Kasus antrian: M/M/n tanpa jockeying

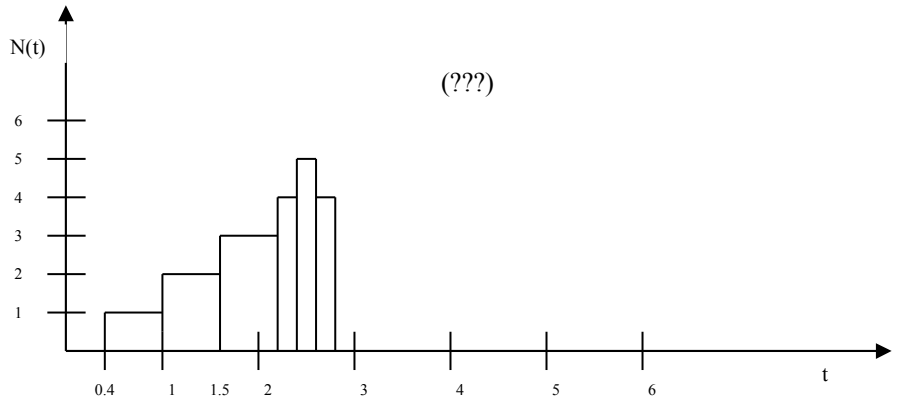
Sistem antrian dengan dua buah loket.

Entitas	Waktu antar kedatangan	Waktu pelayanan
1	0.4	2.0
2	0.6	1.5
3	0.5	1.4
4	0.6	1.0
5	0.2	1.5
6	0.8	1.8
7	1.5	0.5
8	0.5	0.5

Antrian dilayani secara FIFO dan bila kedua loket kosong maka loket pertama mendapat prioritas untuk melayani terlebih dahulu.

(a) Events: - kedatangan customer
- kepergian customer
State: - loket idle / busy
- antrian kosong / isi

(b) Grafik jumlah customer setiap waktu



Jumlah customer pada saat t

Entitas	Waktu kedatangan	Waktu dilayani
1	0.4	0.4 – 2.4
2	1.0	1.0 – 2.5
3	1.5	2.4 – 3.9
4	2.1	2.5 – 3.5
5	2.3	3.5 – 5.0
6	3.1	3.9 – 5.7
7	4.6	5.0 – 5.5
8	5.1	5.5 – 6.0

(c) Panjang antrian setiap saat

Waktu	0.4	1.0	1.5	2.1	2.3	2.4	2.5	3.1	3.5	3.9	4.6	5.0	5.1	5.5	5.7
Panjang	0	0	1	2	3	2	1	2	1	0	1	0	1	0	0

Panjang antrian rata-rata:
(hitung luas daerah dari grafik)

(d) Delay customer

Entitas	1	2	3	4	5	6	7	8
Delay	0	0	0.9	0.4	1.2	???		

Delay customer rata-rata:

(e) Utilisasi server 1 =

Utilisasi server 2 =

(f) State sistem selama simulasi

t	S1	S2	Q1	Q2	t_A	t_{D1}	t_{D2}	C1	C2
0.4	1	0	0	0	1.0	2.4		1	
1.0	1	1	0	0	1.5	2.4	2.5	1	2
1.5	1	1	1 (3)	0	2.1	2.4	2.5	1	2
2.1	1	1	1	1 (4)	2.3	2.4	2.5	1	2
2.3	1	1	2 (3,5)	1	3.1	2.4	2.5	1	2
2.4	1	1	1 (5)	1	3.1	3.9	2.5	3	2
2.5	1	1	1	0	3.1	3.9	3.5	3	4
3.1	1	1	1	1 (6)	4.6	3.9	3.5	3	4
3.5	1	1	1	0	4.6	3.9	5.3	5	6
3.9	1	1	0	0	4.6	5.4	5.3	3	6
4.6	1	1	1 (7)	0	5.1	5.4	5.3	5	6
5.1	1	1	1	1 (8)		5.4	5.3	5	6
5.3	1	1	1	0		5.4	5.8	5	8
5.4	1	1	0	0		5.9	5.8	7	8
5.8	1	0	0	0		5.9		7	
5.9	0	0	0	0					

Keterangan:
t: waktu simulasi
S1: status server 1
S2: status server 2
Q1: panjang antrian 1
Q2: panjang antrian 2
t_A: waktu kedatangan berikutnya
t_{D1}: waktu kepergian customer dari server 1
t_{D2}: waktu kepergian customer dari server 2
C1: customer yang sedang dilayani di server 1
C2: customer yang sedang dilayani di server 2

