

Type TQueue = <wadah:array[1..10] of character,
 head:integer,
 tail:integer >

{Queue model I, kondisi head 0 atau 1}

{pergeseran maju pada elemen ketika dequeue}

Procedure CreateQueue(output Q:TQueue)

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

{Proses: mengisi elemen wadah dengan ' ', head 0, tail 0}

Function Head(Q:TQueue) -> integer

{mengembalikan posisi elemen terdepan}

Function Tail(Q:TQueue) -> integer

{mengembalikan posisi elemen terakhir}

Function InfoHead(Q:TQueue) -> character

{mengembalikan nilai elemen terdepan}

Function InfoTail(Q:TQueue) -> character

{mengembalikan nilai elemen terakhir}

Function isEmptyQueue(Q:TQueue) -> boolean

{mengembalikan true bila Q kosong}

Function isFullQueue(Q:TQueue) -> boolean

{mengembalikan true bila Q penuh}

Procedure Enqueue(input/output Q:TQueue, 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}

Procedure Dequeue(input/output Q:TQueue, 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}

{lalu geser maju 1 langkah semua elemen di belakang head}

Procedure PrintQueue(input Q:TQueue)

{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }

Procedure ViewQueue(input Q:TQueue)

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

P.Edy: pelajari ADT Queue, buatlah body subrutin sesuai kelompok. identitas grup = angka kedua dari belakang NIM ditambah 2. contoh 92 maka frame 11

2

☐ ☒

```
if (not isFullQueue(Q)) then
```

```
Q.tail <- Q.tail + 1
Q.wadah[ Q.tail ] <-- e
```

↓



Procedure Dequeue(input/output Q:TQueue, output e:character) 3
{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}
{lalu geser maju 1 langkah semua elemen di belakang head}

Kamus Lokal

i: integer

Algoritma

```
if(not isEmptyQueue(Q)) then  
    e <-- Q.wadah[Q.head]  
    Q.wadah[Q.head] <-- null  
    if()  
    ↓
```

```
if Q.tail > 1 then  
    i traversal 1.. Q.tail-1  
        Q.wadah[i] <- Q.wadah[i+1]  
    Q.wadah[Q.tail] <-- ''  
    Q.tail <-- Q.tail - 1  
    if Q.tail = 0 then  
        Q.head <-- 0  
    {end if tail}
```


10 4

Procedure PrintQueue(input Q:TQueue)

{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }

Kamus Lokal

i: integer

Algoritma

```
if(isEmptyQueue(Q)) then
    output("Kosong")
else
    i traversal 1...10
        output(Q.wadah[i])
```

Coba diteliti
udah bener
apa belum?

```
Procedure Enqueue(input/output Q:TQueue, 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}
Procedure Dequeue(input/output Q:TQueue, 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}
{lalu geser maju 1 langkah semua elemen di belakang head}
Procedure PrintQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }
Procedure ViewQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan info elemen tak kosong Q}
```

Type TQueue = <wadah:array[1..10] of character,
head:integer,
tail:integer >
{Queue model I, kondisi head 0 atau 1}
{pergeseran maju pada elemen ketika dequeue}
Procedure CreateQueue(output Q:TQueue)
{I.S: - ; F.S: Q terdefinisi}
{Proses: mengisi elemen wadah dengan ' ', head 0, tail 0}
Function Head(Q:TQueue) -> integer
{mengembalikan posisi elemen terdepan}
Function Tail(Q:TQueue) -> integer
{mengembalikan posisi elemen terakhir}
Function InfoHead(Q:TQueue) -> character
{mengembalikan nilai elemen terdepan}
Function InfoTail(Q:TQueue) -> character
{mengembalikan nilai elemen terakhir}
Function isEmptyQueue(Q:TQueue) -> boolean
{mengembalikan true bila Q kosong}
Function isFullQueue(Q:TQueue) -> boolean
{mengembalikan true bila Q penuh}

Masih lanjut
ndak nih??
Apa udah
segini aja?

Oke
kalau notasi
pake output,
kalau bahasa
c pake print

setauku si
output, kalau
print bahas c

okeoke

Ini lho Mas -->
Print semua
array (10 array)

Procedure ViewQueue(input Q:TQueue)

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

Kamus Lokal

i : integer ✓

gmn gan?
apa yang
diganti?

halo?

Algoritma

if (not isEmptyQueue) then
 i traversal 1...Q.tail ✓
 output(Q.wadah[i]) ✓

Procedure Enqueue(input/output Q:TQueue, 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}
Procedure Dequeue(input/output Q:TQueue, 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}
{lalu geser maju 1 langkah semua elemen di belakang head}
Procedure PrintQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }
Procedure ViewQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan info elemen tak kosong Q}

Type TQueue = <wadah:array[1..10] of character,
 head:integer,
 tail:integer >
{Queue model I, kondisi head 0 atau 1}
{pergeseran maju pada elemen ketika dequeue}
Procedure CreateQueue(output Q:TQueue)
{I.S: - ; F.S: Q terdefinisi}
{Proses: mengisi elemen wadah dengan ' ', head 0, tail 0}
Function Head(Q:TQueue) -> integer
{mengembalikan posisi elemen terdepan}
Function Tail(Q:TQueue) -> integer
{mengembalikan posisi elemen terakhir}
Function InfoHead(Q:TQueue) -> character
{mengembalikan nilai elemen terdepan}
Function InfoTail(Q:TQueue) -> character
{mengembalikan nilai elemen terakhir}
Function isEmptyQueue(Q:TQueue) -> boolean
{mengembalikan true bila Q kosong}
Function isFullQueue(Q:TQueue) -> boolean
{mengembalikan true bila Q penuh}

5
9

Procedure CreateQueue(output Q:TQueue)

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

{Proses: mengisi elemen wadah dengan ' ', head 0, tail 0}

Kamus Lokal

i : integer

Algoritma

~~Q.head -> 0~~
~~Q.tail -> 0~~

Q.head <-- 0

Q.tail <-- 0

i traversal 1..10

Q.wadah[i] <-- ' '

6

6

Function Head(Q:TQueue) -> integer
{mengembalikan posisi elemen terdepan}

kamus lokal

Algoritma

-> Q.head



Function Tail(Q:TQueue) -> integer
{mengembalikan posisi elemen terakhir}

kamus lokal

Algoritma

-> Q.tail



7



Function InfoHead(Q:TQueue) -> character
{mengembalikan nilai elemen terdepan}

}

kamus lokal



Algoritma

```
if not isEmptyQueue(Q) then
    -> Q.wadah[Q.Head] ✓
else
    --> ''
```

kamus lokal

Algoritma

```
if (Q.head = 0) then
    -> ''
else
    -> Q[1]
```

kamus lokal

Algoritma

~~-> Q[Q.head]~~

Function InfoTail(Q:TQueue) -> character
{mengembalikan nilai elemen terakhir}

9

kamus lokal



algoritma

```
if not isEmptyQueue(Q) then  
    -> Q.wadah[Q.tail]  
else  
    --> ''
```

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

/0

kamus lokal

10

algoritma

if (Q.head = 0) and (Q.tail = 0) then
 -> True

else
 -> False

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

||

kamus lokal

algoritma

if (Q.head=1 AND Q.tail=10) then

---> true

else

--> false

