

2. DEFINISI DAN SPESIFIKASI

a. type Pasien = \langle Nomor Anggota : integer , Nama : string , Jenis kelamin : character
Tanggal lahir : Tanggal , Prioritas : integer \rangle

type Tanggal = \langle day : integer [1..31] , month : integer [1..12] , year : integer \rangle

1. type Tabel = \langle wadah : array [1..20] of Pasien , top : integer \rangle

b. Cara akses

Nomor anggota \rightarrow A.wadah [index_pasien - P] . NoAnggota

Nomor Pasien \rightarrow A.wadah [index_pasien - P] . NomorPasien

Jenis kelamin \rightarrow A.wadah [index_pasien - P] . Jeniskelamin

Tanggal lahir , hari \rightarrow A.wadah [index_pasien - P] . Tanggal lahir . Hari

Tanggal lahir , bulan \rightarrow A.wadah [index_pasien - P] . Tanggal lahir . bulan

Tanggal lahir , tahun \rightarrow A.wadah [index_pasien - P] . Tanggal lahir . tahun

Prioritas \rightarrow A.wadah [index_pasien - P] . Prioritas

3. procedure TumpukBerkas (input/output S1 : stack, input/output S2 : stack, input/output
S3 : stack, input P : Pasien)

Kamus lokal

Algoritma

if P.Prioritas = 1 then

if S2.top < 10 then

S1.top \leftarrow S1.top + 1

S1.wadah [S1.top] \leftarrow P

else if P.Prioritas = 2 then

if S2.top < 10 then

S2.top \leftarrow S2.top + 1

S2.wadah [S2.top] \leftarrow P

else if P.Prioritas = 3 then

if S3.top < 10 then

S3.top \leftarrow S3.top + 1

S3.wadah [S3.top] \leftarrow P

Aplikasi

S1 : stack

{ kondisi awal : S1.wadah = [P1, P2, P3] , S1.top = 3 }

S2 : stack

{ kondisi awal : S2.wadah = [P4, P7] , S2.top = 2 }

S3 : stack

{ kondisi awal : S3.wadah = [P5, P6, P8, P9] , S3.top = 4 }

P10 : Pasien

TumpukBerkas (S1, S2, S3, P10)

4. type Queue = <wadah: array [1..5] of Pasien, head: integer, tail: integer>

Realisasi

Procedure Mengantre (input/output Q: Queue, input P: Pasien)

Kamus Lokal

i, j = integer

Algoritma

j ← 0

if isEmpty (Q) then

Q.head ← 1

Q.tail ← 1

Q.wadah [Q.head] ← P

else

if isfull (Q) then

output ('Queue Penuh')

else

i traversal [Q.head..Q.tail]

if Q.wadah [i].prioP ≤ P.prioP then

j ← j+1

Q.tail ← Q.tail+1

if Q.wadah [j] = 0 then

Q.wadah [j] ← P

else

i traversal [Q.tail...j]

Q.wadah [i] ← Q.wadah [i-1]

Q.wadah [j] ← P

boolean isEmpty (input Q: Queue)

→ Q.head = 0 and Q.tail = 0

boolean isFull (input Q: Queue)

→ Q.head = 1 and Q.tail = 5

(2).