Kasus antrian: M/M/n dengan jockeying

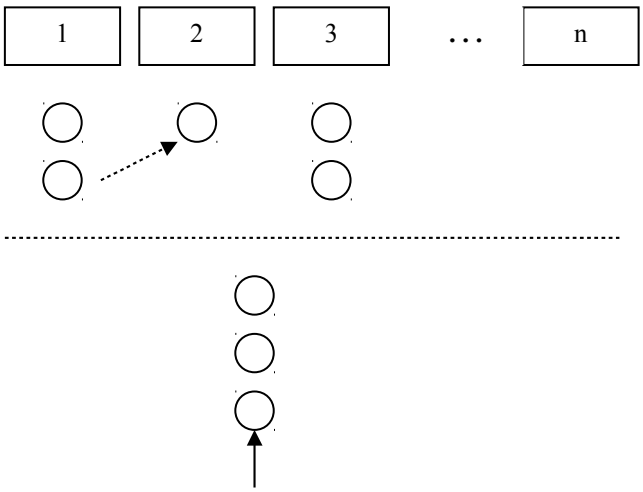
Multiteller Bank

- Permasalahan:

Sistem antrian dengan N buah pelayanan dan masing-masing pelayanan memiliki lajur antrian. Pola pelayanan untuk setiap antrian adalan FIFO. Pengunjung datang dengan memilih antrian yang terpendek.
Jumlah pengunjung dalam antrian ke-i pada suatu saat dinyatakan dengan NCust(i).
Jika dengan selesainya sebuah pelayanan pada server i menyebabkan NCust(i) < NCust(j), maka pengunjung yang berada di ekor antrian ke-j akan pindah ke ekor dari antrian ke-i. Jika ada lebih dari satu j yang memenuhi persyaratan tersebut, maka harga j yang diambil adalah j yang terdekat dengan i, prioritas pencarian di sebelah kiri i dahulu, baru kemudian disebelah kanan i. Pengunjung yang pindah dari antrian ke-j tersebut akan langsung dilayani oleh server ke-i jika status server ke-i adalah idle, atau akan menjadi ekor antrian ke-i jika server ke-i sedang busy.
Pola kedatangan adalah Poisson dengan waktu antar kedatangan berdistribusi eksponensial dengan waktu antar kedatangan rata-rata adalah MArr. Pola pelayanan sama di semua server, yang berdistribusi eksponensial, dengan waktu pelayanan rata-rata MServ.

- Ingin diketahui:

waktu tunggu rata-rata dan panjang antrian rata-rata jika jumlah server N di ubah-ubah.



- Event: - kedatangan customer, kedatangan pengunjung untuk dilayani
  - kepergian (i), selesainya sebuah pelayanan di server ke-i

  Peristiwa jockeying bukan dianggap sebagai event, karena terjadi akibat kepergian/departure.

- Awal simulasi:
  - semua server berstatus idle
  - semua antrian kosong

- Akhir simulasi:
  - setelah waktu buka bank habis (bank tutup)
    (8 jam operasional)

- Contoh kasus:
  1. jam buka bank (=8 jam)
  2. rata-rata waktu antar kedatangan (MArr = 1 menit)
  3. jumlah maksimum teller (= 7 orang)
  4. jumlah minimum teller (= 4 orang)
  5. rata-rata waktu pelayanan (MServ = 4.5 menit)

- Ingin dicari:
  - berapa jumlah teller minimum sehingga waktu tunggu rata-rata <= 25 menit (jika pengunjung terlalu lama menunggu (misal >25 menit) maka dia cenderung untuk membatalkan diri dan pulang).

- Program:

Input: - panjang simulasi (= 8 jam)
- butir 2,3,4,5.

Output: rata-rata jumlah total pengunjung di antrian.

