

**DISCRETE EVENT SIMULATION:
ANTRIAN DI *FOODCOURT***



PAPER SIMULASI

Diajukan untuk Memenuhi Tugas Mata Kuliah Simulasi

Disusun oleh :

Kanaya Tabhita Djie

NIM: 01112180013

PROGRAM STUDI MATEMATIKA TERAPAN

UNIVERSITAS PELITA HARAPAN

TANGERANG

2020

BAB I

Pendahuluan

1.1. Latar Belakang

Satu hal yang perlu kita sadari adalah mengantre merupakan salah satu kegiatan yang sering kita lakukan. Misalnya, seperti mengantre untuk menggunakan mesin ATM, antrean di puskesmas, dan sebagainya. Ini terjadi karena dengan jumlah fasilitas yang terbatas, tidak mungkin dapat melayani seluruh orang secara sekaligus. Antrean yang terlalu lama jelas akan menghambat aktivitas masyarakat.

Topik yang akan dibahas kali ini adalah antrean di *foodcourt* pada saat jam makan siang (12.00-14.00). *Foodcourt* adalah sebuah tempat makan yang terdiri dari banyak restoran. Sering sekali kita temukan berbagai antrean di *foodcourt*, mulai dari ketika ingin memesan makanan, menunggu meja kosong, sampai mengantre untuk membayar di kasir.

Mahasiswa yang memiliki jadwal padat dengan waktu istirahat yang singkat, tentu kegiatan di kampusnya bisa terhambat apabila harus mengantre dengan sangat lama, atau bahkan terpaksa menunda makan siang supaya tidak terlambat, sehingga diperlukan sebuah solusi efisien untuk mengatasi masalah ini.

Untuk mempersingkat waktu, akan dibuat sebuah model antrean berdasarkan situasi yang dapat disimulasikan sebanyak yang kita perlukan. Model ini akan dibuat sesuai dengan kenyataan, seperti jumlah *resource* yang tersedia dan jumlah *entity* yang datang. Setelah dilakukan simulasi, kita akan mengetahui di bagian apa yang perlu kita tingkatkan jumlahnya agar antrean menjadi tidak terlalu panjang dan akan diterapkan di dunia nyata.

Oleh karena itu, simulasi ini bisa membantu kita untuk memecahkan masalah antrean di *foodcourt* dan akan mempermudah kita.

1.2. Tujuan

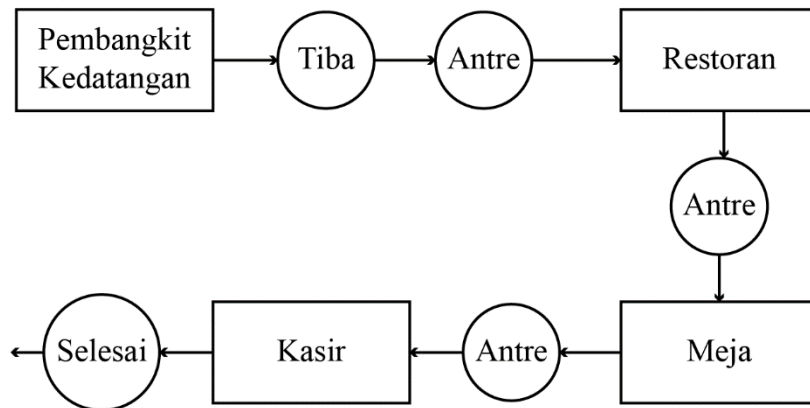
Berdasarkan latar belakang, tujuan yang akan dicapai dalam simulasi ini adalah sebagai berikut:

1. Mensimulasikan model antrean di *foodcourt*.
2. Memberikan solusi model antrean yang efisien.

BAB II

Deskripsi Sistem

2.1. Diagram



2.2. Penjelasan Diagram

Sebuah sistem antrean di *foodcourt* dimodelkan seperti gambar diagram di atas. *Foodcourt* tersebut memiliki tiga buah *resource*, yaitu restoran, meja, dan kasir (ditandai dengan bentuk persegi). Berikut adalah penjelasan setiap objek diagram:

1. Pembangkit Kedatangan

Waktu antara kedatangan pelanggan (*source*) tersebar secara eksponensial dengan *rate* tiga pelanggan per menit ($\lambda = 3$).

2. Restoran

Pelanggan akan memilih salah satu restoran dan memesan makanan yang mereka inginkan. Waktu yang diperlukan untuk proses pemesanan menyebar secara eksponensial dengan *rate* satu pelanggan per dua menit ($\lambda = 0.2$). Jumlah restoran yang tersedia diasumsikan sebanyak lima.

