

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

SINGLE LINKED LIST NON CIRCULAR

TUJUAN

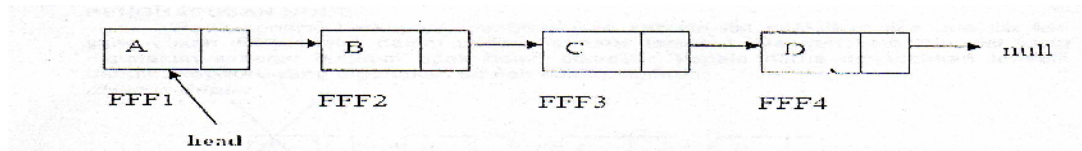
1. Mahasiswa memahami dan mengerti mengenai *single linked list non circular* dalam C++
2. Mahasiswa mampu membuat program dengan menggunakan *single linked list non circular* dalam pemrograman C++

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 :

<i>Array</i>	<i>Linked List</i>
Statis	Dinamis
Penambahan dan penghapusan data terbatas	Penambahan dan penghapusan data tidak terbatas
<i>Random access</i>	<i>Sequential access</i>
Penghapusan <i>array</i> tidak mungkin	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.

```

/*
sllnc
*/

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

//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();

//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;
    }
    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;
    if(isEmpty()==1)
    {
        head=baru;
        head->next = NULL;
    }
    else
    {
        bantu=head;
        while(bantu->next!=NULL)
        {
            bantu=bantu->next;
        }
        bantu->next = baru;
    }
    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;
            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 != 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("%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("single linked list non circular\n1. inialisasi 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

1. Apakah kegunaan dari fungsi *delete* pada program sebelumnya? Berada pada header apakah fungsi *delete*?
2. Buatlah program menggunakan *Single Linked List Non Circular* untuk menyimpan Nama dan NIM data mahasiswa berikut ini :

Nama	NIM
Dede Fahri	12347867
Kiki Fatmala	98765674
Yuki	67453279

Tampilkan data dari *list*-nya dengan format sebagai berikut :

Nama	NIM
Dede Fahri	12347867
Kiki Fatmala	98765674
Yuki	67453279