

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: 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) 2
{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 :

algoritma :

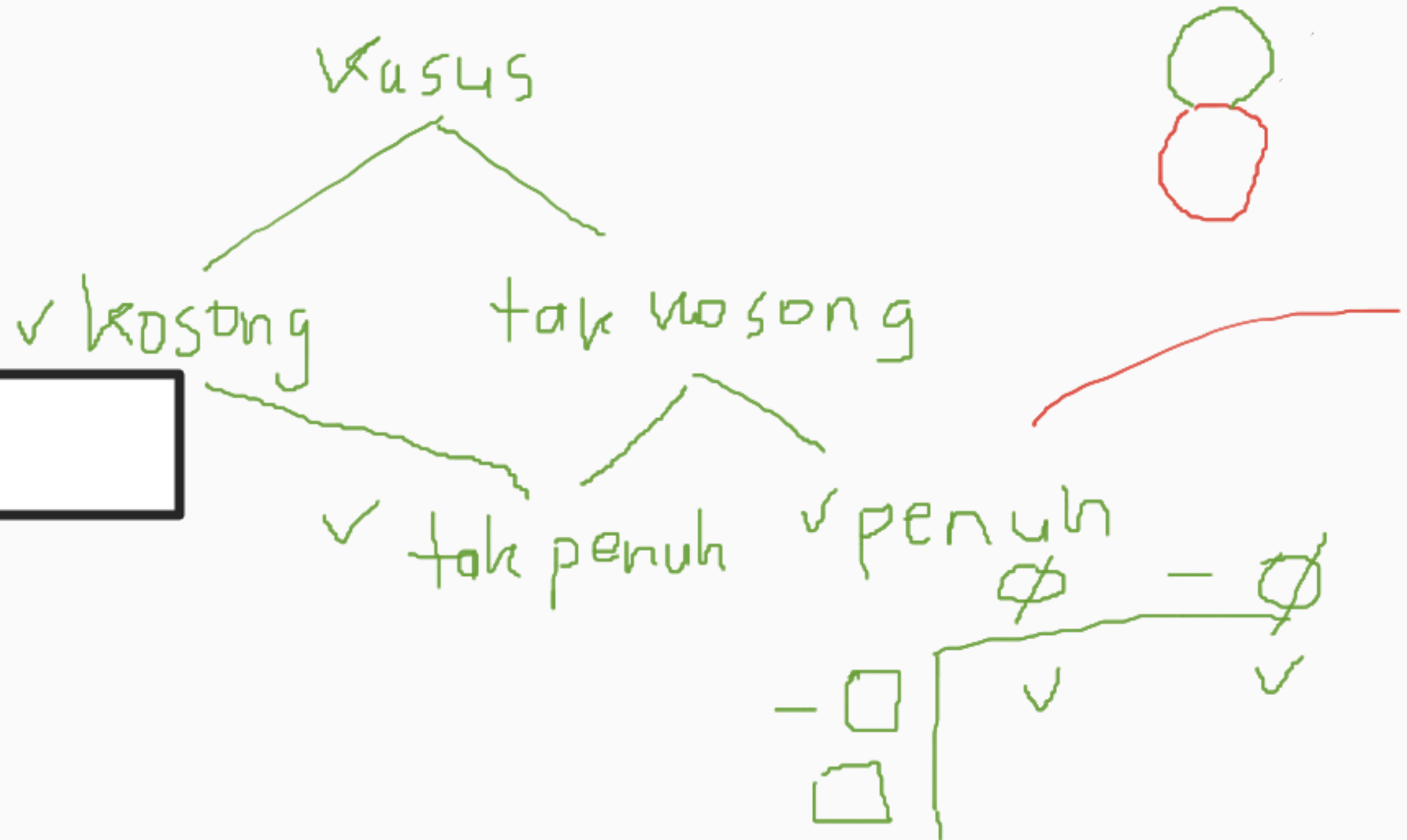
if (not isFullQueue(Q)) then

if isEmptyQueue(Q) then
Q.head \leftarrow Q.head + 1

Q.tail \leftarrow Q.tail + 1

Q.wadah[Q.tail] \leftarrow e

{endif}



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:

if not isEmptyQueue(Q) then ✓

e <- infoHead(Q) ✓

i traversal 2...Q.tail

Q.wadah[i-1] <- Q.wadah[i] ✓

? Q.wadah[Q.tail] <-- '' {membersihkan tail lama}

Q.tail <- Q.tail - 1

if Q.tail = 0 then ✓

Q.head <- 0 {bila antrian habis, ubah head dan tail ke 0 lagi}

8,5

Procedure CreateQueue(output Q:TQueue)

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

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

Kamus Lokal :

j : integer

Algoritma

~~Head~~[Q] <- 0

~~Tail~~[Q] <- 0

Q.head <-- 0

Q.tail <-- 0

j traversal 1..10

Q.wadah[j] <-- ' '

pembatas parameter

pembatas struktur

pembatas posisi

pembatas komentar

**Outputnya
harus 0 Ya?**

4

5

()

< >

[]

{ }

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

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

Kamus Lokal

Algoritma

If (not IsEmptyQueue(Q)) then
-> Q.wadah[Head(Q)]

else
-> "

6



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

sebelum mulai mari kita berdoa
terlebih dahulu

7

Kamus lokal

Algoritma:

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

Docor bila
Q kosong

Kamus lokal

posTail : integer

Algoritma

```
posTail <- Tail(Q)  
-> Q.wadah[posTail]
```

6

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

Kamus lokal

Algoritma

if Head(Q) = 0 and Tail(Q) = 0 then
 -> True
else
 -> False

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

9

Kamus lokal

9.5

Algoritma

if (Q.tail = 10) then
 --> True
else
 --> False

8 Q.head = 1

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

**Kamus
Lokal**

i : integer

parameter formal

Algoritma

i traversal (1...10)

output T.Queue[i]

i traversal (1...Tail(Q))

ouput Q.wadah[i] ✓

i traversal [1..10]
 output Q.wadah[i]



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

Kamus lokal

i : integer

Algoritma

```
{ i traversal [1..]
  if not isEmptyQueue(Q) then
    i traversal [1..10]
    output Q.wadah[i]
  if not isEmptyQueue(Q) then
    i traversal [Q.head..Q.tail]
    output Q.wadah[i]
```