

TUMPUKAN (STACK)

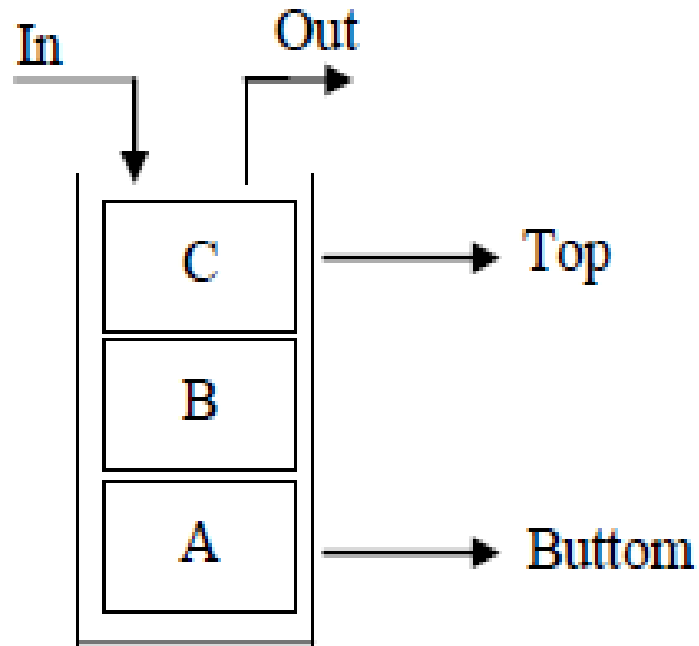
Dosen : Sulistyowati, ST., M.Kom.

PENDAHULUAN STACK

- ❑ Dalam kehidupan sehari-hari kita sering menjumpai tumpukan, seperti tumpukan piring, tumpukan buku, tumpukan batu bata, dll
- ❑ Pada contoh diatas, kita akan bisa mengambil satu elemen tumpukan jika elemen tumpukan diatasnya dapat diambil. Dengan kata lain, elemen terakhir yang ditumpuk adalah elemen yang akan diambil pertama kali.

- Dalam ilmu komputer, stack (tumpukan) bisa diartikan sebagai kumpulan data yang seolah-olah data diletakkan di atas data yang lain.
- Prinsipnya kita hanya bisa menambah data dan menghapus data pada ujung yang sama, yaitu ujung atas. Dimana data yang terakhir masuk adalah data yang akan keluar/diambil pertama kali.
Konsep ini dikenal dengan **LIFO (Last In First Out)**.
- Sifat tumpukan adalah data yang akan dimasukkan ke dalam stack ataupun yang akan dikeluarkan dari stack, hanya boleh satu per satu.

□ Ilustrasi tumpukan :



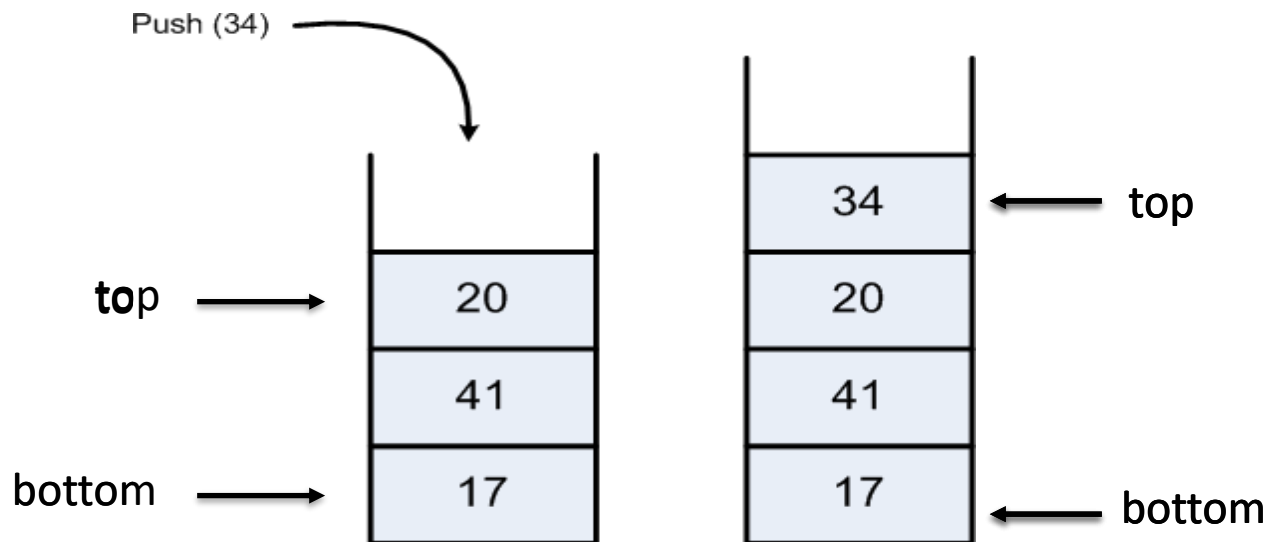
- Ada 2 operasi dasar yang bisa dilakukan pada stack yaitu :
- Push (menambah/menyisipkan data)
 - Pop (menghapus/mengurangi data)

