

PERTEMUAN X

LINKED LIST

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan pengertian dan pembuatan Linked List dengan C++.
- b) Mahasiswa dapat melakukan operasi penyisipan maupun penghapusan simpul pada Linked List dengan C++.
- c) Mahasiswa dapat mengimplementasikan Linked List dengan C++.

TEORI DASAR

a) Pendahuluan

Linked List adalah struktur berupa rangkaian elemen saling berkait dimana tiap elemen dihubungkan ke elemen yang lain melalui pointer. Pointer adalah alamat elemen. Penggunaan pointer untuk mengacu elemen berakibat elemen-elemen bersebelahan secara logic walaupun tidak bersebelahan secara fisik di memori.

Linked List merupakan kumpulan komponen yang saling berkaitan satu dengan yang lain melalui pointer. Masing-masing komponen sering disebut dengan simpul atau verteks.

b) Singly Linked List

Merupakan Linked List yang paling sederhana. Setiap simpul dibagi menjadi dua bagian yaitu bagian isi dan bagian pointer. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, sedangkan bagian pointer merupakan bagian yang berisi alamat dari simpul berikutnya.

Deklarasi Doubly Linked List :

```
typedef struct node *simpul
struct node
{
    type_data Isi;
    simpul Next;
};
```

c) Doubly Linked List

Doubly Linked List merupakan Linked List dimana setiap simpul dibagi menjadi tiga bagian, yaitu bagian isi, bagian pointer kiri, dan bagian pointer kanan. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, bagian pointer

kiri merupakan bagian yang berisi alamat dari simpul sebelumnya dan bagian pointer kanan merupakan bagian yang berisi alamat dari simpul berikutnya.

Deklarasi Doubly Linked List :

```
typedef struct node *simpul
struct node
{
    char Isi;
    simpul kanan;
    simpul kiri;
};
```

TUGAS PRAKTIKUM

- a) Buatlah program untuk Operasi Singly Linked (simpan dengan nama lat10_1.cpp)

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
typedef struct node *simpul;
struct node
{
    char Isi;
    simpul Next;
};
//=====
//==Prototype Function=
//=====

void Sisip_Depan (simpul &L, char elemen );
void Sisip_Belakang (simpul &L, char elemen );
void Sisip_Tengah1 (simpul &L, char elemen1, char elemen2 );
void Sisip_Tengah2 (simpul &L, char elemen1, char elemen2 );
void Hapus_Depan (simpul &L);
void Hapus_Belakang (simpul &L);
void Hapus_Tengah (simpul &L, char elemen);
void Cetak (simpul L);

//=====
//==Function Main=====
//=====

main ()
{
```

```

char huruf, huruf2;
simpul L = NULL; //Pastikan Bahwa L kosong
cout<<"==OPERASI PADA SINGLE LINKED LIST=="<<endl<<endl;
//=====
//==Sisip Depan==
//=====
cout<<"Penyisipan Simpul Di Depan"<<endl<<endl;
cout<<"Masukan Huruf : "; cin>>huruf;
Sisip_Depan (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Depan (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Depan (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Depan (L, huruf );
Cetak (L);
//=====
//==Sisip Belakang=
//=====
cout<<"\n\nPenyisipan Simpul Di Belakang"<<endl<<endl;
cout<<"Masukan Huruf : "; cin>>huruf;
Sisip_Belakang (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Belakang (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Belakang (L, huruf );
cout<<"Masukan Huruf :" ; cin>>huruf ;
Sisip_Belakang (L, huruf );
Cetak (L);
//=====
//==Sisip Simpul Setelah Simpul Tertentu=
//=====
cout<<endl<<endl<<"Masukan Huruf : "; cin>>huruf;
cout<<"Disisip Setelah Huruf : "; cin>>huruf2;
cout<<huruf<<" Disisip Setelah "<<huruf2<<endl;
Sisip_Tengah1 (L, huruf, huruf2);
Cetak (L) ;
//=====
//==Sisip Simpul Sebelum Simpul Tertentu=
//=====

```

