

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

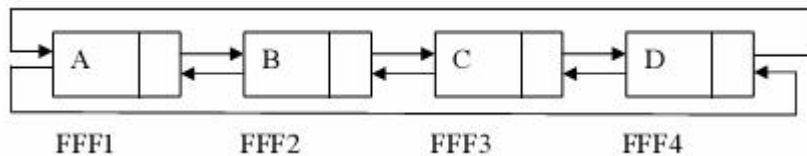
### DOUBLE LINKED LIST CIRCULAR

#### TUJUAN

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

#### DASAR TEORI

*Double Linked List Circular* adalah linked list dengan menggunakan *pointer*, dimana setiap *node* memiliki 3 *field*, yaitu 1 *field pointer* yang menunjuk *pointer* berikutnya (*next*), 1 *field* menunjuk *pointer* sebelumnya (*prev*), serta sebuah *field* yang berisi data untuk *node* tersebut. Ilustrasi dari *linked list* ini dapat dilihat 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 dirinya sendiri. Jika sudah lebih dari satu *node*, maka *pointer prev* akan menunjuk ke *node* sebelumnya, dan *pointer next* akan menunjuk ke *node* sesudahnya.

Pada dasarnya struktur data dan operasi pada *double linked list circular* mirip dengan *double linked list non circular*. Hanya pada *double linked list circular*, maka *node* terakhir selalu menunjuk ke *node* terdepan. *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.

```
/*
dllc
*/

//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, *bantu;
    baru = new TNode;
    baru->data = databaru;
    baru->next = baru;
    baru->prev = baru;

    if(isEmpty()==1)
    {
        head=baru;
        head->next = head;
        head->prev = head;
    }
    else
    {
        bantu = head->prev;
        baru->next = head;
        head->prev = baru;
        head = baru;
        head->prev = bantu;
        bantu->next = head;
    }
    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 = baru;
    baru->prev = baru;
    if(isEmpty()==1)
    {
        head=baru;
        head->next = head;
        head->prev = head;
    }
    else
    {
        bantu=head->prev;
        bantu->next = baru;
        baru->prev = bantu;
        baru->next = head;
        head->prev = baru;
    }
    printf("Data baru telah dimasukkan di belakang\n");
}

```

```

}

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

//hapus data terdepan
void hapusDepan()
{
    TNode *hapus,*bantu;
    int d;
    if (isEmpty()==0)
    {
        if(head->next != head)
        {
            hapus = head;
            d = hapus->data;
            bantu = head->prev;
            head = head->next;
            bantu->next = head;
            head->prev = bantu;
            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,*bantu;
    int d;
    if (isEmpty()==0)
    {
        if(head->next != head)
        {
            bantu = head;
            while(bantu->next->next != head)
            {
                bantu = bantu->next;
            }
            hapus = bantu->next;
            d = hapus->data;
            bantu->next = head;
            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;
    if (isEmpty()==0)
    {
        bantu = head;
        while(bantu->next!=head)
        {
            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 circular* untuk menyimpan dan menampilkan data berikut dari data ke-3 dengan urutan terbalik :

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