□ PRINSIP KERJA STACK :

- Semula (pada saat stack kosong), Top dan Bottom selalu berada di index 0.
Top adalah posisi data teratas, sedangkan **Bottom** adalah posisi data terbawah
- Pada saat diisi satu data, maka posisi Top dan Bottom bergerak ke index 1. Jika diisi data lagi dan seterusnya, maka posisi Top akan bergerak naik ke index berikutnya sedangkan posisi Bottom akan tetap di index 1.
- Pada saat pengambilan/penghapusan sebuah data, maka posisi Top akan bergerak turun ke index sebelumnya dan begitu seterusnya.

PUSH

- Operasi Push bisa dilakukan jika STACK BELUM PENUH
- Ilustrasi operasi Push pada stack adalah sebagai berikut :



Pada gambar di atas mula-mula stack berisi data 17, 41, 20. Kemudian dilakukan operasi push data 34 sehingga tumpukan data pada stack menjadi 17, 41, 20, 34

□ Deklarasi stack dengan array :

```
data_type stack[20];  
int top, bottom;
```

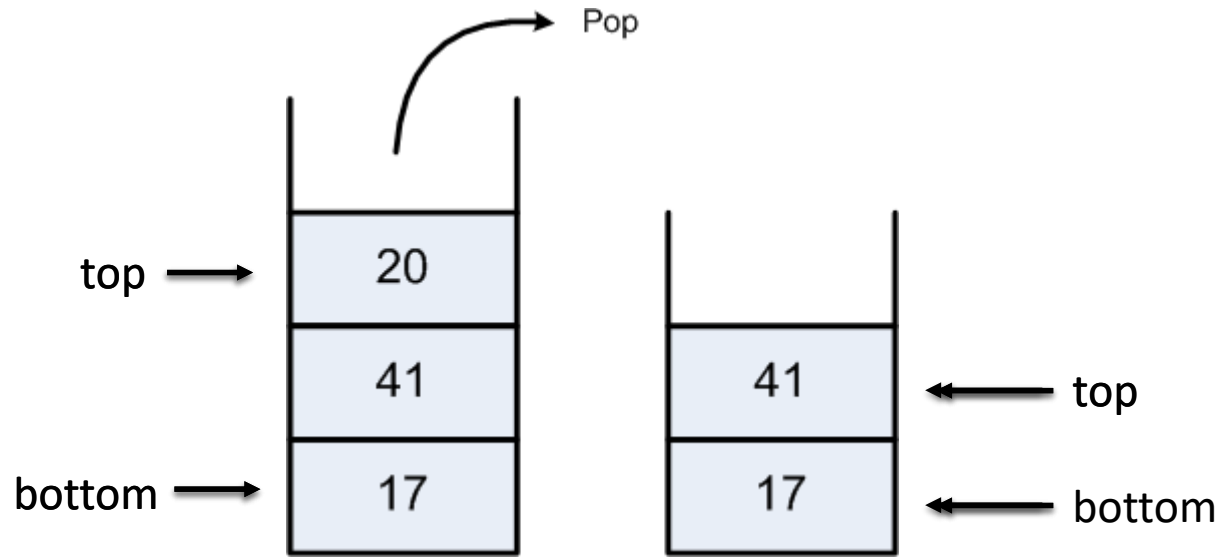
Misal jumlah maksimum
tumpukan adalah 20 data

□ Algoritma PUSH :

```
if (top < max)  
{  
    top++;  
    cin >> stack[top];  
    if (top == 1)  
        bottom++;  
}  
else  
    cout << "Stack sudah penuh";
```

POP

- Ilustrasi operasi Pop pada stack adalah sebagai berikut :



Pada gambar di atas mula-mula tumpukan berisi data 17, 41, 20.

Kemudian dilakukan operasi Pop yaitu data yang paling atas dihapus, sehingga data pada tumpukan menjadi 17, 41

□ Algoritma POP :

```
if (top>0)
{
    hapus=stack[top];
    top--;
    if (top==0)
        bottom--;
}
else
    cout<<"Stack telah kosong";
```

Perlu diingat, bahwa algoritma PUSH ataupun POP diatas, hanya berlaku untuk satu kali menambahkan atau mengambil data

LATIHAN SOAL

Sediakan dua buah stack.

1. Stack pertama, isi/PUSH dengan data karakter :
'S','T','R','U','K','T','U','R'
2. Stack kedua isi/PUSH dengan karakter :
'D','A','T','A'

Kemudian lakukan POP data dari stack secara bergantian, dimulai dari stack pertama lalu ke stack kedua, dst dan cetak data yang sudah di-POP.

Sehingga output dari proses tersebut adalah : **RAUTTAKDURTS**