```

cout<<endl<<endl<<"Masukan Huruf : "; cin>>huruf;
cout<<"Disisip Sebelum Huruf : "; cin>>huruf2;
cout<<huruf<<" Disisip Sebelum "<<huruf2<<endl;
Sisip_Tengah2 (L, huruf, huruf2);
Cetak (L) ;
//=====
//==Hapus Simpul Depan=
//=====
cout<<endl<<endl<<"Setelah Hapus Simpul Depan "<<endl;
Hapus_Depan (L);
Cetak (L);
//=====
//==Hapus Simpul Belakang=
//=====
cout<<endl<<endl<<"Setelah Hapus Simpul Belakang "<<endl;
Hapus_Belakang (L);
Cetak (L);
//=====
//==Hapus Simpul TENGAH==
//=====
cout<<"\n\nMasukkan Huruf Tengah Yang akan dihapus : ";
;cin>>huruf;
Hapus_Tengah (L,huruf);
Cetak (L);
getch ( ) ;
}

//*****
//**FUNCTION SISIP SIMPUL DI DEPAN**
//*****
void Sisip_Depan (simpul &L, char elemen)
{
simpul baru; // = new simpul ;
baru = (simpul) malloc (sizeof(simpul));
baru->Isi = elemen ;
baru-> Next = NULL;
if (L== NULL)
    L=baru;
else
{
    baru->Next = L;
}

```

```

L= baru;
}
}
//*****
//**FUNCTION SISIP SIMPUL SETELAH SIMPUL TERTENTU**
//*****
void Sisip_Tengah1 (simpul &L, char elemen1, char elemen2)
{
    simpul bantu,baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen1 ;
    baru->Next = NULL;
    if (L== NULL)
        cout << "List Kosong ....." << endl;
    else
    {
        bantu = L;
        while (bantu ->isi != elemen2) bantu = bantu -> Next;
        baru->Next = bantu ->Next ;
        bantu->Next = baru ;
    }
}
//*****
//**FUNCTION SISIP SIMPUL SEBELUM SIMPUL TERTENTU*****
//*****
void Sisip_Tengah2 (simpul &L, char elemen1, char elemen2)
{
    simpul bantu, baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen1 ;
    baru->Next = NULL;
    if (L== NULL)
        cout<<"List Kosong....." << endl;
    else
    {
        bantu = L;
        while (bantu->Next->isi != elemen2) bantu = bantu -> Next;
        baru->Next = bantu ->Next ;
        bantu->Next = baru ;
    }
}

```

```

//*****
//**FUNCTION SISIP SIMPUL DI BELAKANG*****
//*****
void Sisip_Belakang (simpul &L, char elemen)
{
    simpul bantu, baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen ;
    baru-> Next = NULL;
    if (L== NULL)
        L=baru;
    else
    {
        bantu = L;
        while (bantu->Next != NULL)
            bantu= bantu -> Next ;
        bantu->Next = baru ;
    }
}
//*****
//**FUNCTION MENCETAK ISI LINKED LIST*****
//*****
void Cetak (simpul L)
{
    simpul bantu ;
    if (L==NULL)
        cout<<"Linked List Kosong ....."<<endl;
    else
    {
        bantu =L;
        cout<<"Isi Linked List : ";
        while (bantu ->Next != NULL)
        {
            cout<<bantu->isi<<"-->";
            bantu=bantu->Next;
        }
        cout<<bantu->isi;
    }
}

```

```

//*****
//**FUNCTION HAPUS SIMPUL DEPAN*****
//*****

void Hapus_Depan (simpul &L)
{
    simpul Hapus ;
    if (L==NULL)
        cout<<"Linked List Kosong.....";
    else
    {
        Hapus = L;
        L = L-> Next ;
        Hapus -> Next = NULL;
        free (Hapus);
    }
}

//*****
//**FUNCTION HAPUS SIMPUL BELAKANG*****
//*****



void Hapus_Belakang (simpul &L)
{
    simpul bantu, hapus;
    if (L==NULL)
        cout<<"Linked List Kosong.....";
    else
    {
        bantu = L;
        while (bantu ->Next->Next != NULL) bantu=bantu->Next;
        hapus = bantu -> Next;
        bantu -> Next = NULL;
        free (hapus);
    }
}

