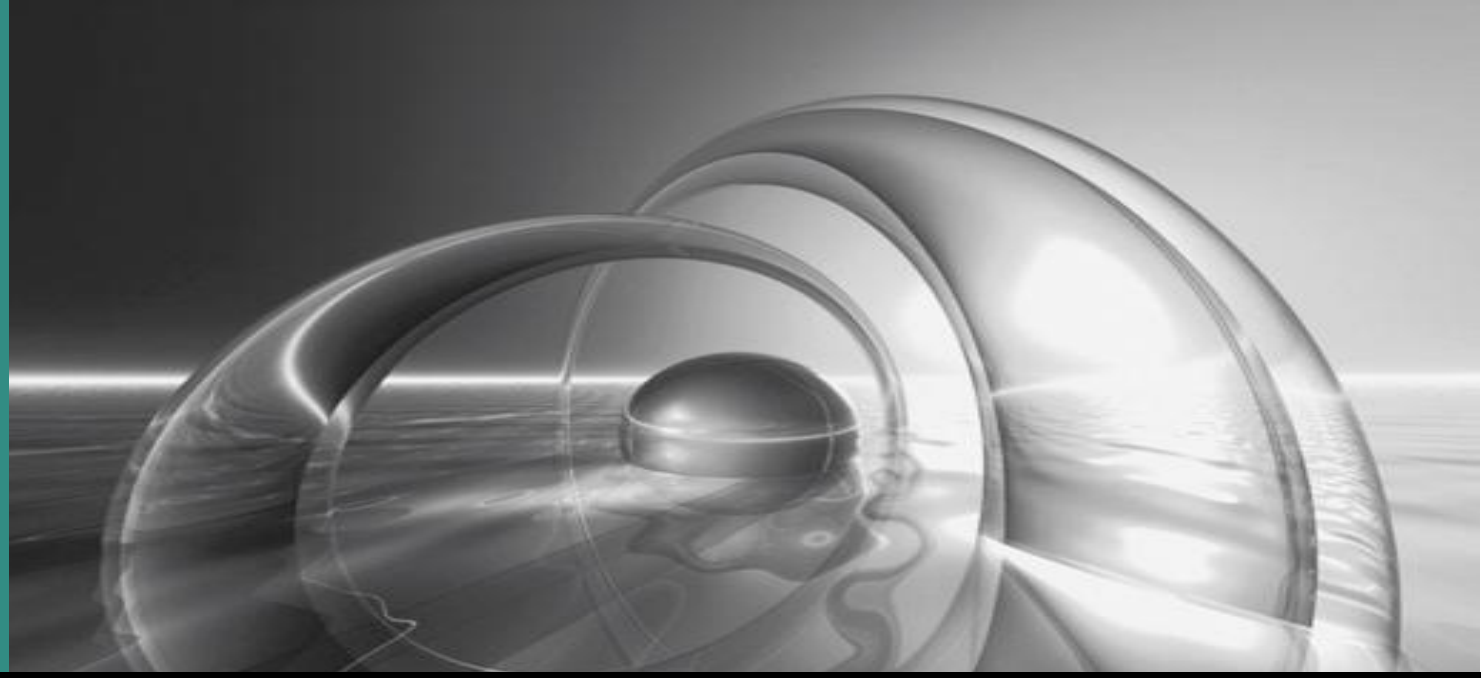
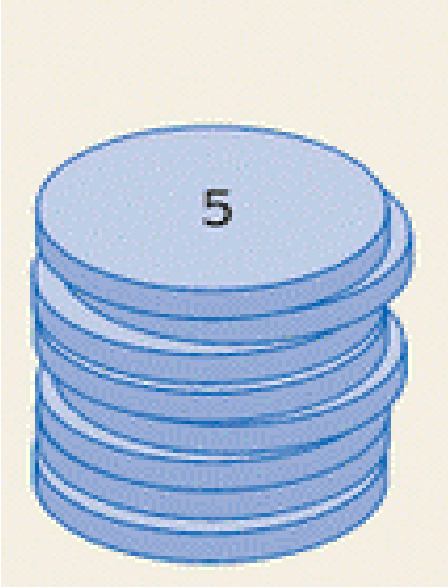


