

ADT STACK

Danang Wahyu Utomo

danang.wu@dsn.dinus.ac.id

+6285 740 955 623

RENCANA KEGIATAN PERKULIAHAN SEMESTER

W	Pokok Bahasan
1	ADT Stack
2	ADT Queue
3	List Linear
4	List Linear
5	List Linear
6	Representasi Fisik List Linear
7	Representasi Fisik List Linear
8	Ujian Tengah Semester

W	Pokok Bahasan
9	Variasi List Linear
10	Variasi List Linear
11	Stack dengan Representasi List
12	Queue dengan Representasi List
13	List Rekursif
14	Pohon dan Pohon Biner
15	Multi List
16	Ujian Akhir Semester



Overview

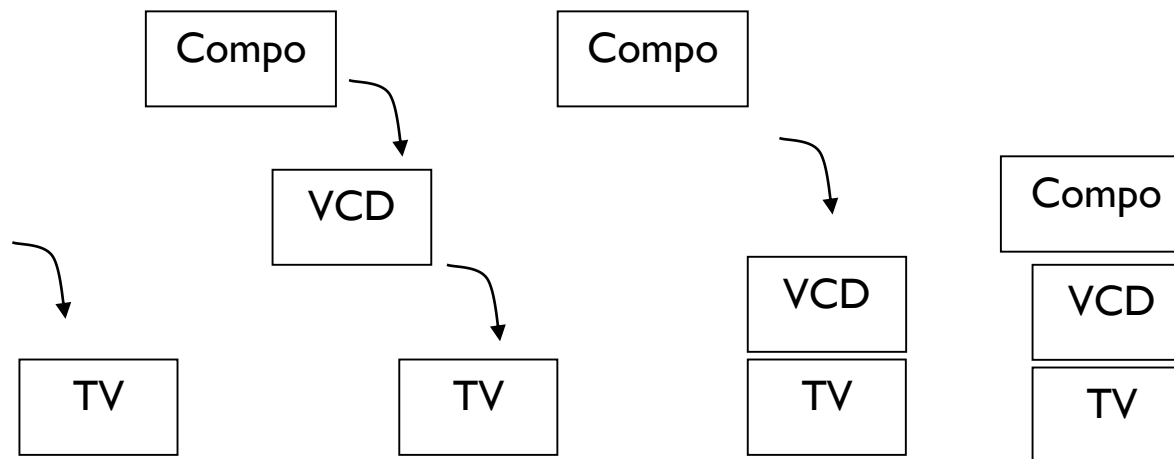
- ▶ Cetak data berikut menggunakan array of structure (struct)
- ▶ Lakukan terhadap array statis dan dinamis

NIM	Nama	Jurusan	Tahun Lulus
A11.2020.01234	Andi	Broadcasting	2023
A11.2010.01234	Budi	Sistem Informasi	2013
A11.2000.01234	Ali	DKV	2003
A11.1990.01234	Siti	Kesehatan	1993



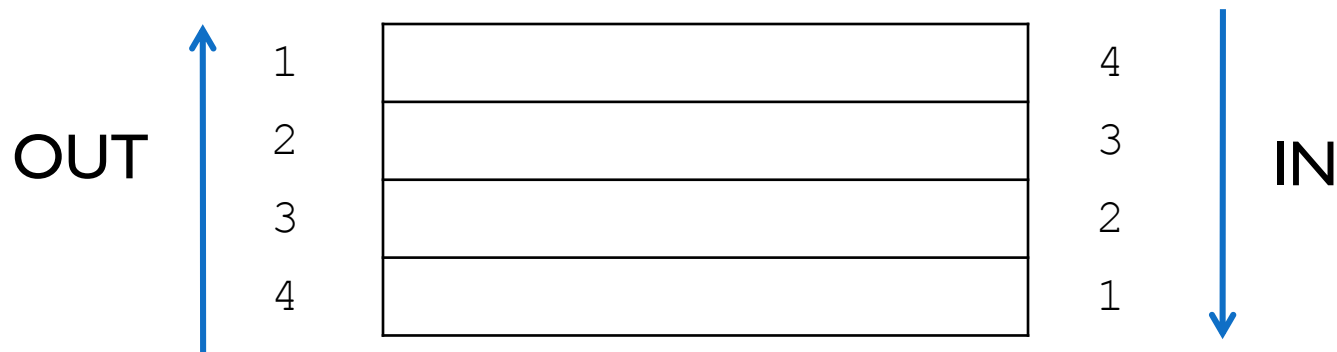
Stack

- ▶ Suatu susunan koleksi data dimana data dapat ditambahkan dan dihapus selalu dilakukan pada bagian akhir data, disebut dengan **top of stack**



