



---

# **Algoritma dan Struktur Data 2**

## **Modul 6 Stack (Tumpukan)**

---

Disusun oleh:

**Dwi Intan Af'idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA  
TAHUN AJARAN 2020/2021**



## Daftar Isi

---

Daftar Isi .....	ii
1 SINGLE & DOUBLE LINKED LIST .....	1
1.1 Target Pembelajaran .....	1
1.2 Dasar Teori .....	1
1.3 Latihan .....	3
1.3.1 Source Code Stack .....	3
1.3.2 Ilustrasi .....	5
1.4 Tugas Praktikum 5 .....	6



## 1 SINGLE & DOUBLE LINKED LIST

### 1.1 Target Pembelajaran

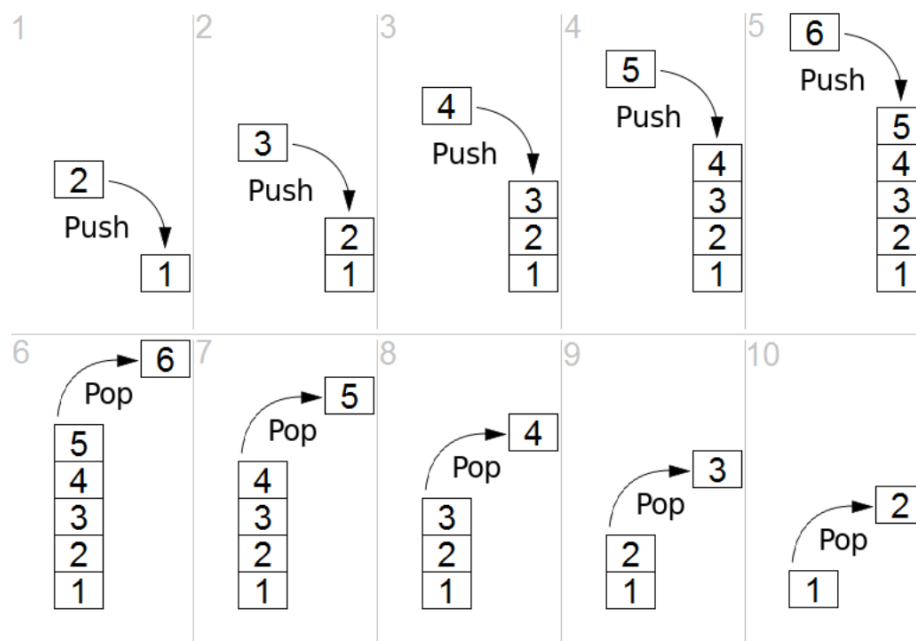
1. Memahami konsep dan operasi pada Stack.
2. Mampu mengimplementasikan struktur data Stack pada Array.

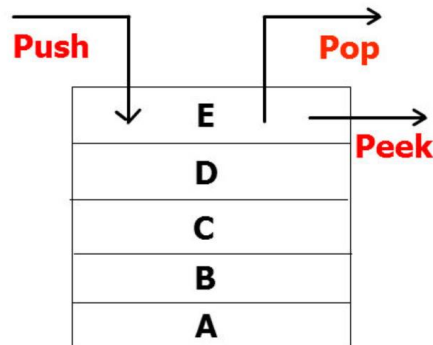
### 1.2 Dasar Teori

Stack(tumpukan) adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO. Elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas stack disebut dengan **push**. Dan untuk memindahkan dari tempat teratas tersebut, dilakukan dengan cara **pop**.

Konsep **LIFO** (*Last In First Out*) menjadikan data terakhir yang masuk adalah yang diproses terlebih dahulu. Berkebalikan dengan *Queue* atau Antrian yang mana data yang masuk terlebih dahulu akan diproses terlebih dahulu. Berikut ilustrasi dari struktur data Stack.

Gambar di bawah ini merupakan ilustrasi dari Stack.





Perhatikan gambar ilustrasi di atas. Data **E** adalah data yang terakhir dimasukkan (method **Push**) oleh karenanya berada pada posisi paling atas. Jika ingin mengambil data **B** maka terlebih dahulu harus mengeluarkan (method **Pop**) berturut-turut data **E**, **D** dan **C**.

Berikut adalah method-method yang wajib ada pada sebuah struktur data Stack:

1. Push, digunakan untuk memasukkan data ke dalam Stack
2. Pop, digunakan untuk mengeluarkan data teratas dari Stack
3. Peek, digunakan untuk melihat data yang berada di posisi paling atas
4. Count, digunakan untuk mengetahui jumlah isi data pada Stack
5. Clear, digunakan untuk menghapus seluruh data yang ada pada Stack



## 1.3 Latihan

### 1.3.1 Source Code Stack

#### 1. Membuat class Tumpukan

```
1 package Stack;
2
3 public class Tumpukan {
4     public int ukuran;
5     public long [] tumpukan;
6     public int top;
7
8     public Tumpukan(int s){
9         ukuran = s;
10        tumpukan = new long [ukuran];
11        top = -1;
12    }
13    public void push(long j){
14        tumpukan[++top] = j;
15    }
16    public long pop(){
17        return tumpukan[top--];
18    }
19    public long peek(){
20        return tumpukan[top];
21    }
22    public boolean isEmpty(){
23        return (top == ukuran-1);
24    }
25    public boolean isFull(){
26        return(top == ukuran-1);
27    }
28    }
29    public void baca(){
30        int i=top;
31        while (i>=0){
32            System.out.print(tumpukan[i]);
33            System.out.print(" ");
34            i--;
35        }
36        System.out.println(" ");
37    }
38    }
39
```



## 2. Membuat class TumpukanApp

```
3 public class TumpukanApp {  
4     public static void main (String [] args){  
5         Tumpukan tumpukan=new Tumpukan(10);  
6         tumpukan.push(45);  
7         tumpukan.baca();  
8         tumpukan.push(56);  
9         tumpukan.baca();  
10        tumpukan.push(67);  
11        tumpukan.baca();  
12        long nilail=tumpukan.pop();  
13        System.out.println("nilail = "+nilail);  
14        System.out.println(" ");  
15        tumpukan.push(83);  
16        tumpukan.baca();  
17        tumpukan.push(27);  
18        tumpukan.baca();  
19        long nilai2 = tumpukan.pop();  
20        System.out.print("nilai2 = "+nilai2);  
21        System.out.println(" ");  
22        long nilai3 = tumpukan.pop();  
23        System.out.println("nilai3 = "+nilai3);  
24        System.out.println(" ");  
25        tumpukan.baca();  
26    }  
27 }
```



### 1.3.2 Ilustrasi

Ilustrasi dari stuktur data tumpukan dengan operasi pada sub bab 1.3.2 adalah sebagai berikut sebagai berikut:

1. Push 45		2. Push 56		3. Push 67		4. Pop	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7	67	7	
8		8	56	8	56	8	56
9	45	9	45	9	45	9	45
5. Push 83		6. Push 27		7. Pop		7. Pop	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6	27	6		6	
7	83	7	83	7	83	7	
8	56	8	56	8	56	8	56
9	45	9	45	9	45	9	45



### 1.4 Tugas Praktikum 5

- Buatlah

1. kode program
2. ilustrasi

sehingga menghasilkan output seperti di bawah ini:

```
run-single:
50

100 50

50

60 50

80 60 50

60 50

nilai top = 60
BUILD SUCCESSFUL (total time: 0 seconds)
```