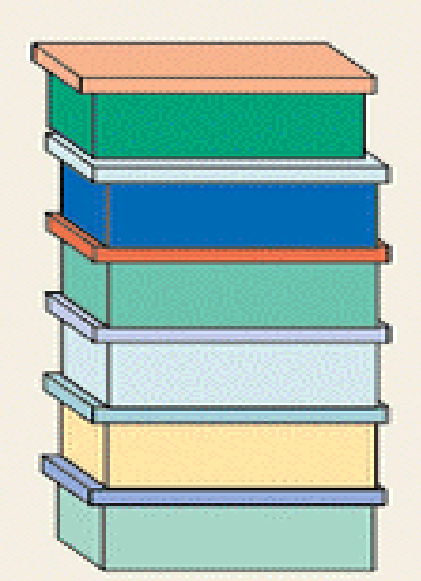
STRUKTUR DATA

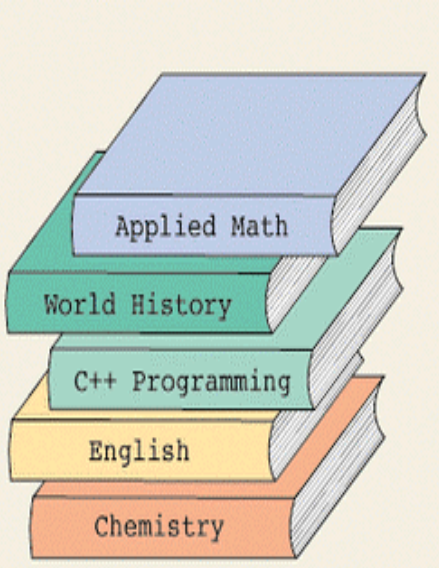
STACK

PENGERTIAN STACK

Stack (tumpukan) adalah suatu urutan elemen yang elemennya dapat diambil dan ditambah hanya pada posisi akhir (top) saja → **LIFO**







Representasi Stack

1. Array Statis

2. Linked List

Pendeklarasian Stack

Suatu Stack memiliki beberapa bagian yaitu :

- **Top** yang menunjuk posisi data terakhir (top) pada Stack
- **Elemen** yang berisi data yang ada dalam Stack. Bagian inilah yang berbentuk array.
- **MaxStack** yaitu variabel yang menunjukan maksimal banyaknya elemen dalam Stack.

Pendeklarasian Stack (Array Statis)

Const

MaxStack=……..

Type

NamaStack= array [1..MaxStack] of tipedata

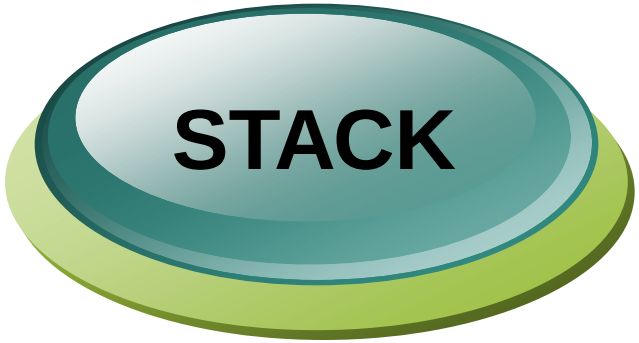
Stack : NamaStack

Top : **Integer**

Pendeklarasian Stack (Linked List)

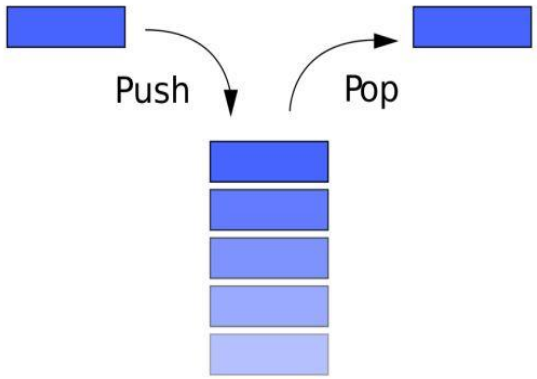
```
Type
    NamaPointer = ↑Stack
    Stack  =  Record
        MedanData              : tipedata,
        MedanSambungan         : NamaPointer
    EndRecord

Top  : NamaPointer
```

OPERASI UTAMA



PUSH

operasi menambahkan/ memasukkan sebuah data ke dalam stack.



POP

operasi mengambil / mengeluarkan sebuah data dari stack

OPERASI STACK

1. Inisialisasi
2. PUSH
3. POP

Inisialisasi

- **Array**

  Proses memberi harga awal terhadap variabel Top dengan harga 0 (nol) jika indeks pertama array diawali dengan nomor 1.

  Jika indeks pertama array dimulai dengan 0 (nol) (contoh bahasa C), maka variabel Top diberi harga awal dengan harga -1.

- **List**

  Proses memberi harga awal terhadap variabel Top dengan harga nil/NULL.

