

Linked List



Sisilia Thya Safitri, ST., MT

Jl. DI Panjaitan 128 Purwokerto

* Untuk kalangan sendiri

Praktikum 5

Materi : Linked List

Waktu : 100 menit

Dasar Teori

Linked List merupakan suatu bentuk struktur data yang berisi kumpulan data yang disebut sebagai node yang tersusun secara sekuensial, saling sambung menyambung, dinamis, dan terbatas. Linked List sering disebut sebagai senarai berantai. Untuk menghubungkan satu node dengan node lainnya maka Linked List menggunakan pointer sebagai penunjuk node selanjutnya. Node sendiri merupakan sebuah struct yang menempati suatu lokasi memori secara dinamis yang terdiri dari beberapa field, minimal 2 buah field yaitu field untuk isi dari struct datanya sendiri, dan 1 field arbitari bertipe pointer sebagai penunjuk node selanjutnya. Array dan Linked List memiliki perbedaan sebagai berikut :

Array

Statis

Penambahan dan penghapusan data Terbatas

Random access

Penghapusan array tidak mungkin

Linked List

Dinamis

Penambahan dan penghapusan data tidak terbatas

Sequential access

Penghapusan mudah

Salah satu tipe Linked List yang sederhana yaitu Single Linked List. Single Linked List merupakan Linked List yang memiliki hanya satu pointer penunjuk dengan arah data hanya satu arah juga. Single Linked List memiliki 2 macam bentuk yaitu Non Circular dan Circular. Non Circular Linked List merupakan Linked List di mana antara kepala dan node terakhir tidak memiliki hubungan. Pada Linked List ini maka pointer terakhir selalu menunjuk NULL sebagai pertanda data terakhir dalam list-nya. Single Linked List Non Circular dapat digambarkan sebagai gerbong kereta api seperti berikut ini :

Langkah membuat dan operasi pada sebuah Linked List adalah sebagai berikut :

1. Mendeklarasikan struct node
2. Membuat node head
3. Menginisialisasi node head
4. Menambah node baru baik di depan maupun di belakang
5. Menghapus node

Linked List banyak dimanfaatkan pada pemrograman kecerdasan buatan, fuzzy, maze solving, dan sebagainya.

PROSEDUR PERCOBAAN

Kompilasi program berikut ini dan amati outputnya pada layar Anda. Perhatikan baik-baik pemanggilan dan penggunaan fungsi-fungsi serta prosedurnya agar dapat mengerjakan tugas yang diberikan !

Membuat header pada program

```
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

//global var/const
typedef struct TNode{
    int data;
    TNode *next;
};

TNode *head; //head node
//proto func/proc

void initHead();
int isEmpty();
void insertDepan(int databaru);
void insertBelakang (int databaru);
void tampilList();
void hapusDepan();
void hapusBelakang();
void clearList();
```

Menuliskan detail setiap fungsi

```
//detil func/proc
//init head

void initHead()
{
    head = NULL; //NULL <> null!!!
}

//cek list kosong atau tdk
int isEmpty()
{
    return (head == NULL) ? 1:0;
}
```

```
//tambah data di depan
```

```
void insertDepan(int databaru)
```

```
{
    TNode *baru;
    baru = new TNode;
    baru->data = databaru;
    baru->next = NULL;
    if(isEmpty()==1)
    {
        head=baru;
        head->next = NULL;
    }
    else
    {
        baru->next = head;
        head = baru;
    }
    cout<<"Data baru telah dimasukkan di depan\n";
}
```

```
//tambah data di belakang
```

```
void insertBelakang (int databaru)
```

```
{
    TNode *baru,*bantu;
    baru = new TNode;
    baru->data = databaru;
    baru->next = NULL;
    if(isEmpty()==1)
    {
        head=baru;
        head->next = NULL;
    }
    else
    {
        bantu=head;
        while(bantu->next!=NULL)
        {
            bantu=bantu->next;
        }
        bantu->next = baru;
    }
    cout<<"Data baru telah dimasukkan di belakang\n";
}
```

```

//menampilkan list
void tampilList()
{
    TNode *bantu;
    bantu = head;
    if (isEmpty() == 0)
    {
while (bantu != NULL)
    {
        cout<<bantu->data<<" ";
        bantu=bantu->next;
    }
        cout<<"\n";
    }
    else
        cout<<"Masih kosong\n";
}

