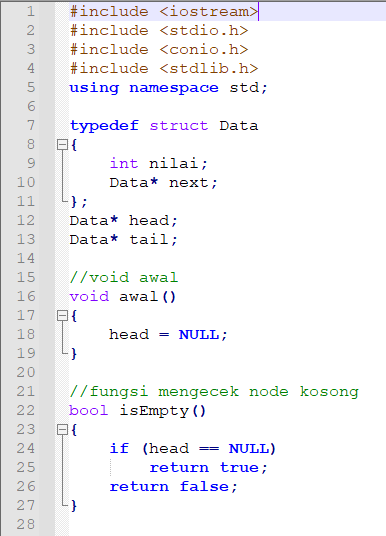
LAPORAN PRAKTIKUM

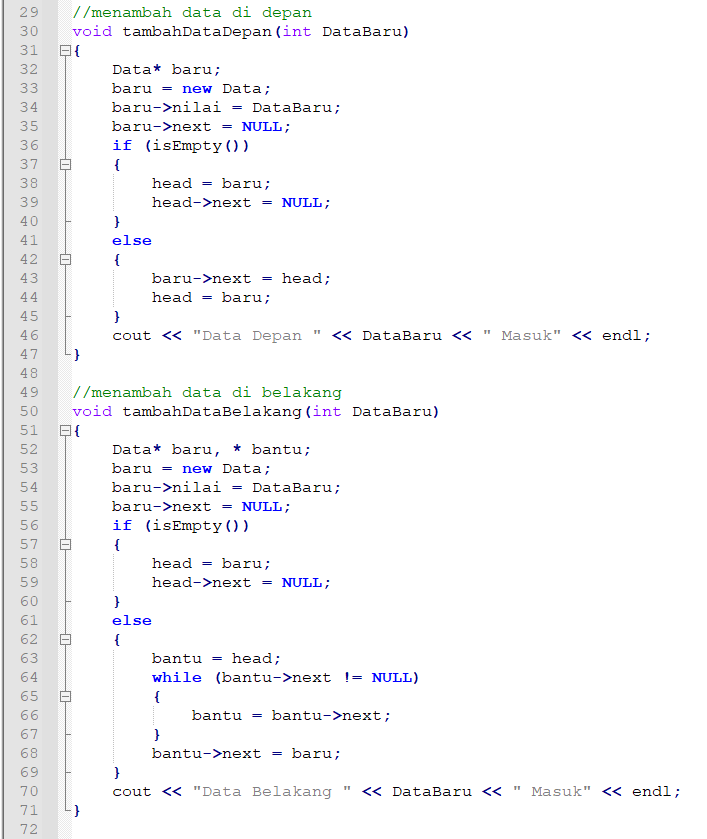
|  |  |
| --- | --- |
| Matakuliah | Struktur Data |
| Pertemuan ke | 8 |
| Nama Praktikan | Wijayanto Agung Wibowo |
| NIM | 22.11.4552 |
| NILAI (diisi oleh dosen / asisten praktikum) |  |

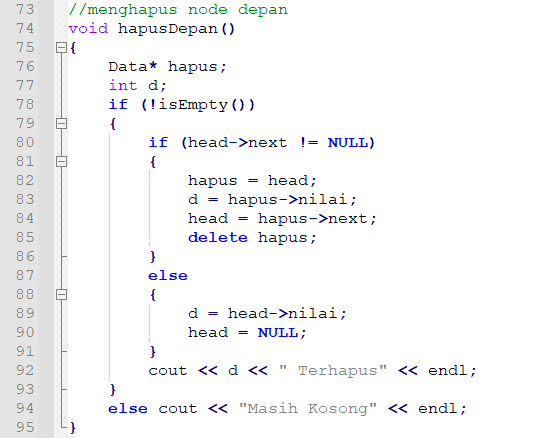
1. Tujuan

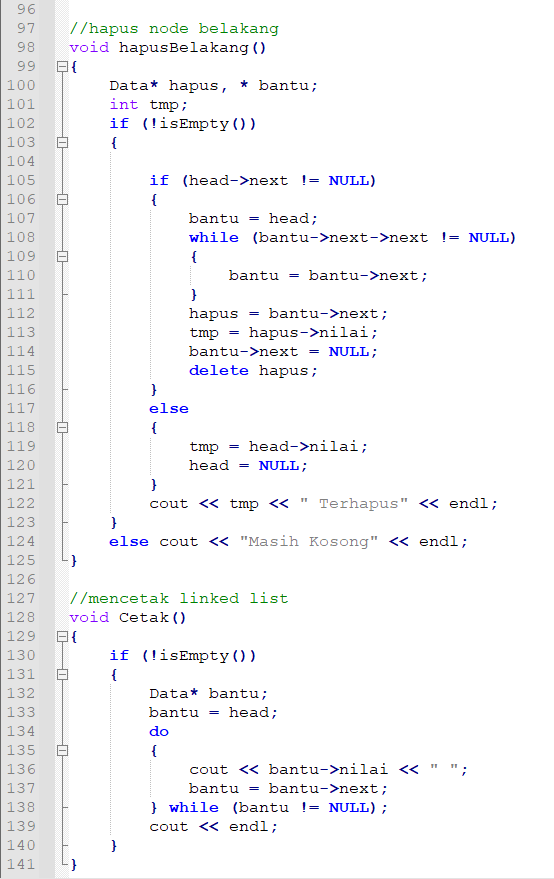
Setelah praktikum ini, praktikan diharapkan dapat:

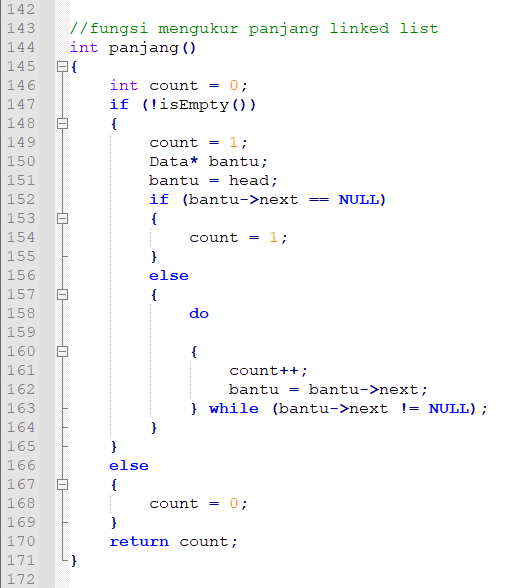
1. Memahami tipe data dasar
2. Memahami tipe data bentukan
3. Stuktur Program menggunakan bahasa C++ Praktikum Linked list Non circular dan Circular
4. Hasil Percobaan
   1. Percobaan Praktikum 1
5. Tampilan Coding

