

Type Tstack = <wadah:array[1..10] of character, top:integer>

Procedure CreateStack(output S:Tstack)

{I.S: - ; F.S: S terdefinisi }

{Proses mengisi elemen wadah dengan ' ', top 0}

/\*Kamus Lokal\*/

i : integer

/\*Algoritma\*/

top <-- 0

i traversal [1..10]

S.wadah[i] <-- ' '

Type Tstack = <wadah:array[1..10] of character, top:integer>

Function Infotop(S:Tstack) -> character  
{mengembalikan nilai elemen puncak}

/\*Kamus Lokal\*/

/\*Algoritma\*/

--> S.wadah[top]

Type Tstack = <wadah:array[1..10] of character, top:integer>

Function Top(S:Tstack) -> integer  
{mengembalikan posisi puncak}

/\*Kamus Lokal\*/

/\*Algoritma\*/  
--> S.Top

cek dulu kosong atau tidak

if NOT isEmptyStack(S) then

--> S.top

else

--> ' ' {asumsi: karakter kosong/spasi}

Type Tstack = <wadah:array[1..10] of character, top:integer>

Function isEmptyStack(S:Tstack) -> boolean  
{mengembalikan true bila S kosong}

/\*Kamus Lokal\*/

/\*Algoritma\*/  
if(S.top = 0) then  
    --> true  
else { S.top /= 0 }  
    --> false

Type Tstack = <wadah:array[1..10] of character, top:integer>

Function isFullStack(S:Tstack) -> boolean  
{mengembalikan true bila S penuh}

/\*Kamus Lokal\*/

/\*Algoritma\*/

if (S.top = 10) then

--> true

else { S.top /= 10 }

--> false

Type Tstack = <wadah:array[1..10] of character, top:integer>

Procedure Push(input/output S:Tstack, input e:character)

{I.S: S,e terdefinisi, S mungkin kosong }

{F.S: S tetap, atau infotop(S)=e }

{Proses mengisi elemen e ke puncak S, bila belum penuh}

/\*Kamus Lokal\*/

/\*Algoritma\*/

if (S.top = 10) then

~~output ("Stack sudah terisi penuh")~~

else {Stack belum terisi penuh atau kosong}

S.top <-- S.top + 1 ✓

S.wadah[S.top] <-- e ✓

Type Tstack = <wadah:array[1..10] of character, top:integer>  
Procedure Pop(input/output S:Tstack, output e:character)  
{I.S: S terdefinisi, mungkin kosong }  
{F.S: S tetap, atau e berisi infotop(S) lama }  
{Proses menghapus elemen e dari puncak S, bila belum kosong}

/\*Kamus Lokal\*/

/\*Algoritma\*/

if (S.top = 0) then  
~~output ("Stack Kosong")~~  
else {S.top /= 0}  
~~output (S.wadah[S.top])~~  
e <-- S.wadah[S.top]  
S.wadah[S.top] <-- ' '  
S.top <-- S.top - 1

if NOT isEmptyStack(S) then  
e <-- S.wadah[S.top]  
S.wadah[S.top] <-- "  
S.top <--- S.top - 1

if NOT isEmptyStack(S) then  
e <-- infotop(S)  
infotop(S) <-- "  
top(S) <--- top(S) - 1

Type Tstack = <wadah:array[1..10] of character, top:integer>

Procedure PrintStack(input S:Tstack)

{I.S:-; F.S:-; Proses: menampilkan info elemen S }

/\*Kamus Lokal\*/

i : integer

/\*Algoritma\*/

i traversal [1..S.top]

output (S.wadah[i])







