

## PERTEMUAN 10: APLIKASI LINKED LIST UNTUK STACK

### A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai aplikasi dan penerapan linked list pada stack dan queue. Di modul ini, Anda harus mampu:

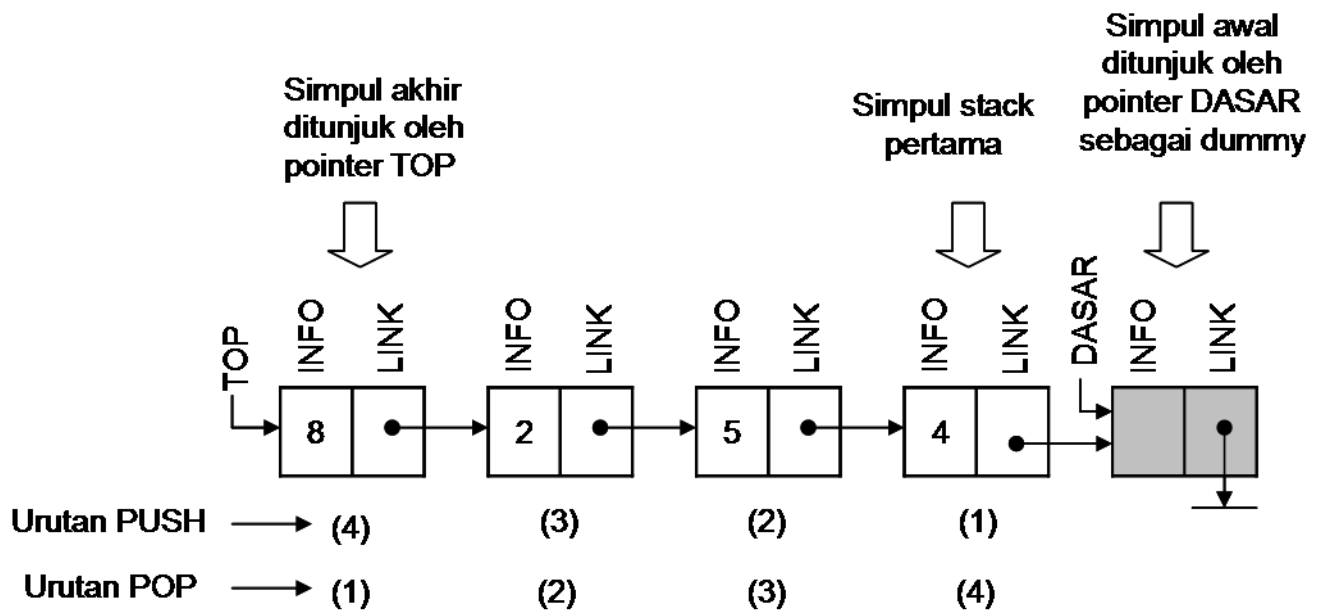
- 10.1 Merepresentasikan dan membuat aplikasi Stack dan Queue menggunakan Linked List dalam bahasa pemrograman .

### B. URAIAN MATERI

*Tujuan Pembelajaran 10.1:*

*Aplikasi Linked List Pada Stack*

#### I. Ilustrasi untuk STACK dengan menggunakan Simpul Head



PUSH : Selalu Insert Kiri

POP : Selalu Delete Kiri

Fungsi-fungsi yang diperlukan :

- 1) Deklarasi struktur simpul dan pointer yang diperlukan

```
struck SEMPUL{
    int INFO;
    struck SEMPUL *LINK;
};
SEMPUL *P,*Q,*TOP,*DASAR;
```

- 2) Inisialisasi stack

```
TOP = NULL;
DASAR = NUL;
```

- 3) Fungsi pembuatan simpul baru

```
void BUAT_SEMPUL(int X)
{
    P=(SEMPUL *)malloc(sizeof(SEMPUL));
    if(P!=NULL)
        P->INFO=X;
    else
    {
        cout<<"Membuat simpul gagal";
        exit(1);
    }
}
```

- 4) Fungsi pembuatan simpul Head

```
void BUAT_HEAD(void)
{
    if(DASAR==NULL)
    {
        DASAR=P;
        TOP=DASAR;
        DASAR->LINK=NULL;
        DASAR->INFO=0;
    }
    else
```

```
    cout<<"Head sudah ada";  
}
```

#### 5) Fungsi PUSH/Insert Kiri

```
void PUSH(void)  
{  
    if(DASAR!=NULL)  
    {  
        P->LINK=TOP;  
        TOP=P;  
    }  
    else  
        cout<<"Stack belum ada";  
}
```