Algoritma Inisialisasi (Array)

Procedure Inisialisasi(**Output** Top : **Integer**)

{I.S. : Memberi harga awal terhadap variabel penunjuk stack (top)}

{F.S. : menghasilkan Stack yg siap digunakan}

Kamus:

Algoritma:

Top ← 0

EndProcedure

Algoritma Inisialisasi (List)

Procedure Inisialisasi(**Output** Top : NamaPointer)
{I.S. : Memberi harga awal terhadap variabel penunjuk stack (top)}
{F.S. : menghasilkan Stack yg siap digunakan}
Kamus:

Algoritma:
    Top ← nil
EndProcedure

Push

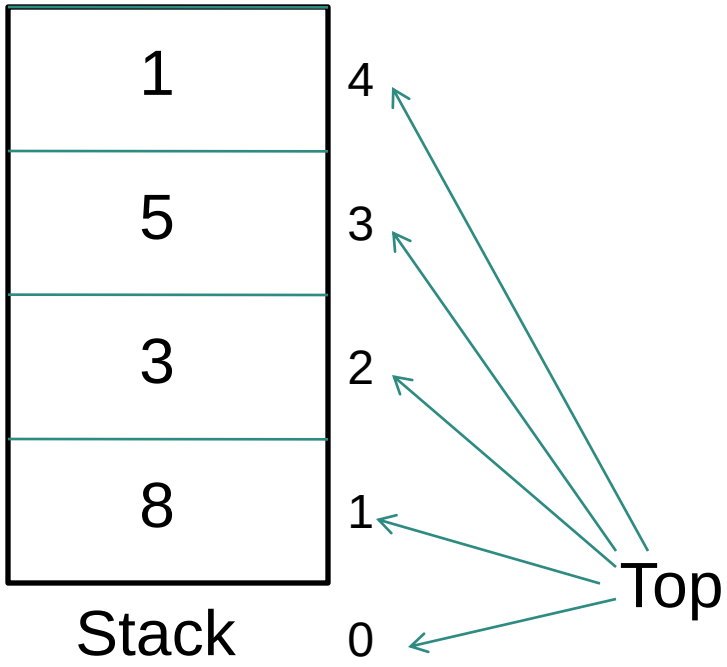
Langkah operasi push dalam array adalah dengan cara :

- Stack dapat ditambah jika stack belum penuh
- Tambahkan penunjuk Stack (top) dengan 1
- Elemen Stack pada posisi top diisi dengan data baru.

Push (lanjutan)

Operasi push pada stack yang menggunakan linked list adalah sama dengan proses penyisipan di awal/depan

Illustrasi Push



Push(Top,Stack,8)
Push(Top,Stack,3)
Push(Top,Stack,5)
Push(Top,Stack,1)
Push(Top,Stack,7)

“Stack Penuh”

Algoritma Push

Procedure Push(I/O Top : **Integer,** I/O Stack : NamaStack, **Input** databaru : tipedata)

{I.S. : data yg baru, Stack dan penunjuk stack (top) sudah terdefinisi}

{F.S. : menghasilkan Stack yg sudah ditambah sebuah data}

Kamus:

Algoritma:

```
If(Top < MaxStack)
  Then
     Top ← Top + 1
     Stack(Top) ← databaru
  Else
     output('Stack Sudah Penuh, Push Gagal')
EndIf
EndProcedure
```

Pop

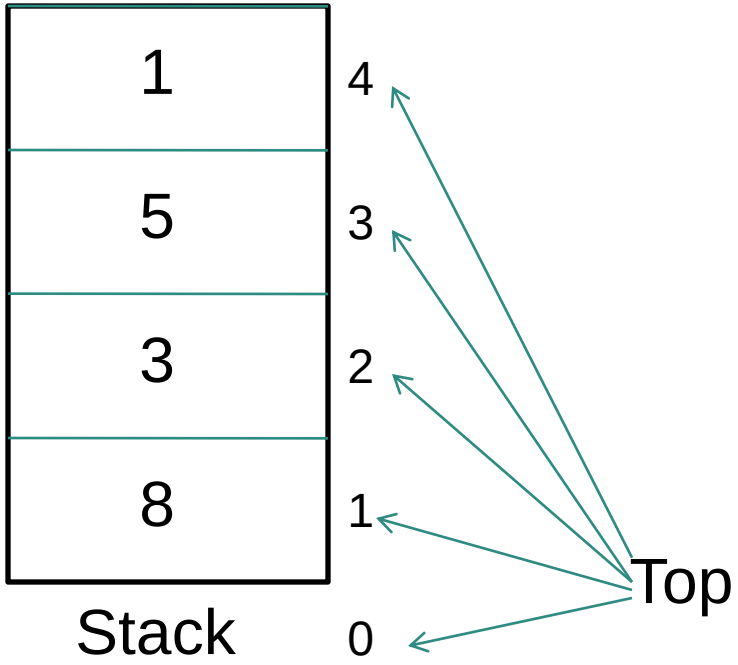
Langkah operasi pop pada stack yang menggunakan array adalah

- Stack dapat mengeluarkan elemennya jika stack tidak kosong
- Elemen yang dikeluarkan disimpan pada suatu variabel
- Nilai Top berkurang 1

Pop (lanjutan)

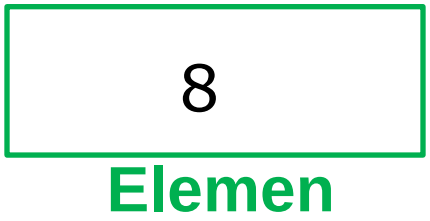
Operasi pop pada stack yang menggunakan **linked list** adalah sama dengan proses **penghapusan di awal/depan**.

