

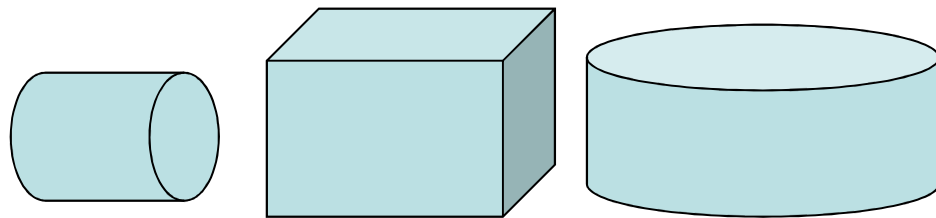
## Pertemuan 5

# **STACK atau TUMPUKAN**

# STACK (TUMPUKAN)

Merupakan bentuk khusus dari Linier List yang pemasukan dan penghapusan elemennya hanya dapat dilakukan pada satu posisi, yaitu posisi akhir dari List (Top)

Prinsip Stack adalah ***LAST-IN-FIRST-OUT (LIFO)***.



Klik untuk  
Ilustrasi Stack

# OPERASI STACK

- **IEMPTY**  
Untuk memeriksa apakah stack kosong
- **ISFULL**  
Untuk memeriksa apakah stack sudah penuh
- **PUSH**  
Untuk menambahkan item pada posisi paling atas (TOP)
- **POP**  
Untuk menghapus item paling atas (TOP)
- **CLEAR**  
Untuk mengosongkan stack

# STACK PADA ARRAY

## **Deklarasi MAX\_STACK**

```
#define MAX_STACK 5
```

## **Deklarasi STACK dengan struct dan array data**

```
typedef struct STACK{  
    int top;  
    int data[5];  
};
```

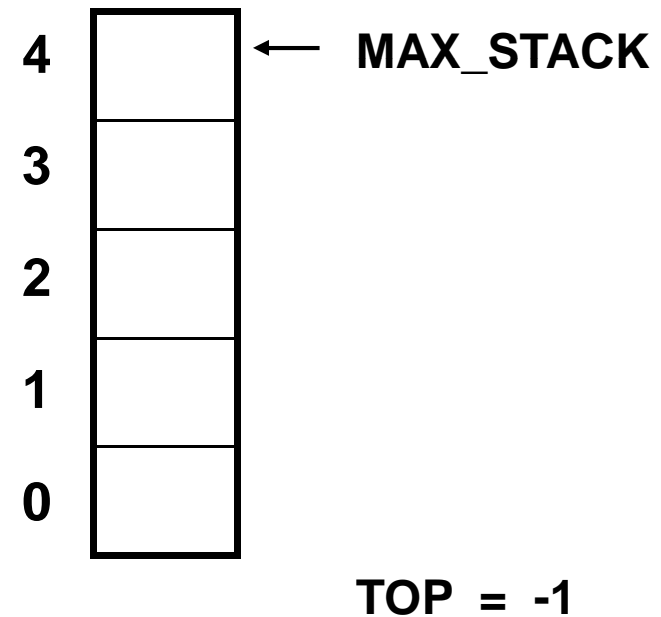
## **Deklarasi variabel stack dari struct**

```
STACK tumpuk;
```

## Inisialisasi

- Pada mulanya isi top dengan -1, karena array dalam C/C++ dimulai dari 0, berarti stack adalah KOSONG
- TOP adalah variabel penanda dalam STACK yang menunjukkan elemen teratas Stack.
- TOP of STACK akan selalu bergerak hingga mencapai MAX of STACK sehingga menyebabkan stack PENUH

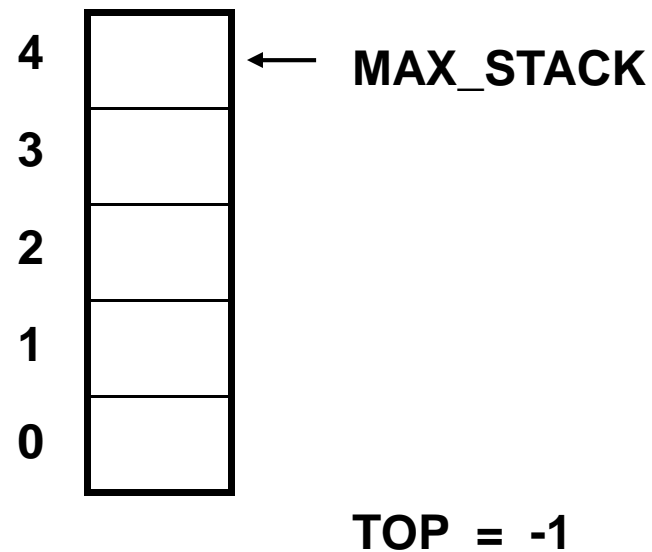
```
void inisialisasi ()  
{  
    tumpuk.top = -1  
}
```



