Nama : Tiara Yemelda

NPM : G1F021005

Matkul : Algoritma & Struktur Data

Pengertian Double Linked List

Double Linked List

Double Link List adalah elemen-elemen yang dihubungkan dengan dua pointer dalam satu elemen dan list dapat melintas baik di depan atau belakang.

Elemen double link list terdiri dari tiga bagian:

- Bagian data informasi

- Pointer next yang menunjuk ke elemen berikutnya

- Pointer prev yang menunjuk ke elemen sebelumnya

Untuk menunjuk head dari double link list, pointer prev dari elemen pertama menunjuk NULL. Sedangkan untuk menunjuk tail, pointer next dari elemen terakhir menunjuk NULL.

Contoh Membuat TDA(Tipe Data Abstrak) dari Double Linked Circular List tersebut.

Instan :

Double Linked Circular List

Operasi :

Buat\_node(char x) : membuat node baru dengan informasi x

Tambah\_elemen\_didepan() : menambah elemen paling depan (pointernya menunjuk

elemen pertama link list)

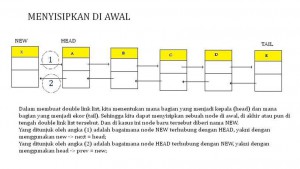
Tambah\_elemen\_dibelakang() : menambah elemen paling belakang (pointer elemen

yang baru menunjuk elemen pertama)

Hapus\_elemen\_() : Menghapus elemen (pointer menunjuk elemen yang akan dihapus)

Cetak () : menelusuri elemen satu demi dan menampilkan informasinya.

Ilustrasinya :





Ada 2 jenis Double Linked List yaitu :

· Double Linked List Circular

· Double Linked List Non Circular

1. Double Linked List Circular

Pengertian secara umumnya DLLC itu Linked list yang menggunakan pointer, dimana setiap node memiliki 3 field, yaitu:

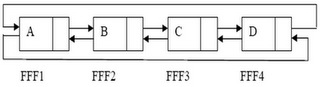
1 field pointer yang menunjuk pointer berikutnya “next“,

1 field menunjuk pointer sebelumnya ” prev “,

1 field yang berisi data untuk node tersebut .

Double Linked List Circular pointer next dan prev nya menunjuk kedirinya sendiri secara circular. Bentuk Node DLLC

Ilustrasi Double Linked List Circular



Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya

Untuk pembentukan node baru , mulanya pointer next dan prev akan menunjuk ke nilai NULL

Selanjutnya pointer prev akan menunjuk ke node sebelumnya , dan pointer next akan menunjuk ke node selanjutnya pada list.

Contoh program Double Linked List Circular :

#include <conio.h>

#include <stdio.h>

struct TNode{

char data[30];

TNode \*next;

TNode \*prev;

};

TNode \*head, \*tail;

void init(void);

int isEmpty(void);

void insertDepan(char databaru[30]);

void insertBelakang (char databaru[30]);

void inserttengah(char databaru[30], int pilihdepan, int pilihbelakang);

void tampil(void);

void hapusDepan(void);

void hapusBelakang(void);

void deletetengah(int pilih);

void clear(void);

int cari(char elemen[30]);

main()

{

////////////////////////////////

char pilih;

char elm[30];

int depan,belakang;

init();

do

{

printf("\n");

printf("\t\t===================================\n");

printf("\t\t|| CONTOH PROGRAM ||\n");

printf("\t\t|| DOUBLE LINK LIST CIRCULAR ||\n");

printf("\t\t===================================\n");

printf(" \t\tMENU PILIHAN : \n");

printf("\t\t===================================\n");

printf("\t\t[1] MASUKKAN DATA DARI DEPAN\n");

printf("\t\t[2] MASUKKAN DATA DARI BELAKANG\n");

printf("\t\t[3] MASUKKAN DATA DI TENGAH\n");

printf("\t\t[4] TAMPILKAN DATA\n");

printf("\t\t[5] HAPUS DATA PALING DEPAN\n");

printf("\t\t[6] HAPUS DATA PALING BELAKANG\n");

printf("\t\t[7] HAPUS DATA DI TENGAH\n");

printf("\t\t[8] HAPUS SEMUA DATA\n");

printf("\t\t[9] CARI DATA\n");

printf("\t\t[0] KELUAR\n");

printf("\t\t===================================\n\n");

printf("\t\t===================================\n\n");

printf("\n");

printf("\t\t->->PILIHAN ANDA : ");

scanf("%s",&pilih);

switch(pilih)

