SENARAI BERANTAI TUNGGAL TIDAK BERPUTAR (SINGLE LINKED LIST NON CIRCULAR)

DEKLARASI LINKED LIST

Sebelum membuat sebuah senarai (Link List) kita harus mendeklarasikan tipe data dan pointer yang akan kita gunakan. Selain itu kita juga harus mendeklarasikan list-list yang nantinya akan kita gunakan. Sebagai contoh:

```
#include <iostream.h>
typedef struct Gerbong         //membuat sebuah tipe data baru yang terdiri dari
2 field
{
      int data;          //field data bertipe integer
      Gerbong *next;           //field next merupakan pointer dari list
};
Gerbong *baru;                 //variable baru bertipe pointer dari list
Gerbong *kepala;         //variable kepala bertipe pointer dari list
Gerbong *ekor;                 //variable ekor bertipe pointer dari list
Gerbong *bantu;          //variable bantu bertipe pointer dari list
Gerbong *hapus;          //variable hapus bertipe pointer dari list
```

catatan:

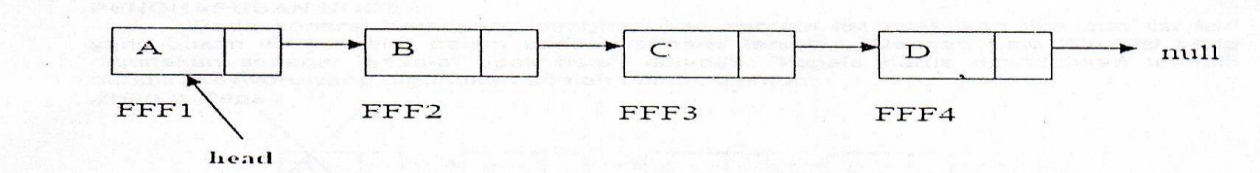
- 'Gerbong' diatas merupakan nama dari tipe data bentukan.
- kepala, ekor, baru, bantu,hapus merupakan variable pointer yang berisi alamat memori.
- kepala merupakan node paling awal dan selalu tetap(tidak boleh berpindah).
- ekor merupakan node paling akhir jadi ketika terjadi penambahan sebuah node yang diletakan di belakang maka pointer dari ekor akan ikut berpindah.
- baru merupakan nama untuk node yang diciptakan atau ditambahkan.
- bantu, hapus digunakan untuk memudahkan operasi edit dan delete sebuah node.

ILUSTRASI

Senarai berantai dapat diilustrasikan seperti satu kesatuan rangkaian kereta api. Kereta api terdiri dari beberapa gerbong, masing-masing dari gerbong itulah yang disebut struct / tipe data bentukan. Agar gerbong-gerbong tersebut dapat saling bertaut dibutuhkan minimal sebuah kait yang dinamakan sebagai pointer.

Setelah mendeklarasikan tipe data dan pointer pada list, selanjutnya kita akan mencoba membuat senarai / linked list tunggal tidak berputar atau sebuah gerbong. Ada beberapa operasi yang dapat kita buat pada senarai tersebut, diantaranya: tambah, hapus dan edit dari gerbong tersebut.

Inti dari linked list adalah proses (tambah, edit, hapus) dari gerbong / node dan bagaimana menyambungkan antar gerbong / node tersebut.



Gambar diatas dilustrasikan sebuah rangkaian kereta api dengan 4 buah gerbong. Gerbong A akan disebut sebagai kepala / head (walaupun penamaan ini bebas) dan gerbong D adalah ekor / tail. Tanda panah merupakan kait atau pointer yang menghubungkan satu gerbong dengan yang lainnya.

Pointer yang dimiliki D menuju ke NULL,inilah yang membedakan antara senarai berputar dengan yang tidak berputar. Kalau senarai berputar maka pointer dari D akan menuju ke A lagi.

