

Pertemuan 05

STACK

Learning Outcomes

Pada akhir pertemuan ini, diharapkan mahasiswa akan mampu :

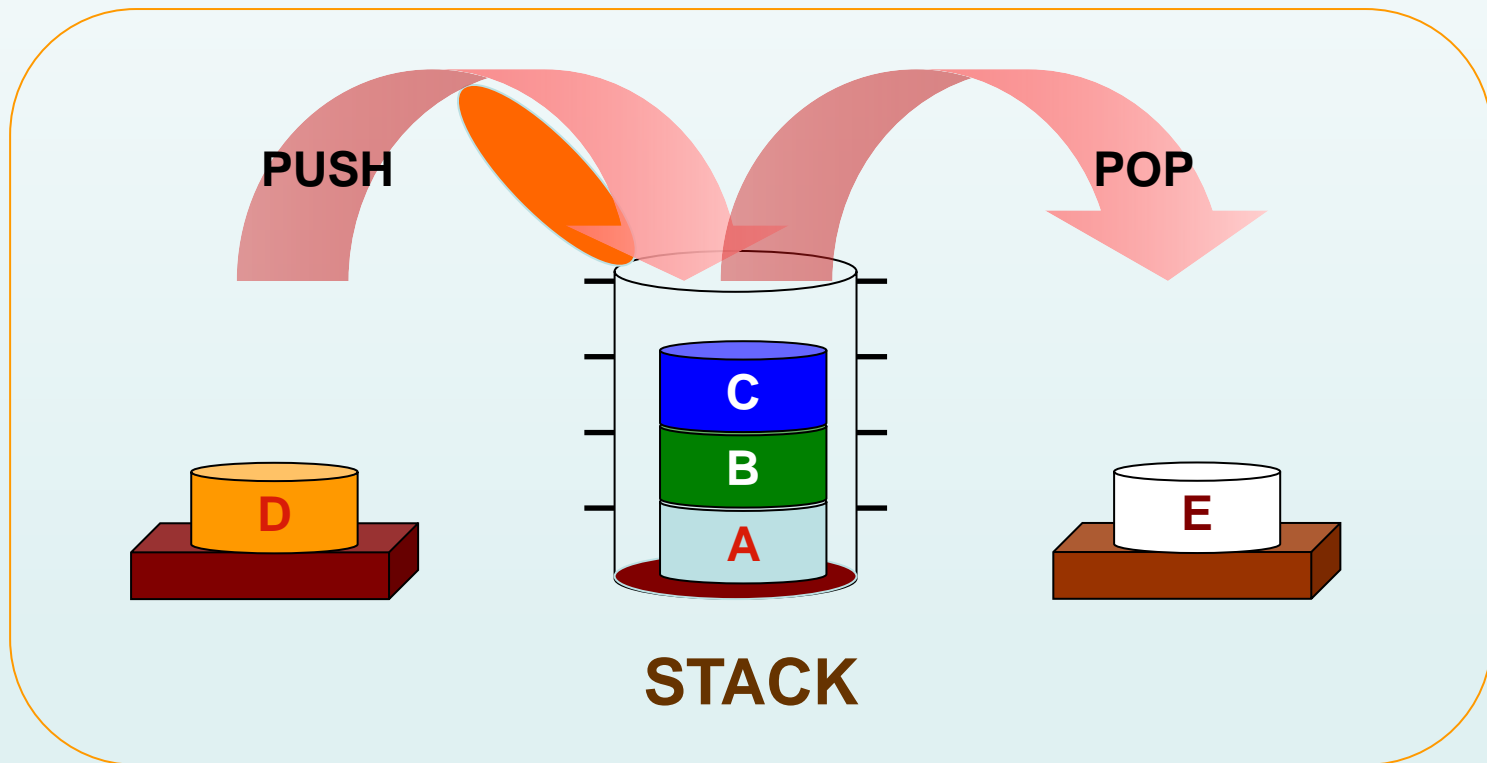
- mendemonstrasikan TDA Stack.
- menerapkan STACK pada program aplikasi komputer.