//*****
//**FUNCTION HAPUS SIMPUL DI TENGAH****
//*****



void Hapus_Tengah(simpul &L, char elemen)
{
    simpul bantu,hapus;

```

```

if (L==NULL)
    cout<<"Linked List Kosong.....";
else
{
    bantu = L;
    while (bantu ->Next->Isi != elemen) bantu=bantu->Next;
    hapus = bantu -> Next;
    bantu ->Next = bantu -> Next ->Next;
    hapus -> Next = NULL;
    free (hapus);
}
//=====

```

- b) Buatlah program untuk Operasi Doubly Linked (simpan dengan nama lat10_2.cpp)

```

#include<iostream>
#include<conio.h>
#include<stdlib.h>
#define true 1
#define false 0
using namespace std;
typedef struct node *simpul;
struct node
{
    char Isi;
    simpul kanan;
    simpul kiri;
};
//=====
//==Prototype Function=
//=====
void Sisip_Depan (simpul &DL, char elemen );
void Sisip_Belakang (simpul &DL, char elemen ) ;
void Sisip_Tengah1 (simpul &DL, char elemen1, char elemen2 ) ;
void Sisip_Tengah2 (simpul &DL, char elemen1, char elemen2 ) ;
void Hapus_Depan (simpul &DL);
void Hapus_Belakang (simpul &DL);
void Hapus_Tengah (simpul &DL, char elemen);
void Cetak (simpul DL);

```

```

//=====
//==Function Main==
//=====

main ( )
{
    char huruf, huruf2;
    simpul DL = NULL; //Pastikan Bahwa DL kosong
    int i;
    cout<<"\t\t==OPERASI PADA DOUBLY LINKED LIST==\n\n";
    //=====
    //==Sisip Depan==
    //=====

    cout<<"Penyisipan Simpul Di Depan"<<endl<<endl;
    for (i=1;i<=4;i++)
    {
        cout<<"Masukan Huruf : "; cin>>huruf;
        Sisip_Depan (DL, huruf );
    }
    Cetak (DL);
    //=====

    //==Sisip Belakang=
    //=====

    cout<<"\n\nPenyisipan Simpul Di Belakang"<<endl<<endl;
    for (i=1;i<=4;i++)
    {
        cout<<"Masukan Huruf : "; cin>>huruf;
        Sisip_Belakang (DL, huruf );
    }
    Cetak (DL);
    //=====

    //==Sisip Simpul Setelah Simpul Tertentu=
    //=====

    cout<<endl<<endl<<"Masukan Huruf : "; cin>>huruf;
    cout<<"Disisip Setelah Huruf : "; cin>>huruf2;
    cout<<huruf<<" Disisip Setelah "<<huruf2<<endl;
    Sisip_Tengah1 (DL, huruf, huruf2);
    Cetak (DL) ;
    //=====

    //==Sisip Simpul Sebelum Simpul Tertentu=
    //=====

    cout<<endl<<endl<<"Masukan Huruf : "; cin>>huruf;
    cout<<"Disisip Sebelum Huruf : "; cin>>huruf2;
    cout<<huruf<<" Disisip Sebelum "<<huruf2<<endl;
    Sisip_Tengah2 (DL, huruf, huruf2);
}

```

```

Cetak (DL) ;
//=====
//==Hapus Simpul Depan=
//=====
cout<<endl<<endl<<"Setelah Hapus Simpul Depan "<<endl;
Hapus_Depan (DL);
Cetak (DL);
//=====
//==Hapus Simpul Belakang=
//=====
cout<<endl<<endl<<"Setelah Hapus Simpul Belakang "<<endl;
Hapus_Belakang (DL);
Cetak (DL);
//=====
//==Hapus Simpul TENGAH==
//=====
cout<<"\n\nMasukkan Huruf Tengah Yang akan dihapus : ";
cin>>huruf;
Hapus_Tengah (DL,huruf);
Cetak (DL);
getch ( );
}

//*****
//**FUNCTION SISIP SIMPUL DI DEPAN****
//*****
void Sisip_Depan (simpul &DL, char elemen)
{
    simpul baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen ;
    baru-> kanan = NULL;
    baru-> kiri = NULL;
    if (DL== NULL)
        DL=baru;
    else
    {
        baru->kanan = DL;
        DL->kiri = baru;
        DL= baru;
    }
}
//*****
//**FUNCTION SISIP SIMPUL SETELAH SIMPUL TERTENTU **
//*****

```

