

Praktikum Struktur Data

Pertemuan Ke-7

STACK

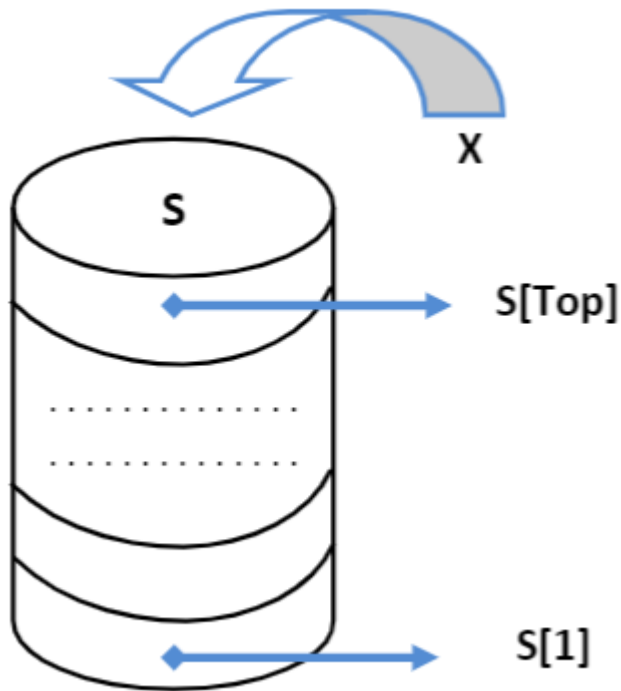
- ❑ Stack (tumpukan) adalah dapat diartikan sebagai list (urutan) dimana penambahan dan pengambilan elemen hanya dilakukan pada satu sisi yang disebut top (puncak) dari stack.
- ❑ Konsep utama dari Stack adalah LIFO (*Last In First Out*).

Operasi Dasar Pada Stack

Operasi dasar pada Stack terdiri dari:

1. CREATE, membuat stack baru dengan nilai 0
2. IsEmpty, mengecek apakah stack kosong (null)
3. IsFull, mengecek apakah stack penuh (null)
4. PUSH, memasukkan elemen ke dalam stack
5. POP, menghapus 1 elemen pada stack

Ilustrasi Stack



S merupakan nama dari Stack.
S[Top] merupakan elemen stack puncak dan **S[1]** merupakan elemen dasar dari Stack

Sedangkan **X** adalah nilai yang akan dimasukkan ke dalam elemen stack.

Deklarasi tipe untuk tumpukan (stack)

1. Deklarasi Stack Dengan Array

```
type tumpukan = record
  banyak : 0..maxelm;
  elemen : array[1..maxelm] of typeelemen;
end;
```

2. Deklarasi Stack Dengan Array

```
type
  tumpukan = record
    Stacks: array[0..10] of integer;
    top: integer;
end;
```

Deklarasi Stack di Pascal

CONST

Max = 5;

TYPE

Stack = record

isi : array[1..Max] of integer;

Top : integer

end;

Var

S : Stack;

Procedure dan Fungsi Untuk Stack

```
Procedure BuatStack(Var S : Stack);  
Begin  
    S.Top := 0;  
End;  
Function IsFull(S : Stack) : boolean;  
Begin  
    IsFull := (S.Top=Max);  
End;  
Function IsEmpty(S : Stack) : boolean;  
Begin  
    IsEmpty := (S.Top=0);  
End;
```