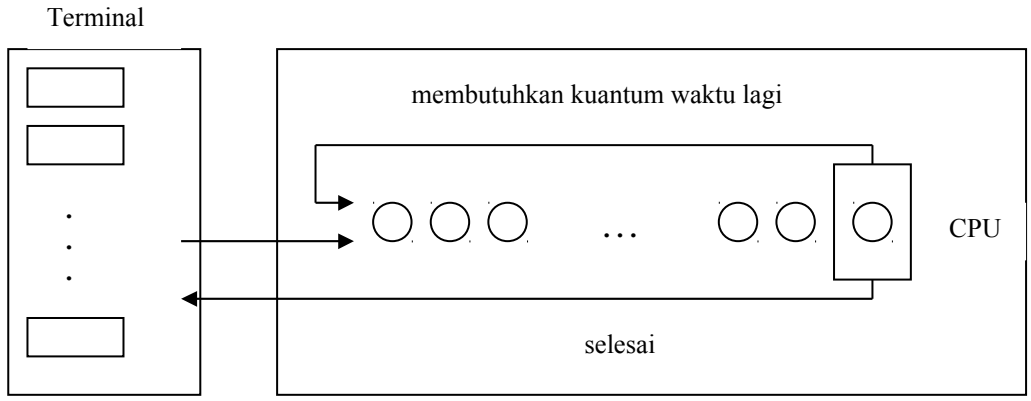
Kasus antrian: Time Shared Computer Model

- Permasalahan:

Sebuah komputer mainframe dapat digunakan untuk N terminal. Setiap terminal akan mengirim job untuk diproses oleh CPU dengan waktu antar pengiriman berdistribusi eksponensial, dengan waktu antar pengiriman rata-rata sebesar MThink. Setiap job yang dikirim akan masuk ke antrian dan diproses oleh CPU dengan cara"round robin" (setiap job akan mendapat alokasi kuantum waktu sebesar Q). Jika satu job belum selesai untuk satu kuantum waktu, maka dia akan dimasukkan lagi ke dalam antrian.
Untuk setiap kuantum waktu, CPU membutuhkan waktu sebesar Q + T, dengan T adalah overhead (konstant) yaitu waktu untuk memindahkan job ke antrianlagi (swap).
Waktu proses total setiap job adalah S yang berdistribusi eksponensial dengan waktu proses rata-rata adalah MServ.

- Ingin diketahui:

  Kinerja sistem jika jumlah terminal diubah-ubah.
  Yaitu:
  - response time rata-rata
    R_i, adalah response time job ke-i, yaitu selisih waktu antara job ke-i meninggalkan terminal sampai selesai diproses oleh CPU.
  - jumlah rata-rata job dalam antrian.



- Event: - Kedatangan job dari terminal ke CPU
  - Akhir sebuah kuantum waktu, setelah sebuah job menerima kuantum waktu proses, ditambah overhead T.

- Awal simulasi:
  - CPU idle
  - antrian kosong
  - semua terminal dalam kondisi 'think'.

- Akhir simulasi:
  - pada saat job ke-n selesai dilayani/diproses.

- Contoh kasus:
  - rata-rata waktu 'think' = 25'
  - rata-rata waktu proses CPU = 0.8'
  - kuantum waktu Q = 0.1'

- overhead waktu = 0.015’

➢ Ingin dicari:
Berapa jumlah terminal maksimum sehingga response time rata-rata <= 30’
(untuk jumlah terminal N = 10, 20, 30,…, 80).

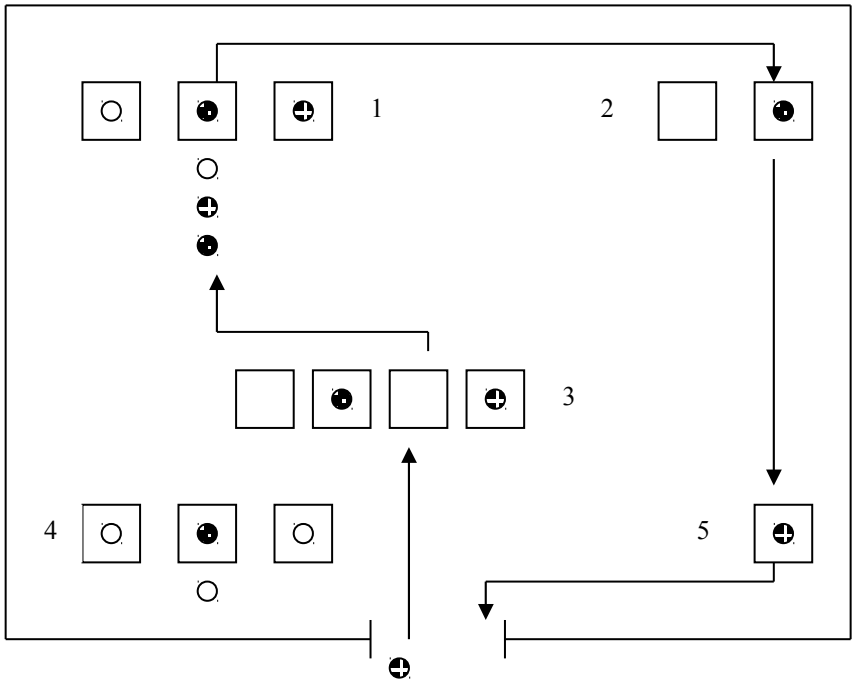
➢ Program:
Input:
- minimum terminal (= 10)
- maksimum terminal (= 80)
- increment (= 10)
- rata-rata waktu proses (= 0.8’)
- rata-rata waktu ‘think’ (= 25’)
- kuantum waktu (0.1’)
- swap time (0.015’)
- jumlah total job yang diamati (= 1000).

