**Penjelasan program single linked list**

Muhammad Fadlan Harahap

222310026

TI 22 PA 1

1. Deklarasi node dengan struct pada single linked Single Linked List yang pointer next-nya menunjuk ke dirinya sendiri, jika terdiri dari beberapa node maka pointer terakhirnya(tail) akan menunjuk ke pointer terdepannya(head).

1. Deklarasi Single Linked List Circular:

struct tnode

{

     int data;

     tnode \*next;

};

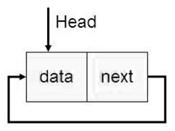
void main()

{

     head = new tnode;

     head->next = head;

}

[](https://3.bp.blogspot.com/-GoO7JOilAtY/WQA2FtcLnQI/AAAAAAAABNA/7aVKQ9NUfS47uXrBE89VUKpAagyyEfMBwCLcB/s1600/single%2Blinked%2Blist%2Bcircular1.PNG)

2. Menambah node dan membuat tail dari single linked list circular

Deklarasi penambahan node baru:

void main()

{

     node = new tnode;

     tail = new tnode;

     //head->next di bawah ini "masih" menunjuk ke head itu sendiri

     //jadi, kode di bawah ini maksudnya node->next menunjuk ke head

     node->next = head->next;

     //baru di sini head->next diubah menunjuk ke node

     //hasilnya akan seperti "Gambar 6"

     head->next = node;

     tail = node;

}

|  |
| --- |
| [https://2.bp.blogspot.com/-WVmMCrqNZo8/WQA4FoL-7lI/AAAAAAAABNU/mGnomQxSGfoHgb_7_t7bnE2zYXUNhRvKQCLcB/s400/single%2Blinked%2Blist%2Bcircular2.PNG](https://2.bp.blogspot.com/-WVmMCrqNZo8/WQA4FoL-7lI/AAAAAAAABNU/mGnomQxSGfoHgb_7_t7bnE2zYXUNhRvKQCLcB/s1600/single%2Blinked%2Blist%2Bcircular2.PNG) |
| Gambar Penambahan Node Baru |

3. Deklarasi menyisipkan node baru menggunakan sintak berikut:  
void main()  
{  
  node= new tnode;  
 node->next=head->next;  
 head->next=node;  
}

|  |
| --- |
| <https://3.bp.blogspot.com/-i3OjQZ0WmR0/WQA3sx2KowI/AAAAAAAABNY/HqUoI877fSQfBSsEAKQj959Pj0bz76N_QCPcB/s1600/single%2Blinked%2Blist%2Bcircular3.PNG> |
| Gambar Menyisipkan Node Baru |

4. Deklarasi menghapus node dari single linked list circular, menggunakan sintaks berikut:

void main()

{

     hapus = new tnode;

     if( head != tail)

     {

           hapus = head;

           head = head->next;

           tail->next = head;

           delete hapus;

     }else

     {

           head = NULL;

           tail = NULL;

     }

}

|  |
| --- |
| [https://3.bp.blogspot.com/-hlk4CMHT4_c/WQA23JrZLmI/AAAAAAAABNM/2q9Q_ndX0eon4IsX9Nlp82H2OmGikaPyACPcB/s320/single%2Blinked%2Blist%2Bcircular4.PNG](https://3.bp.blogspot.com/-hlk4CMHT4_c/WQA23JrZLmI/AAAAAAAABNM/2q9Q_ndX0eon4IsX9Nlp82H2OmGikaPyACPcB/s1600/single%2Blinked%2Blist%2Bcircular4.PNG) |
| Gambar Menghapus Node dari SLLC |

**Contoh Program Single linked list :**

#include <iostream>

using namespace std;

struct node

{

      int data;

      node\* next; // untuk menghubungkan dengan node lain, tipe data dibuat sama seperi aturan penggunaan pointer.

};

node\* head;

node\* tail;

node\* curr;

node\* entry;

node\* del;

void inisialisasi()

{

      head = NULL;

      tail = NULL;

}

void input(int dt)

{

      entry = (node\* )malloc(sizeof(node)); //alokasi memori

      entry->data = dt;

      entry->next = NULL;

      if(head==NULL)

      {

            head = entry;

            tail = head;

      }

      else

      {

            tail->next = entry;

            tail = entry;

      }

}

void hapus()

{

      int simpan;

      if(head==NULL)

      {

            cout<<"\nlinked list kosong, penghapusan tidak bisa dilakukan"<<endl;

      }

      else

      {

            simpan  = head ->data;

            //hapus depan

            del = head;

            head = head->next;

            delete del;

            cout<<"\ndata yang dihapus adalah "<<simpan<<endl;

      }

}

void cetak()

{

      curr = head;

      if(head == NULL)

            cout<<"\ntidak ada data dalam linked list"<<endl;

      else

      {

            cout<<"\nData yang ada dalam linked list adalah"<<endl;

            cout<<setw(6);

            while(curr!=NULL)

            {

                  cout<<curr->data<<"->";

                  curr = curr->next;

            }

cout<<"NULL";

            cout<<endl;

      }

}

void menu()

{

      char pilih, ulang;

      int data;

      do

      }

      cout<<"SINGLE LINKED LIST "<<endl;

      cout<<"-------------------------------"<<endl;

      cout<<"Menu : "<<endl;

      cout<<"1. Input data"<<endl;

      cout<<"2. Hapus data"<<endl;

      cout<<"3. Cetak Data"<<endl;

      cout<<"4. Exit"<<endl;

      cout<<"Masukkan pilihan Anda : ";

      cin>>pilih;

      switch(pilih)

      {

      case '1' :

            cout<<"\nMasukkan data : ";

            cin>>data;

            input(data);

            break;

      case '2' :

            hapus();

            break;

      case '3' :

            cetak();

            break;

      case '4' :

            exit(0);

            break;

      default :

            cout<<"\nPilih ulang"<<endl;

      }

      cout<<"\nKembali ke menu?(y/n)";

      cin>>ulang;

      }while(ulang=='y' || ulang=='Y');

}

int main()

{

      inisialisasi();

      menu();

      return EXIT\_SUCCESS;

}