Stack

- ▶ Stack bersifat LIFO (*Last In First Out*)
- ▶ Benda yang **terakhir masuk** dalam stack, akan menjadi yang **pertama keluar** dari stack



Operasi Stack

Push

- Digunakan untuk menambah item pada stack pada tumpukan paling atas

Pop

- Digunakan untuk mengambil item pada stack pada tumpukan paling atas

IsEmpty

- Fungsi untuk mengecek apakah stack kosong

IsFull

- Fungsi untuk mengecek apakah stack penuh



Stack – Array of Struct

- ▶ Definisikan STACK dengan struct
- ▶ Definisikan konstanta MAX_STACK untuk menyimpan maksimum isi stack
- ▶ Elemen struct Stack adalah array data dan top untuk menandakan posisi data teratas
- ▶ Buat variabel (**variabel tumpuk**) sebagai implementasi dari struct STACK
- ▶ Deklarasikan operasi – operasi / function diatas



Program Stack

```
#define MAX_STACK 10
```

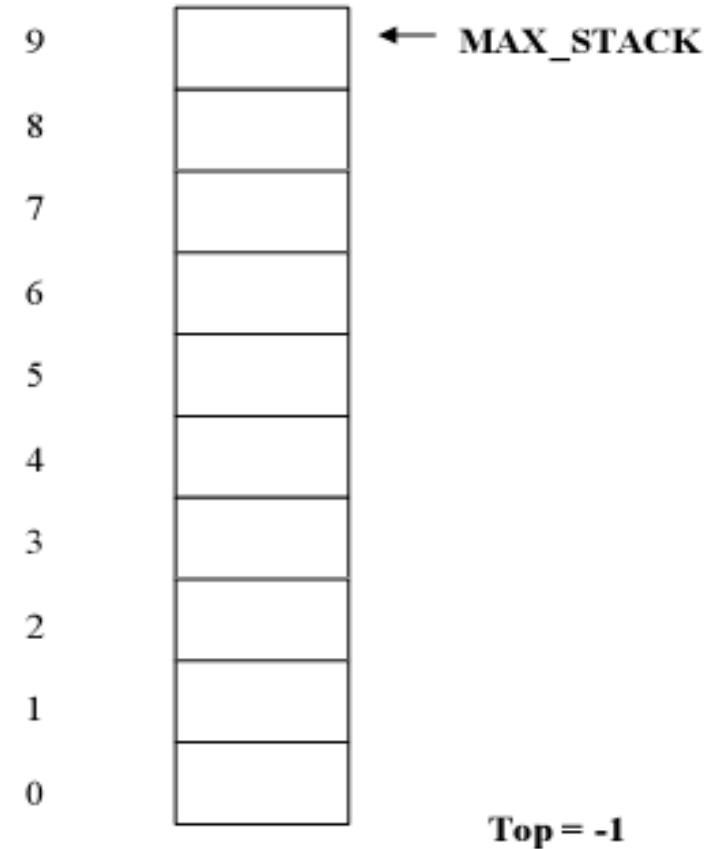
```
typedef struct {  
    int top;  
    char data[10][10];  
} stack;
```



Program Stack

► Inisialisasi Stack

- Top diisi dengan **-1**, karena array (dalam bahasa C) dimulai dari 0, yang berarti bahwa data stack **kosong**
- Top adalah variabel penanda dalam stack yang menunjukkan elemen teratas data stack (**Top Of Stack**)