Outline Materi

- Basis operasi
- Abstraksi
 - Obyek
 - Operasi/fungsi
- Implementasi
 - Array
 - Single
 - Double
 - Linked List

Basis Operasi

Basis operasi pada STACK adalah LIFO (Last In First Out), Elemen yang diambil / dihapus adalah elemen yang baru dimasukkan / diinsert



TDA Stack

ADT Stack

Structure *Stack* is

Objects : a finite ordered list with zero or more elements.

function :

For all *stack* \in *Stack*, *item* \in *element*, *max_stack_size* \in *positive integer*.

Stack Create(*max_stack_size*) ::= create an empty *stack* whose maximumsize is *max_stack_size*.

Boolean *IsFull*(*stack* , *max_stack_size*) ::= if (number of elements in *stack* == *max_stack_size*) return TRUE else return FALSE

Stack Push(*stack* ,*item*) ::= if (*IsFull*(*stack*)) *stack -full* else insert *item* into top of *stack* and return.

Boolean *IsEmpty*(*stack*) ::= if (*stack* == *Create*(*max_stack_size*)) return TRUE else return FALSE

Element *Pop*(*stack*) ::= if (*IsEmpty*(*stack*)) return else remove and return the *item* on the top of *stack*.

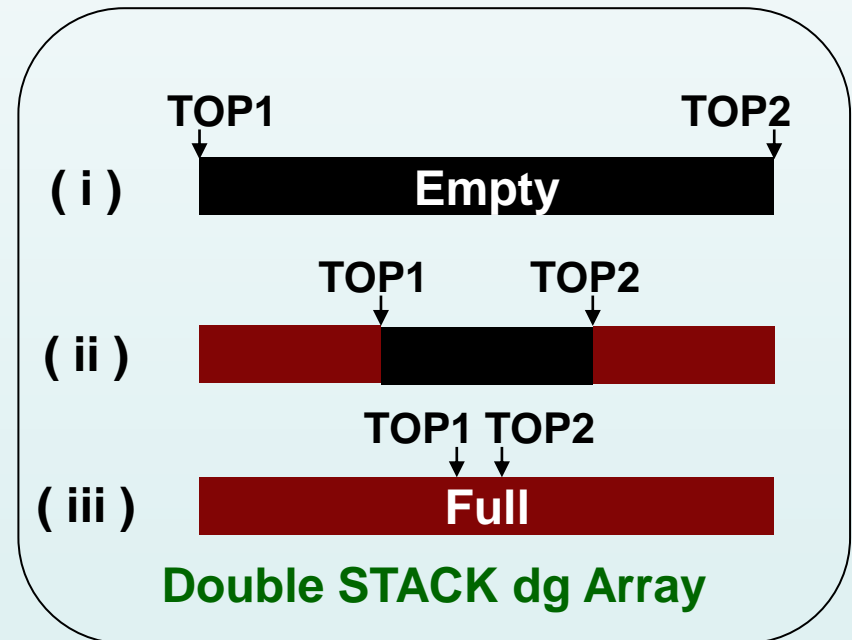
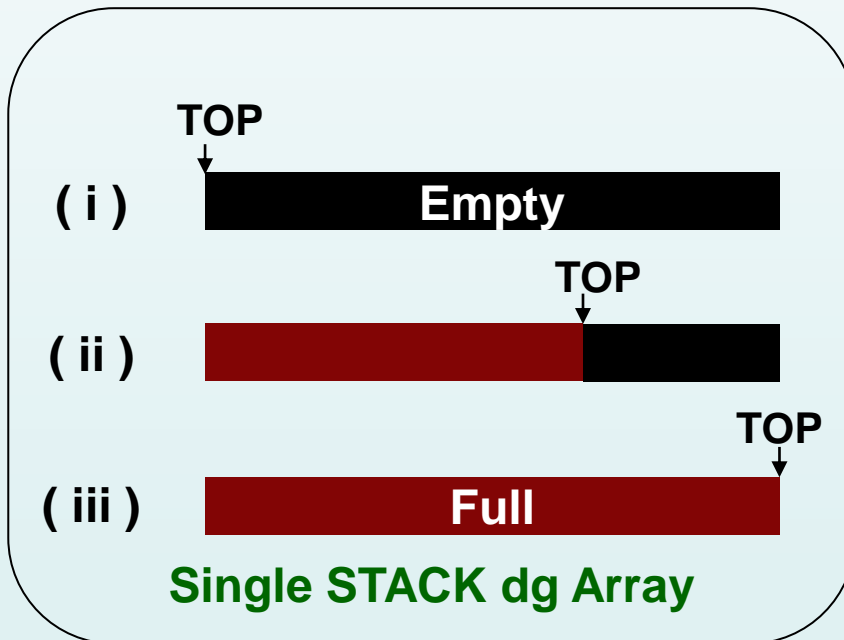
Operasi