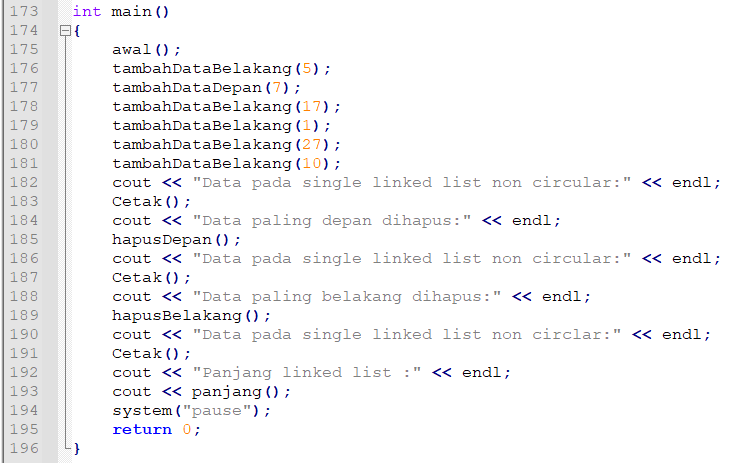




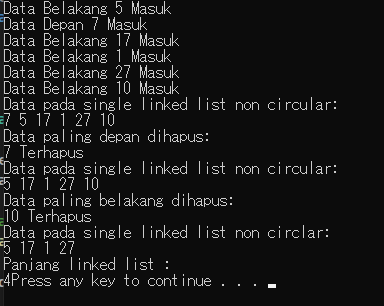








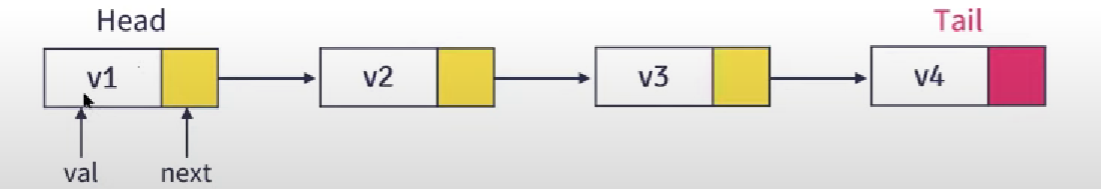
1. Hasil Running



1. 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

1. Bool is Empty

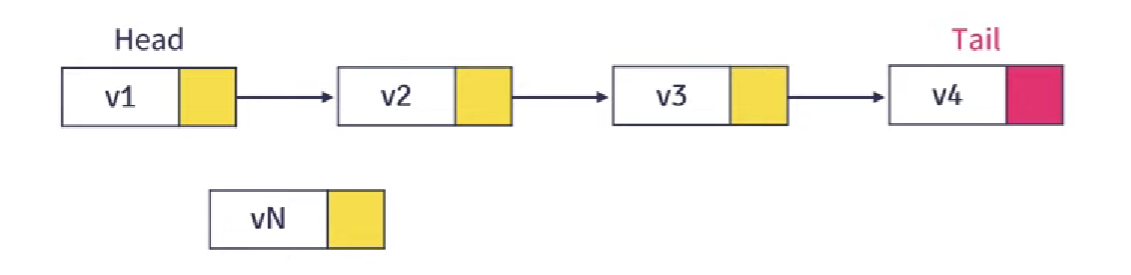
Fungsi untuk mengecek node berisi data kosong atau tidak. Algoritma nya akan return true bila data kosong dan false bila data sudah ada isinya.

Apabila data kosong, maka akan membuat node baru ini akan menjadi head di setiap

1. Menambah data depan / Menyisipkan Node di awal.

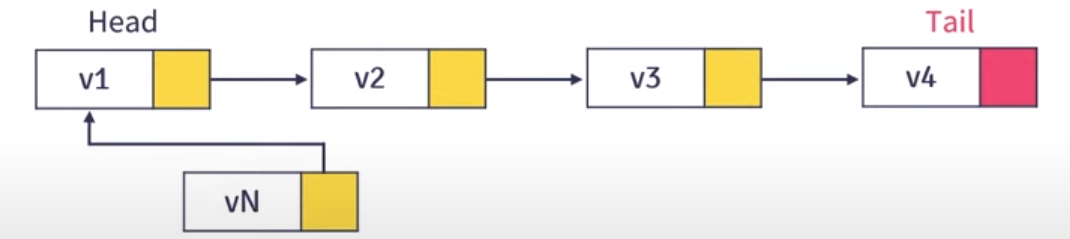
Langkah algoritmanya yaitu:

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



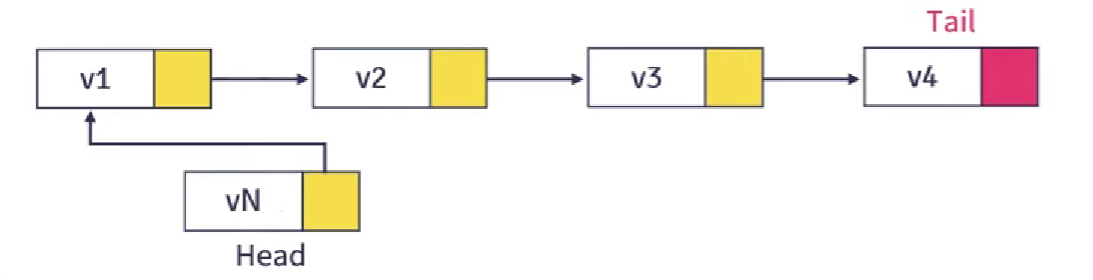
-Mengecek, Bila ada data maka:

Node Vn menunjuk ke Head



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

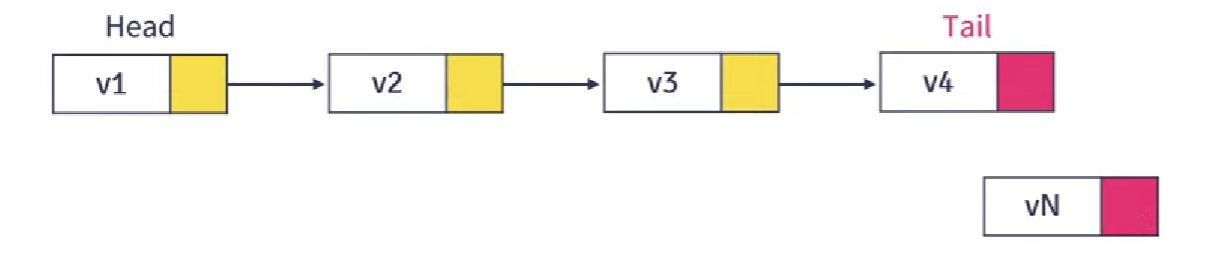
-Vn menjadi head



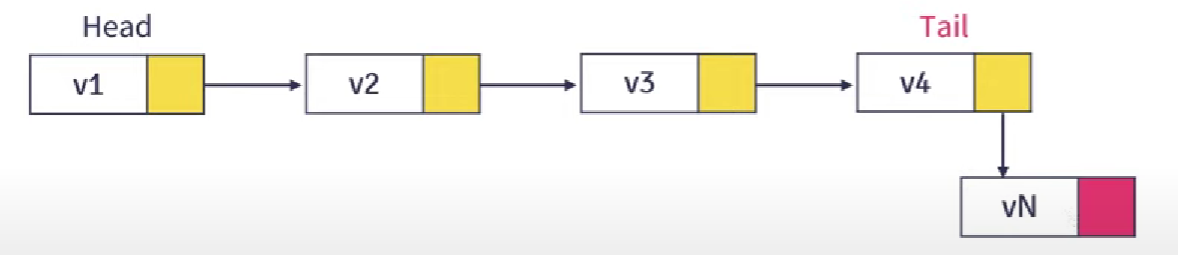
1. Menambah data belakang / Menyisipkan Node di akhir

Langkah Algoritmanya yaitu:

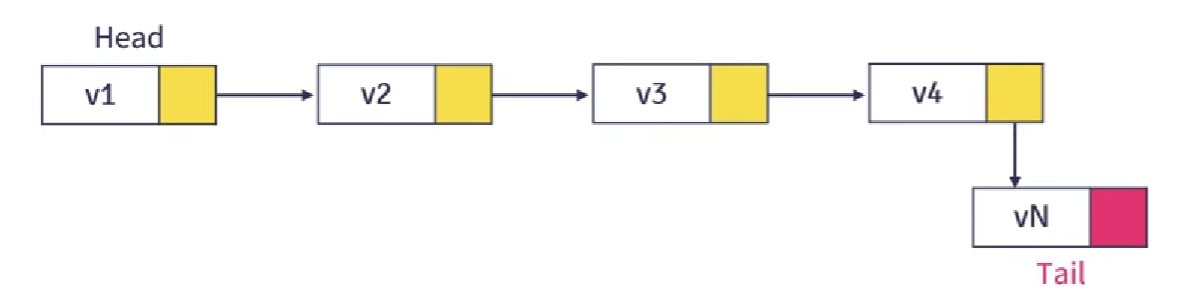
-Membuat Node baru dengan data baru (Vn)



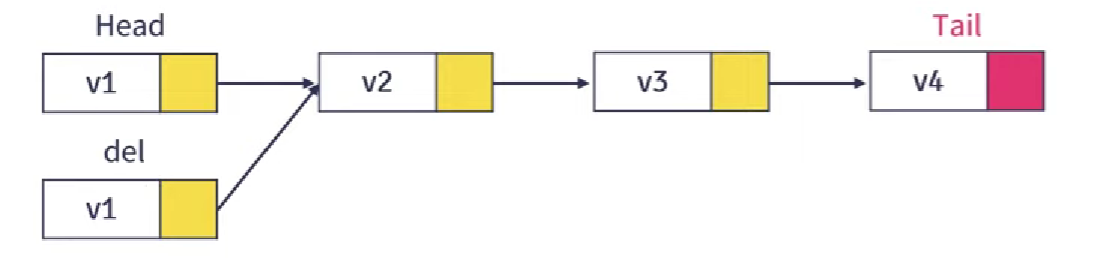
* Tail node terakhir , set ke node baru

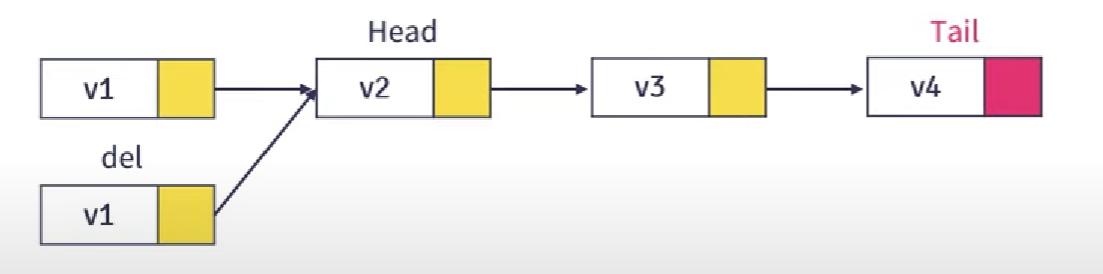


* Tail node baru diset menjadi NULL

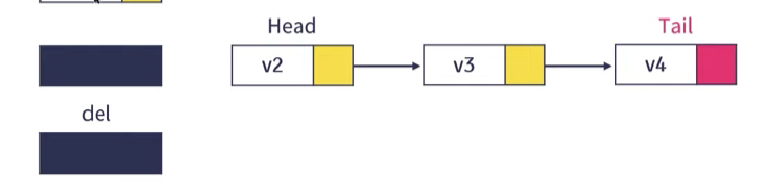


1. Menghapus node depan

* 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



1. 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

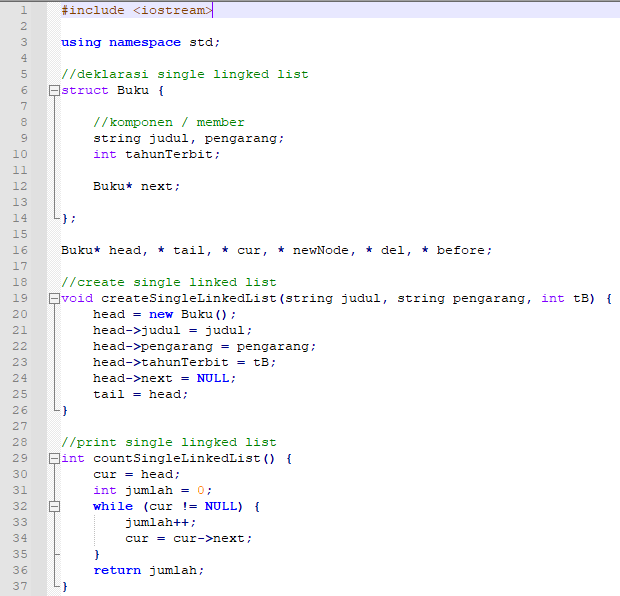
1. Mencetak linked list

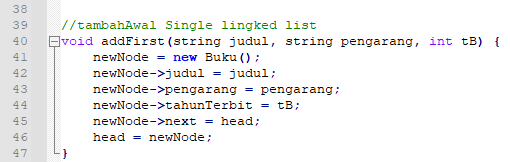
* Membuat node bantu
* Head akan diubah ke node bantu
* Head ini akan berubah terus dan mencetak terus nilai dari node nya, dan akan berjalan berantai terus sampai bertemu kondisi dimana head -> tail = NULL.

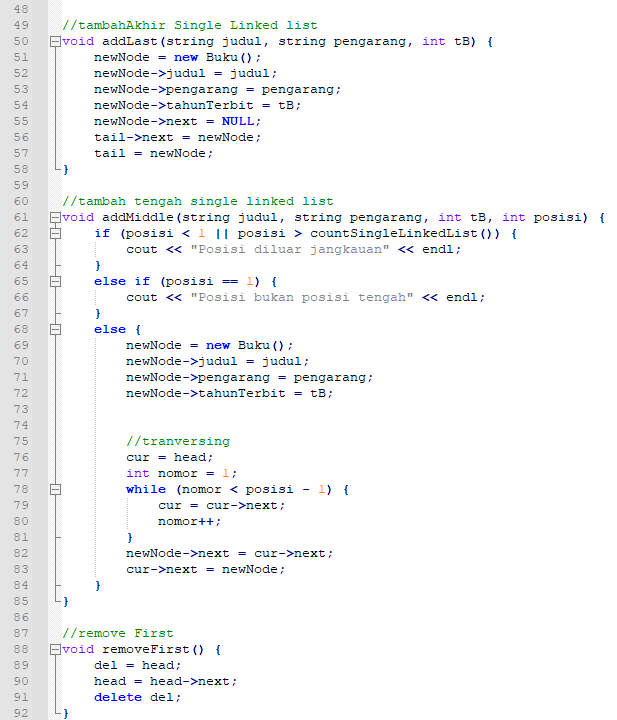
1. Menghitung Panjang node

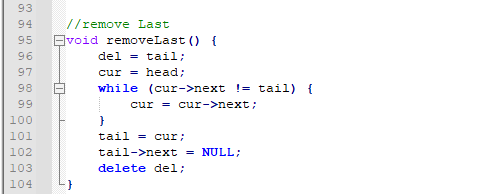
* Membuat node bantu
* Head akan diubah ke node bantu
* Head ini akan terus menambahkan nilai dari count sejalannjya node berantai terus sampai bertemu kondisi head->tail = NULL
* Return nilai dari count.
  1. Studi kasus