{

case '1':clrscr();

tampil();

printf("\n");

printf(" \t\tMASUKKAN DATA DARI DEPAN\n");

printf("\t\t----------------------------\n");

printf("\t\t:: MASUKKAN DATA : ");

scanf("%s",&elm);

insertDepan(elm);

getch();

clrscr();

break;

case '2':clrscr();

tampil();

printf("\n");

printf(" \t\tMASUKKAN DATA BELAKANG\n");

printf("\t\t----------------------------\n");

printf("\t\t:: MASUKKAN DATA : ");

scanf("%s",&elm);

insertBelakang(elm);

clrscr();

break;

case '3':clrscr();

tampil();

printf("\n");

printf(" \t\tMASUKKAN DATA DARI TENGAH\n");

printf("\t\t----------------------------\n");

printf("\t\t:: MASUKKAN DATA : ");

scanf("%s",&elm);

printf("\t\t:: DATA DEPAN : ");

scanf("%i",&depan);

printf("\t\t:: DATA BELAKANG : ");

scanf("%i",&belakang);

inserttengah(elm,depan,belakang);

getch();

clrscr();

break;

case '4':clrscr();

tampil();

printf("\t\t-------------\n\n");

printf("\t\tPress Enter to Continue..");

getch();

clrscr();

break;

case '5':clrscr();

tampil();

hapusDepan();

printf("\n");

printf("\t\t-------------\n");

printf("\t\tPress Enter to Continue..");

getch();

clrscr();

break;

case '6':clrscr();

tampil();

hapusBelakang();

printf("\n");

printf("\t\t-------------\n");

printf("\t\tPress Enter to Continue..");

getch();

clrscr();

break;

case '7':clrscr();

tampil();

printf("\n");

printf(" \t\tMENGHAPUS DATA DARI TENGAH\n");

printf("\t\t----------------------------\n");

printf("\t\t:: DATA NO : ");

scanf("%i",&depan);

deletetengah(depan);

getch();

clrscr();

break;

case '8':clrscr();

clear();

printf("\t\tDATA TELAH DIHAPUS SEMUA\n");

printf("\t\t-------------\n");

printf("\t\tPress Enter to Continue..");

getch();

clrscr();

break;

case '9':clrscr();

printf("\n");

printf(" \t\t MASUKKAN DATA YANG DICARI\n");

printf("\t\t----------------------------\n");

printf("\t\t:: MASUKKAN DATA : ");

scanf("%s",&elm);

if(cari(elm)==1){

printf("\n\t\tdata success ditemukan");

}else{

printf("\n\t\tMaaf data tidak ditemukan");

}

getch();

clrscr();

break;

case '0': break;

getch();

clrscr();

default:printf("\t\tSalah pilih...\n");

break;

}

}while(pilih!='0');

}

/////////////////////////////////

void init(void){

head = NULL;

tail = NULL;

}

/////////////////////////////////

int isEmpty(void){

if(tail == NULL) return 1;

else return 0;

}

//////////////////////////////////

void insertDepan(char databaru[30]){

TNode \*baru;

int i;

baru = new TNode;

for(i=0;i<=30;i++){

baru->data[i] = databaru[i];

}

baru->next = baru;

baru->prev = baru;

if(isEmpty()==1){

head=baru;

tail=baru;

head->next = head;

head->prev = head;

tail->next = tail;

tail->prev = tail;

}

else {

baru->next = head;

head->prev = baru;

head = baru;

head->prev = tail;

tail->next = head;

}

printf("\n\t\t\Data masuk\n");

printf("\t\tPress Enter to Continue..");

}

////////////////////////////////////////////////////

void insertBelakang (char databaru[30]){

TNode \*baru,\*bantu;

int i;

baru = new TNode;

for(i=0;i<=30;i++){

baru->data[i] = databaru[i];

}

baru->next = baru;

baru->prev = baru;

if(isEmpty()==1){

head=baru;

tail=baru;

head->next = head;

head->prev = head;

tail->next = tail;

tail->prev = tail;

}

else {

tail->next = baru;

baru->prev = tail;

tail = baru;

tail->next = head;

head->prev = tail;

}

printf("\t\tData masuk\n");

printf("\t\tPress Enter to Continue..");

}

//////////////////////////////////////////////

void tampil(void){

int i=0;

if(isEmpty()==0){

do{

i++;

printf("\t\t%i. %s\n",i,head->data);

printf("\t\t===================\n");

head=head->next;

}while(head!=tail->next);

printf("\n");

} else printf("\t\t\t..Masih kosong..\n\n");

