Modul Praktikum Proses Stokastik: Teori Antrean (Queueing Theory)

Tim Pengampu Mata Kuliah Pemodelan Stokastik

Pertemuan 8

Contents

1. Pendahuluan

1.1 Tujuan Pembelajaran

Modul praktikum ini dirancang untuk mahasiswa Program Sains Data agar mampu:

1. Menjelaskan konsep dasar teori antrean (M/M/1 dan M/M/c).
2. Menghitung kuantitas penting (ρ, L, L_q, W, W_q) secara analitik.
3. Mengimplementasikan simulasi event-driven untuk M/M/1 dan M/M/c di R.
4. Mengestimasi parameter λ dan μ dari data simulasi atau data sintetis.
5. Membandingkan hasil simulasi dengan hasil teori melalui tabel dan grafik.

2. Teori Antrean

2.1 Definisi

Sistem antrean dimodelkan oleh triplet $A/B/c$ dimana A = distribusi kedatangan, B = distribusi pelayanan, c = jumlah server. Pada modul ini kita fokus pada M/M/1 dan M/M/c:

- M: Markovian (kedatangan yang menyebar Poisson) dengan laju λ.
- M: Markovian (pelayanan yang menyebar eksponensial) dengan laju μ.
- c: jumlah server identik.

Asumsi umum: Antrian tidak dibatasi dan pelanggan menunggu sampai dilayani dan tidak ada yang mendahului antrean (*First In First Out*).

2.2 Antrean M/M/1

Proses ini merupakan suatu sistem antrean dengan satu server (seperti satu kasir di swalayan). Pada proses ini state i mewakili total pelanggan dalam sistem (menunggu + dilayani). Diasumsikan pelanggan tiba dengan laju konstan: $\lambda_i = \lambda$ untuk semua $i = 0, 1, 2, \ldots$ dan pelanggan menyelesaikan layanan dengan laju: $\mu_i = \mu$ untuk semua i.

Berikut ini adalah ilustrasi bagaimana proses antrean ini berjalan. Secara intuitif, proses antrean ini berjalan sebagai berikut: pelanggan datang satu per satu, kemudian jika server sedang sibuk, mereka akan menunggu dalam antrean. Jika server sedang kosong, pelanggan langsung dilayani. Ketika suatu layanan selesai, pelanggan keluar dari sistem dan server melayani pelanggan berikutnya (jika ada).

Figure 1: Antrean dengan 1 server

2.3 Antrean M/M/c

Proses ini merupakan suatu sistem antrean dengan c buah server (seperti 2 teller pelayanan di suatu bank). Sama seperti M/M/1, pada proses ini state i mewakili total pelanggan dalam sistem (menunggu + dilayani). Laju kedatangan tetap konstan: $\lambda_i = \lambda$ untuk semua i. Ketika $i < c$ (lebih banyak server daripada pelanggan) laju pelayanan diberikan oleh: $\mu_i = i\mu$. Sedangkan ketika $i \geq c$ (semua server sibuk) laju pelayanan diberikan oleh: $\mu_i = c\mu$.

Perhatikan bahwa terdapat perbedaan penting pada laju pelayanan. Karena terdapat lebih dari satu server, jumlah pelanggan yang sedang dilayani pada suatu waktu bergantung pada state i:

- Ketika jumlah pelanggan kurang dari jumlah server ($i < c$), berarti tidak semua server sedang sibuk. Dalam kondisi ini, hanya ada i pelanggan yang sedang dilayani, sehingga laju total pelayanan adalah
$$\mu_i = i\mu$$
Ini mencerminkan bahwa setiap pelanggan yang sedang dilayani berada di server yang berbeda dan masing-masing server memiliki laju eksponensial μ.

- Ketika jumlah pelanggan sama dengan atau melebihi jumlah server ($i \geq c$), seluruh server berada dalam keadaan sibuk. Oleh karena itu, laju total pelayanan mencapai kapasitas maksimal, yaitu
$$\mu_i = c\mu$$
Dalam keadaan ini, pelanggan yang baru datang tidak bisa langsung dilayani dan harus menunggu dalam antrean hingga salah satu dari server selesai melayani pelanggan sebelumnya.

