

Stack / Tumpukan



Akmal, S.Si, MT

Mata Kuliah : Struktur Data

Tujuan

- ❑ Mahasiswa dapat : menganalisis tumpukan dan menyajikannya dengan operasi push dan pop baik dengan array maupun dengan list berkait dengan benar

Pokok Bahasan

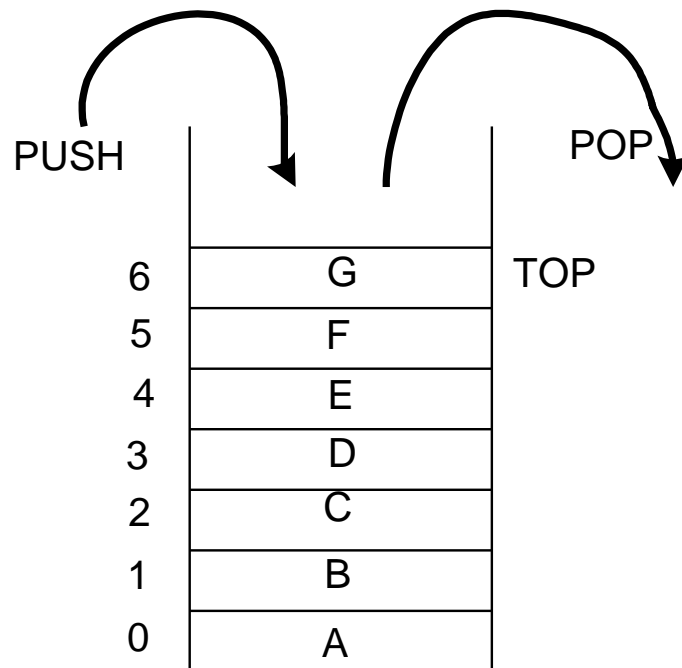
- ❑ Pengertian Tumpukan
- ❑ Penyajian tumpukan dengan Array dan List
- ❑ Operasi Push dan Pop
- ❑ Pemakaian Tumpukan

Pengertian Stack

- ❑ Tumpukan / Stack adalah satu set atau urutan elemen data dimana manipulasi data dari elemen-elemen hanya diperbolehkan pada tumpukan teratas dari data tersebut. Hal ini merupakan perintah pengumpulan data secara linier yang disebut "Last In, First Out" (LIFO).
- ❑ Stack berguna untuk bermacam-macam aplikasi seperti pattern recognition dan pengkonversian antar notasi infix, postfix dan prefix.
- ❑ Dua operasi yang dihubungkan dengan stack adalah operasi **push** dan **pop**. Push berarti memasukkan data ke dalam stacks yang paling atas sedangkan pop untuk mengambil data atas tumpukan. Sementara **top** digunakan sebagai penunjuk/pointer yang selalu menunjuk elemen paling atas.

Ilustrasi Stack

- Untuk memahami bagaimana cara kerja stack, pikirkan bagaimana menambah atau memindahkan sebuah piring dari tumpukan piring. Pikiran kita akan memberitahu untuk menambah atau memindahkan data hanya pada stack yang paling atas karena jika menggunakan cara lain, dapat menyebabkan tumpukan / stack akan terjatuh



Penyajian Tumpukan Menggunakan Array

- Disajikan tumpukan sebagai record dengan atributnya yaitu atribut pertama untuk menyimpan elemen tumpukan dan atribut yang kedua suatu nilai integer untuk mencatat posisi ujung (TOP) tumpukan.

```
const int maxElemen = 255;
struct Stack {
    char isi[maxElemen];
    int TOP;
}
Stack S;
```

- Fungsi yang perlu dibuat terlebih dahulu adalah fungsi create Stack yaitu membuat sebuah stack kosong dengan ciri posisi TOP adalah -1:

```
void createStack (Stack& S){
    S.TOP = -1;
}
```

PUSH

Operasi Push adalah fungsi untuk menambahkan elemen ke dalam stack.

Ada 2 kasus yang perlu ditangani yaitu :

1. Stack sudah penuh dengan ciri nilai $TOP = \text{maxElemen}-1$, maka tidak ada penambahan elemen
2. Stack belum penuh, maka langkah yang dilakukan adalah :
 - * Menaikkan posisi TOP ke atas ($TOP++$)
 - * Menyisipkan elemen baru ke posisi TOP yang baru tsb

```
void push(Stack& S, char elemenBaru) {  
    if (S.TOP == maxElemen-1){  
        cout<<"Tumpukan sudah penuh "<<endl;  
    }  
    else{  
        S.TOP=S.TOP + 1;  
        S.isi[S.TOP] = elemenBaru;  
    }  
}
```

POP

- ❑ Operasi Pop adalah fungsi untuk menghapus elemen yang terletak pada posisi paling atas dari tumpukan.