# Fungsi IsEmpty

- Digunakan untuk memeriksa apakah stack masih dalam kondisi kosong
- Dengan cara memeriksa TOP of STACK.  
Jika TOP masih = -1 maka berarti stack masih kosong

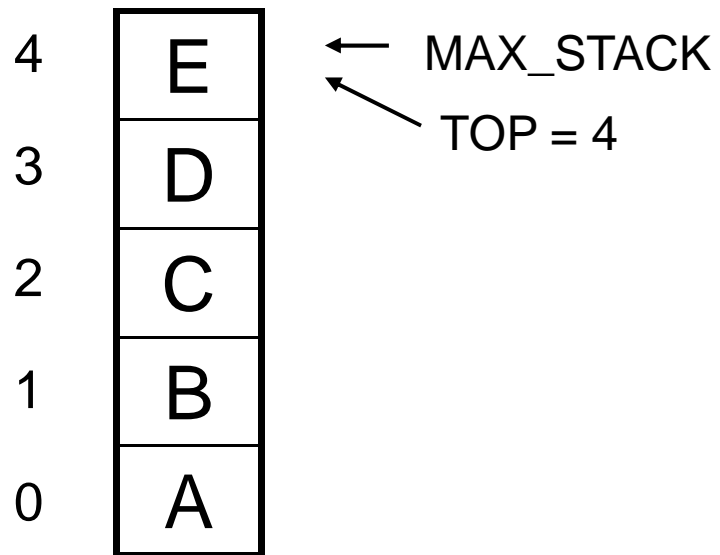
```
int IsEmpty ()  
{  
    if (tumpuk.top == -1  
        return 1;  
    else  
        return 0;  
}
```



# Fungsi IsFull

- Digunakan untuk memeriksa apakah kondisi stack sudah penuh
- Dengan cara memeriksa TOP of Stack.  
Jika  $TOP\ of\ STACK = MAX\_STACK - 1$  maka FULL (Penuh).  
Jika  $TOP\ of\ STACK < MAX\_STACK - 1$  maka belum penuh

```
int IsFull ()
{
    if (tumpuk.top == MAX_STACK-1
        return 1;
    else
        return 0;
}
```