Berikut ini adalah ilustrasi bagaimana proses antrean ini berjalan dimana terdapat dua teller yang melayani seluruh pelanggan.

Figure 2: Antrean dengan 2 server

3. Antrean M/M/1

3.1 Formula Penting: M/M/1

Untuk M/M/1 (c = 1), kita memiliki beberapa formula yang umum untuk dihitung:

- Intensitas lalu linta s pelangagan (rasio laju kedatangan dan pelayanan):

$$\rho = \lambda/\mu$$

.

- Probabilitas proses berada di keadaan n pelanggan dalam jangka waktu panjang:

$$P_n = (1-\rho)\rho^n, n \geq 0$$

.

- Ekspektasi jumlah pelanggan di sistem (menunggu dan dilayani) dalam jangka waktu panjang:

$$L = \frac{\rho}{1-\rho}$$

.

- Ekspektasi jumlah di antrean dalam jangka waktu panjang:

$$L_q = \frac{\rho^2}{1-\rho}$$

.

- Ekspektasi waktu yang dihabiskan seseoarang di sistem dalam jangka waktu panjang:

$$W = \frac{1}{\mu-\lambda}$$

.

- Ekspektasi waktu yang dihabiskan seseorang menunggu di antrean:

$$W_q = \frac{\rho}{\mu-\lambda}$$

.

Agar sistem M/M/1 ini stabil, diperlukan $\rho < 1$ atau $\lambda < \mu$, yaitu laju kedatangan lebih kecil dari laju pelayanan. Pada bagian ini kita akan melakukan komputasi dan simulasi untuk proses antrean M/M/1 ini.