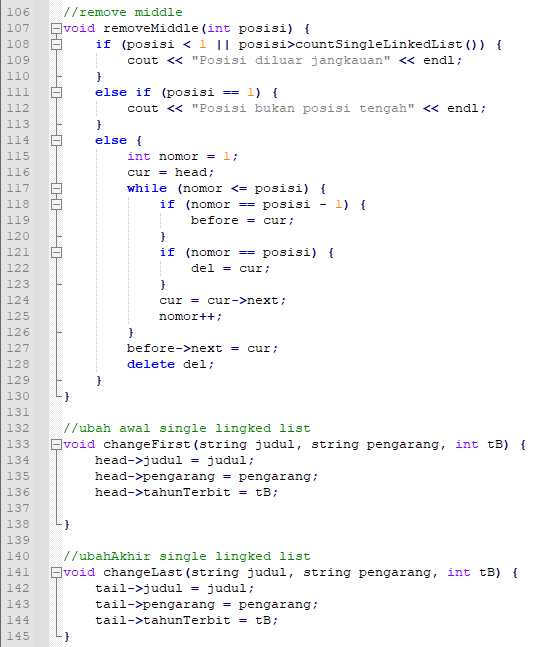
1. Tampilan Coding

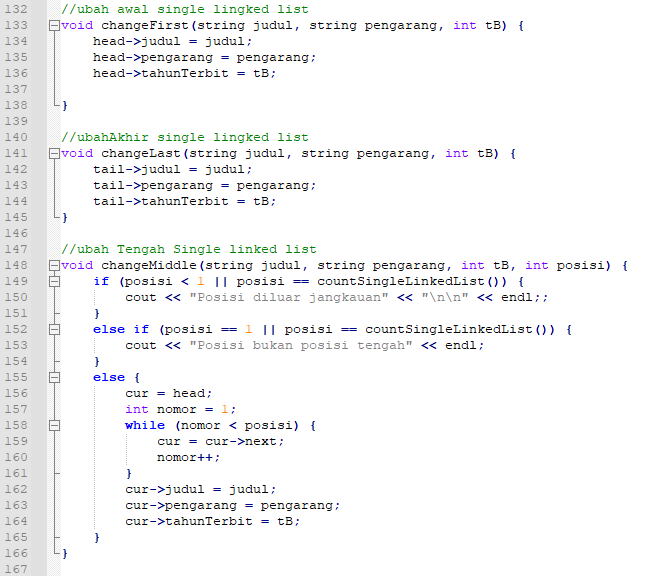


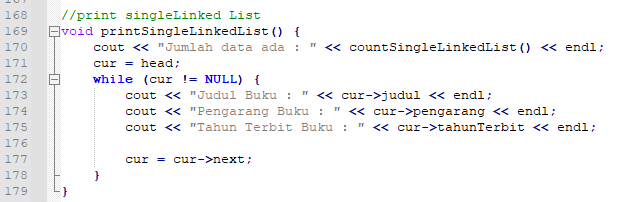


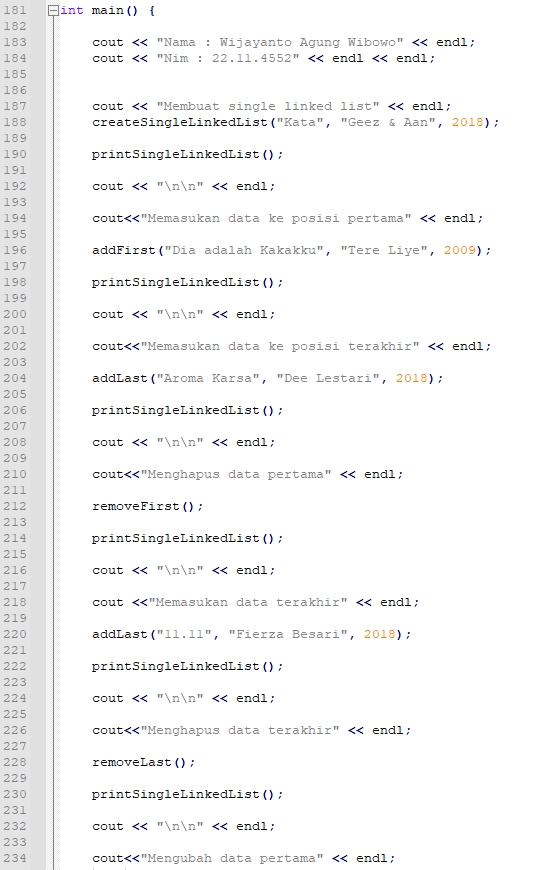


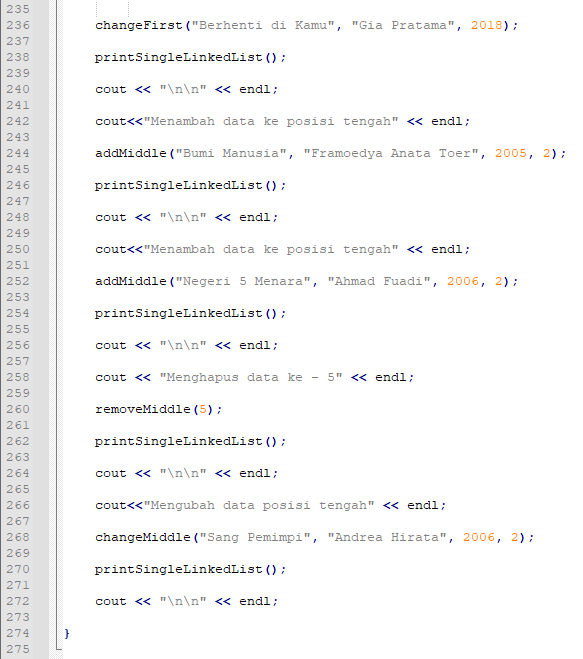




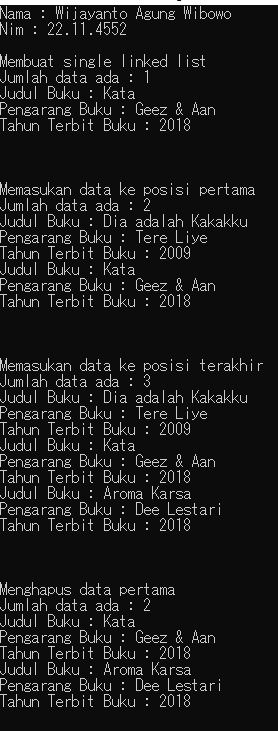
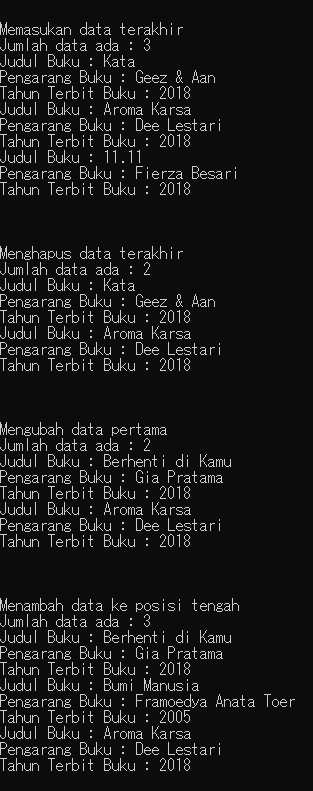