Pop (Lanjutan)



Pop(Top,Stack,Elemen)
Pop(Top,Stack,Elemen)
Pop(Top,Stack,Elemen)
Pop(Top,Stack,Elemen)
Pop(Top,Stack,Elemen)

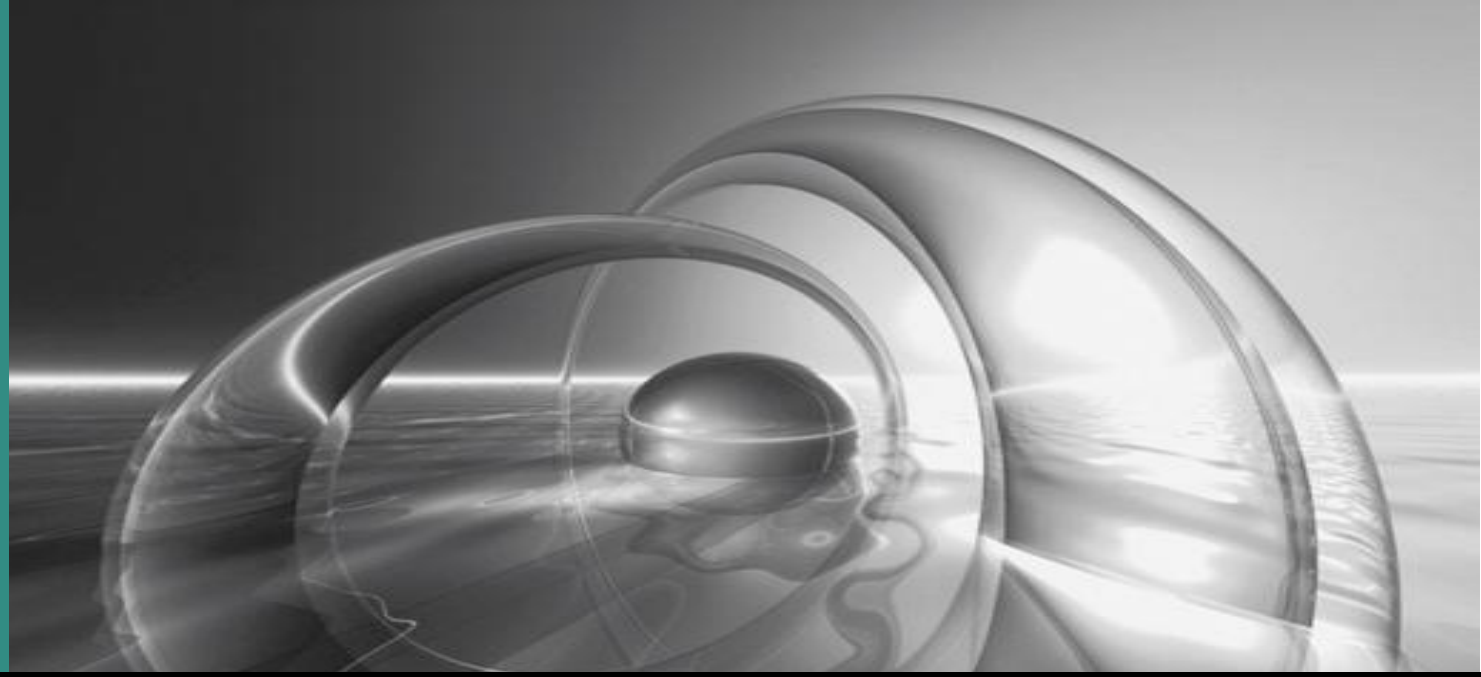
“Stack Kosong”



Algoritma Pop

Procedure Pop(**I/O** Top : **integer**, **I/O** Stack:NamaStack,
Output Elemen : tipedata)
{I.S. : Stack, dan penunjuknya (Top) sudah terdefinisi}
{F.S. : menghasilkan Stack yang sudah dikeluarkan sebuah datanya}
Kamus:

Algoritma:
If (Top ≠0)
Then
Elemen ← Stack(Top)
Top ← Top - 1
Else
output('Stack Kosong')
EndIf
EndProcedure



Terima Kasih