

## Pertemuan 6

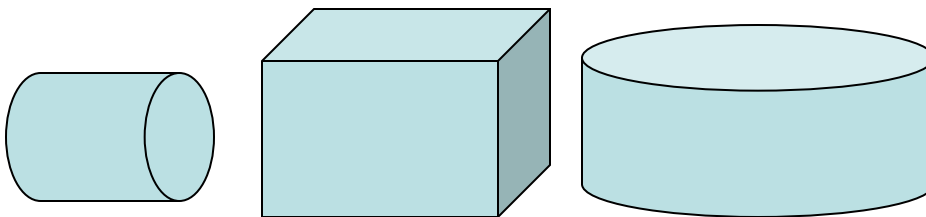
# STACK atau TUMPUKAN



# 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

- **EMPTY**  
Untuk memeriksa apakah stack kosong
- **SIZE**  
mengembalikan nilai stack
- **PUSH**  
Untuk menambahkan item pada posisi paling atas
- **POP**  
Untuk menghapus item paling atas (TOP)
- **TOP**  
mengembalikan posisi ke elemen paling atas
- **NOEL**  
menampilkan jumlah elemen pada stack

# Inisialisasi

```
stack = []

def push(value):
    stack.append(value)

def pop():
    stack.pop()

def noel():
    print len(stack)

def top():
    top = len(stack) - 1
    if top < 0:
        print "Tidak terdefinisi"
    else:
        print stack[top]

def isempty():
    if len(stack) == 0:
        print "True"
    else:
        print "False"

def tampilkan(stack):
    print stack

while True:
    value = input("--> ")
```

**stack = [ ]** ==> Digunakan untuk membuat stack dengan nama "stack" yang masih kosong

**def push(value):** ==> Digunakan untuk mendefinisikan sebuah fungsi "push" yang diisi dengan nilai

**stack.append(value)** ==> Digunakan untuk menambahkan elemen pada puncak stack dengan nilai dari fungsi "push"

**def pop():** ==> Digunakan untuk mendefinisikan sebuah fungsi "pop"

**stack.pop()** ==> Digunakan untuk menghapus elemen dari puncak stack dari fungsi "pop"

**def noel():** ==> Digunakan untuk mendefinisikan sebuah fungsi "noel"

**print len(stack)** ==> Digunakan untuk menampilkan jumlah elemen di dalam stack dengan memanggil nama "stack" yang telah dibuat

**def top():** ==> Digunakan untuk mendefinisikan sebuah fungsi "top"

**top = len(stack) - 1** ==> Sebagai perhitungan untuk mengetahui elemen pada puncak stack

**print stack[top]** ==> Maka akan menampilkan elemen pada puncak stack

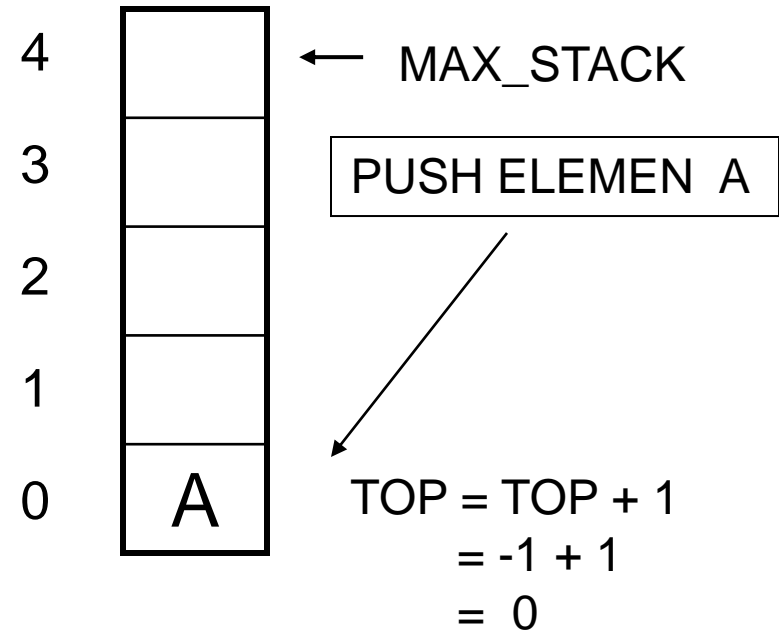
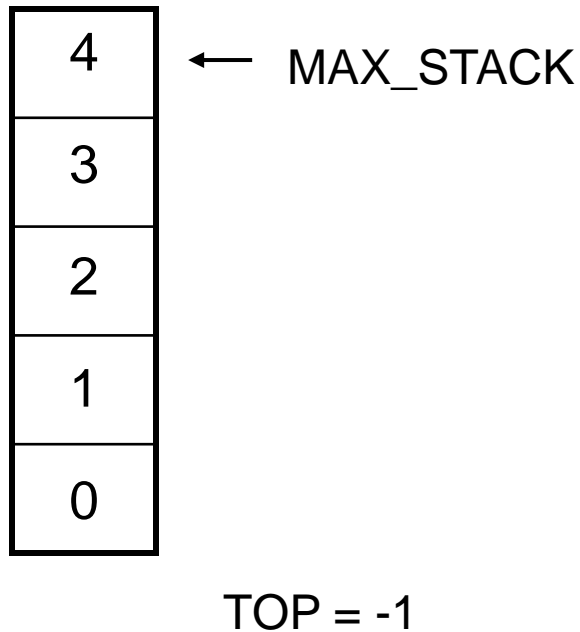
**def isempty():** ==> Digunakan untuk mendefinisikan sebuah fungsi "isempty"

