

KONSEP STACK

STACK

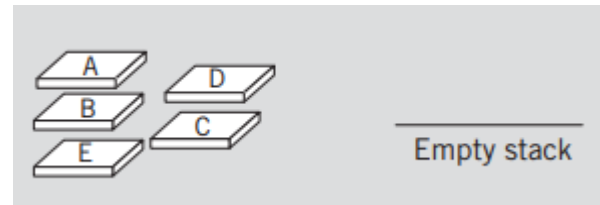
1. Stack atau tumpukan merupakan struktur data yang dapat menambahkan dan menghapus elemen hanya dapat dilakukan pada satu posisi saja, yaitu posisi akhir dari List (Top).
2. Prinsip Stack adalah struktur data Last In First Out (LIFO).

OPERASI STACK

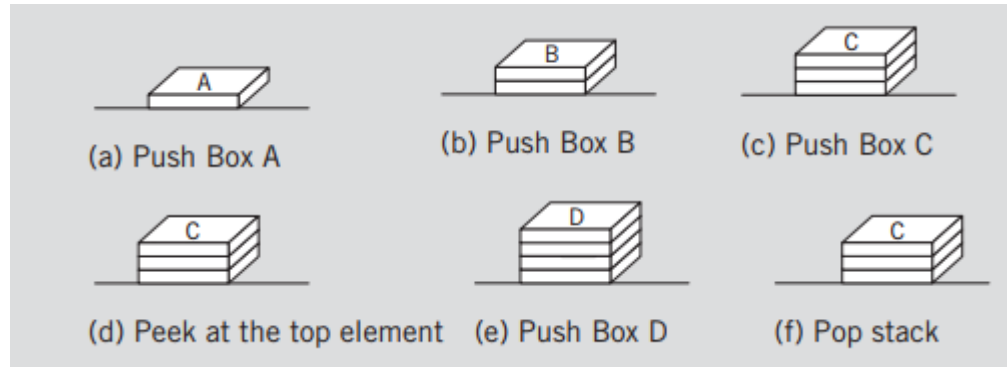
- **ISEMPTY**
Berfungsi untuk memeriksa apakah stack kosong.
- **ISFULL**
Berfungsi untuk memeriksa apakah stack sudah penuh.
- **PUSH**
Berfungsi untuk menambahkan item pada posisi paling atas (TOP).
- **TOP**
Berfungsi untuk menunjukkan item pada stack teratas.
- **POP**
Berfungsi untuk menghapus item paling atas (TOP).
- **CLEAR**
Untuk mengosongkan stack.

OPERASI STACK

ILUSTRASI



Empty Stack



Operasi utama Stack
(Push, Top, and Pop)

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**

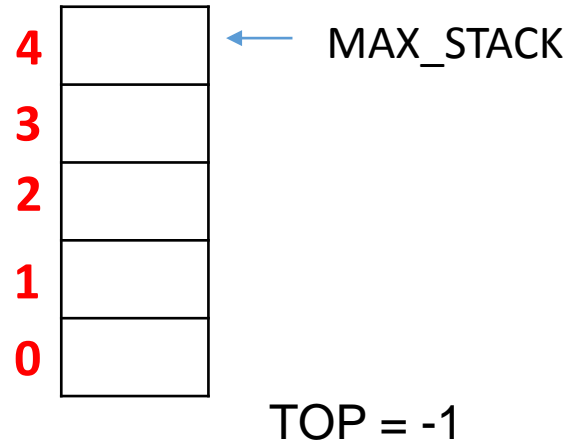
```
STACK tumpuk;
```

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.

Contoh kode:

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

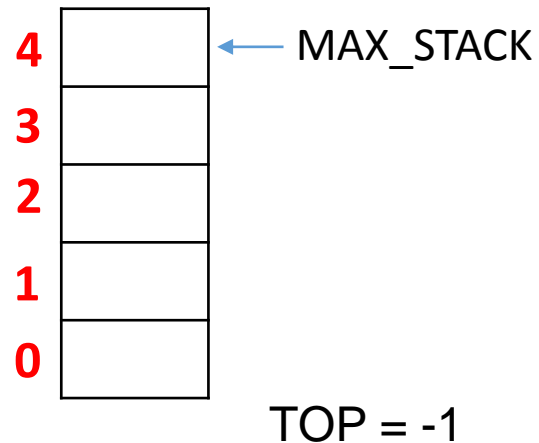


ISEMPTY

- Digunakan untuk memeriksa apakah stack masih dalam kondisi kosong.
- Penggunaan IsEmpty dilakukan dengan cara memeriksa TOP of STACK yaitu jika TOP masih = -1 maka berarti stack masih kosong.

Contoh kode:

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



ISFULL

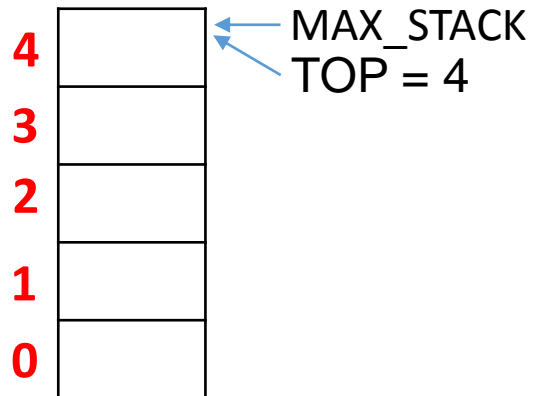
- Digunakan untuk memeriksa apakah kondisi stack penuh.
- Penggunaan IsFull dilakukan dengan cara memeriksa TOP of STACK.

