

Type TQueue2 = <wadah:array[1..10] of character,
head:integer,
tail:integer >
{Queue model II, kondisi head bisa geser 0..kapasitas}
{pergeseran maju pada elemen ketika Tail terhenti}

Function isEmptyQueue(Q:TQueue2) -> boolean
{mengembalikan true bila Q kosong}

Function isFullQueue(Q:TQueue2) -> boolean
{mengembalikan true bila Q penuh}

Type TQueue2 = wadah:array[1..10] of character,
head:integer,
tail:integer >
{Queue model II, kondisi head bisa geser 0..kapasitas}
{pergeseran maju pada elemen ketika Tail terhenti}

Kamus Lokal

~~tail : integer~~

~~isEmptyQueue : boolean~~

Algoritma

if Q.tail == " " salah tipe
 return true
else

if head(Q)=0 and tail(Q)=0 then
 --> true
else
 --> false

if head(Q)=1 and tail(Q)=10 then
 --> true
else
 --> false

Sonni

Function IsTailStop(Q:TQueue2) -> boolean
{mengembalikan true jika Tail tidak dapat lagi geser}
{karena sudah di posisi kapasitas}

Kamus lokal

Algoritma **AND Q.head > 1**
if (Q.tail = 10) then
 --> true
else
 --> false

pikirkan apakah perlu cek kondisi head?

Type TQueue2 = <wadah:array[1..10] of character,
head:integer,
tail:integer >
{Queue model II, kondisi head bisa geser 0..kapasitas}
{pergeseran maju pada elemen ketika Tail terhenti}



Procedure ResetHead(input/output Q:TQueue2)
 {I.S: Tail=kapasitas, head>1; F.S: head=1;
 {Proses: mengembalikan Head ke indeks 1 }
 {Elemen selain head ikut bergeser menyesuaikan}

kamus lokal
 i : integer

algoritma
 if (Q.Head > 1) and Q.Tail = 10 {isTailStop(Q)}
 ~~for traversal 1 .. Q.tail~~
 ~~Q.wadah[i] <- Q.wadah[i+1]~~
 ~~Q.tail <- Q.tail - 1~~
 ~~Q.tail <- i~~
 ~~Q.head <- 1~~

coba pikirkan kondisi ini



Vicky

Procedure Enqueue(input/output Q:TQueue2, input e:character)

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

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

{Proses menambah elemen e ke ekor Q bila belum penuh}

cek spesifikasi, adakah reset?

kamus lokal

i: integer

Type TQueue2 = <wadah:array[1..10] of character,
head:integer,
tail:integer >
{Queue model II, kondisi head bisa geser 0..kapasitas}
{pergeseran maju pada elemen ketika Tail terhenti}

algoritma

if NOT isFullQueue(Q) then

if isTailStop(Q) then { asumsi jika pemanggilan enqueue dengan kondisi queue belum penuh namun tail mencapai kapasitas }

ResetHead(Q)

if isEmptyQueue(Q) then

Q.head <-- 1

Q.tail <-- Q.tail + 1

Q.wadah[Q.tail] <-- e



Procedure Dequeue(input/output Q:TQueue2, output e:character)

{I.S: Q terdefinisi, mungkin kosong }

{F.S: Q tetap, atau e berisi infoHead(Q) lama }

{Proses menghapus elemen e dari head Q bila belum kosong}

{bila tail di kapasitas, head direset 1 diikuti elemen lain}

{asumsi: jika head dan tail sama maka akan melakukan resetHead(Q) atau mengembalikan head ke indeks 1}

kamus lokal

kosong

penuh

satu elemen

lainnya

~~bocor bila Q kosong?~~

algoritma

if (not isEmptyQueue(Q)) then

e <- infoHead(Q)

if (Q.head != 10) then

Q.head <- Q.head + 1

else if (Q.head = 10 OR Q.head = Q.tail) then

resetHead(Q)

else

Q.head <- 0

Q.tail <- 0

f

if isEmptyQueue(Q) then

e <- '@'

else {tidak kosong}

e <- Q.wadah[Q.head]

Q.wadah[Q.head] <- '@'

if (Q.head=Q.tail) then {1 elemen --> kosong}

Q.head <- 0

Q.tail <- 0

else {banyak elemen}

Q.head <- Q.head + 1

if (Q.head=Q.tail AND Q.tail=10) then

resetHead(Q) {2 elemen -> 1 elemen}