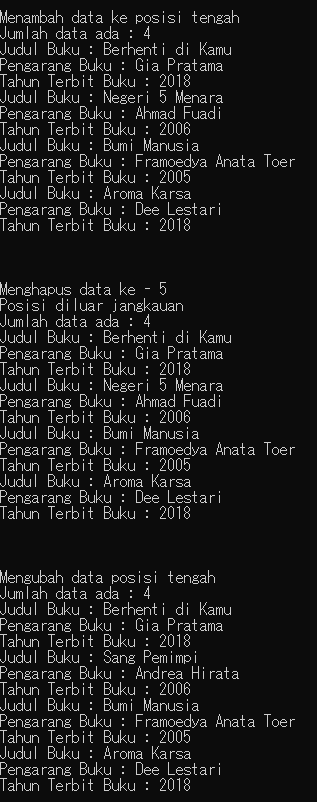






1. Hasil Running



1. 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

1. 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)

1. 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

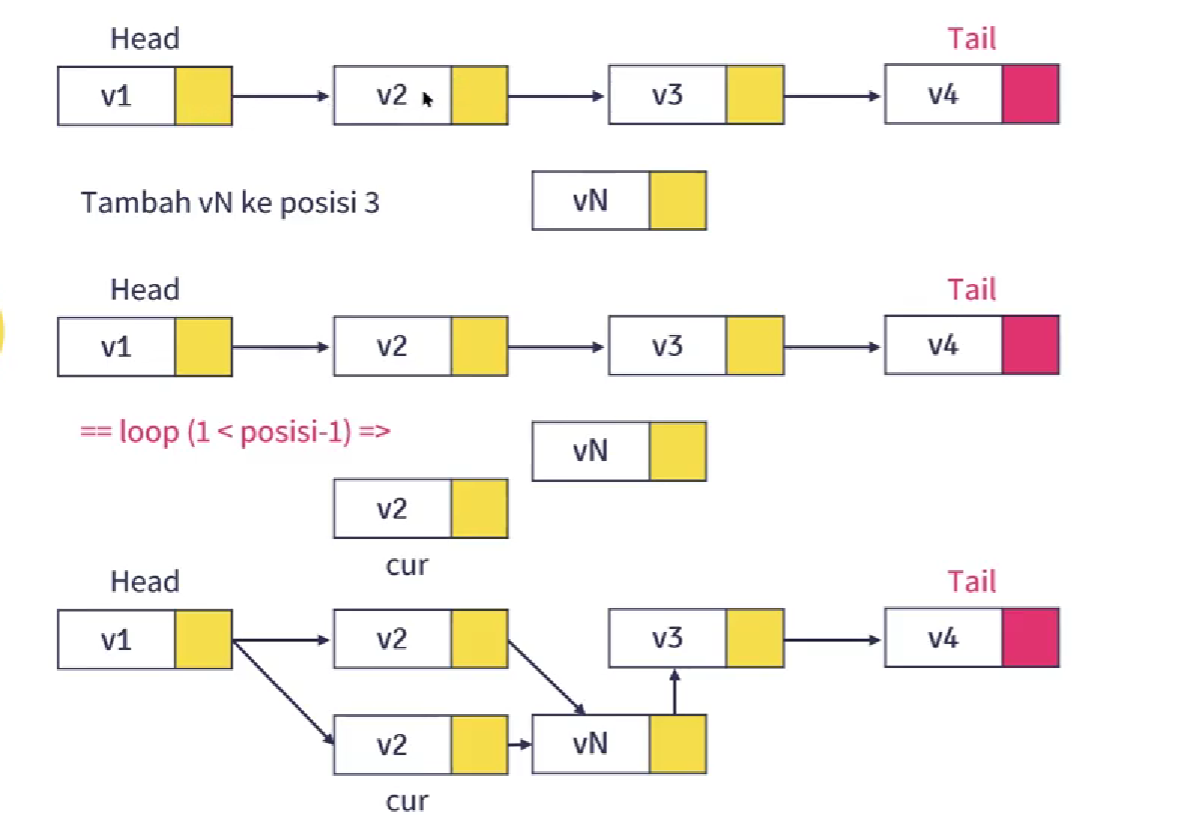
- tail next menunjuk node baru

- node baru next menunjuk ke NULL

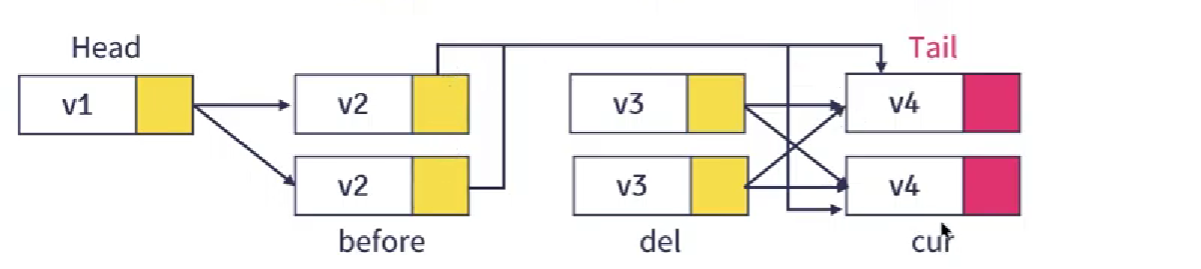
5) Tambah tengah single linked list

- Mengecek dulu apakah Tambah posisi nodenya nya melebihi maksimal node atau tidak

Jika bisa maka akan dilakukan operasi penambahan data Berikut:



Tranversing:



6) Remove first

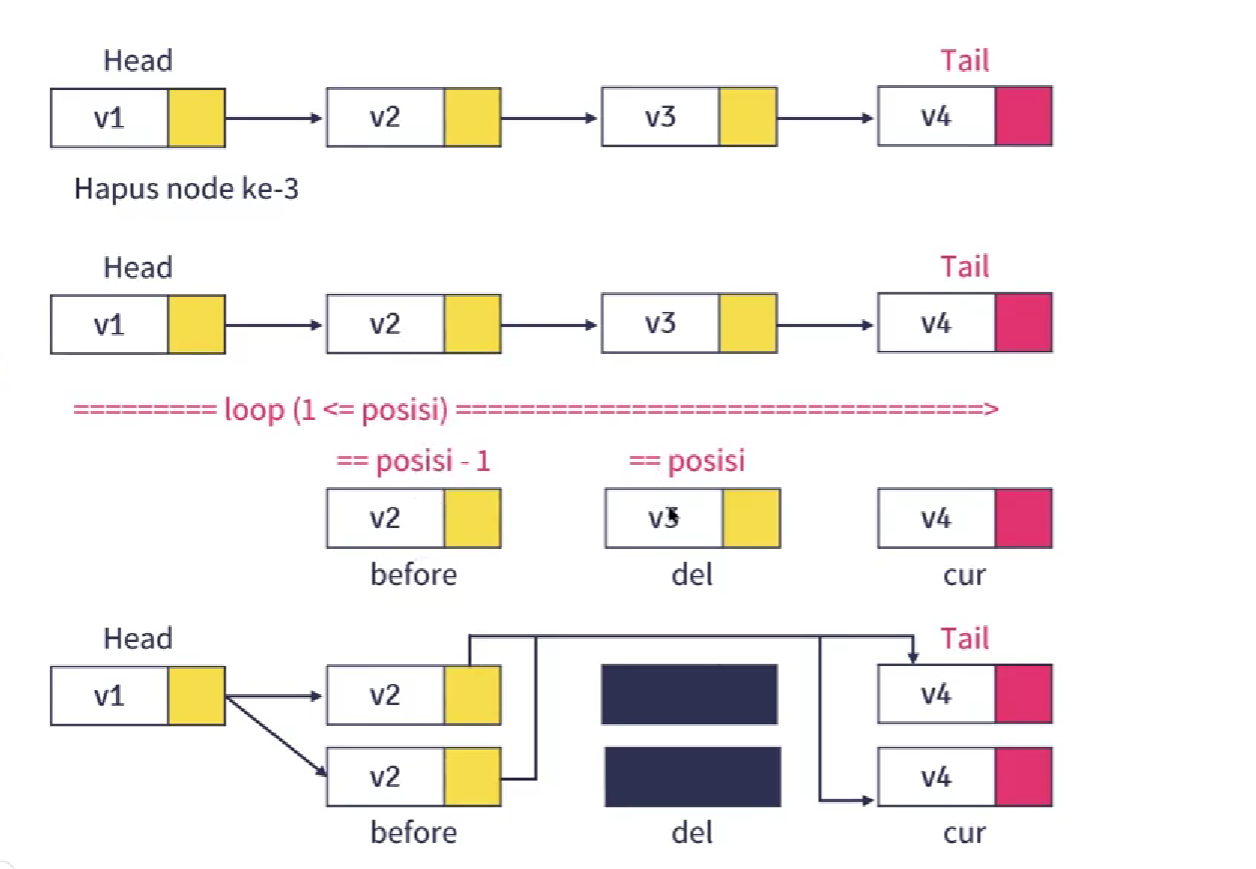
Menghapus node pertama

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 dihabus ke menunjuk node yang dihapus selanjutnya



1. Ubah awal singleLinked List

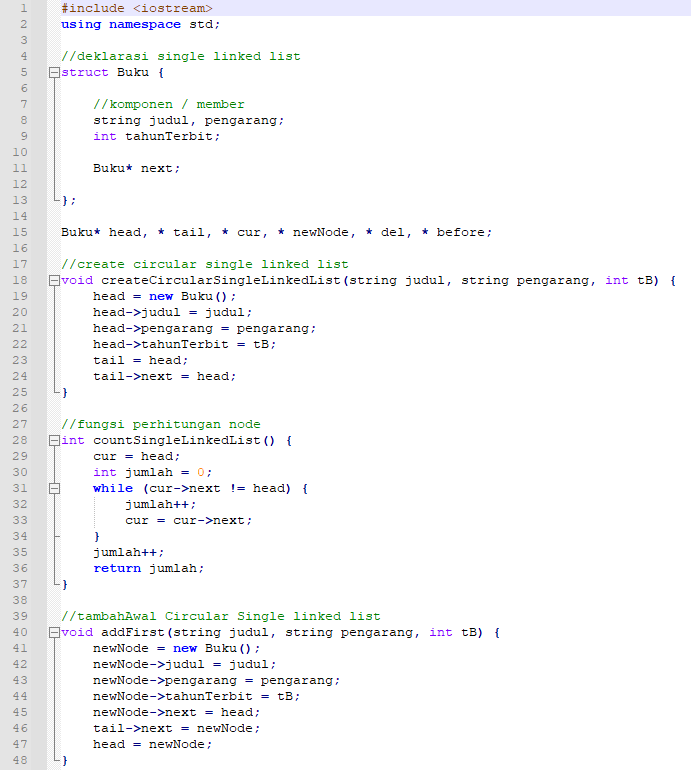
Mengubah data single linked list urutan pertama

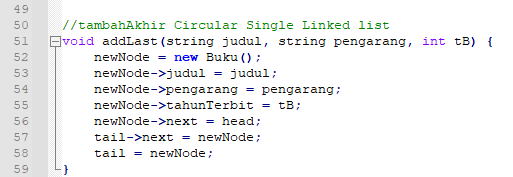
1. Ubah tengah single linked list

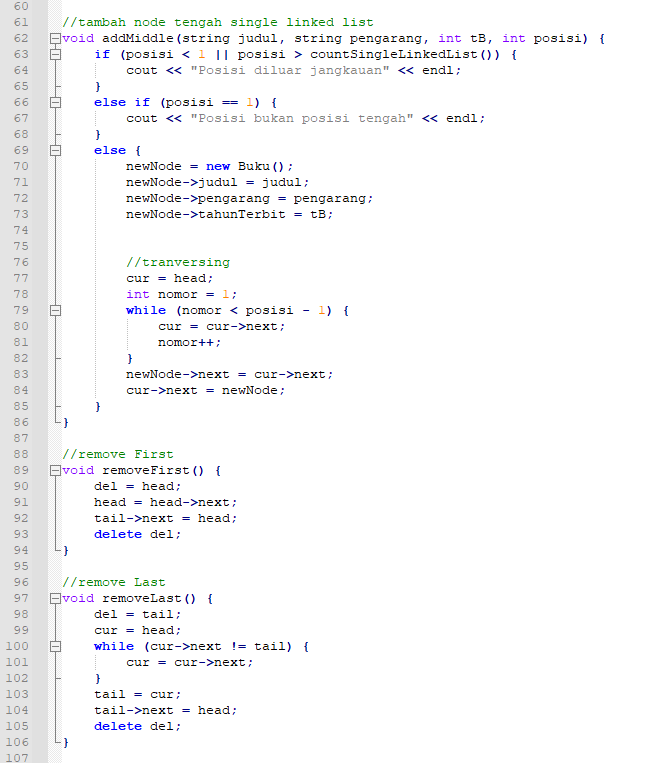
Mengubah data single linked list urutan tengah