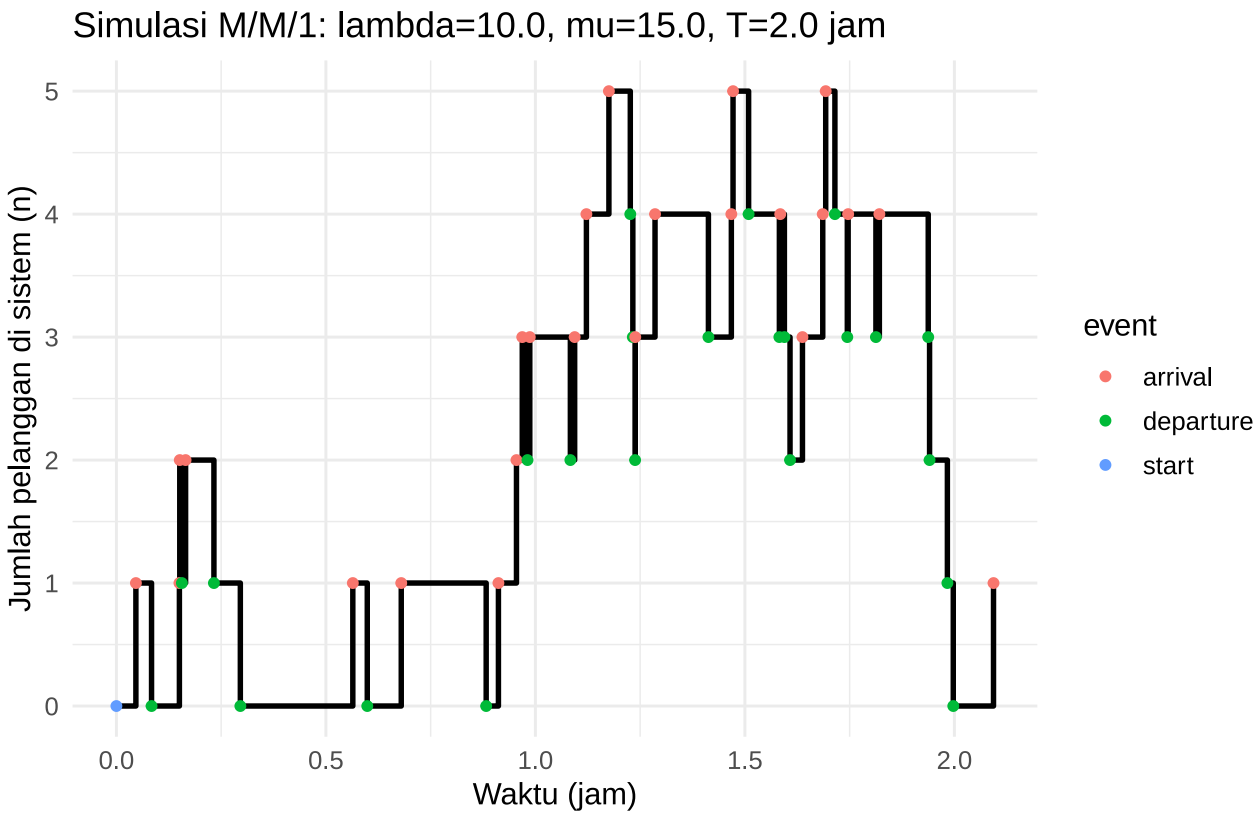
3.2 Simulasi M/M/1 dengan R

```
library(tidyverse)
library(gridExtra)
library(ggplot2)
set.seed(2025)
```

```r
simulate_mm1 <- function(lambda, mu, T = 10){
  t <- 0    # time variable
  n <- 0    # jumlah pelanggan
  # records is a dataframe containing when customer arrive or depart
  records <- tibble(time = 0, n = n, event = "start")
  next_arrival <- t + rexp(1, rate = lambda)  # inter-arrival time is exponential
  next_departure <- Inf
  while(t < T){
    if(next_arrival <= next_departure){
      t <- next_arrival
      n <- n + 1
      records <- add_row(records, time = t, n = n, event = "arrival")
      next_arrival <- t + rexp(1, rate = lambda)
      if(n == 1) next_departure <- t + rexp(1, rate = mu)
    } else {
      t <- next_departure
      n <- max(0, n - 1)
      records <- add_row(records, time = t, n = n, event = "departure")
      if(n > 0) next_departure <- t + rexp(1, rate = mu) else next_departure <- Inf
    }
    if(t > T) break
  }
  # ensure final time T recorded for plotting
  if(t < T) records <- add_row(records, time = T, n = records$n[nrow(records)],
                                event = "end")
  records
}

# contoh simulasi
lambda <- 10; mu <- 15; T <- 2 # waktu dalam jam
sim1 <- simulate_mm1(lambda, mu, T = T)
head(sim1)
```

```
## # A tibble: 6 x 3
##     time     n event
##    <dbl> <dbl> <chr>
## 1 0          0 start
## 2 0.0465     1 arrival
## 3 0.0839     0 departure
## 4 0.150      1 arrival
## 5 0.151      2 arrival
## 6 0.156      1 departure
```

```r
# buat step plot jumlah di sistem terhadap waktu
ggplot(sim1, aes(x = time, y = n)) +
  geom_step(linewidth = 1) +
  geom_point(aes(col = event), size = 1.5) +
  labs(title = sprintf("Simulasi M/M/1: lambda=%.1f, mu=%.1f, T=%.1f jam",
                       lambda, mu, T),
       x = "Waktu (jam)", y = "Jumlah pelanggan di sistem (n)") +
  theme_minimal(base_size = 12)
```