```

```

//hapus data terdepan
void hapusDepan()
{
    TNode *hapus;
    int d;
    if (isEmpty() == 0)
    {
        if (head->next != NULL)
        {
            hapus = head;
            d = hapus->data;
            head = head->next;
            delete hapus;
        }
        else
        {
            d = head->data;
            head = NULL;
        }
        cout<<d<<" terhapus\n";
    }
    else
        cout<<"Masih kosong\n";
}

```

```

//hapus data terakhir
void hapusBelakang()
{
    TNode *hapus,*bantu;
    int d;
    if (isEmpty()==0)
    {
        if(head->next != NULL)
        {
            bantu = head;
            while(bantu->next->next!=NULL)
            {

                bantu = bantu->next;
            }
            hapus = bantu->next;
            d = hapus->data;
            bantu->next = NULL;
            delete hapus;
        }
        else
        {
            d = head->data;
            head = NULL;
        }
        cout<<d<<" terhapus\n";
    }
    else
        cout<<"Masih kosong\n";
}

```

```

//clear semua node
void clearList()
{
    TNode *bantu,*hapus;
    bantu = head;
    while(bantu!=NULL)
    {
        hapus = bantu;
        bantu = bantu->next;
        delete hapus;
    }
    head = NULL;}

```

Menuliskan main program

```

//main prog
int main()
{
    cout<<"single linked list non circular\n1. inisialisasi head ... \n";
    initHead();
    cout<<"done\n tampilkan isi list :\n";
    tampilList() ;

    //entry data di depan
    cout<<"\nEntri data di depan list\n";
    int data_baru;
    for(int i=1;i<=5;i++)
    {
        cout<<"masukkan data ke- "<<i<<" : ";
        cin>> data_baru;
        insertDepan(data_baru);
    }
    cout<<"tampilkan isi list :\n";
    tampilList() ;

    //entry data di belakang
    cout<<"\nEntri data di belakang list\n";
    for(int i=1;i<=5;i++)
    {
        cout<<"masukkan data ke- " <<i<<" : ";
        cin>>data_baru;
        insertBelakang(data_baru);
    }
}

```

```

cout<<"tampilkan isi list :\n";
tampilList() ;

//hapus data di depan
cout<<"\nhapus 2 data terdepan\n";

for(int i=1;i<=2;i++)
{
    hapusDepan();
}
cout<<"tampilkan isi list :\n";
tampilList() ;

//hapus data di belakang
cout<<"\nhapus 2 data terakhir\n";

for(int i=1;i<=2;i++)
{
    hapusBelakang();
}
cout<<"tampilkan isi list :\n";

tampilList();

.

//clear semua list
cout<<"\n hapus semua node \n";
clearList();
cout<<"\n tampilkan isi list :\n";
tampilList();
getch();
return 0;
}

```

Praktik

1. Buatlah program menggunakan Single Linked List Non Circular untuk menyimpan

Nama dan NIM data mahasiswa berikut ini :

Nama	NIM
Dede	12347867
Kiki	98765674
Nina	67453279
Andi	83450120

2. Hapus list Andi !
3. Tampilkan di layar hasilnya sbb :

PROGRAM SENARAI BERANTAI

Masukkan nama ke-1 : Dede
Masukkan NIM ke-1 : 12347867

Dst

DATA MAHASISWA

Nama	NIM
Dede	12347867
Kiki	98765674
Nina	67453279
Andi	83450120

Andi terhapus

DATA MAHASISWA

Nama	NIM
Dede	12347867
Kiki	98765674
Nina	67453279

Resume

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

S1 TEKNIK INFORMATIKA

Hari/Tanggal Praktikum :

Modul :

NIM :

Nama Praktikan :

Nama Asistant : 1.....

2.....

Nilai dan Parat :

Hasil Analisa Praktikum