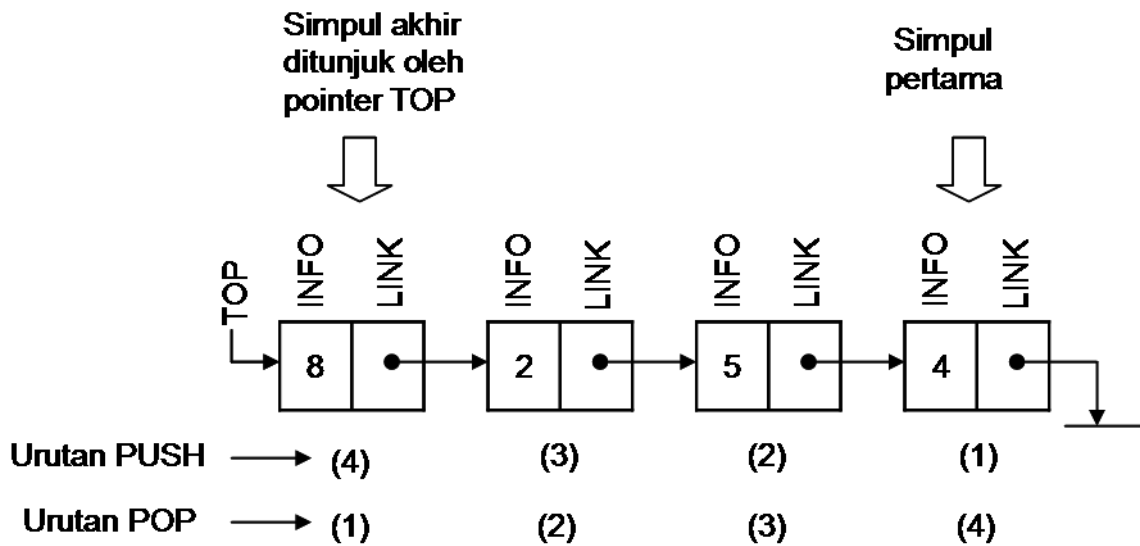
#### 6) Fungsi POP/Delete Kiri

```
int POP(void)  
{  
    int X;  
    if(TOP!=DASAR)  
    {  
        X=TOP->INFO;  
        Q=TOP->LINK;  
        free(TOP);  
        TOP=Q;  
        return(X);  
    }  
    else  
        cout<<"Stack kosong";  
}
```

**TUGAS I:**

Buat program animasi Stack menggunakan Linked List untuk mengelola data mahasiswa dengan struktur mahasiswa sbb : NAMA, NIM, GENDER, NILAI STRUKTUR DATA. Program dibuat dalam bentuk menu dengan pilihan : INSERT DATA, HAPUS DATA, CETAK DATA, EXIT

## II. Ilustrasi untuk STACK tanpa menggunakan simpul Head



PUSH : selalu insert kiri

POP : selalu delete kiri

Jika TOP->LINK=NULL berarti isi stack tinggal satu simpul (simpul pertama) dan bila simpul ini dihapus (POP) maka TOP dibuat sama dengan NULL.

Fungsi-fungsi yang diperlukan :

1) Deklarasi struktur simpul dan pointer yang diperlukan

```
struct SIMPUL{
    int INFO;
    struct SIMPUL *LINK;
};
SIMPUL *P,*Q,*TOP;
```

2) Inisialisasi stack

```
TOP=NULL
```

3) Fungsi pembuat simpul baru

```
void BUAT_SIMPUL(int X)
{
    P=(SIMPUL *)malloc(sizeof(SIMPUL));
    if(P!=NULL)
```

```
        P->INFO=X;
    else
    {
        cout<<"Membuat simpul gagal";
        exit(1);
    }
}
```

#### 4) Fungsi PUSH (Insert Kiri atau Buat Awal)

```
void PUSH(void)
{
    if(TOP==NULL)
    {
        TOP=P;
        TOP->LINK=NULL;
    }
    else
    {
        P->LINK=TOP;
        TOP=P;
    }
}
```

#### 5) Fungsi POP (Delete Kiri)

```
int POP(void)
{
    int X;
    if(TOP!=NULL)
    {
        X=TOP->INFO;
        Q=TOP->LINK;
        free(TOP);
        TOP=Q;
        return(X);
    }
    else
```

```
cout<<"Stack kosong";  
}
```

#### TUGAS II :

Buat program animasi Stack menggunakan Linked List tanpa Head untuk mengelola data mahasiswa dengan struktur mahasiswa sbb : NAMA, NIM, GENDER, NILAI STRUKTUR DATA. Program dibuat dalam bentuk menu dengan pilihan : INSERT DATA, HAPUS DATA, CETAK DATA, EXIT.

## **C. DAFTAR PUSTAKA**

### **Buku**

1. Esakov, Jeffrey, Tom Weiss, Data Structures An Advanced Approach Using C, Prentice-Hall, Inc. 1989
2. Hariyanto, Bambang, Struktur Data, Informatika Bandung, Pebruari 2000
3. Kadir, Abdul, Pemrograman Dasar Turbo C, Andi Offset, Yogyakarta, 1991
4. Kruse, Robert L. Data Structures & Program Design, Prentice-Hall, Inc. 1987
5. Standish, Thomas A. Data Structures, Algorithms & Software Principles In C, Addison Wesley, 1995