}

//////////////////////////////////////////////

void hapusDepan(void){

TNode \*hapus;

char d[30];

int i;

if (isEmpty()==0){

if(head != tail){

hapus = head;

for (i=0;i<=30;i++){

d[i] = hapus->data[i];

}

head = head->next;

tail->next = head;

head->prev = tail;

delete hapus;

} else {

for (i=0;i<=30;i++){

d[i] = head->data[i];

}

head = NULL;

tail = NULL;

}

printf("\t\t%s terhapus\n",d);

} else printf("\t\tMasih kosong\n");

}

//////////////////////////////////////////

void hapusBelakang(void){

TNode \*hapus;

char d[30];

int i;

if (isEmpty()==0){

if(head != tail){

hapus = tail;

for (i=0;i<=30;i++){

d[i] = hapus->data[i];

}

tail = tail->prev;

tail->next = head;

head->prev = tail;

delete hapus;

} else {

for (i=0;i<=30;i++){

d[i] = head->data[i];

}

head = NULL;

tail = NULL;

}

printf("\t\t%s terhapus\n",d);

} else printf("\t\tMasih kosong\n");

}

//////////////////////////////////////////

void clear(void){

TNode \*bantu,\*hapus;

if (isEmpty()==0){

bantu = head;

while(bantu->next!=head){

hapus = bantu;

bantu = bantu->next;

delete hapus;

}

head = NULL;

}

}

//////////////////////////////////////////

//////////////////////////////////////

int cari(char elemen[30]){

int i=0;

int status=0;

if(isEmpty()==0){

do{

i++;

if(head->data[i]==elemen[i]){

status=1;

}else head=head->next;

}while(head!=tail->next && i<=30);

return(status);

} else printf("\t\tMasih kosong\n");

}

/////////////////////////////////////

void inserttengah(char databaru[30], int pilihdepan, int pilihbelakang){

TNode \*baru,\*bantu,\*depan,\*belakang;

char elemen[30];

int i;

int j;

baru = new TNode;

for(i=0;i<=30;i++){

baru->data[i] = databaru[i];

}

baru->next = baru;

baru->prev = baru;

if(isEmpty()==1){

head=baru;

tail=baru;

head->next = head;

head->prev = head;

tail->next = tail;

tail->prev = tail;

}else{

depan = head;

belakang = head;

for(i=1;i<pilihdepan;i++){

depan=depan->next;

}

for(i=1;i<pilihbelakang;i++){

belakang=belakang->next;

}

depan->next = baru;

baru->prev = depan;

baru->next = belakang;

belakang->prev = baru;

}

printf("\t\tData masuk\n");

printf("\t\tPress Enter to Continue..");

}

////////////////////////////////////////////////////////

void deletetengah(int pilih){

TNode \*hapusdepan,\*hapusbelakang,\*hapustengah;

char d[30];

int i,j;

if (isEmpty()==0){

if(head != tail){

hapusdepan = head;

hapusbelakang = head;

hapustengah = head;

for(i=1;i<pilih;i++){

hapusdepan=hapusdepan->next;

}

for(i=1;i<(pilih+2);i++){

hapusbelakang=hapusbelakang->next;

}

for(i=1;i<=pilih;i++){

hapustengah=hapustengah->next;

}

for(j=1;j<pilih;j++){

for (i=0;i<=30;i++){

d[i] = hapustengah->data[i];

}

hapustengah = hapustengah->next;

}

delete hapustengah;

hapusdepan->next = hapusbelakang;

hapusbelakang->prev=hapusdepan;

} else {

for (i=0;i<=30;i++){

d[i] = hapustengah->data[i];

}

head = NULL;

tail = NULL;

}

printf("\t\t%s Success terhapus\n",d);

} else printf("\t\tMasih kosong\n");

}

2. Double Linked List Non Circular

DLLNC "Double linked list non circular" adalah Double Linked List yang memiliki 2 buah pointer yaitu pointer next dan prev.

Pointer next menunjuk pada node setelahnya dan pointer prev menunjuk pada node sebelumnya.

Pengertian:

Double : artinya field pointer-nya dua buah dan dua arah, ke node sebelum dan sesudahnya.

Linked List : artinya node-node tersebut saling terhubung satu sama lain.

Non Circular : artinya pointer prev dan next-nya akan menunjuk pada NULL.

Ilustrasi DLLNC

Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya

Untuk pembentukan node baru , mulanya pointer next dan prev akan menunjuk ke nilai NULL

Selanjutnya pointer prev akan menunjuk ke node sebelumnya , dan pointer next akan menunjuk ke node selanjutnya pada list.

