

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

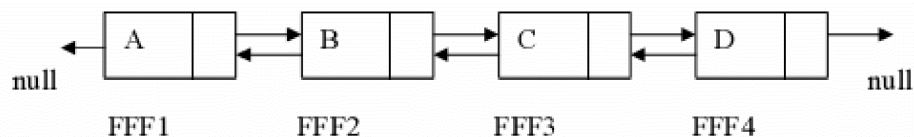
DOUBLE LINKED LIST NON CIRCULAR

TUJUAN

1. Mahasiswa memahami dan mengerti mengenai *double linked list non circular* dalam C++
2. Mahasiswa mampu membuat program dengan menggunakan *double linked list non circular* dalam pemrograman C++

DASAR TEORI

Double linked list merupakan senarai berantai bersambung yang memiliki 2 buah pointer yang menunjuk ke node sebelum dan sesudahnya. Linked list ini memiliki 2 arah yang ditunjukkan dengan 2 buah pointer penunjuk yaitu *prev* dan *next* seperti pada gambar berikut ini :



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.

Pada dasarnya struktur data dan operasi pada *double linked list non circular* mirip dengan *single linked list non circular*. Hanya pada *double linked list non circular*, maka ditambahkan sebuah pointer sebagai penunjuk node sebelumnya yaitu *prev*. Pointer ini nantinya akan diubah dan diperbaharui isi dan posisinya sesuai dengan operasi yang dilakukan terhadap list-nya. Operasi-operasinya meliputi penambahan data baru, penghapusan data, dan penampilan isi list-nya. Sama seperti pada *single linked list*, maka *double linked list* juga dapat disusun menggunakan *head* dan *tail*. Pointer-pointer ini nantinya juga harus diperbaharui jika terjadi operasi-operasi pada list-nya.

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.

```
/*
ddlnc
*/

//lib
#include <stdio.h>
#include <conio.h>

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

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();

//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;
    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 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;
    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 baru telah dimasukkan di belakang\n");
}

```

```

//menampilkan list
void tampilList()
{
    TNode *bantu;
    bantu = head;
    if(isEmpty()==0)
    {
        while(bantu!=NULL)
        {
            printf("%d\t",bantu->data);
            bantu=bantu->next;
        }
        printf("\n");
    }
    else
        printf("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;
            head->prev = NULL;
            delete hapus;
        }
        else
        {
            d = head->data;
            head = NULL;
        }
        printf("%d terhapus\n",d);
    }
    else
        printf("Masih kosong\n");
}

```

```

//hapus data terakhir
void hapusBelakang()
{

```

```

    TNode *hapus;

```

```

    int d;
    if (isEmpty()==0)
    {
        if(head->next != NULL)
        {
            hapus = head;
            while(hapus->next!=NULL)
            {
                hapus = hapus->next;
            }
            d = hapus->data;
            hapus->prev->next = NULL;
            delete hapus;
        }
        else
        {
            d = head->data;
            head = NULL;
        }
        printf("%d terhapus\n",d);
    }
    else printf("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;
}

//main prog
int main()
{
    printf("double linked list non circular\n1. inisialisasi head ... \t");
    initHead();
    printf("done\ntampilkan isi list :\n");
    tampilList() ;

    //entry data di depan
    printf("\nentry data di depan list\n");
}

```

```

int data_baru;
for(int i=1;i<=5;i++)
{
    printf("masukkan data ke-%d : ",i);
    scanf("%d",&data_baru);
    insertDepan(data_baru);
}
printf("tampilkan isi list :\n");
tampilList() ;

//entry data di belakang
printf("\nentry data di belakang list\n");
for(int i=1;i<=5;i++)
{
    printf("masukkan data ke-%d : ",i);
    scanf("%d",&data_baru);
    insertBelakang(data_baru);
}
printf("tampilkan isi list :\n");
tampilList() ;

//hapus data di depan
printf("\nhapus 2 data terdepan\n");
for(int i=1;i<=2;i++)
{
    hapusDepan();
}
printf("tampilkan isi list :\n");
tampilList() ;

//hapus data di belakang
printf("\nhapus 2 data terakhir\n");
for(int i=1;i<=2;i++)
{
    hapusBelakang();
}
printf("tampilkan isi list :\n");
tampilList();

//clear semua list
printf("\nhapus semua node\n");
clearList();
printf("tampilkan isi list :\n");
tampilList();

getch();
return 0;
}

```

TUGAS

Buatlah program menggunakan double linked list non circular untuk menyimpan dan menampilkan data berikut dari belakang (urutan terbalik) :

1. Janu
2. Dadu
3. Ayu
4. Kiki
5. Budi