3. Meja

Setelah memesan makanan, pelanggan akan makan di meja yang telah disediakan. Lama setiap meja makan yang digunakan menyebar eksponensial dengan *rate* satu pelanggan per dua puluh menit ($\lambda = 0.05$). Jumlah restoran yang tersedia diasumsikan sebanyak lima belas.

4. Kasir

Setelah makan, pelanggan akan mengantre di kasir. Lama proses pembayaran menyebar secara eksponensial dengan *rate* satu pelanggan per satu menit ($\lambda = 1$). Jumlah kasir yang tersedia diasumsikan sebanyak tiga.

Asumsi-asumsi lainnya yaitu jam makan siang adalah selama seratus dua puluh menit (12.00-14.00) dan pelanggan datang secara individual (bukan *batch*).

BAB III

Script R

```
rm(list=ls())

library(simmer)
library(simmer.plot)
library(ggplot2)
set.seed(123)
env <- simmer("Foodcourt")
env

order <- function() round(rexp(1, 0.5),3) #melayani 1 pelanggan
dalam 2 menit
eat <- function() round(rexp(1, 0.05),3) #melayani 1 pelanggan dalam
20 menit
cashier <- function() round(rexp(1, 1),3) #melayani 1 pelanggan
dalam 1 menit
arrival <- function() round(rexp(1,3),3) #datang 3 pelanggan dalam 1
menit

pelanggan <- trajectory("antrian pelanggan") %>%
  # log_("datang...") %>%
  seize("restoran") %>%
  timeout(order) %>%
  release("restoran") %>%

  #log_("makan...") %>%
  seize("meja") %>%
  timeout(eat) %>%
  release("meja") %>%

  #log_("ke kasir...") %>%
  seize("kasir") %>%
  timeout(cashier) %>%
  release("kasir") %>%

#log_("pulang...")

env2 <- lapply(1:100, function(i){
  simmer("Foodcourt") %>%
    add_resource("restoran", 7) %>% #ada 7 restoran
    add_resource("meja", 15) %>% #ada 15 meja
    add_resource("kasir", 2) %>% #ada 2 kasir
    add_generator("pelanggan", pelanggan, arrival) %>%
    run(120) %>% invisible #jam makan siang selama 2 jam
})

#melihat proses kedatangan pelanggan
output1 <- get_mon_arrivals(env2)
sub1 <- subset(output1, replication==1)
sub2 <- subset(output1, replication==2)

#melihat proses pemakaian server
```

```

output2 <- get_mon_resources(env2)
sub3 <- subset(output2, replication==1)
sub4 <- subset(output2, replication==2)

#output1
k <- NULL; rata <- NULL
for(j in 1:100){
  dt <- subset(output1, replication == j)
  k <- c(k, nrow(dt))
  lsist <- dt$end_time - dt$start_time
  rata <- c(rata, mean(lsist))
}
k %>% head()
mean(k) #rata2 jmlh pelanggan yang selesai dilayani dlm 120 menit
hist(k)

rata %>% head()
mean(rata) #rata2 lama org di dalam sistem
hist(rata)

#output2
rantri <- NULL
for(j in 1:100){
  dt <- subset(output2, replication == j)
  rantri <- c(rantri, mean(dt$queue))
}
rantri %>% head()
mean(rantri) #rata2 jml antrian
hist(rantri)

#plot
plot(output2, metric = "utilization")
plot(output2, metric = "usage", c("restoran", "meja", "kasir"),
items = "server")
plot(output2, metric = "usage", c("restoran", "meja", "kasir"),
items = "queue")
plot(output1, metric = "activity_time")
plot(output1, metric = "waiting_time")

#selang kepercayaan
SE1 <- sd(k)
xbar1 <- mean(k)
Z.97.5 <- qnorm(0.975)
(SK1 <- xbar1+c(-1,1)*Z.97.5 * SE1)

SE2 <- sd(rata)
xbar2 <- mean(rata)
Z.97.5 <- qnorm(0.975)
(SK2 <- xbar2+c(-1,1)*Z.97.5 * SE2)

SE3 <- sd(rantri)
xbar3 <- mean(rantri)
Z.97.5 <- qnorm(0.975)
(SK3 <- xbar3+c(-1,1)*Z.97.5 * SE3)

