

## PERTEMUAN 4: DOUBLE STACK

### A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai tumpukan (Stack) yang terdapat pada struktur data. Di modul ini, Anda harus mampu:

3.1 Merepresentasikan Stack dalam bahasa pemrograman .

### B. URAIAN MATERI

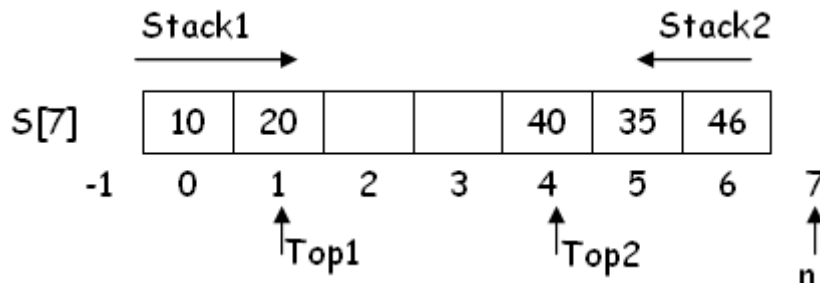
*Tujuan Pembelajaran 3.1:*

*Aplikasi Stack*

#### II. Double Stack.

- Dua stack dalam satu array
- Dasar stack1 berada pada index terkecil
- Dasar stack2 berada pada index terbesar

Ilustrasi :



Proses :

- a) AWAL(Inisialisasi)
- b) PUSH1, Push untuk stack1
- c) POP1, Pop untuk stack1
- d) PUSH2, Push untuk stack2
- e) POP2, Pop untuk stack2

a. Fungsi dasar proses AWAL :

```
void AWAL(void)
{
    Top1 = -1;
    Top2 = n;
}
```

b. Fungsi dasar proses PUSH1 :

```
void PUSH1(void)
{
    S[++Top1] = x;
}
```

c. Fungsi dasar proses POP1 :

```
void POP1(void)
{
    x = S[Top1--];
}
```

d. Fungsi dasar proses PUSH2 :

```
void PUSH2(void)
{
    S[--Top2] = x;
}
```

e. Fungsi dasar proses POP2 :

```
void POP2(void)
{
    x = S[Top2++];
}
```

#### Kondisi Stack

	Kondisi Stack	Ciri
1	Stack1 kosong	Top1 = -1
2	Stack2 kosong	Top2 = n
3	Stack Penuh	Top2 - Top1 = 1
4	Stack bisa diisi	Top2 - Top1 > 1
5	Stack1 ada isinya	Top1 > -1
6	Stack2 ada isinya	Top2 < n

a) Algoritma lengkap proses PUSH1 :

Periksa apakah  $Top2 - Top1 > 1$ ,

- a. jika ya :
  - Naikan Top1 dengan 1
  - Isikan data kedalam elemen yang ditunjuk oleh Top1
- b. Jika tidak
  - Cetak komentar "Stack Penuh"

```
void PUSH1(void)
{
    if(Top2 - Top1 > 1)
        S[++Top1] = x;
    else
        cout<< "Stack Penuh";
}
```

b) Algoritma lengkap proses POP1 :

Periksa apakah  $Top1 > -1$ ,

- Jika ya,
  - Copy data dari elemen yang ditunjuk Top1
  - Turunkan Top1
- Jika tidak,
  - Cetak komentar "Stack Kosong"

```
void POP1(void)
{
    if(Top1 > -1)
        x = S[Top1--];
    else
        cout<< "Stack Kosong";
}
```

c) Algoritma lengkap proses PUSH2 :

Periksa apakah  $Top2 - Top1 > 1$ ,

- c. jika ya :
  - i. Turunkan Top2 dengan 1
  - ii. Isikan data kedalam elemen yang ditunjuk oleh Top2
- d. Jika tidak
  - i. Cetak komentar "Stack Penuh"

```
void PUSH2(void)
{
    if(Top2 - Top1 > 1)
        S[--Top2] = x;
    else
        cout<< "Stack Penuh";
}
```

d) Algoritma lengkap proses POP2 :

Periksa apakah  $Top2 < n$ ,

- Jika ya,
  - Copy data dari elemen yang ditunjuk Top2
  - Naikan Top2
- Jika tidak,
  - Cetak komentar "Stack Kosong"

```
void POP2(void)
{
    if(Top2 < n)
        x = S[Top2++];
    else
        cout<< "Stack Kosong";
}
```

Soal :

1. Susunlah program untuk menginput data dari keyboard terus menerus hingga stack1 penuh
2. Susunlah program untuk menginput data dari keyboard terus menerus hingga stack2 penuh
3. Susunlah program untuk menghapus stack1 hingga kosong
4. Susunlah program untuk menghapus stack2 hingga kosong

## **C. DAFTAR PUSTAKA**

### **Buku**

1. Esakov, Jeffrey, Tom Weiss, Data Structures An Advanced Approach Using C, Prentice-Hall, Inc. 1989
2. Hariyanto, Bambang, Struktur Data, Informatika Bandung, Pebruari 2000
3. Kadir, Abdul, Pemrograman Dasar Turbo C, Andi Offset, Yogyakarta, 1991
4. Kruse, Robert L. Data Structures & Program Design, Prentice-Hall, Inc. 1987
5. Standish, Thomas A. Data Structures, Algorithms & Software Principles In C, Addison Wesley, 1995