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}

halo

jangan disini

```
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
    i traversal [1..10]
        Q.wadah[i] <-- '@'
```

Head(Q) <--0

```
i traversal 1..10

i travesal [1..10]
    Q.wadah[i]<--'@'
Q.head <-- 0
Q.tail<-- 0
```

saya pindah ke sebelah ges semangat

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

Kamus Lokal

Algoritma
--> Q.head

bismillah bener

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

Kamus Lokal

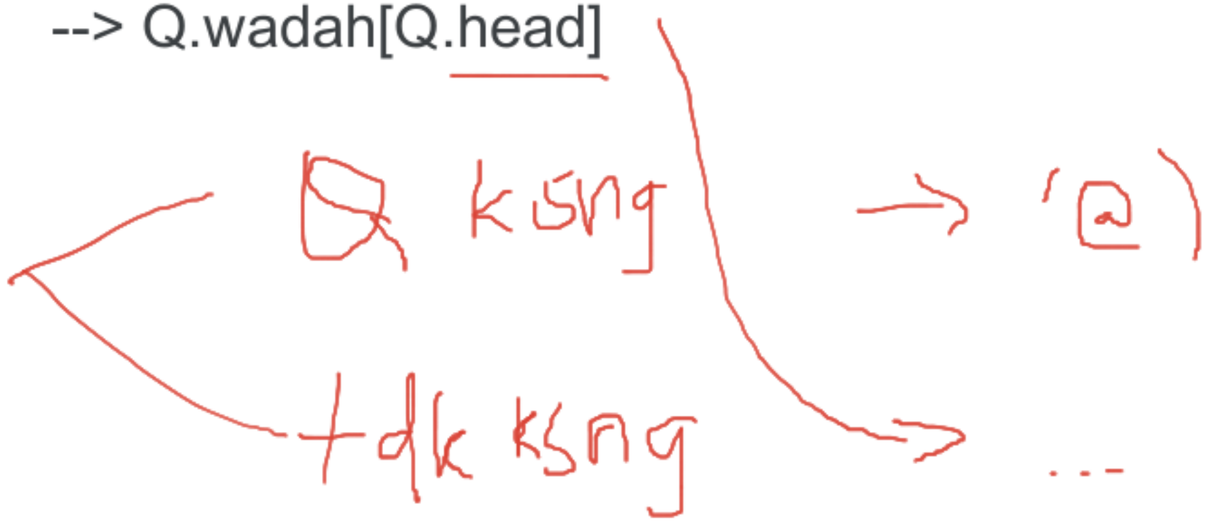
Algoritma
--> Q.tail

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

Kamus Lokal

Asumsi

Algoritma:
--> Q.wadah[Q.head]




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

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

Kamus Lokal
i : integer

Algoritma:
--> Q.wadah[Q.tail]

```
if (Q.tail = 0) then
    --> '@'
else
    --> Q.wadah[Q.tail]
```

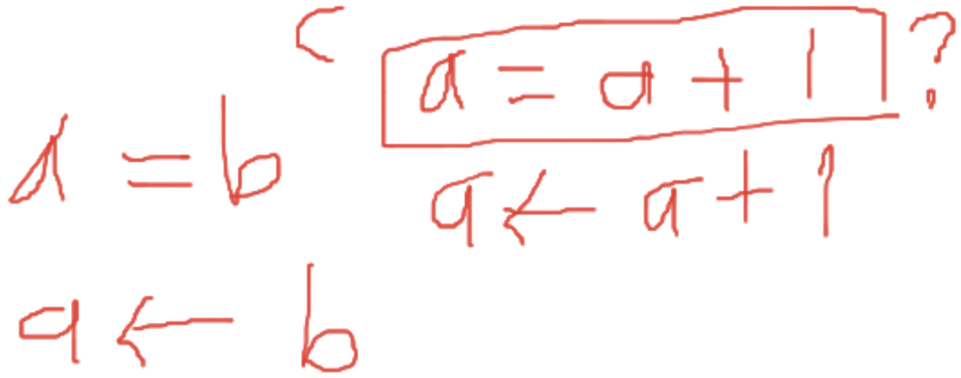
Function isFullQueue(Q:TQueue) -> boolean
{mengembalikan true bila Q penuh}

kamus lokal

algoritma:
if Q.tail = 10 then
--> true
else
--> false

--> Q.tail == 10

Terima kasih sudah mengingatkan Pak



ohh okayy makasii

head kan tetep di elemen pertama

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

Kamus Lokal

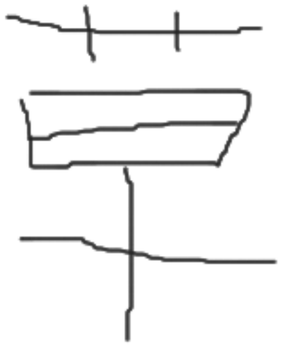
Algoritma:

if (Q.head= 0) then
--> true

else
--> false

--> Q.tail =\= 10

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



hai

hai juga

OLA

```
Procedure PrintQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }

Kamus Lokal
  i: integer

Algoritma
   i traversal 1..10
      output Q.wadah[i]                                    konpeko~
```