3.3 Estimasi parameter dari data

Berikut ini adalah code untuk mengestimasi nilai λ dan μ dari tabel data seperti di atas.

```r
estimate_mm1 <- function(records) {
  # --- Estimasi lambda ---
  arr_times <- records$time[records$event == "arrival"]
  lambda_hat <- 1 / mean(diff(arr_times))

  # --- Estimasi mu ---
  # service time dari periode ketika server tidak menganggur:
  # arrival yang terjadi saat n=0 langsung mulai servis
  rec <- records[order(records$time), ]
  start_service <- c()
```

```r
  end_service <- c()

  for (i in 2:nrow(rec)) {
    if (rec$event[i] == "arrival" && rec$n[i-1] == 0) {
      start_t <- rec$time[i]
      dep_idx <- which(rec$event == "departure" & rec$time > start_t)[1]
      if (!is.na(dep_idx)) {
        start_service <- c(start_service, start_t)
        end_service   <- c(end_service, rec$time[dep_idx])
      }
    }
  }

  mu_hat <- 1 / mean(end_service - start_service)

  list(lambda_hat = lambda_hat,
       mu_hat = mu_hat)
}


# cara pakai
lambda_true <- 10
mu_true <- 15
sim <- simulate_mm1(lambda_true, mu_true, T = 10)
param_hat <- estimate_mm1(sim)

cat("Nilai sebenarnya lambda:", lambda_true, "\n")
```

```
## Nilai sebenarnya lambda: 10
```

```r
cat("Estimasi lambda:", param_hat$lambda_hat, "\n")
```

```
## Estimasi lambda: 9.272376
```

```r
cat("Nilai sebenarnya mu:", mu_true, "\n")
```

```
## Nilai sebenarnya mu: 15
```

```r
cat("Estimasi lambda:", param_hat$mu_hat, "\n")
```

```
## Estimasi lambda: 14.18904
```

Dapat kita lihat bahwa hasil estimasi parameter lambda dan mu yang kita peroleh memiliki nilai yang cukup dekat dengan nilai lambda dan mu sesungguhnya.

3.4 Perbandingan nilai teori dan simulasi

Kasus (Bank): Nasabah tiba $\lambda = 10$ per jam, teller $\mu = 15$ per jam. Hitung

1. Ekspektasi jumlah nasabah pada kondisi steady state.
2. Peluang seorang nasabah yang baru mulai dilayani masih dalam pelayanan setelah 5 menit.
3. Ekspektasi jumlah nasabah (L) pada steady state:

```
lambda <- 10; mu <- 15
rho <- lambda / mu
L_theory <- rho / (1 - rho)


# hitung L dari data simulasi
compute_L_from_records <- function(records, T) {
  # Ensure ordered by time
  records <- dplyr::arrange(records, time)
  # Compute the duration until next event
  t_next <- dplyr::lead(records$time)
  dt <- t_next - records$time
  # Remove the last NA interval
  valid <- !is.na(dt)
  # Time-average number of customers
  L_sim <- sum(records$n[valid] * dt[valid]) / T
  return(L_sim)
}

rec <- simulate_mm1(lambda = 10, mu = 15, T = 1000)
L_sim <- compute_L_from_records(rec, T=1000)

L_theory
```

```
## [1] 2
```

```
L_sim
```

```
## [1] 2.285548
```

2. Peluang seorang nasabah yang baru mulai dilayani masih dalam pelayanan setelah 5 menit (5/60 jam): Perhatikan bahwa waktu pelayanan menyebar eksponensial dengan parameter $\mu = 15$ menit.

```
mu <- 15
tmin <- 5/60
prob_theory <- exp(-mu * tmin)
prob_theory
```

```
## [1] 0.2865048
```

4. Antrean M/M/c

4.1 Formula Penting: M/M/c

Untuk sistem antrean M/M/c, kita peroleh beberapa formula penting:

- Faktor Erlang C (probabilitas menunggu) dengan $\rho = \lambda/(c\mu)$ dan $a = \lambda/\mu$ adalah:

$$C = \frac{\frac{a^c}{c!}\frac{1}{1-\rho}}{\sum_{k=0}^{c-1}\frac{a^k}{k!} + \frac{a^c}{c!}\frac{1}{1-\rho}}.$$

- Ekspektasi jumlah pelanggan di antrean dalam jangka waktu panjang:

$$L_q = C\frac{\rho}{1-\rho}$$

- Ekspektasi jumlah di antrean dalam jangka waktu panjang:

$$L = L_q + \frac{\lambda}{\mu}$$

- Ekspektasi waktu yang dihabiskan seseorang menunggu di antrean:

$$W_q = \frac{L_q}{\lambda}.$$

- Ekspektasi waktu yang dihabiskan seseoarang di sistem dalam jangka waktu panjang:

$$W = W_q + \frac{1}{\mu}.$$

Agar sistem M/M/1 ini stabil, diperlukan $\lambda < c\mu$, yaitu laju kedatangan lebih kecil dari total laju pelayanan dari c server. Pada bagian ini kita akan melakukan komputasi dan simulasi untuk proses antrean M/M/c ini.