- **CREATE()**
Untuk menciptakan stack yang baru dan kosong.
- **CLEAR()**
Untuk menghapus semua elemen dalam Stack.
- **PUSH (ElementType e)**
Untuk memasukkan 1 elemen ke dalam Stack.
- **POP (ElementType *e)**
Untuk mengambil 1 elemen dari Stack.
- **int EMPTY()**
Untuk mengecek apakah Stack masih kosong atau sudah berisi data.
- **int FULL()**
Untuk mengecek apakah Stack sudah penuh atau masih bisa menampung data.

Implementasi dg Array

- **Dua jenis STACK :**

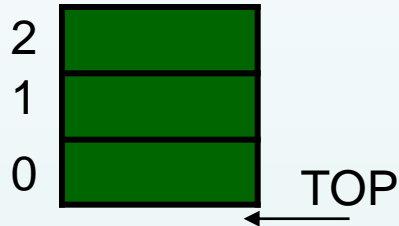
- Single STACK : menggunakan 1 ujung STACK utk PUSH maupun POP
- Double STACK : menggunakan 2 ujung STACK utk PUSH maupun POP



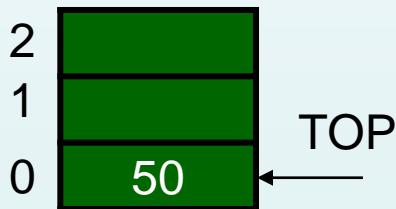
Simulasi dg Array

Contoh sebuah STACK memiliki jumlah maksimum 3 elemen:

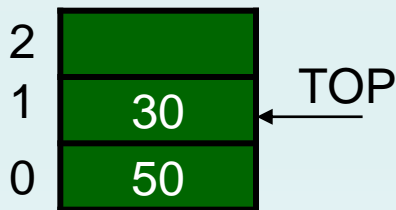
CREATE()



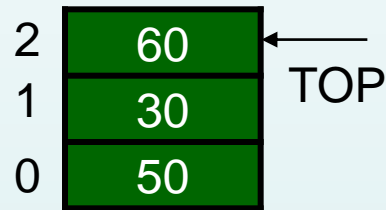
PUSH(50)



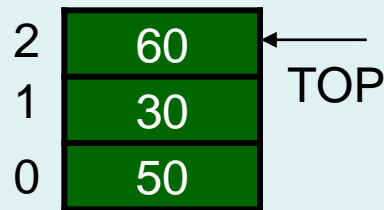
PUSH(30)



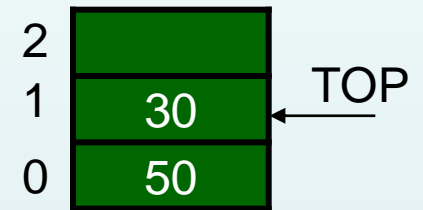
PUSH(60)