```

void Sisip_Tengah1 (simpul &DL, char elemen1, char elemen2)
{
    simpul bantu,baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen1 ;
    baru->kanan = NULL;
    baru->kiri = NULL;
    if (DL== NULL)
        cout << "List Kosong ....." << endl;
    else
    {
        bantu = DL;
        while (bantu ->isi != elemen2)
            bantu = bantu ->kanan;
        baru->kanan = bantu ->kanan;
        baru->kiri = bantu;
        bantu->kanan->kiri = baru;
        bantu->kanan = baru;
    }
}
//*****
//**FUNCTION SISIP SIMPUL SEBELUM SIMPUL TERTENTU*****
//*****
void Sisip_Tengah2 (simpul &DL, char elemen1, char elemen2)
{
    simpul bantu, baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen1 ;
    baru->kanan = NULL;
    baru->kiri = NULL;
    if (DL== NULL)
        cout<<"List Kosong....." << endl;
    else
    {
        bantu = DL;
        while (bantu->kanan->isi != elemen2)
            bantu = bantu ->kanan;
        baru->kanan = bantu ->kanan;
        baru->kiri = bantu;
        bantu->kanan->kiri = baru;
        bantu->kanan = baru;
    }
}

```

```

//*****
//**FUNCTION SISIP SIMPUL DI BELAKANG*****
//*****
void Sisip_Belakang (simpul &DL, char elemen)
{
    simpul bantu, baru;
    baru = (simpul) malloc (sizeof(simpul));
    baru->isi = elemen ;
    baru->kanan = NULL;
    baru->kiri = NULL;
    if (DL==NULL)
        DL=baru;
    else
    {
        bantu = DL;
        while (bantu->kanan != NULL)
            bantu = bantu -> kanan;
        bantu->kanan = baru;
        baru->kiri = bantu;
    }
}
//*****
//**FUNCTION MENCETAK ISI LINKED LIST*****
//*****
void Cetak (simpul DL)
{
    simpul bantu ;
    if (DL==NULL)
        cout<<"Linked List Kosong ..... "<<endl;
    else
    {
        bantu =DL;
        cout<<"Isi Linked List : ";
        while (bantu ->kanan != NULL)
        {
            cout<<bantu->isi<<"<=>";
            bantu=bantu->kanan;
        }
        cout<<bantu->isi;
    }
}

```

```

//*****
//**FUNCTION HAPUS SIMPUL DEPAN*****
//*****

void Hapus_Depan (simpul &DL)
{
    simpul Hapus ;
    if (DL==NULL)
        cout<<"Linked List Kosong.....";
    else
    {
        Hapus = DL;
        DL = DL-> kanan ;
        DL ->kiri = NULL;
        Hapus -> kanan = NULL;
        free (Hapus);
    }
}

//*****
//**FUNCTION HAPUS SIMPUL BELAKANG*****
//*****
```

```

void Hapus_Belakang (simpul &DL)
{
    simpul bantu, hapus;
    if (DL==NULL)
        cout<<"Linked List Kosong.....";
    else
    {
        bantu = DL;
        while (bantu ->kanan->kanan != NULL) bantu=bantu->kanan;
        hapus = bantu -> kanan;
        bantu -> kanan = NULL;
        hapus -> kiri = NULL;
        free (hapus);
    }
}

//*****
//**FUNCTION HAPUS SIMPUL DI TENGAH****
//*****
```

```

void Hapus_Tengah(simpul &DL, char elemen)
{

```

```
simpul bantu,hapus;
if (DL==NULL)
    cout<<"Linked List Kosong.....";
else
{
    bantu = DL;
    while (bantu ->kanan->isi != elemen)
        bantu=bantu->kanan;
    hapus = bantu -> kanan;
    bantu ->kanan->kanan->kiri = bantu;
    bantu ->kanan = bantu->kanan->kanan;
    hapus -> kanan = NULL;
    hapus -> kiri = NULL;
    free (hapus);
}
}//=====eof=====
```

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Linked List!
2. Jelaskan perbedaan antara Singly Linked List, Doubly Linked List, dan Circular Linked List !
3. Jelaskan Operasi-Operasi pada Singly Linked List!
4. Jelaskan Operasi-Operasi pada Doubly Linked List!

TUGAS AKHIR

1. Buatlah program menu untuk menampilkan 2 program di atas !