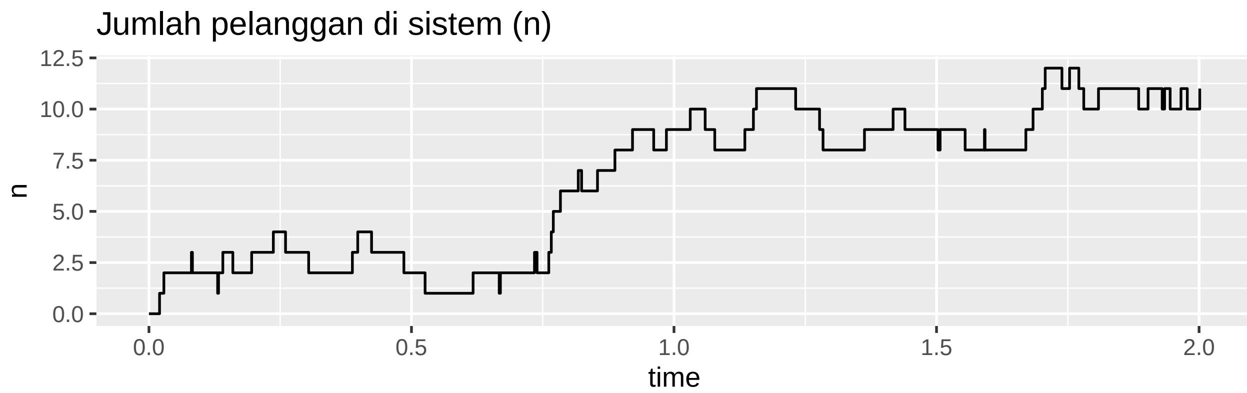
4.2 Simulasi M/M/c dengan R

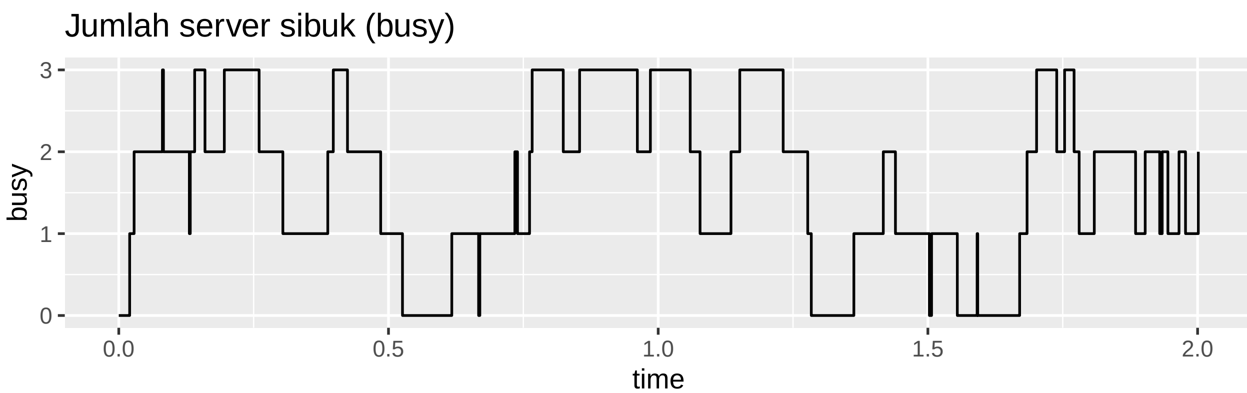
```
simulate_mmc <- function(lambda, mu, c = 2, T = 10){
  t <- 0; n <- 0; busy <- 0
  departures <- numeric(0)
  records <- tibble(time = 0, n = n, busy = busy, event = "start")
  next_arrival <- rexp(1, rate = lambda)
  while(t < T){
    next_departure <- if(length(departures)==0) Inf else min(departures)
    if(next_arrival <= next_departure){
      t <- next_arrival
      n <- n + 1
```

```r
      if(busy < c){
        busy <- busy + 1
        departures <- c(departures, t + rexp(1, rate = mu))
      }
      records <- add_row(records, time = t, n = n, busy = busy,
                         event = "arrival")
      next_arrival <- t + rexp(1, rate = lambda)
    } else {
      t <- next_departure
      # remove one instance of that departure time
      departures <- departures[departures != next_departure]
      n <- max(0, n - 1)
      busy <- max(0, busy - 1)
      records <- add_row(records, time = t, n = n, busy = busy,
                         event = "departure")
    }
    if(t > T) break
  }
  if(t < T) records <- add_row(records, time = T, n = records$n[nrow(records)],
                               busy = records$busy[nrow(records)], event = "end")
  records
}

# contoh: call center dari soal
lambda <- 20; mu <- 8; c <- 3; T <- 2
sim3 <- simulate_mmc(lambda, mu, c, T)
head(sim3)
```

```
## # A tibble: 6 x 4
##     time     n  busy event
##    <dbl> <dbl> <dbl> <chr>
## 1 0          0     0 start
## 2 0.0202     1     1 arrival
## 3 0.0284     2     2 arrival
## 4 0.0809     3     3 arrival
## 5 0.0828     2     2 departure
## 6 0.131      1     1 departure
```

```r
# plot jumlah di sistem dan busy servers
p1 <- ggplot(sim3, aes(x = time, y = n)) + geom_step() +
  labs(title = "Jumlah pelanggan di sistem (n)")
p2 <- ggplot(sim3, aes(x = time, y = busy)) + geom_step() +
  labs(title = "Jumlah server sibuk (busy)")

grid.arrange(p1, p2, nrow = 2)
```