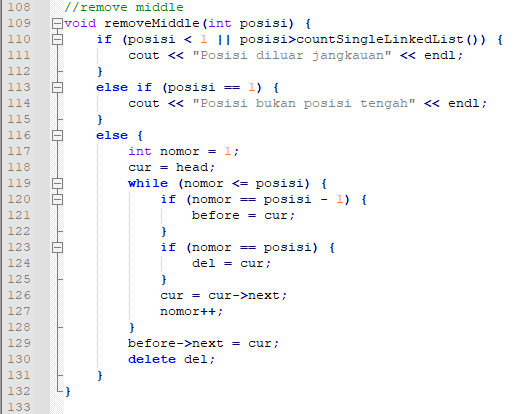
* 1. Percobaan Latihan 1

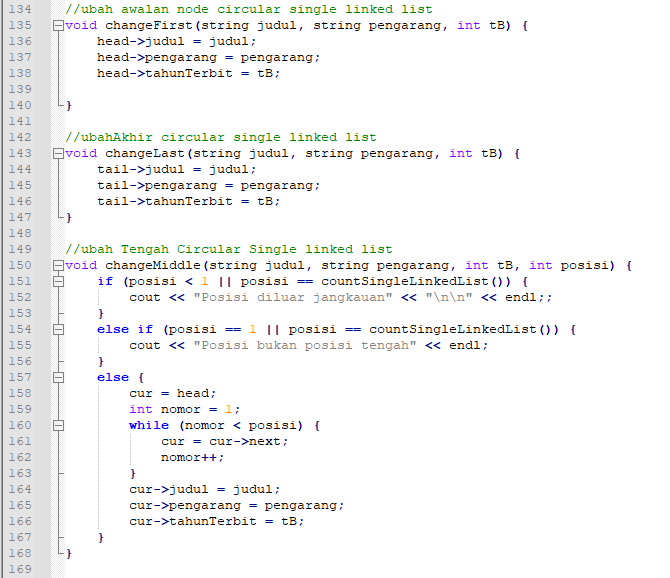
1. Tampilan coding

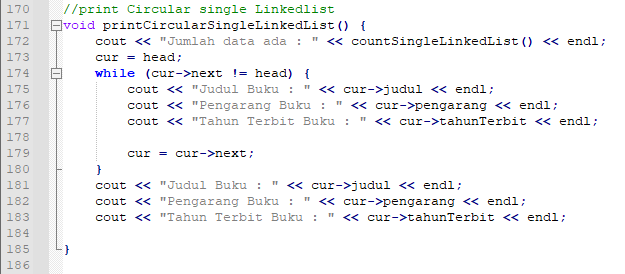


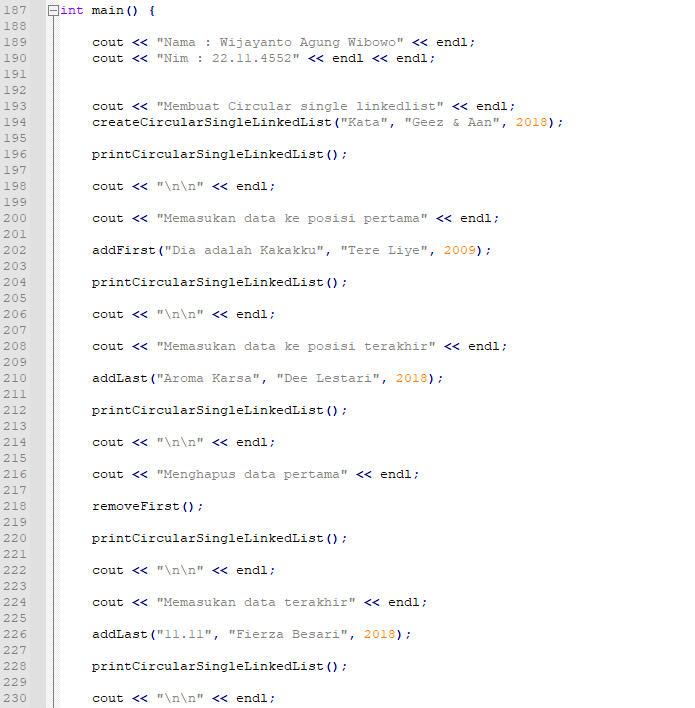


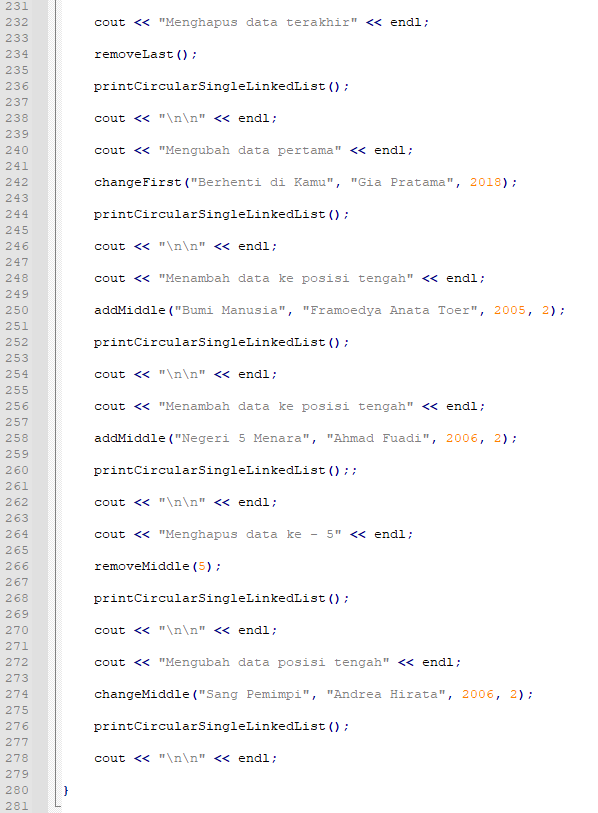




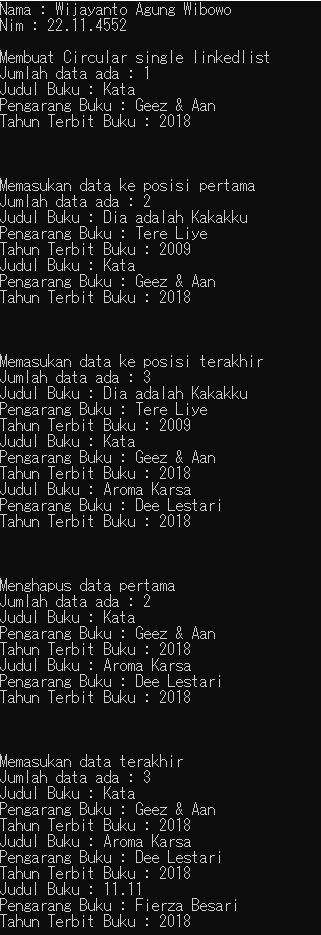
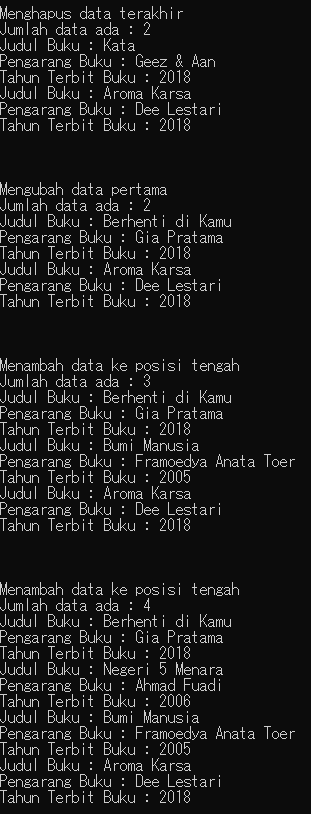


