

STACK (TUMPUKAN)

STRUKTUR DATA

PENGERTIAN STACK

- ▶ STACK (TUMPUKAN) adalah dapat diartikan sebagai list (urutan) dimana penambahan dan pengambilan elemen hanya dilakukan pada satu sisi yang disebut top (puncak) dari stack.
- ▶ Konsep utama dari STACK adalah LIFO (Last In First Out)
- ▶ Dua operasi dasar pada STACK adalah :
 - PUSH (operasi pemasukan elemen ke dalam STACK)
 - POP (operasi pengambilan satu elemen dari dalam STACK)

Operasi yang dilakukan	Isi Stack	Keterangan
Kondisi Awal	Kosong	-
PUSH('A',S)	A	-
PUSH('B',S)	AB	-
PUSH('C',S)	ABC	-
POP(Data,S)	AB	Variabel <i>Data</i> berisi 'C'
PUSH('D',S)	ABD	-
POP(Data,S)	AB	<i>Data</i> berisi 'D'
POP(Data,S)	A	<i>Data</i> berisi 'B'

- ▶ Untuk mempermudah penulisan, di bawah ini isi stack tidak dituliskan secara bertumpuk, tetapi dengan kesepakatan:
 - elemen paling kanan adalah elemen yang ada pada TOS (Top Of the Stack)
 - stack yang dipakai bernama S
 - PUSH(S,B) berarti memasukkan elemen B ke dalam stack S
 - POP(B,S) berarti mengambil elemen dari stack S dan menaruhnya ke dalam variabel B

IMPLEMENTASI STACK DALAM PASCAL

- ▶ Implementasi dalam bahasa Pascal dapat dilakukan dengan memanfaatkan struktur data record dan array. Array dipergunakan untuk menyimpan elemen-elemen yang dimasukkan. Selain itu diperlukan pula suatu variabel untuk mencatat banyaknya elemen yang adadi dalam array yang sekaligus menunjukkan TOS. Pada implementasi di bawah ini:
 - ▶ – konstanta *maxelm* menyatakan banyaknya elemen maksimum yang dapat ditampung oleh stack
 - ▶ – *typeelemen* adalah tipe data yang akan disimpan di dalam stack (bisa integer, word, real, boolean, char , string atau lainnya)
 - ▶ – *banyak* adalah field yang menyatakan banyaknya elemen dalam stack saat itu, yang sekaligus menyatakan TOS

- ▶ Deklarasi tipe untuk tumpukan (stack):
type tumpukan = record
 banyak :0..maxelm;
 elemen : array[1..maxelm] of typeelemen;
end;
- ▶ Selain prosedur untuk **POP** dan **PUSH**, kita dapat pula menambahkan sejumlah fungsi untuk membantu penanganan kesalahan diantaranya adalah fungsi **PENUHS** (untuk mengecek apakah stack penuh) fungsi **KOSONGS** (untuk mengecek apakah stack kosong) dan fungsi **SIZES** (untuk mengetahui banyaknya elemen di dalam stack). Masing-masing sub program di atas dapat disajikan pseudocode-nya sebagai berikut:

- ▶ **Procedure Inisialisasi**(var S : tumpukan);
begin
 S.banyak = 0
end;
- ▶ **Function PENUHS**(S : tumpukan): boolean;
begin
 Jika S.banyak = maxelm maka PENUHS = true
 else PENUHS = false
end;
- ▶ **Function KOSONGS**(S : tumpukan):boolean;
begin
 If S.banyak = 0 then KOSONGS = true
 else KOSONGS = false
end;

- ▶ **Procedure PUSH**(data : tipeelemen; var S : tumpukan);
begin
 If not PENUHS(S) then
 begin
 S.banyak = S.banyak + 1
 S.elemen[S.banyak]=data
 end
 else
 Tampilkan pesan kesalahan "Stack Penuh"
 end;
end;

- ▶ **Procedure POP**(var S : tumpukan; var data : tipeelemen);
begin
 If not KOSONGS(S) then
 begin
 data=S.elemen[S.banyak]
 S.banyak = S.banyak - 1
 end
 else
 Tampilkan pesan kesalahan "Stack kosong"
 End;

- ▶ POP
Mengambil elemen teratas dari stack.
Syarat: Stack tidak boleh kosong.
- ▶ Procedure Pop (elemen:string);
Begin
 If not empty then
 Begin
 Elemen:=stack.data;
 Stack.top:=top - 1;
 End;
End;

- ▶ Latihan :
- ▶ Buat program pascal dengan menggunakan fungsi dan prosedur diatas, jika ouput yang diinginkan adalah :
- ▶ A
- ▶ B
- ▶ C
- ▶ B
- ▶ D
- ▶ E