**def tampilkan(stack):** ==> Digunakan untuk mendefinisikan sebuah fungsi "tampilkan"

# 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)

# Fungsi PUSH (Lanjutan)

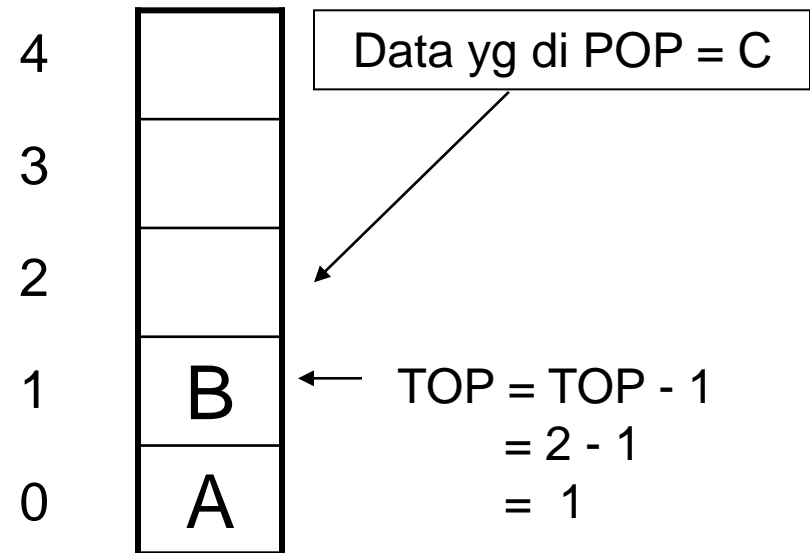
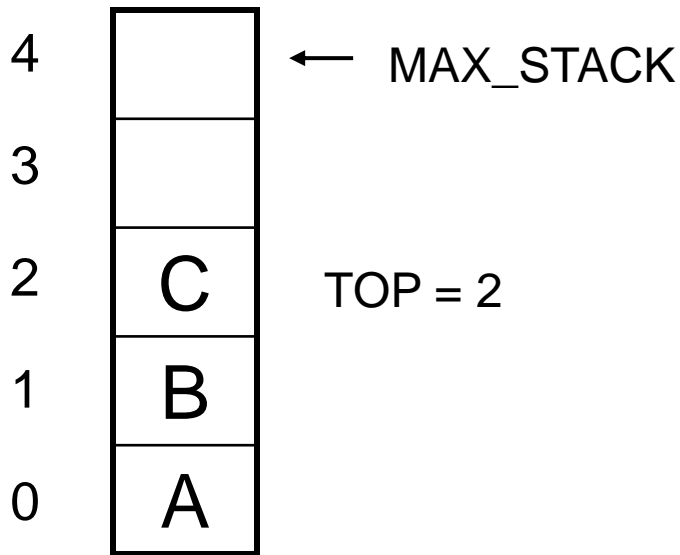