PEMBENTUKAN NODE (GERBONG)

Digunakan keyword new yang berarti mempersiapkan sebuah node baru berserta alokasi memorinya, kemudian node tersebut diisi data dan pointer nextnya ditunjuk ke NULL. Pembentukan node tidak dapat dilakukan sekaligus namun harus satu persatu, hal ini berkaitan dengan bagaimana cara menyambungkan antar node tersebut.

```
kepala= new Gerbong;      //membuat sebuah node dengan nama kepala
kepala->next==NULL;       //pointer milik kepala diarahkan ke NULL
kepala->data=101;         //isi field data dengan 101
```

Kode diatas apabila dicompile maka akan membentuk sebuah node bertipe Gerbong dengan nama kepala. Karena bertipe list maka kepala juga memiliki 2 field seperti list yaitu field data dan field next. Bila diasumsikan dengan gerbong maka pada proses ini dibentuk gerbong A.

Setelah membentuk sebuah node dengan nama kepala, maka langkah selanjutnya adalah membentuk gerbong yang lain (gerbong B dan selanjutnya).

```
baru=new Gerbong;              //membuat sebuah node dengan nama baru
baru->data=5;
```

//Tambah Depan

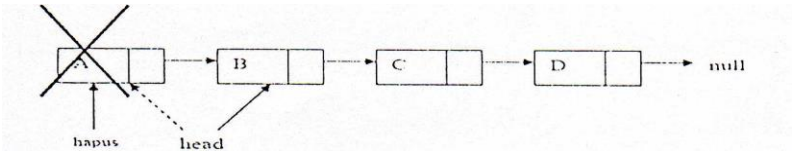
```
if (kepala->next==NULL)
{
      baru-->next=kepala;           //rnenyambung node bare dengan node kepala
      kepala=baru;                  //memindahkan kepala ke node baru
}
```

//Tambah Belakang

```
ekor->next=baru;           //menyambung node paling akhir/ekor ke node
baru
ekor=baru;             //meniindahkan ekor ke node terakhir yaitu node baru
.
ekor->next=NULL;
```

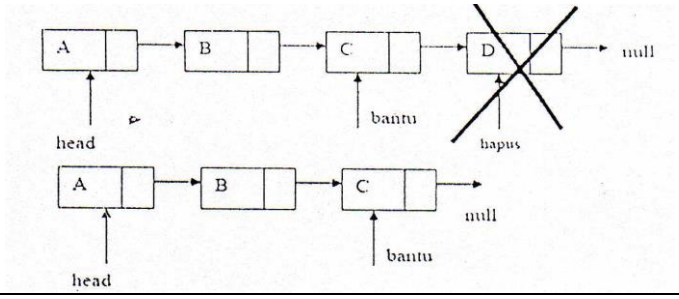
Tambah belakang berarti node baru yang terbentuk akan diletakan di posisi paling belakang.
PENGHAPUSAN NODE

Pada senarai berkepala, penghapusan sebuah list dilakukan jika ada list lain yang bukan list "kepala" dalam barisan senarai tersebut. Dengan kata lain, list yang digunakan sebagai "kepala" tidak boleh dihapus, "kepala harus dipindahkan terlebih dahulu. Keyword yang digunakan adalah delete. Contoh:
//Hapus Depan




```
if(kepala->next!=NULL)
{     if(kepala->next==ekor)
      {     hapus=kepala;
      kepala=kepala->next;
      delete hapus;
      }
}
else
{
      kepala=NULL;
}
```

//Hapus Belakang




```
if(head->next!=NULL)
{
      bantu=kepala;
      while(bantu->next->next!=NULL)
      {     bantu=bantu->next;
      }
      hapus=bantu->next;
      bantu->next = NULL;
      delete hapus;
}
else
{
      head =NULL;
}
```

MENAMPILKAN ISI NODE

Untuk menampilkan isi suatu node dengan cara mengakses field yang ada di dalam node tersebut. Yaitu dengan → Contoh :

```
bantu =head;
while(bantu!=NULL)
{
      printf("%i ",bantu->data);
      bantu=bantu->next;
}
```