Output:
- rata-rata jumlah antrian
- rata-rata response time
- utilisasi CPU.

Kasus antrian: Job Shop (Produksi Berantai)

- Permasalahan:

Dalam sebuah pabrik, untuk mendapatkan sebuah barang jadi, barang harus melewati beberapa pos kerja secara berturut-turut. Jumlah mesin pada setiap pos kerja berbeda-beda. Contoh situasi:



tipe 1

Sistem ini sebenarnya adalah sebuah jaringan dari antrian multiserver.
Pada situasi di atas, terdapat tiga tipe produksi, dan setiap tipe harus melewati pos kerja tertentu yang juga disebut dengan rute.

Tipe	Rute	
1	3,1,2,5	(tipe 1 harus melewati 4 buah task untuk diselesaikan)
2	4,1,3	(tipe 2 harus melewati 3 buah task untuk diselesaikan)
3	2,5,1,4,3	(tipe 3 harus melewati 5 buah task untuk diselesaikan)

Setiap pos kerja terdiri atas sejumlah mesin yang sejenis dan terdiri hanya antrian tunggal.

Pos kerja	Jumlah mesin
1	3
2	2
3	4
4	3
5	1

Jika produk sampai pada salah satu pos kerja, maka produk akan dikerjakan pada salah satu mesin yang sedang idle. Distribusi waktu proses pada setiap mesin adalah Erlang-2 dengan waktu proses rata-rata bergantung pada tipe produk (Mserv) seperti berikut:

Tipe	Waktu proses
1	0.50, 0.60, 0.85, 0.50
2	1.10, 0.80, 0.75
3	1.20, 0.25, 0.70, 0.90, 1.00

- Ingin diketahui:
  - delay rata-rata pada setiap pos kerja
  - panjang antrian rata-rata di depan sebuah pos kerja
  - utilitas rata-rata setiap pos kerja

- Event:
  - arrival produk ke pos kerja → 'baru', mulai awal produk dikerjakan atau
    → setelah selesai dari sebuah task
  - departure dari sebuah pos kerja, yang akan membangkitkan arrival ke pos kerja berikutnya, jika task yang harus dilalui belum selesai.

- Awal simulasi:
  - semua mesin pada semua pos kerja berstatus idle
  - semua antrian kosong

- Akhir simulasi:
  Setelah satu siklus kerja selesai (1 minggu, 1 bulan, atau 1 tahun)

- Ingin diketahui:
  - di pos kerja mana terjadi 'bottleneck' atau
  - jika ada alokasi dana untuk membeli sebuah mesin,maka mesin yang mana yang perlu dibeli (dengan asumsi harga mesin-mesin tidak terlalu jauh berbeda).

- Program:

I:
- panjang waktu simulasi (= 365 hari @ 8 jam)
- rata-rata waktu antar kedatangan job (= 0.25 jam)
- rata-rata waktu pelayanan dari task j untuk job tipe i (dalam jam)
- jumlah pos kerja (= 5)
- jumlah mesin di pos kerja ke-i
- jumlah task untuk job tipe i
- jumlah tipe job (= 3)
- probabilitas dari job tipe i

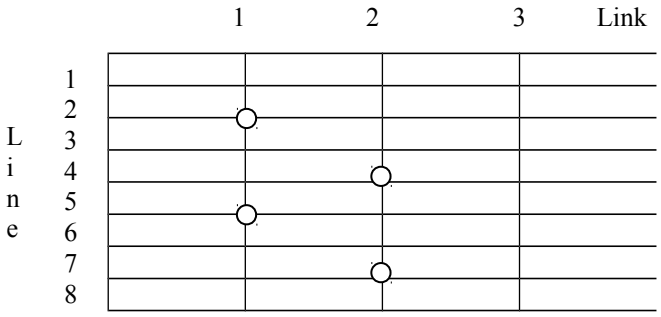
O:
- rata-rata total delay di antrian untuk job tipe i
- rata-rata delay di antrian pos kerja ke-i
- rata-rata utilisasi mesin di pos kerja ke-i
- rata-rata jumlah antrian di pos kerja ke-i
- keseluruhan rata-rata delay job di antrian.