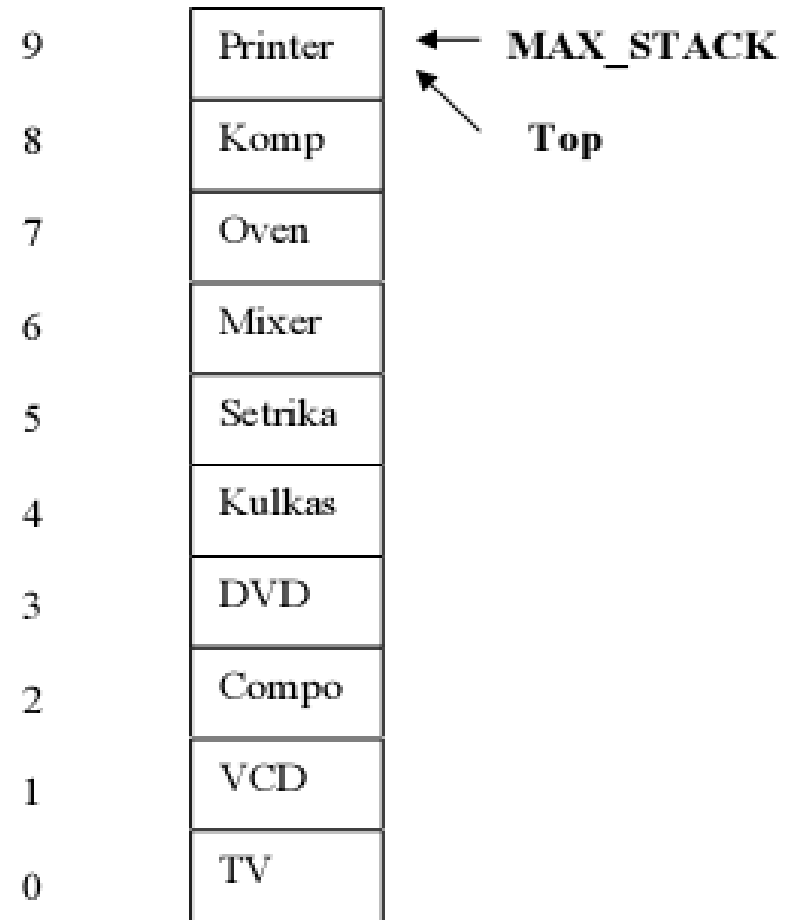


Program Stack

► Fungsi IsFull

- Apakah stack penuh ?
- Dengan cara memeriksa ToS, jika sudah sama dengan **MAX_STACK-1** maka stack full,
- Jika masih kecil dari **MAX_STACK-1** maka belum full

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



Program Stack

▶ Fungsi IsEmpty

- Apakah stack kosong ?
- Memeriksa ToS, jika masih **-1** maka stack masih kosong