4.3 Estimasi parameter dari data

Berikut ada kode yang dapat kita gunakan untuk estimasi parameter proses M/M/c.

```
library(dplyr)

estimate_rates <- function(records, T) {
  rec <- arrange(records, time)
  # Arrival rate
  arrivals <- sum(rec$event == "arrival")
  lambda_hat <- arrivals / T

  # Compute total busy time
  dt <- diff(rec$time)
  busy_time <- sum(dt * rec$busy[-nrow(rec)])

  # Service rate
  departures <- sum(rec$event == "departure")
  mu_hat <- departures / busy_time

  tibble(lambda_hat = lambda_hat, mu_hat = mu_hat)
}
```

```r
# Example
param_mmc <- estimate_rates(sim3, T = 2)
param_mmc
```

```
## # A tibble: 1 x 2
##   lambda_hat mu_hat
##        <dbl>  <dbl>
## 1       19.5   8.60
```

5. Latihan Praktikum

Berikut soal praktikum beserta petunjuk singkat implementasi di R.

Soal 1

Simulasikan M/M/1 $\lambda = 8, \mu = 12$ selama T=100 jam. Hitung $\hat{L}$ sebagai time-average. Bandingkan dengan nilai teoretis.

Soal 2

Simulasikan suatu sistem antrean M/M/3 dengan laju kedatangan $\lambda = 20$ pelanggan per jam dan laju pelayanan $\mu = 10$ pelanggan per jam per server. Estimasi parameter dari data simulasi yang diperoleh dan bandingkan dengan parameter awal.

Soal 3

Bandingkan distribusi waktu antar-kedatangan (inter-arrival time) dari simulasi M/M/1 pada soal nomor 1 dengan distribusi teori. Lakukan uji visual (histogram atau QQ plot atau keduanya).

Petunjuk: Gunakan fungsi *hist()* dan *qqplot()*

6. Kesimpulan

Modul ini menggabungkan teori antrean (M/M/1 dan M/M/c) dengan praktik simulasi event-driven di R. Grafik dan hasil simulasi membantu melihat perilaku stokastik yang tidak selalu langsung terlihat dari rumus analitik.