```

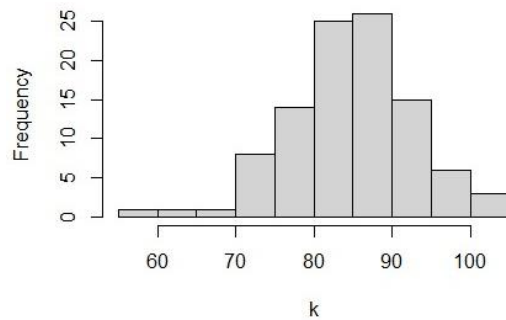
```
#ringkasan statistik
hasil <- env2 %>%
  get_mon_arrivals() %>%
  transform(waiting_time = end_time - start_time -
activity_time)
paste("Rata-rata total", sum(hasil$finished), "orang pelanggan
dalam menyelesaikan aktivitas di Foodcourt adalah",
  round(mean(hasil$waiting_time),3), "menit")
```

BAB IV

Hasil Simulasi

4.1. Rata-rata Jumlah Pelanggan yang Selesai Dilayani dalam 120 Menit

Histogram of k



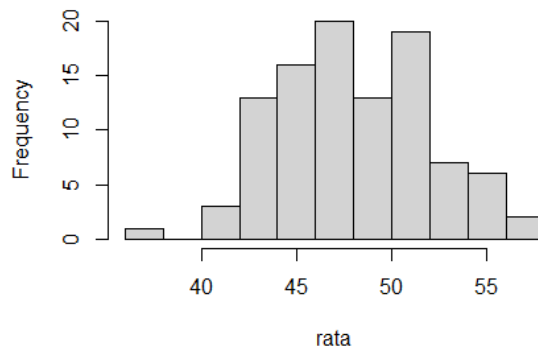
```
> mean(k)
[1] 85.49

> SE1
[1] 8.027094

> SK1
2.5%      97.5%
69.75718 101.22282
```

4.2. Rata-rata Lama Pelanggan dalam Antrean

Histogram of rata

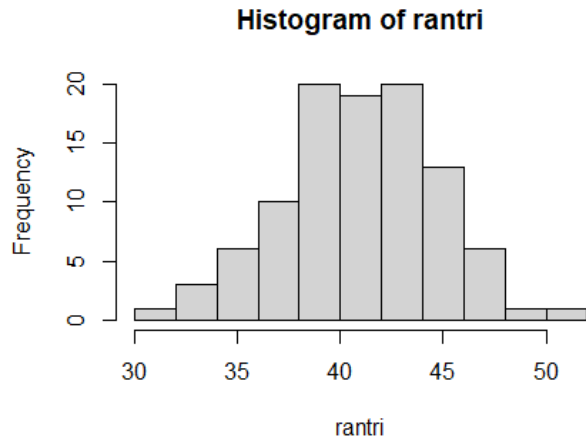


```
> mean(rata)
[1] 48.01138

> SE2
[1] 3.929742

> SK2
2.5%      97.5%
40.30923 55.71353
```

4.3. Rata-rata Jumlah Antrean

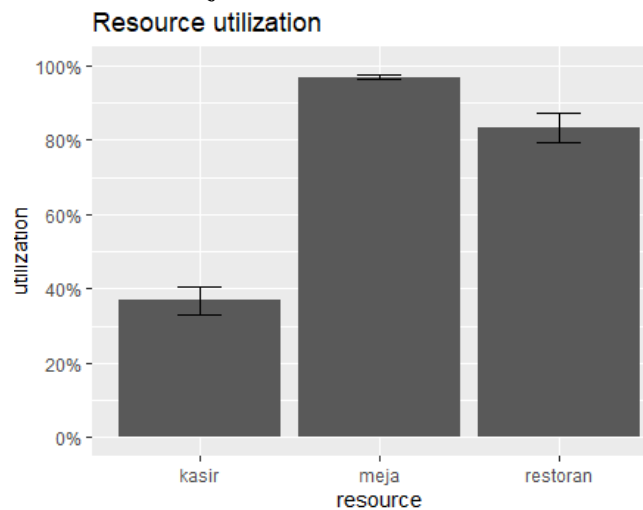


```
> mean(rantri) #rata2 jml antrian  
[1] 40.94903
```

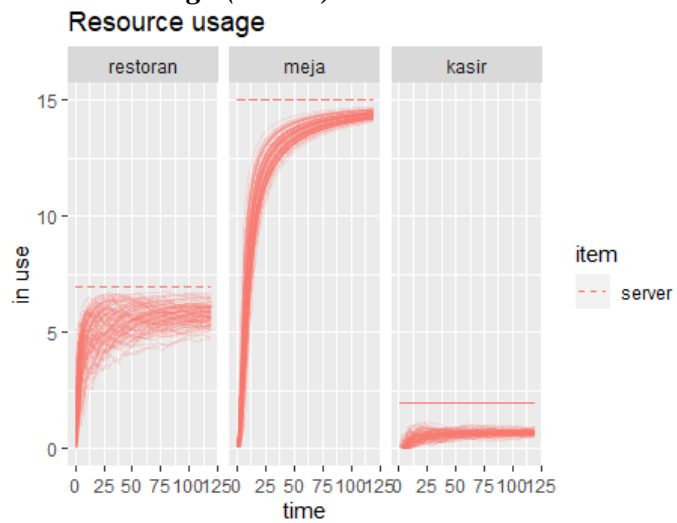
```
> SE3  
[1] 3.796934
```

```
> SK3  
2.5%      97.5%  
33.50717  48.39088
```