# Fungsi POP

- 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



# Fungsi POP (Lanjutan)



## Latihan

Diketahui suatu stack dgn  $\text{max\_stack} = 6$

1. Bila dilakukan PUSH 3 elemen kedalam stack, kemudian di PUSH lagi 2 elemen dan di POP 3 elemen. Maka dimana posisi Top of Stack ?
2. IsEmpty pada kondisi terakhir adalah ?
3. Dari kondisi diatas, Berapa elemen yg hrs di PUSH unt mencapai kondisi penuh Top of Stack =  $\text{max\_stack}$  ?
4. Berapa elemen yg hrs di POP unt mencapai kondisi Empty = True

```
stack = []
```

```
# append() function to push
```

```
# element in the stack
```

```
stack.append('a')
```

```
stack.append('b')
```

```
stack.append('c')
```

```
print('Initial stack')
```

```
print(stack)
```

```
# pop() function to pop
```

```
# element from stack in
```

```
# LIFO order
```

```
print('\nElements popped from stack:')
```

```
print(stack.pop())
```

```
print(stack.pop())
```

```
print(stack.pop())
```

```
print('\nStack after elements are  
popped:')
```

```
print(stack)
```

```
# uncommenting print(stack.pop())
```

```
# will cause an IndexError
```

```
# as the stack is now empty
```

## Output:

```
Initial stack  
['a', 'b', 'c']
```

```
Elements popped from stack:
```

```
c
```

```
b
```

```
a
```

```
Stack after elements are popped:
```

```
[]
```

# Latihan Soal Struktur Data (Pertemuan 6)

1. Operasi Stack yang digunakan untuk memeriksa apakah stack kosong, adalah .....

- a. PUSH
- b. POP
- c. CLEAR
- d. EMPTY
- e. SIZE

2. 

```
def noel():  
    print len(stack)
```

Merupakan fungsi...

- a. Is Full
- b. NOEL
- c. PUSH
- d. POP
- e. CLEAR

2. `def noel():`  
    `print len(stack)`

Merupakan fungsi...

- a. Is Full
- b. NOEL
- c. PUSH
- d. POP
- e. CLEAR

3. Jika pada stack terdapat kondisi TOP of STACK = MAX\_STACK - 1 maka stack berada dalam keadaan ...

- a. EMPTY
- b. CLEAR
- c. TOP
- d. RETREIVE
- e. FULL

3. Jika pada stack terdapat kondisi TOP of STACK = MAX\_STACK - 1 maka stack berada dalam keadaan ...
- a. EMPTY
  - b. CLEAR
  - c. TOP
  - d. RETREIVE
  - e. FULL
4. Prinsip dari stack adalah
- a. First in First Out (FIFO)
  - b. ISFULL
  - c. PUSH
  - d. Last in First Out (LIFO)
  - e. CLEAR

4. Prinsip dari stack adalah
- a. First in First Out (FIFO)
  - b. ISFULL
  - c. PUSH
  - d. Last in First Out (LIFO)
  - e. CLEAR

5.

```
def top():  
    top = len(stack) - 1  
    if top < 0:  
        print "Tidak terdefinisi"  
    else:  
        print stack[top]
```

Maksud dari perintah/instruksi tersebut adalah ....

- a.  $Top = Top + 1$
- b.  $Top = Top - 1$
- c.  $Top = 0$
- d.  $Top = - 1$
- e.  $Top = 1$



5.

```
def top():  
    top = len(stack) - 1  
    if top < 0:  
        print "Tidak terdefinisi"  
    else:  
        print stack[top]
```

Maksud dari perintah/instruksi tersebut adalah ....

- a.  $\text{Top} = \text{Top} + 1$
- b.  $\text{Top} = \text{Top} - 1$
- c.  $\text{Top} = 0$
- d.  $\text{Top} = - 1$
- e.  $\text{Top} = 1$

1. Operasi Stack yang digunakan untuk memeriksa apakah stack sudah penuh, adalah .....
- a. PUSH
  - b. POP
  - c. CLEAR
  - d. EMPTY
  - e. ISFULL