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



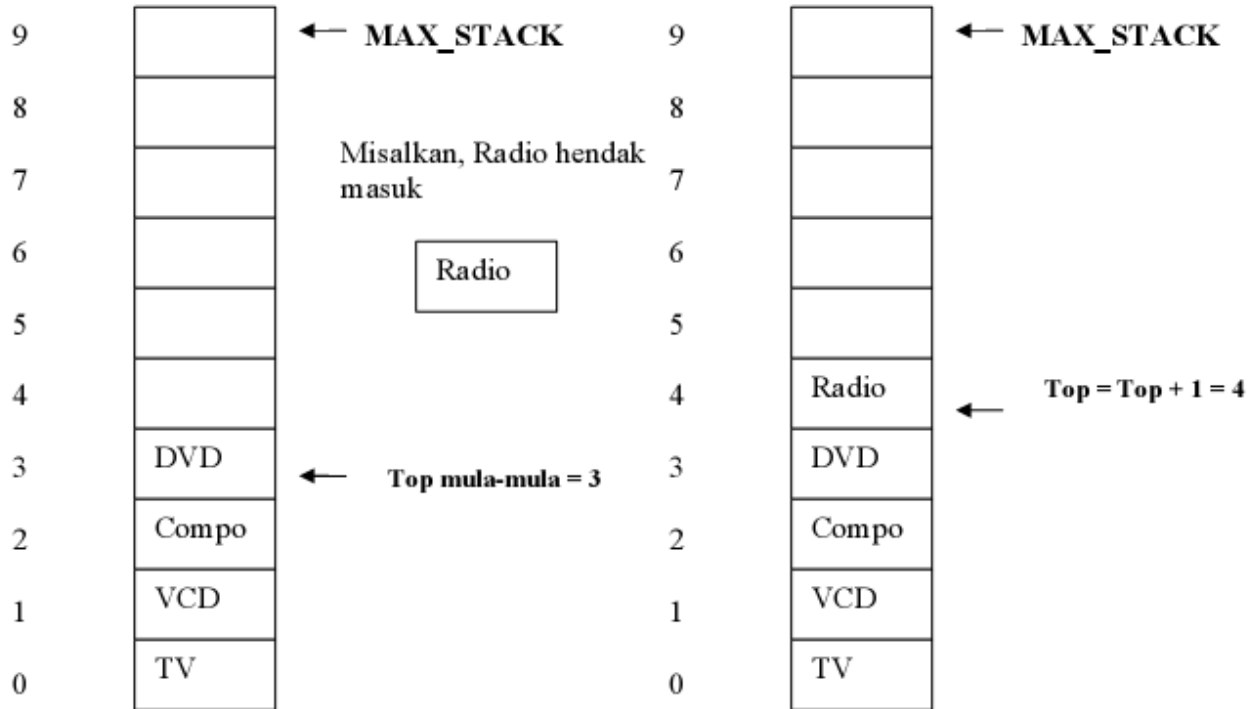
Program Stack

▶ Fungsi Push

- Memasukkan data ke stack
- Data yang diinput menjadi data teratas dari stack (ditunjuk oleh ToS)
- Jika data **belum penuh**, tambah satu (increment) nilai ToS lebih dahulu setiap kali ada penambahan ke dalam array data stack
- Isikan data baru ke stack berdasarkan index ToS yang telah di-increment



Program Stack



```
void Push(char d[10]){
    tumpuk.top++;
    strcpy(tumpuk.data[tumpuk.top], d);
}
```

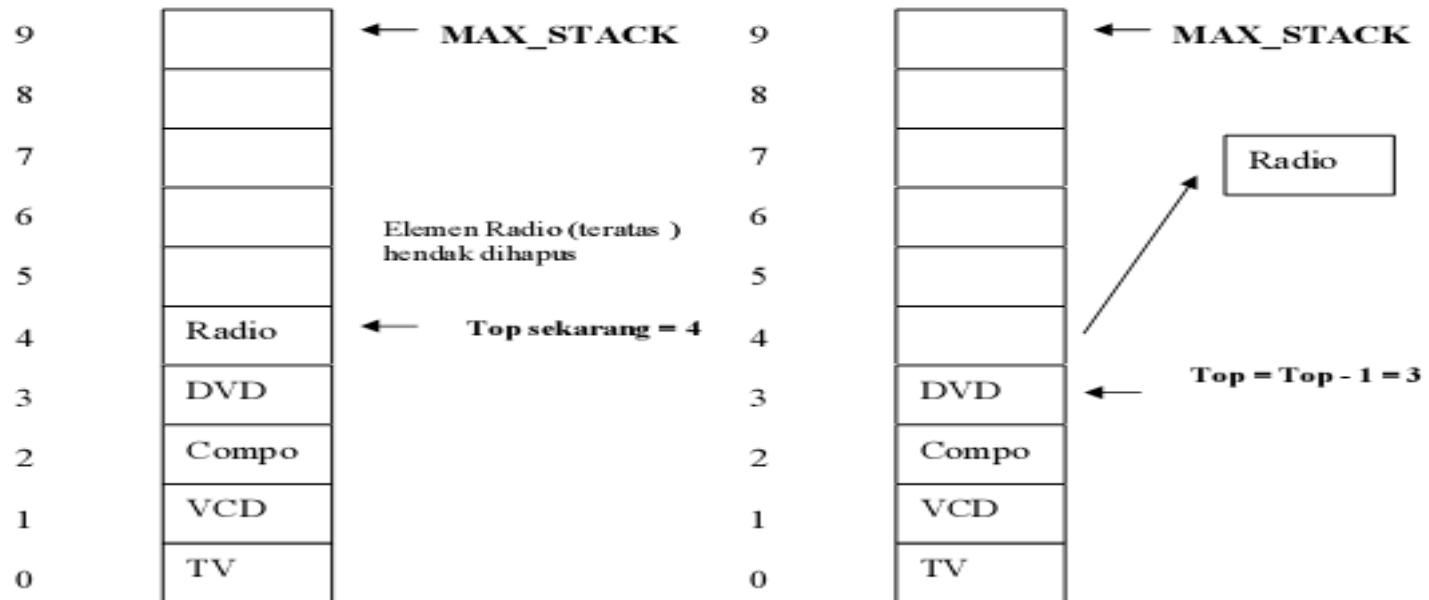
Program Stack

▶ Fungsi Pop

- Mengambil data stack yang terletak paling atas (data yang ditunjuk ToS)
- Tampilkan terlebih dahulu nilai elemen teratas dengan mengakses index sesuai dengan ToS
- Decrement nilai ToS sehingga elemen stack berkurang



Program Stack



Programnya:

```
void Pop(){  
    printf("Data yang terambil = %s\n",tumpuk.data[tumpuk.top]);  
    tumpuk.top--;  
}
```

▶ danang.wu@dsn.dinus.ac.id

