

# Pemahaman Modul

## Praktikum Struktur Data.

Kode\_Asi sten\_PJ\_: \_MTN

Kel ompok = A11. 4301U

NIM : A11. 2012. 06758

NIM : A11. 2012. 06889

Nama : Muhammad Adhi D

Nama : Agung Madra Jaya

### PSDA – 10 (Stack dengan Rep. List Linier).

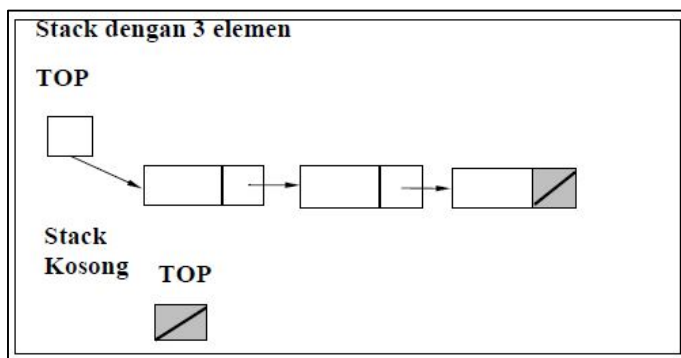
#### A. Penjelasan mengenai List Stack.

Stack yaitu tumpukan , dimana operasi penambahan selalu dilakukan pada **TOP** ( elemen paling atas dari stack) , dan penghapusan juga dilakukan pada elemen teratas *konsep LIFO*.

Namun pada topik kali ini , susunan struktur Stack disusun dengan menggunakan struktur ADT dari List Linier. dimana operasi :

- ✓ "PUSH" dapat di asumsikan sebagai **"InserVFi rst"**.
- ✓ "POP" dapat di asumsikan sebagai **"Del VFi rst"**.

Konsep dari Stack sendiri masih sama dengan stack pada pembahasan sebelumnya. Hanya saja perbedaanya terdapat pada struktur ADTnya yang menggunakan representasi **ADT List Linier**.



Pada gambar di samping adalah ilustrasi Stack dengan kondisi 3 elemen, dimana elemen terdepan akan di tunjuk sebagai **"TOP"** (el emen teratas), setiap kali penambahan akan di lakukan pada **"TOP"** , sehingga nilai **"TOP"** akan sel al u berubah.

Sedangkan jika penghapusan elemen (**POP**) juga di lakukan pada el emen **"TOP"**. Kondisi stack kosong adalah jika :

$$TOP(*S) = NIL.$$

```
typedef struct
{
    address TOP;
}Stack;
```

<<< ADT Stack

ADT ElmtList >>>

```
typedef struct tElmtList
{
    InfoType INFO;
    address NEXT;
}ElmtList;
```

### B. Penjelasan Mengenai Fungsi.

#### a. PUSH.

```
void PUSH (Stack *S, InfoType X)
{
    address newP;

    ALOKASI (&newP, X);

    if (newP != NIL)
    {
        NEXT(newP) = TOP(*S);
        TOP(*S) = newP;
    }
}
```

Pada gambar di samping merupakan isi prosedur dari **PUSH**, dimana prosedur tersebut mirip dengan "**InsVFirst**" yang ada pada bab List Linier.

Tahap pertama dari prosedur ini adalah, memesan sebuah tempat (1 elemen), dimana elemen yang pertama akan di tunjuk sebagai **TOP**. Dan seiring dengan pemanggilan prosedur ini nilai dari **TOP**, akan selalu naik (selalu berubah).

Misalnya jika terdapat stack "**8 > 7 > 1**", dan akan di **PUSH** elemen "**2**", maka akan menjadi "**2 > 8 > 7 > 1**".

#### b. POP.

```
void POP (Stack *S, InfoType *X)
{
    address P;

    P = TOP(*S);

    if (IsEmpty(*S))
    {
        puts("Stack kosong tuh ...!");
    }
    else
    {
        *X = INFO(P);
        TOP(*S) = NEXT(TOP(*S));
        NEXT(P) = NIL;

        DEALOKASI(P);
    }
}
```

Pada prosedur ini merupakan prosedur **POP**, dimana system kerja prosedur ini mirip dengan "**DelVFirst**". Dimana proses penghapusan juga selalu dilakukan pada **TOP** elemen.

Tahap pertama pada prosedur ini adalah, memutuskan sambungan dari Elemen pertama (**TOP**) dengan elemen yang ada pada selanjutnya. Kemudian elemen (address) yang telah diputuskan tadi akan di hapus (dealokasi), kemudian elemen yang selanjutnya akan menjadi **TOP** yang baru.

Misalkan terdapat sebuah Stack dengan elemennya "**8 > 7 > 9 > 1**", kemudian dilakukan pemanggilan prosedur "**POP**", maka Stack yang baru akan menjadi "**7 > 9 > 1**". Dimana elemen "**8**" yang telah diputuskan sambungannya tadi akan di dealokasi.