Kasus antrian: Sistem Telepon

- Permasalahan:

Sebuah sistem telepon terdiri atas N buah Line yang dapat berhubungan satu sama lain melalui Link pada sentral telepon. Setiap Line mewakili sebuah pesawat. Hubungan antara sebuah pesawat dengan pesawat lain disambung melalui panel yang dilengkapi dengan Link. Pada umumnya, untuk N buah Line hanya ada sejumlah Link yang lebih kecil dari N. Setiap ada permintaan pembicaraan antara sebuah pesawat dengan pesawat lainnya akan diperiksa Link yang sedang tidak digunakan. Jika semua Link sedang digunakan maka permintaan percakapan tidak dapat dilayani.
Pola permintaan sambungan adalah pola kedatangan Poisson dengan waktu permintaan sambungan rata-rata MeanArr.
Lama pembicaraan terdistribusi eksonensial dengan harga rata-rata MCallLength.

Contoh: sistem dengan 3 Link dan 8 Line



- Line 2 sedang berkomunikasi dengan Line 5 menggunakan Link 1.
- Line 4 berkomunikasi dengan Line 7 menggunakan Link 2.
- Link 3 sedang tidak digunakan.

Asumsi:

- pada satu saat, sebuah pesawat (Line) hanya dapat menerima sebuah sambungan.
- sistem "hilang" yang artinya jika ada permintaan percakapan tetapi tidak dapat dipenuhi, maka permintaan tersebut diabaikan (tidak dimasukkan antrian). Kemungkinan "hilang" tersebut adalah karena:
  → Line yang akan dihubungi sedang bicara / "busy"
  → Link penuh / "blocked".

- Ingin diketahui:

Jumlah permintaan percakapan yang: sukses, busy, atau blocked,untuk menentukan jumlah Link yang optimal untuk sejumlah Line, karena harga Link yang mahal.

- Event:
  - arrival sebuah permintaan percakapan
  - selesainya percakapan

- Awal simulasi:
  - semua Line idle
  - semua Link idle

- Akhir simulasi:

setelah permintaan sambungan percakapan mencapai MaxCall.

- **Contoh kasus (snapshot)**

Time	Percakapan yang akan datang				Percakapan sedang berlangsung			Line								Link		Statistik			
	dari	ke	durasi	waktu Arr	dari	ke	waktu berakhir	1	2	3	4	5	6	7	8	maks	digunakan	diproses	sukses	busy	blocked
…	…	…	…	…	…	…	…														
1027	3	7	120	1057	4	7	1075	F	T	F	T	T	F	T	F	3	2	131	98	5	28
					2	5	1053														
1053	3	7	120	1057	4	7	1075	F	F	F	T	F	F	T	F	3	1	132	99	5	28
1057	3	6	98	1063	4	7	1075	F	F	F	T	F	F	T	F	3	1	133	99	6	28
1063	1	5	132	1082	3	6	1161	F	F	T	T	F	T	T	F	3	2	133	99	6	28
					4	7	1075														
…	…	…	…	…	…	…	…														

Sketsa kerangka program untuk Fixed Time Increment:

```
{Baca parameter}
{Inisialisasi state sistem}
{Tuliskan state awal sistem}
Time ← Tawal
While Time <= Takhir do
      {Next_State}
      {Tulis_State}
      Time ← Time + Increment Time
{endwhile}
{Tulis Report / laporan}
```

Sketsa kerangka program untuk Event Oriented Simulation:

```
{Baca parameter}
{Inisialisasi state sistem}
{Tuliskan state awal sistem}
while not EndSimulation do
      Timing(Event)
      Case
            EventType = 1:    ProcEvent-1
            EventType = 2:    ProcEvent-2
                  .
                  .
                  .
            EventType = n:    ProcEvent-n
      {endcase}
{endwhile}
{Tulis Report / laporan}
```

Catatan:
Prosedur `Timing(Output: Event)` akan mengambil sebuah event yang paling petensial untuk dilaksanakan, yaitu yang waktunya paling kecil dan mengajukan clocktime. Setiap event memiliki atribut EventTime, EventType, dll.