Jika TOP of STACK = MAX_STACK-1 maka FULL (Penuh).

Jika TOP of STACK < MAX_STACK-1 maka belum penuh

Contoh kode:

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



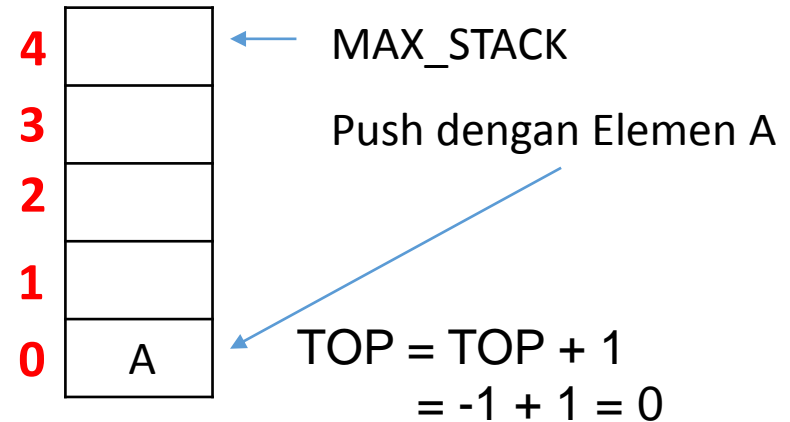
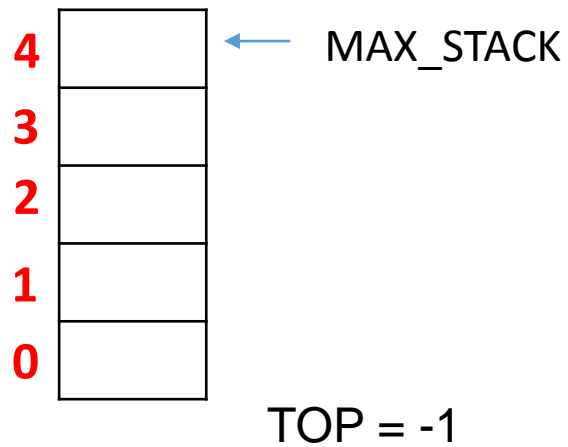
PUSH

- Digunakan untuk memasukkan elemen ke dalam stack dan selalu menjadi elemen paling atas pada stack.
- Penggunaan Push dilakukan dengan cara :
 1. Menambah satu (increment) nilai TOP of STACK setiap ada penambahan elemen stack selama stack masih belum penuh
 2. Mengisikan nilai baru ke stack berdasarkan indeks TOP of STACK setelah ditambah satu (diincrement).

PUSH

- Contoh kode:

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



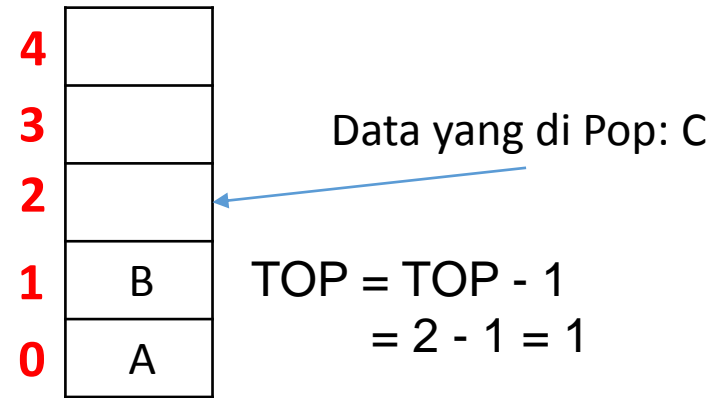
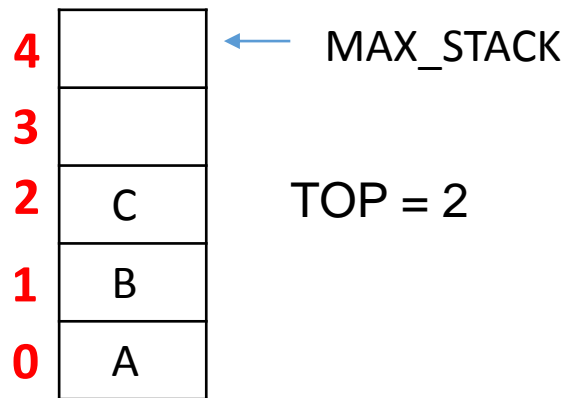
POP

- Digunakan untuk menghapus elemen yang berada pada posisi paling atas stack.
- Penggunaan Pop dilakukan dengan cara :
 1. Mengambil dahulu nilai elemen teratas stack dengan mengakses TOP of STACK.
 2. Menampilkan nilai yang akan diambil.
 3. Melakukan decrement nilai TOP of STACK sehingga jumlah elemen stack berkurang 1

POP

- Contoh kode:

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



CLEAR

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

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



PEMBELAJARAN MANDIRI

Pelajari pada buku dan sumber-sumber informasi lainnya tentang contoh program pada masing-masing operasi di Stack. Jangan lupa untuk dipraktekkan dan disimpan pada PC atau laptop kalian masing-masing.

*****SELAMAT BELAJAR*****

TERIMA KASIH