LATIHAN

1. Buatlah sebuah linked list non circular yang berisi nim anda dan nama lengkap anda.
2. Buat fungsi untuk menambahkan node single linked list non circular dimana tiap node mengandung informasi nim dan nama. Peletakan posisi node urut berdasar nim secara ascending, jadi bisa tambah depan, belakang maupun tambah di tengah.
   Isikan data nim dan nama lengkap teman sebelah kiri dan kanan anda!!!
3. Buatlah fungsi untuk menampilkan data 3 buah node yang telah anda bentuk sebelumnya.
   Contoh tampilan

NIM	Nama Lengkap
22053766	Hernawan
22053768	Andrew S
22053791	Anthony S

4. Buatlah fungsi untuk mencari nama yang telah diinputkan dengan menggunakan NIM.
   Contoh tampilan:
   Inputkan nim yang dicari = 22053768
   Nama yang tercantum Andrew S
5. Buatlah sebuah fungsi untuk menghapus nim yang diinputkan oleh user.
   Contoh tampilan:
   NIM yang mau dihapus = 2205376
   NIM dengan nama Andrew S ditemukan dan telah dihapus
6. Buatlah sebuah program dengan menggunakan single linked list non circular dengan fungsi-fungsi:
   - menambah data(dari depan dan dari belakang)
   - search data yang telah diinputkan
   - menghapus data(dari depan dan dari belakang)
   - mencetak data

   Buat dengan menggunakan menu ya…..
7. (soal extended) Dengan menggunakan soal no 6 tambahkan 4 fungsi tambahan:
   - menambah data(di tengah)
   - menghapus data tertentu(berada di tengah)
   - mengurutkan data acak secara acending

- mengurutkan data acak secara decending

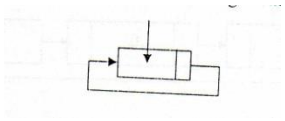
SELAMAT MENGERJAKAN

SENARAI BERANTAI TUNGGAL BERPUTAR (SINGLE LINKED LIST CIRCULAR)

✓ Perbedaan Single Linked list Circular dengan Non Circular

Pada saat kita membahas Single Linked List non Circular telah dijelaskan bahwa kita membutuhkan sebuah kait untuk menghubungkan node-node / gerbong-gerbong data yang ada. Perbedaan dengan Single Linked List Circular adalah pada node terakhir/ ekor yang semula menunjuk ke NULL diganti menunjuk kembali ke kepala atau head.

Contoh Struct Single Link List Circular



Dengan melihat gambar kita dapat membayangkan bahwa setiap node akan membentuk lingkaran, dan semakin banyak node lingkaran akan semakin besar. **Pada intinya tail akan selalu terhubung dengan head.**

Membuat Single Linked List Circular berkepala berputar

Untuk menginisialisasikan Single Linked List Circular berkepala berputar cukup mudah karena kita hanya seperti memberikan label pada suatu list. Cara mendeklarasikanna seperti contoh dibawah ini:

```
#include "stdio.h"
typedef struct Tnode
{  int data;
   Tnode *next;
};
Tnode *bantu;         //mendeklarasikan pointer bantu
Tnode *baru;          //mendeklarasikan pointer baru
Tnode *head;          //mendeklarasikan pointer head
Tnode *tail;          //mendeklarasikan pointer tail
void main()
{
   head=new Tnode;
   head ->next=head;
}
```

//memasukkan data Single Linked list Circular berkepala berputar

```
head -> data = 50;
```

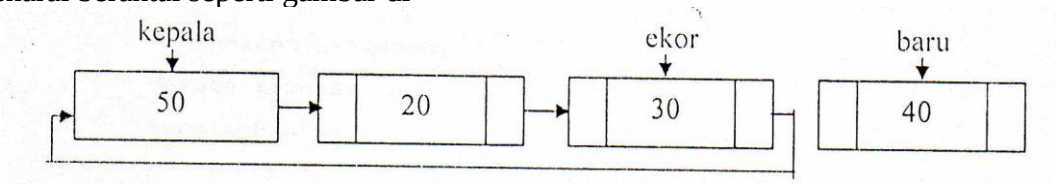
//menambah simpul baru dan menghubungkannya dengan simpul kepala

```
baru = new Tnode;
baru -> data = 20;                apakah perintah ini masih dapat digunakan
untuk penambahan
baru -> next = head;              simpul berikutnya???
head ->next = head;
```