Ada 2 kasus yang harus ditangani yaitu :

1. Kasus Stack sudah kosong, dengan ciri nilai TOP sama dengan -1 ($TOP < 0$), maka tidak ada penghapusan
2. Stack masih ada isi , maka langkah yang dilakukan adalah :
 - * Mengamankan elemen pada posisi TOP tsb
 - * Menurunkan posisi TOP ke bawah (TOP --)

```
void pop(Stack& S, char& elemenHsl){  
    if (S.TOP < 0) {  
        cout<<"Tumpukan sudah kosong "<<endl;  
    }  
    else {  
        elemenHsl= S.isi[S.TOP];  
        S.TOP=S.TOP - 1;  
    }  
}
```



```

void traversal(Stack& S){
    int idxBantu=S.TOP;                //idxbantuan mulai dari TOP
    while (idxBantu >=0) {
        cout << S.isi[idxBantu] <<"    " ;
        idxBantu=idxBantu-1;
    }
    cout << endl;
}

```

```

main() {
    Stack myTumpukan;
    char hasil;
    cout<<"Program STACK menggunakan array"<<endl;

    createStack(myTumpukan);
    push(myTumpukan, 'A');
    push(myTumpukan, 'B');
    push(myTumpukan, 'C');

    traversal(myTumpukan);                // C    B    A

    pop(myTumpukan, hasil);
    cout<<"Setelah pop : " << hasil<<endl;    // C
    traversal(myTumpukan);                // B    A

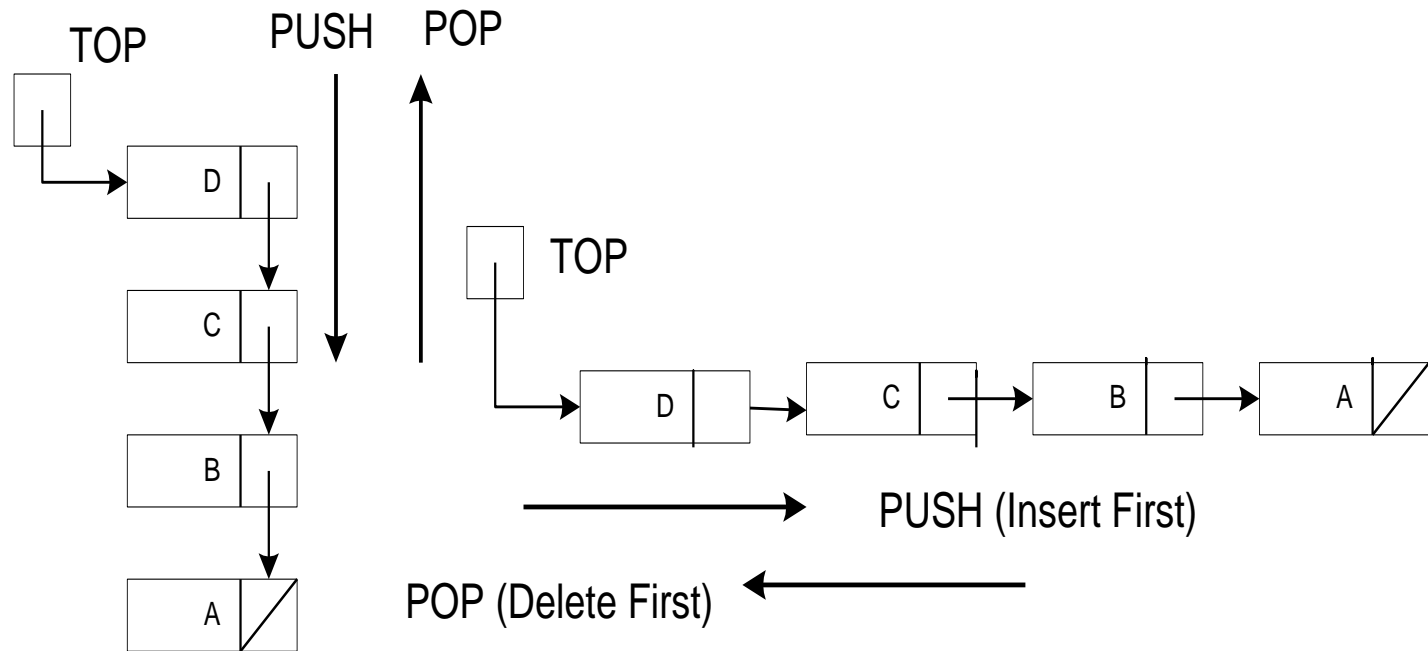
    pop(myTumpukan, hasil);
    cout<<"Setelah pop : " << hasil<<endl;    // B
    traversal(myTumpukan);                // A

    pop(myTumpukan, hasil);
    cout<<"Setelah pop : " << hasil<<endl;    // A
    traversal(myTumpukan);                // Stack Kosong
}

```

Penyajian Tumpukan Menggunakan List Berkait

- Dengan menggunakan List Berkait, maka karakteristik operasi PUSH adalah Operasi Insert First dan Operasi POP adalah Operasi dari Delete First;



PUSH

Seperti Insert First pada Singly Linked List maka ada 2 kasus yang perlu ditangani yaitu

- kasus stack kosong
- kasus stack ada isi

```
void push(Stack& Top, pointer pBaru) {  
    cout<<"Push Stack "<<endl;  
    if (Top==NULL)                                // Stack kosong  
        Top=pBaru;  
    else {                                        // Stack ada isi  
        pBaru->next=Top;  
        Top=pBaru;  
    }  
}
```

POP

- Seperti Delete First pada Singly Linked List, maka ada 3 kasus yang perlu ditangani yaitu :
 - Kasus stack kosong
 - Kasus Stack dengan isi 1 elemen
 - Kasus Stack dengan isi lebih dari 1 elemen

```
void Pop(Stack& Top, pointer& pHapus){
    cout<<"Pop Stack "<<endl;
    if (Top==NULL){
        cout<<"Stack kosong "<<endl;
        pHapus=NULL;
    }
    else if (Top->next==NULL){
        pHapus=Top;
        Top=NULL;
    }
    else {
        pHapus=Top;
        Top=Top->next;
        pHapus->next=NULL;
    }
}
```

Latihan dan Tugas

- ❑ Buatlah program untuk mengolah data pegawai dengan menggunakan operasi PUSH dan POP suatu Tumpukan.
 - a. Gunakan Struktur Array
 - b. Gunakan Struktur List berkait