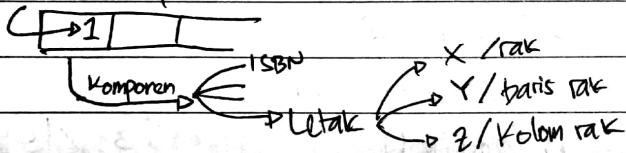


1 a) Asumsi

- Jika pada posisi baru, belum terdapat buku, maka langsung pindahkan
- jika pada posisi baru sudah ada buku, maka posisinya ditukar

misal Buku 1 pada index ke 1



Procedure UpdatePosisi (i/o: T:Book; i: X:integer, B:integer, C: Integer, D:integer)

UpdatePosisi (T,X,B,C,D) prosedur untuk mengubah posisi buku ke-X ke posisi di rak B, baris C, kolom D; Apabila pada posisi baru, letaknya ditukar?

Kamus

Ada: boolean

i : integer

kmp : integer

Algoritma

Ada \leftarrow False

i = 1

while ((Ada = False) AND (i \leq T.Neff)) do { pengecekan apakah posisi baru terdapat buku }

if (T.T[i].letak.X = B) AND (T.T[i].letak.Y = C) AND (T.T[i].letak.Z = D) then

Ada \leftarrow True

else

i \leftarrow i + 1

if (Ada = True) then

tmp \leftarrow T.T[i].letak.X

T.T[i].letak.X \leftarrow T.T[x].letak.X

T.T[x].letak.X \leftarrow tmp

tmp \leftarrow T.T[i].letak.Y

T.T[i].letak.Y \leftarrow T.T[x].letak.Y

T.T[x].letak.Y \leftarrow tmp

tmp \leftarrow T.T[i].letak.Z

T.T[i].letak.Z \leftarrow T.T[x].letak.Z

T.T[x].letak.Z \leftarrow tmp

else:

T.T[x].letak.X \leftarrow B

T.T[x].letak.Y \leftarrow C

T.T[x].letak.Z \leftarrow D

b) function JmlBuku (B: Book)

{ menghitung jumlah buku dengan rumus $\sum_{i=1}^n B.T[i].jumlah$ }

kamus

Jumlah: integer , i: integer

Algoritma

Jumlah \leftarrow 0

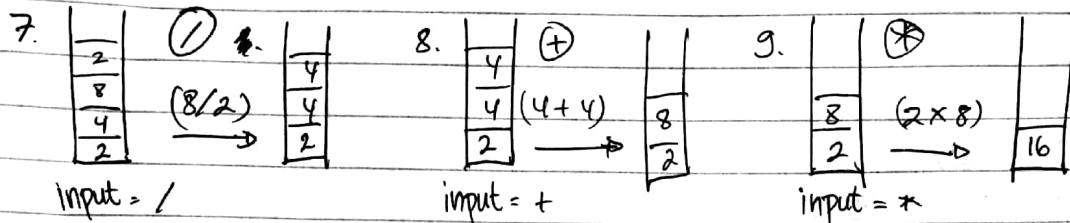
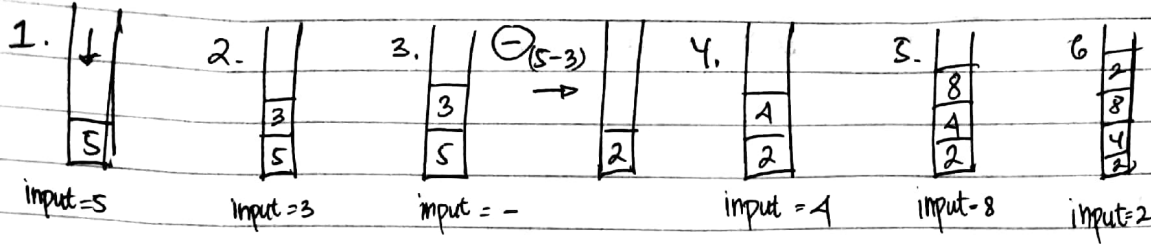
i traversal [1.. B.Neff]

Jumlah \leftarrow Jumlah + B.T[i].jumlah

\rightarrow Jumlah

2) a) Postfix = 5 3 - 4 8 2 / + *

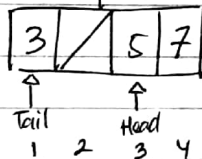
Stack



Hasilnya 16

- melakukan iterasi dari awal postfix, jika angka, lakukan operasi Push ke Stack. jika operator, Pop elemen pada stack dua kali dan masukkan ke dalam variabel, lalu lakukan operasi dengan operator, hasilnya di-Push ke dalam Stack.

b) Isi akhir dari queue



Head() = 3

Tail() = 1

infoHead() = 5

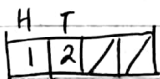
infoTail() = 3

(proses, FYI) optional

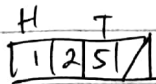
Enqueue (1)



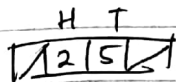
Enqueue (2)



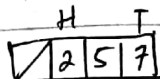
Enqueue (5)



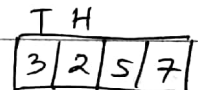
Dequeue ()



Enqueue (7)

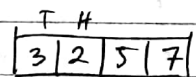


Enqueue (3)



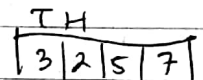
Enqueue (9)

↳ Queue penuh

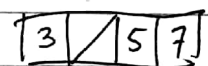


Enqueue (2)

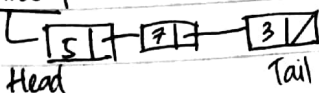
↳ Queue penuh



Dequeue ()



3) a) First



36) Algoritma

if (first (L) = Nil) then

Output ("Under flow")

q Karena list sudah kosong?

Else

max \leftarrow 0

Q \leftarrow Nil

R \leftarrow Nil

S \leftarrow Nil

P \leftarrow First (L)

while (P \neq Nil) do :

if (info (P) > max) then

max \leftarrow info (P)

Q \leftarrow P

P \leftarrow Next (P)

S \leftarrow Q

R \leftarrow P

if (S = nil) then

First (L) \leftarrow R

Else

Next (S) \leftarrow Next (R)

Next (R) \leftarrow Nil

X \leftarrow info (R)