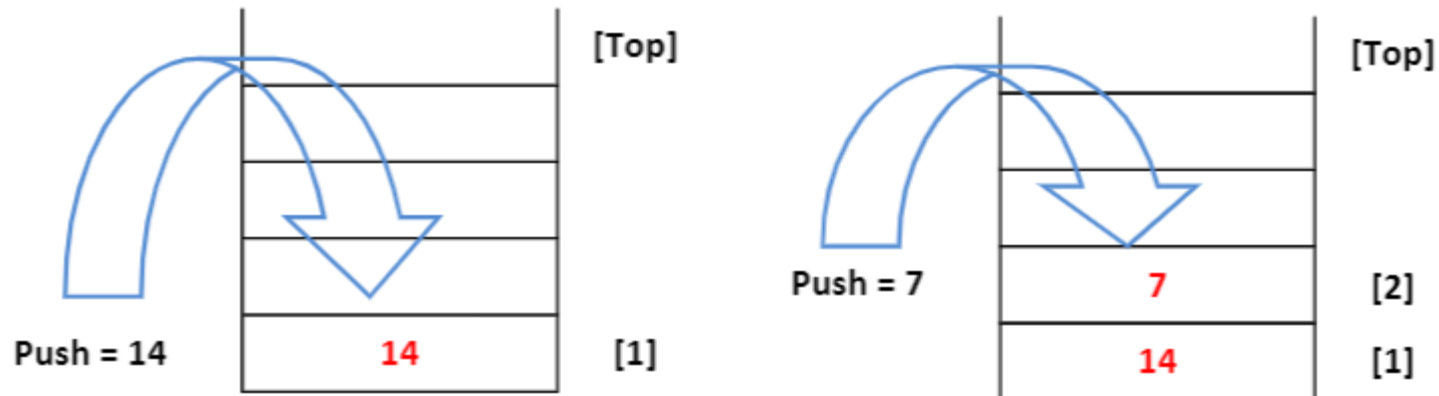
Operasi Pada Stack

```
Procedure Pop (Var S : Stack);  
Begin  
    S.isi[S.Top] := 0;  
    S.Top := S.Top - 1;  
End;
```

```
Procedure Tampil (S : Stack);  
Var  
    i : integer;  
Begin  
    ??????  
  
End;
```


Algoritma PUSH

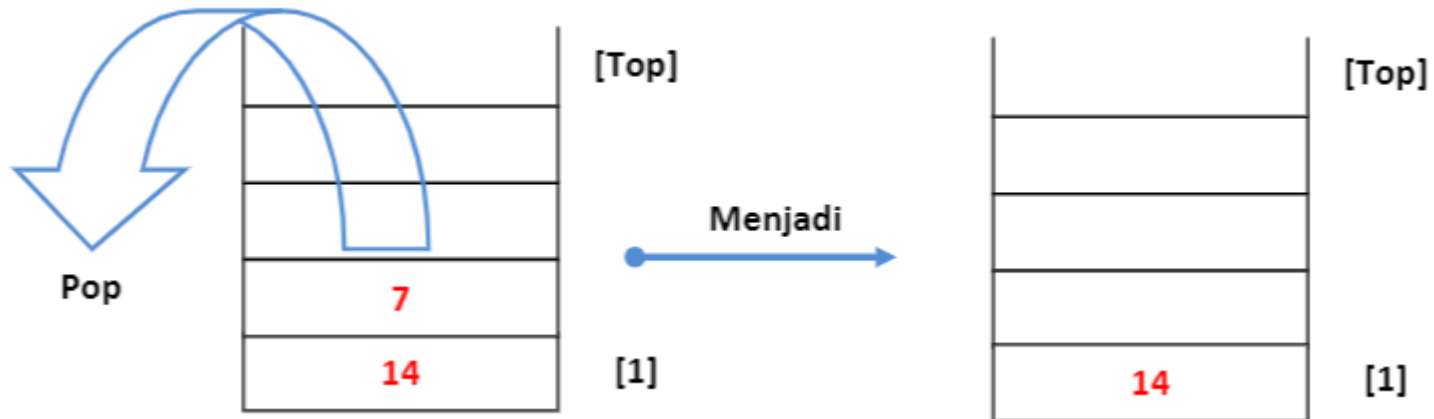
Misalkan akan mengisi nilai 14 dan 7 pada Stack dengan Max = Top 5



```
Procedure Push (Var S : Stack; X : integer);  
Begin  
    S.Top := S.Top + 1;  
    S.isi[S.Top] := X;  
End;
```

Algoritma POP

Misalkan akan mengeluarkan 1 elemen dari stack



```
Procedure Pop (Var S : Stack);  
Begin  
    S.isi[S.Top] := 0;  
    S.Top := S.Top - 1;  
End;
```

Procedure Untuk Menampilkan Data

```
Procedure Tampil (S : Stack);  
Var  
  i : integer;  
Begin  
    ?????  
  
End;
```

Contoh: Program Stack

```
1  Program OperasiSTACK;
2  uses crt;
3
4  const maks = 10;
5  Type
6      nilai_matkul = record
7          npm : string[8];
8          nama : string[30];
9          nilai : integer;
10     End;
11     stack = record
12         top : integer;
13         data : array[1..maks] of nilai_matkul;
14 End;
15
16 Procedure Inisialisasi(var s : stack);
17 Begin
18     s.top := 0;
19 End;
20
21 Function Isempty(var s : stack) : boolean;
22 var
23     hasil : boolean;
24 Begin
25     hasil := false;
26     if s.top = 0 then
27     begin
28         hasil := true;
29     end;
30     isempty := hasil;
31 End;
32
```