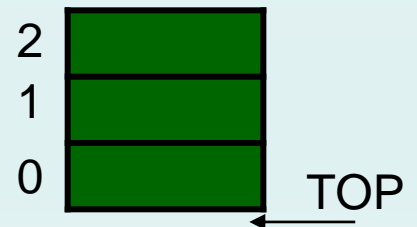
PUSH(20)



POP(*e)



CLEAR()



Implementasi dg Linked List

Contoh Implementasi Stack dengan Linked List menggunakan bahasa C

```
#include <stdio.h>
#include <malloc.h>
typedef int ElemenType;           //misal data yg disimpan bertipe integer
struct NODE{
    ElemenType Data;
    struct NODE *Next; } *HEAD;

struct Node *Head;
void CREATE()
{    HEAD = NULL; }

int EMPTY()
{    if(HEAD == NULL) return(1);
    else return(0); }
```

Implementasi dg Linked List(2)

```
void PUSH( ElementType e )
{
    struct NODE *P;
    P=(struct NODE *)malloc(sizeof(struct NODE));
    P->Data=e;
    P->Next = HEAD;
    HEAD = P; }
```

```
void POP( ElementType *e )
{
    struct NODE *P;
    *e = HEAD->Data;
    P=HEAD;
    Head=HEAD->Next;
    free(P); }
```

```
void CLEAR()
{
    ElementType X;
    while(!EMPTY()) POP(&X); }
```

Simulasi dg Linked List

CREATE()

HEAD → NULL

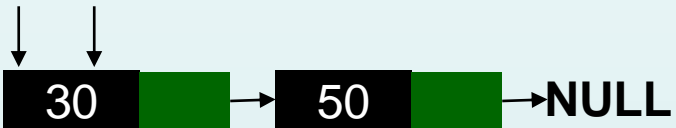
PUSH(50)

P HEAD



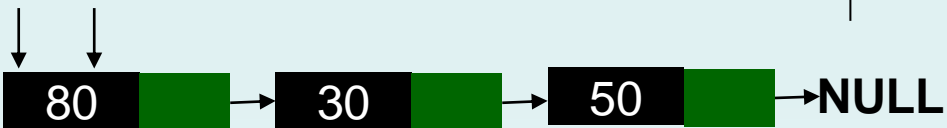
PUSH(30)

P HEAD



PUSH(80)

P HEAD



POP(*e)

HEAD



CLEAR()

HEAD → NULL

//CONTOH PROGRAM STACK

// DG ARRAY, DALAM BAHASA C

```
#include <stdio.h>  
#define max 10
```

```
int S[max];  
int i,k,top;  
void create()  
{  
    top=0;  
}
```

```
int full()  
{  
    if (top == max-1) return 1;  
    else return 0;
```

```
void push( int e )  
{  
    if (full()) printf(" Stack penuh Coy \n");  
    else {  
        top = top + 1;  
        S[top] = e;  
    }  
}  
  
int empty()  
{  
    if (top == 0 ) return 1;  
    else return 0;  
}  
  
void pop( int *e)  
{  
    if (empty()) printf(" Stack kosong mas ! \n");  
    else {  
        *e = S[top];  
        top = top - 1;  
    }
```

```
void main()  
{  
    create();  
    pop (&k);  
    push (70);  
    push(80);  
    push (30);  
    push(40);  
    for (i=4; i<=10; i++) push(i);  
    printf ("lihat 4 elemen stack yang teratas \n");  
    for(i=1;i<=4;i++) {  
        pop(&k);  
        printf("%5d \n",k);  
    }  
    clear();  
    pop(&k);  
    for (i=50; i<=55; i++) push(i);
```

LATIHAN:

**COBA JALANKAN PROGRAM TERSEBUT
DAN ANALISA HASILNYA**

Selesai