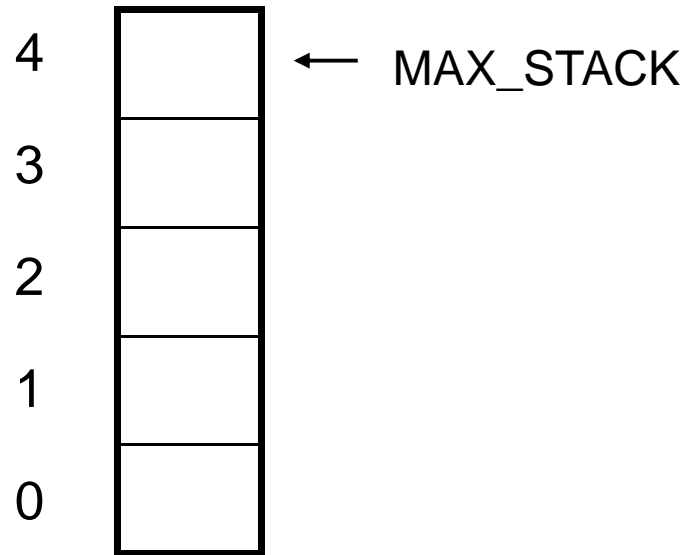




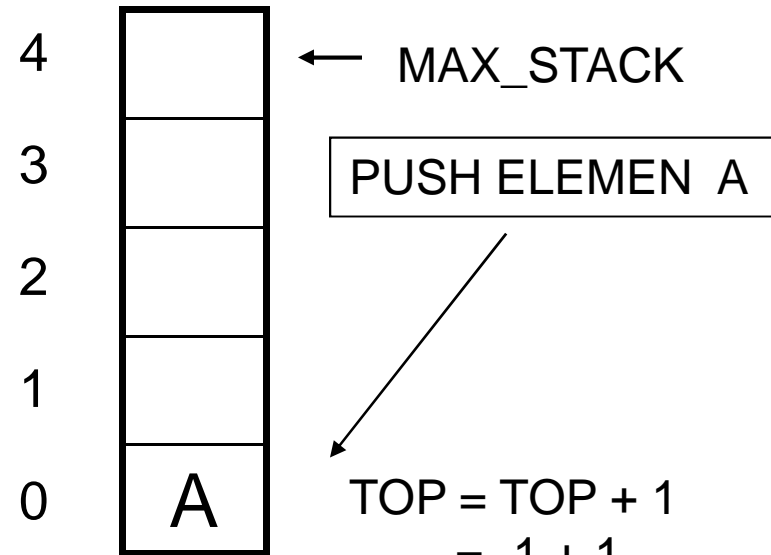
# Fungsi PUSH

- Digunakan untuk memasukkan elemen ke dalam stack dan selalu menjadi elemen teratas stack
- Dengan cara :
  1. Menambah satu (increment) nilai TOP of STACK setiap ada penambahan elemen stack selama stack masih belum penuh
  2. Isikan nilai baru ke stack berdasarkan indeks TOP of STACK setelah ditambah satu (diincrement)

```
void push (char d[5])
{
    tumpuk.top++
    strcpy(tumpuk.data[tumpuk.top],d);
}
```



TOP = -1

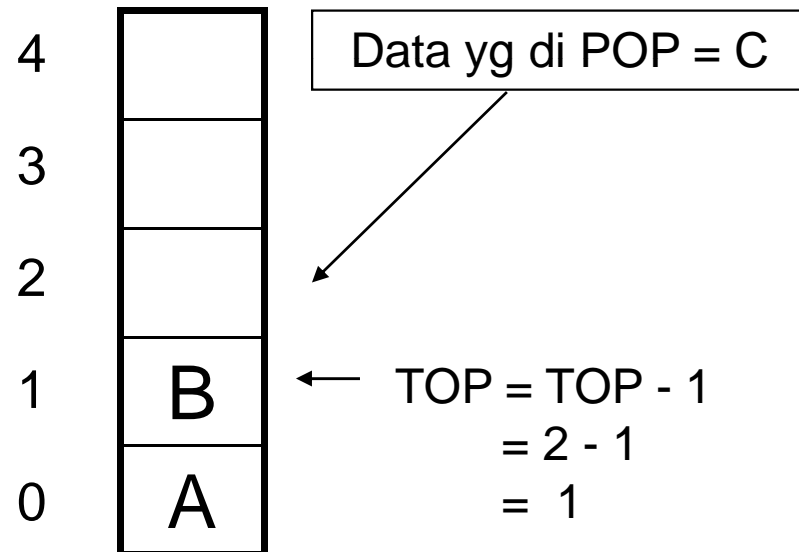
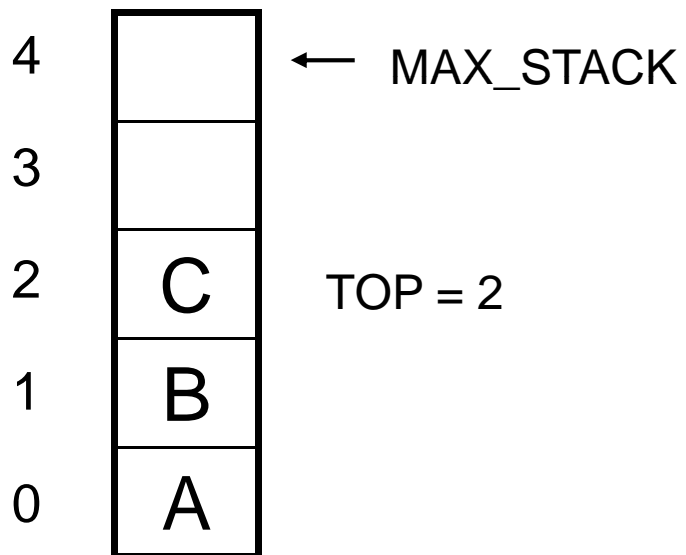


TOP = TOP + 1  
= -1 + 1  
= 0

# Fungsi POP

- Digunakan untuk menghapus elemen yang berada pada posisi paling atas dari stack.
- Dengan cara :
  1. Ambil dahulu nilai elemen teratas stack dengan mengakses TOP of STACK.
  2. Tampilkan nilai yang akan diambil.
  3. Lakukan decrement nilai TOP of STACK sehingga jumlah elemen stack berkurang 1

```
void pop ()
{
    printf("Data yg di POP = %s\n", tumpuk.data[tumpuk.top]);
    tumpuk.top--;
}
```



# Fungsi CLEAR

- Digunakan untuk mengosongkan stack / membuat stack hampa sehingga Top pada Stack berada kembali di posisi Top = -1

```
void clear ()  
{  
    tumpuk.data=tumpuk.top=-1  
    printf("Data clear");  
}
```