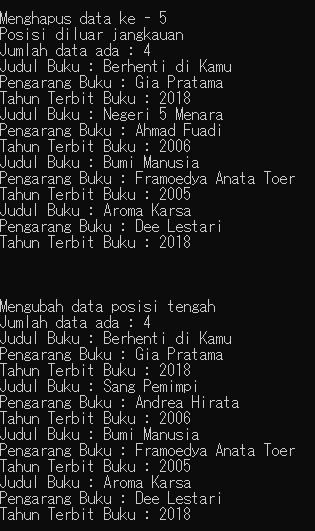






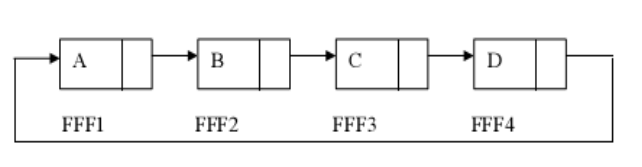
1. Tampilan Hasil Runing



1. Penjelasan

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

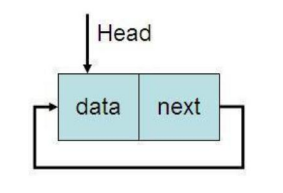


* Setiap node pada linked list mempunyai field yang berisi pointer ke node berikutnya, dan juga memiliki field yang berisi data.
* Pada akhir linked list, node terakhir akan menunjuk ke node terdepan sehingga linked list tersebut berputar.
* Node terakhir akan menunjuk lagi ke head

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



1. Count Single linked list

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

1. 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

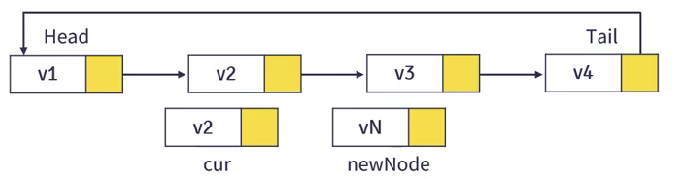
1. Menambah data belakang

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

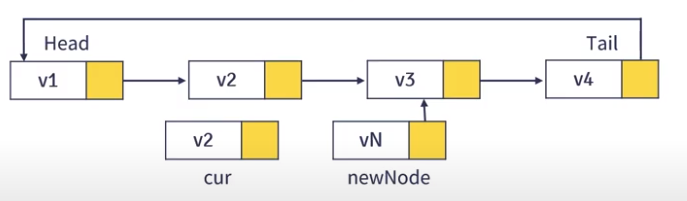
1. Menambah Node tengah Circular Single Linked list

-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

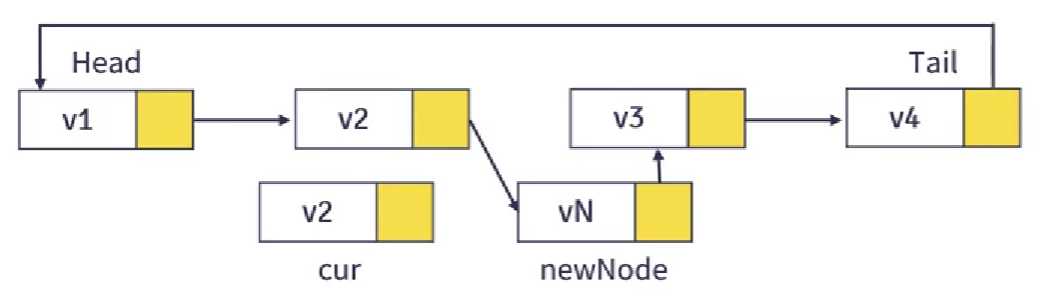
- Membuat newNode berisi data baru



-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

1. Change middle

Mengubah node di antara node head dan node tail

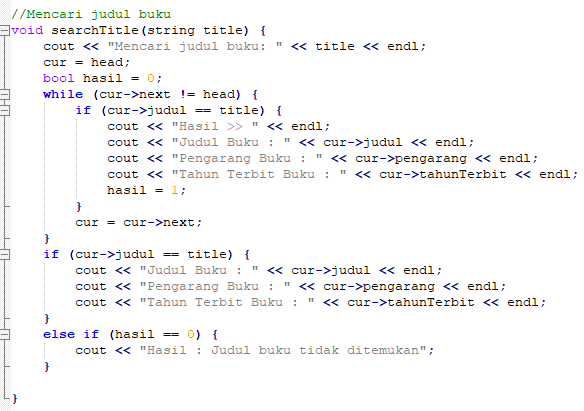
1. Print circular node

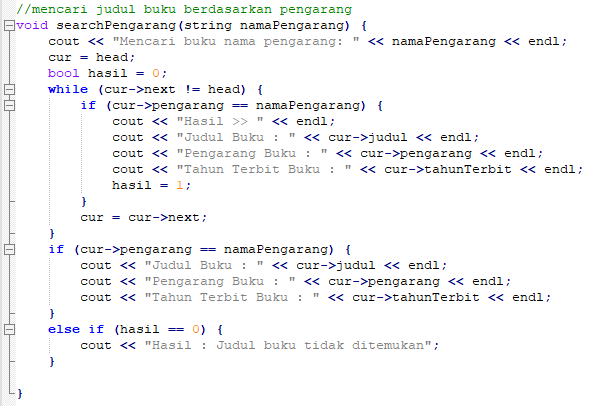
Mencetak Semua node dari head sampai tail

* 1. Percobaan Latihan 2

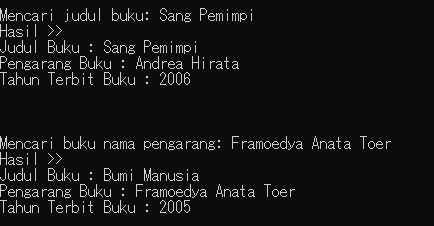
Menambahkan fungsi pencarian untuk judul buku dan nama pengarang di single linked list circular

1. Tampilan coding





1. Hasil Running



1. Penjelasan

Diatas adalah code algoritma untuk mencari nilai data dalam node.

-Mulai node pertama sebagai head

- melakukan perulangan while terus menerus sampai bertemu head

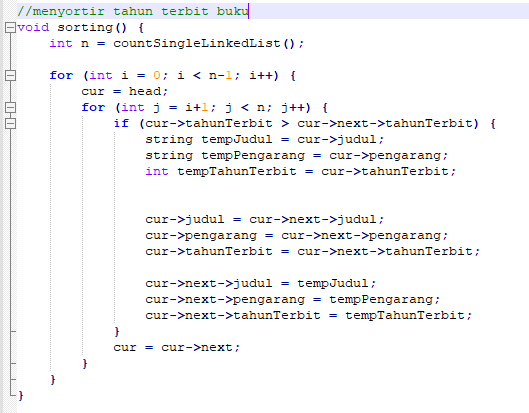
- Jika data pada node saat ini cocok dengan data yang diinput, maka data di node tersebut akan ditampilkan.

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

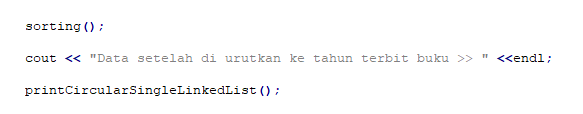
* 1. Latihan 3

Membuat fungsi untuk menyortir buku berdasarkan tahun terbit

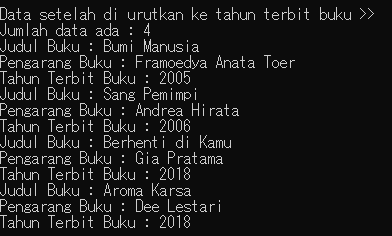
1. Tampilan coding



Dalam fungsi main();



1. Tampilan hasil running



1. 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.

1. Kesimpulan

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

Penggunaan linked list sendiri mempunya berbagai macam kelebihan melebihi array.