//menambah simpul baru dan membuat ekor pada akhir senarai

```
tail = baru;
baru = new Tnode;
baru -> data = 30;                apakah perintah ini masih dapat digunakan
untuk penambahan
baru ->next=head;                 simpul berikutnya???
tail -> next=baru;
tail = baru;
```

- ✓ Pada senarai berantai seperti gambar di bawah:



- ✓ Bagaimana cara meletakkan data baru 40 diantara simpul dengan data 50 dan 20??
  Tambahkan pada program diatas:

```
baru=new Tnode;
baru->data=40;
baru->next=head->next;
head->next=baru;
```

- ✓ Mengedit / update data
  Misalnya kita ingin mengganti data pada ekor menjadi 25, caranya adalah sebagai berikut:

```
tail->data=25;
```

- ✓ Membaca senarai
  Tambahkan pada program diatas

```
bantu=head;
printf("%d", bantu);
bantu=bantu->next;
while(bantu!=head)
{
      printf("%d", bantu);
bantu=bantu->next;
}
```

- ✓ Menghapus Simpul ekor
  Tambahkan pada program diatas:

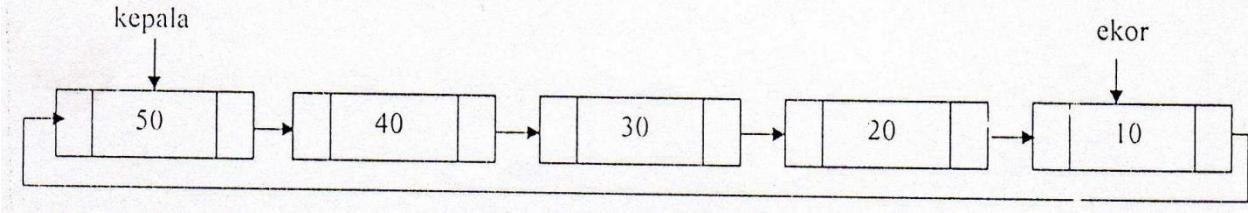
```
bantu=head;
while(bantu->next!=tail)
{
   bantu=bantu->next;
}
bantu->next=head;
delete tail;
tail=bantu;
```

- ✓ Menghapus Simpul kepala/head
  Tambahkan pada program diatas:

```
bantu=head->next;
delete head;
head = bantu;
```

Latihan

1. Buatlah sebuah Single Linked List Circular yang mcmpunyai 5 buah simpul seperti pada contoh dibawah ini



   a. Tulisk an listing program untuk membaca data dari seluruh simpul dengan menggunakan perulangan (for/while)
   b. TuIiskan listing program untuk menambah sebuah simpul baru dengan data 5 yang diletakan diantara data 30 dan data 20.
   c. Tuliskan listing program untuk menghapus simpul dengan data 20 tanpa merusak senarai.
   d. Tuliskan listing program untuk menghapus semua simpul dengan menggunakan variabel bantu dengan menggunakan perulangan (while/for).

2. Buatlah program Single Linked List Circular yang fleksibel dan dinamis. Dimana user dapat menambah simpul baru beserta datanya, membaca semua data pada senarai, menghapus simpul dengan data yang ditentukan oleh user, dan dapat menghapus semua simpul yang ada. Jangan Iupa membuat trapping errornya.
3. (soal extended) Buatlah sebuah link list berisi data-data identitas ktp(nama, alamat, no KTP) buatlah menu sebagai berikut:
   a. Input entry ktp (input akan langsung terurutkan berdasarkan no ktp asending)
   b. Delete file yang diinginkan user.
   c. Searching entry ktp
   d. Cetak semua entry ktp
   e. Cetak entry ktp tertentu
4. (soal extended) Buatlah 3 link list yang merepresentasikan jam(1-12), menit(1-60) dan detik(1-60). Mintalah inputan dari user berupa jumlah detik. Konversikan jumlah detik tersebut dan masukkan ke dalam link list (gunakan pointer jam,menit,detik) kemudian cetak jam yang tertera di link list tersebut.