Deklarasi dan node baru DLLNC

Deklarasi node

Dibuat dari struct berikut ini :

typedef struct TNode {

int data ;

TNode \*next ;

Tnode \* prev ;

};

Pembentukan node baru

Digunakan keyword new yang berarti mempersiapkan sebuah node baru berserta alokasi memorinya .

TNode \* baru ;

baru = new TNode ;

baru ->data = databaru ;

baru ->next = NULL;

baru -> prev = NULL;

Contoh program double linked list non circular

#include <stdio.h>

#include <stdafx.h>

#include <conio.h>

#include <stdlib.h>

typedef struct TNode{ int data;

TNode \*next;

TNode \*prev;

} TNode;

TNode \*head=NULL;

void init();

int IsEmpty();

void InsertDepan(int value);

void InsertBelakang(int value);

void DeleteDepan();

void DeleteBelakang();

void ClearAll();

void Tampil();

void main() // ---> Program Utama

{ int data;

int key;

do

{

printf("\n\n \_\_\_\_MENU PROGRAM\_\_\_\_ \n\n");

printf(" [1] Insert Depan \n");

printf(" [2] Insert Belakang \n");

printf(" [3] Hapus Depan \n");

printf(" [4] Hapus Belakang \n");

printf(" [5] Hapus Semua List \n");

printf("\n Pilihan Anda [1-5] --> ");scanf("%d",&key);

system("CLS");

if(key==1)

{

printf("\n Masukan Data : "); scanf("%d",&data);

InsertDepan(data);

Tampil();

}

else if(key==2)

{

printf("\n Masukan Data : "); scanf("%d",&data);

InsertBelakang(data);

Tampil();

}

else if(key==3)

{

DeleteDepan();

getch();

Tampil();

}

else if(key==4)

{

DeleteBelakang();

getch();

Tampil();

}

else if(key==5)

{

ClearAll();

printf("\n\n Semua List Sudah Di Hapus \n");

}

else

{

printf("\n Pilihan Anda Salah \n");

}

getch();

}

while(key);

} // ---> En Program Utama

void init() // ---> Inisiasi

{

head = NULL;

}

int IsEmpty() // ---> Mengecekan kondisi [kosong/tidak]

{ if(head==NULL)

return 1;

else

return 0;

}

void InsertDepan(int value) // ---> Menambahkan data dari depan

{ TNode \*baru;

baru = new TNode;

baru ->data = value;

baru ->next = NULL;

baru ->prev = NULL;

if(IsEmpty()==1)

{

head = baru;

head ->next=NULL;

head ->prev=NULL;

}

else

{

baru ->next=head;

head ->prev=baru;

head=baru;

}

printf(" Data Masuk --> ");

}

void InsertBelakang(int value) // ---> Menambahkan data dari belakang

{

TNode \*baru, \*bantu;

baru = new TNode;

baru ->data = value;

baru ->next = NULL;

baru ->prev = NULL;

if(IsEmpty()==1)

{

head = baru;

head ->next = NULL;

head ->prev = NULL;

}

else

{

bantu=head;

while(bantu ->next != NULL)

{

bantu = bantu ->next;

}

bantu ->next=baru;

baru ->prev=bantu;

}

printf(" Data Masuk --> ");

}

void DeleteDepan() // ---> Menghapus data dari depan

{

TNode \*hapus;

int d;

if(IsEmpty()==0)

{

if(head ->next !=NULL)

{

hapus = head;

d=hapus ->data;

head=head ->next;

head ->prev = NULL;

delete hapus;

}

else

{

d=head->data;

head=NULL;

}

printf("\n %d Data Terhapus

\n",d);

printf(" Maka Menjadi -->

");

}

else

printf("\n Masih Kosong -->

");

}

void DeleteBelakang() // ---> Menghapus data dari belakang

{

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;

}

printf("\n %d terhapus

\n",d);

printf(" Maka Menjadi -->

");

}

else printf("\n Masih Kosong -->

");

}

void ClearAll() // ---> Mengahapus semua data(hapus list)

{

TNode \*bantu, \*hapus;

bantu=head;

while(bantu !=NULL)

{

hapus=bantu;

bantu=bantu ->next;

delete hapus;

}

head=NULL;

}

void Tampil() // ---> Menapmpilkan list

{

TNode \*bantu;

bantu=head;

if(IsEmpty()==0)

{

while(bantu !=NULL)

{

printf(" [%d]", bantu

->data);

bantu=bantu ->next;

}

printf("\n ");

}

else

printf(" Data Kosong");

}