Contoh: Program Stack

```
33 Function Isfull(var s : stack) : boolean;  
34 var  
35     hasil : boolean;  
36 Begin  
37     hasil := false;  
38     if s.top = maks then  
39     Begin  
40         hasil :=true;  
41     end;  
42     isfull :=hasil;  
43 End;
```

Contoh: Program Stack

```
45 Procedure Push(npm : string; nama:string; nilai:integer; var s:stack);
46 Begin
47     if isfull(s)=true then
48     Begin
49         writeln('Stack Penuh');
50     end
51     else
52     begin
53         if isempty(s)=true then
54         begin
55             s.top := 1;
56             s.data[1].npm := npm;
57             s.data[1].nama := nama;
58             s.data[1].nilai := nilai;
59         end
60         else
61         begin
62             s.top := s.top+1;
63             s.data[s.top].npm :=npm;
64             s.data[s.top].nama :=nama;
65             s.data[s.top].nilai :=nilai;
66         end;
67     end;
68 End;
69
```

Contoh: Program Stack

```
70 Procedure Pop(var s : stack);
71 Begin
72     if s.top <> 0 then
73     begin
74         s.top := s.top-1;
75     end
76     else
77     write('Data Habis');
78     writeln;
79 end;
80
81 Procedure Cetakstack(var s:stack);
82 var
83     i : integer;
84 Begin
85     if s.top<> 0 then
86     begin
87         for i := s.top downto 1 do
88         with s.data[i] do
89         begin
90             writeln;
91             writeln('elemen ke: ',i);
92             wrietln('npm: ',npm);
93             writeln('nama: ',nama);
94             writeln('nilai: ',nilai);
95         end;
96         writeln;
97     end
98     else
99     begin
100         writeln('Stack Kosong');
101     end;
102 end;
```


Contoh: Program Stack

```
104 var
105     s :stack;
106     banyak : byte;
107     i : integer;
108 Begin
109 clrscr;
110 inisialisasi(s);
111 writeln;
112 writeln('-----AWAL-----');
113 Cetakstack(s);
114 writeln;
115 writeln('-----');
116 write('masukkan jumlah data yang akan disimpan: '); readln(banyak);
117 writeln;
118 for i := 1 to banyak do
119     with s.data[i] do
120     begin
121     write('NPM : '); readln(npm);
122     write('Nama: '); readln(nama);
123     write('Nilai: '); readln(nilai);
124 end;
125 readln;
```


Contoh: Program Stack

```
126 writeln('setelah di Push');
127 for i :=1 to banyak do
128 begin
129 push(s.data[i].npm, s.data[i].nama, s.data[i].nilai,s);
130 writeln;
131 end;
132 Cetakstack(s);
133 writeln('—————');
134 writeln;
135 readln;
136 writeln('——setelah di POP——');
137 pop (s);
138 writeln;
139 Cetakstack(s);
140 writeln('—————');
141 writeln;
142 readln;
143 readln;
144 end.
```

Output Program Stack

 Free Pascal

```
-----AWAL-----
```

```
Stack Kosong
```

```
-----
```

```
Masukkan jumlah data yang akan disimpan: 3
```

```
NPM :201043500111
```

```
Nama :Desi Maryani
```

```
Nilai :78
```

```
NPM :201043500222
```

```
Nama :Irianti Hapsari
```

```
Nilai :84
```

```
NPM :201043500333
```

```
Nama :Donny Sukmaji
```

```
Nilai :65_
```

```
setelah di Push
```

```
elemen ke: 3
```

```
NPM : 201043500333
```

```
Nama : Donny Sukmaji
```

```
Nilai : 65
```

```
elemen ke: 2
```

```
NPM : 201043500222
```

```
Nama : Irianti Hapsari
```

```
Nilai : 84
```

```
elemen ke: 1
```

```
NPM : 201043500111
```

```
Nama : Desi Maryani
```

```
Nilai : 78
```

```
-----
```

Output Program Stack

```
-----setelah di POP-----
```

```
elemen ke: 2
```

```
NPM      : 201043500222
```

```
Nama     : Irianti Hapsari
```

```
Nilai    : 84
```

```
elemen ke: 1
```

```
NPM      : 201043500111
```

```
Nama     : Desi Maryani
```

```
Nilai    : 78
```

```
-----
```

Thank you!

