

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

vicky

parameter formal

Procedure CreateStack(output S:Tstack)

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

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

kamus lokal

i : integer

algoritma

S.top <-- 0

i traversal 1..10

S.wadah[i] <-- ' '

{stack terdefinisi}

S.wadah[i] <-- ' '

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

sonni

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

Kamus Lokal
top:integer

Algoritma

InfoTop(S)-> (S).T[(S).Top]
If isEmptyStack(S) = true then
--> "Stack Kosong"
Else
--> S.T[S.Top]

if isEmptyStack(S) then
--> ''
else
--> S.wadah[S.top]

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

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

kamus lokal

algoritma
-->S.top

-

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

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

Kamus lokal

Algoritma

--> S.top = 0

if S.top=0 then

--> true

else

--> false

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

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

kamus lokal
full : boolean



Algoritma
if S.top = 10 then
 full <- true
else
 full <- false
-> full

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

rafli

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

algoritma

~~if (!isFullStack(S)) then~~

~~S.top <- S.top +1~~

~~S.wadah[S.top] <- e~~

if ~~NOT~~ isFullStack(S) then

if S.top<10 then

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

aufarisq &
steven

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 NOT isEmptyStack(S) then

 e <-- S.wadah[S.top]

 S.wadah[S.top] <-- ''

 S.top <-- S.top - 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..10
 output S.wadah[i]

Procedure addX (input/output T:Tabel, input X: integer)
 {I.S.: T terdefinisi, X terdefinisi }
 {F.S.: isi T.wadah bertambah 1 elemen jika belum penuh}
 {Proses: mengisi elemen T.wadah dengan nilai X}

Kamus Lokal :
 i : integer {iterator}

Algoritma:
if T.size != 10 then
 T.size <- T.size+1
 T.wadah[T.size] <- X