Procedure ViewQueue(input Q:TQueue)
{I.S:-; F.S:-; Proses: menampilkan info elemen tak kosong Q}

```
Kamus Lokal
    i : integer
    pos : integer

Algoritma

if (not isEmpty(Q)) then
    pos <- Tail(Q)
  i traversal 1..pos
        --> Q.wadah[i]
```

```
Kamus lokal
    i: integer

Algoritma
    if (not isEmpty(Q) then
        i traversal 1.. Tail(Q)
            output Q.wadah[i]
```

konpeko~

aku mo pake fungsi dari Tail(Q) buat nyari elemen terakhir, terus dilooping sebanyak elemen terakhirnya

Function sizeQueue(Q:TQueue) -> integer
{mengembalikan panjang/banyak elemen}

Kamus Lokal

Algoritma
-->Q.tail - Q.head + 1

jangan lupa ditambah 1
ngab
klo misal isi arraynya 1
kan head dan tail nya
sama klo dikurangi
nanti hasilnya 0 dong

ohh okeh2

iya ngab

```
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

Algoritma
    if not isFullQueue(Q) then
      if isEmptyQueue(Q) then
          Q.head <-- 1
      Q.tail <-- Q.tail + 1
      Q.wadah[Q.tail] <-- e
```

```
if not isFullQueue(Q) then
   if isEmptyQueue(Q) then
      Q.head<--1
      Q.tail<--1
      Q.wadah[1]<--e
   else i ← 1
      while(Q.wadah[i]=/='@') do
         i<--i+1
       {end while}
      Q.wadah[i] <-- e
      Q.tail<--Q.tail+1
```

```
if not isFullQueue(Q) then
    if isEmptyQueue(Q) then
        head(Q) <-- 1
    tail(Q) <-- tail(Q) + 1
    Q.wadah[ tail(Q) ] <-- e
```

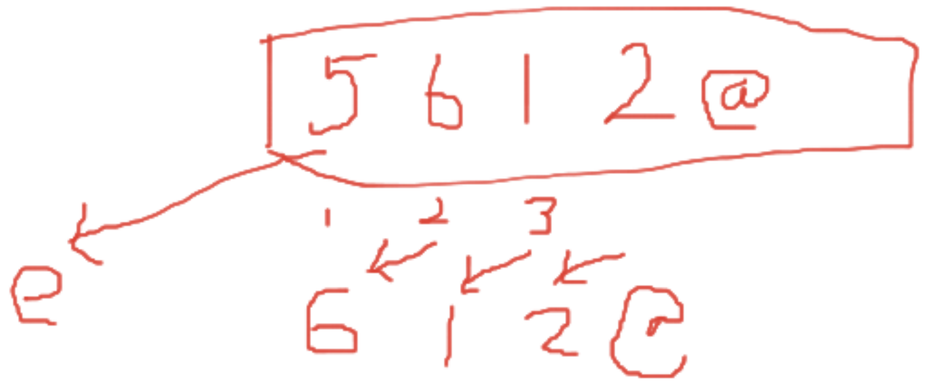
```
Procedure Dequeue(input/output Q:TQueue, output e:character)
{I.S: Q terdefinisi, mungkin kosong }
{F.S: Q tetap, atau e berisi infoHead(Q) lama }
{Proses menghapus elemen e dari head Q bila belum kosong}
{lalu geser maju 1 langkah semua elemen di belakang head}

Kamus Lokal
    i : integer

Algoritma
    e <-- Q.wadah[Q.head]
    if(not isEmptyQueue(Q)) then
        i traversal [Q.head .. (Q.tail-1)]
            Q.wadah[i] <-- Q.wadah[i+1]
        {endtraversal}
        Q.wadah[i+1] <-- '@'
        Q.tail <-- i
```

bocor bila Q kosong

else
e ← '@'



masih bocor bila Q dari 1 elemen menjadi kosong, seharusnya head jadi 0 juga

```
if (not isEmptyQueue(Q)) then
    e <-- Q.wadah[Q.head]
    i traversal [Q.head .. (Q.tail-1)]
            Q.wadah[i] <--    Q.wadah[i+1]
        {endtraversal}
        Q.wadah[i+1] <-- '@'
        Q.tail <-- i
else
    e <-- '@'
```