```
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}
```

P.Edy: pahami ADT Queue, lalu buatlah realisasi body fungsi/prosedur sesuai kelompok. Identitas grup = angka kedua dari belakang NIM ditambah 2.

```
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}
```

2

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}

Kamus lokal
i: integer {menunjukkan sebagai counter}

Algoritma

```
i traversal [1..10]
  if (TQueue.tail<10) then
    if not isFullQueue(Q) then
      if isEmptyQueue(Q) then
        Q.head <-- 1
      Q.tail <-- Q.tail + 1
      Q.wadah[ Q.tail ] <-- e
  else
      infoTail(Q)
```

infoTail(Q)
<-- e

--i traversal [1...10].
. if
TQueue.Tail < 10 then.
. infoTail(Q)
= e. .
else.
.
TQueue.Tail = 10

Koreksi ndess wkwkw

gatau ini salah atau bnr wkwk blm selesai

Kerajakan euy gapaham

3 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}

```
kamus lokal
  i : integer {counter}
Algoritma
if not isEmptyQueue(Q) then
    e <-- infoHead(Q)
    if Q.tail > 1 then
        i traversal [1... (Q.tail-1)]
            Q.wadah[i] <-- Q.wadah[i + 1]  {Antrian bergeser }
    else
        Q.wadah[1] <-- " "  { Antrian kosong, tak ada character }
        Q.tail <-- 0
        Q.head  <-- 0
```

```
Q.wadah[ Q.tail ] <-- ' '
Q.tail <-- Q.tail - 1
```

```
Procedure PrintQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q
{Kamus Lokal}
    i : integer     {counter}
{Algoritma}
    i traversal [1...10]
      output(Q.wadah[i])
```

Kelompok 4

5

Procedure ViewQueue(input Q:TQueue)

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

```
Kamus lokal
    i: integer

Algoritma
    i traversal [1..10]
        if (Q.wadah[i] ≠ ' ') then
            output Q.wadah[i]
```

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

Oke deh lanjut nanti lagi, laper susah buat mikir wwkkw

ngecek kosong gk nya bukan 0 keknya, kan char.. gatau deng || tp wadahnya integer.....gtau ah bingung ak ikan kembung jgn tny aku..... wkwkwk sama aku jg ikan

gtu bukann

Semangat yoshh

klo Q.wadah[i] != null gmn?

ngga ada return ya, pakenya panah kanan. Return kan bahasa C || oooooooooooo ok

Mungkin ngga si kalo head sama tailnya kosong?

punya yg nimnya xxxxxxxxxxx3x (slide 5 :D)

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
i traversal [1..10]
Q.wadah[i] <-- ' '
Q.head <-- 0
Q.tail <-- 0

bener nga ya?

insyaallah

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

kamus lokal

algoritma
-> Q.Head

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

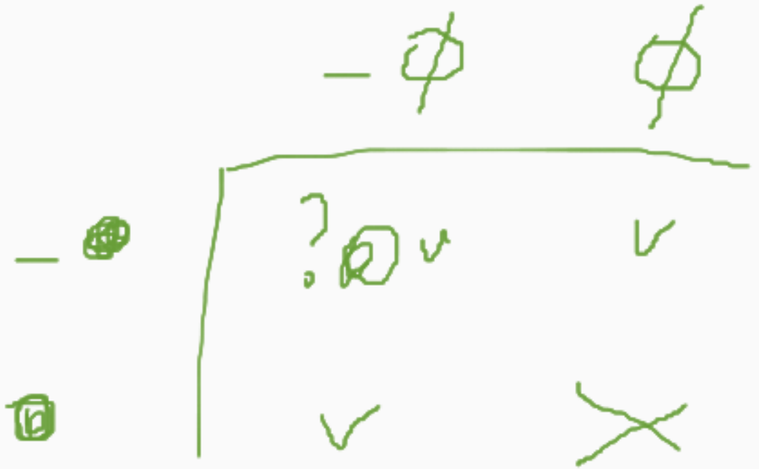
kamus lokal

algoritma
-> Q.Tail

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

kamus lokal

algoritma
  if not isEmptyQueue(Q) then
    --> Q.wadah [Q.head]
  else {Q kosong}
    --> ' '



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

9

```
kamus lokal :
    i : integer

algoritma
```

```
kamus lokal

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

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

Kamus lokal

Algoritma

```
if (Q.wadah[10] = Q.Tail) then
        -> true
else
        -> false
```

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

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

Kamus Lokal

-

Algoritma

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

atau

-> Q.head = 0 and Q.tail = 0

{nanya: kalau head=0, tail auto 0 kan?
mungkin gk perlu ngecek tail? atau sebaliknya.
lebih efisien?}