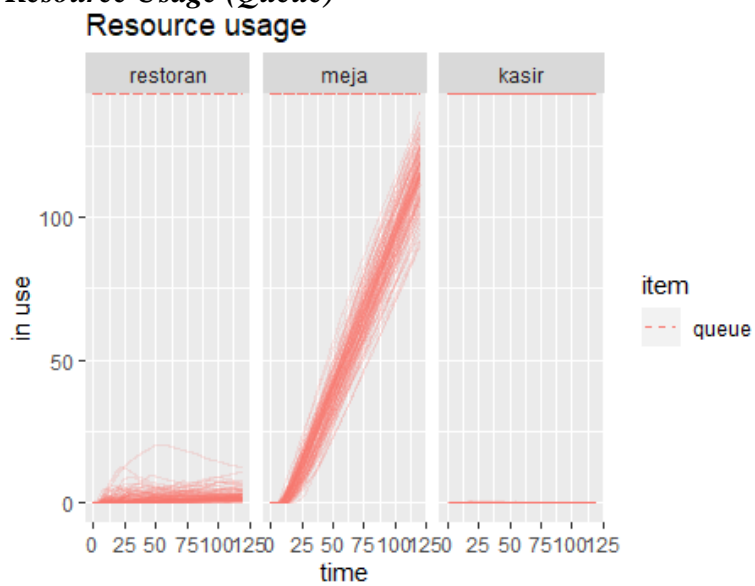
4.4. Resource Utilization



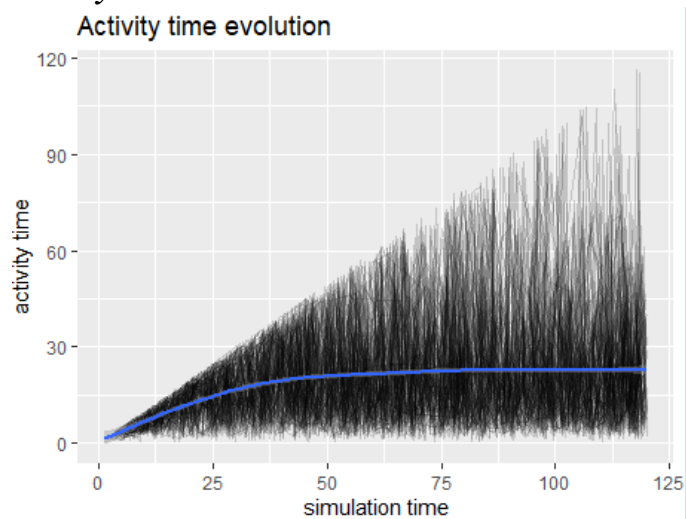
4.5. *Resource Usage (Server)*



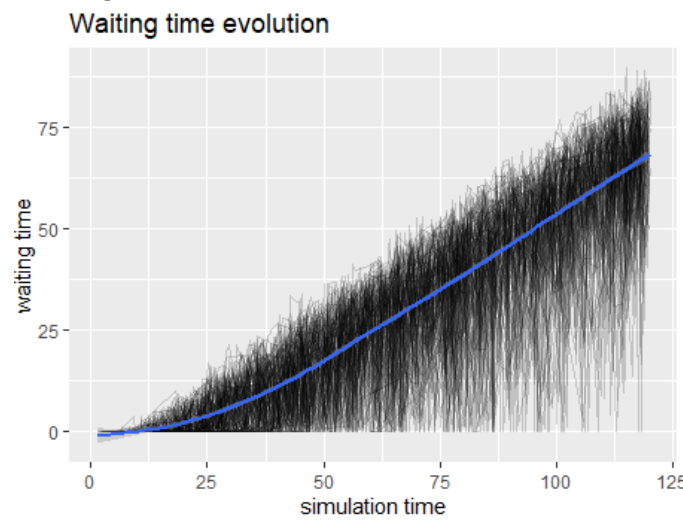
4.6. *Resource Usage (Queue)*



4.7. *Activity Time Evolution*



4.8. *Waiting Time Evolution*



4.9. **Total Rata-rata Pelanggan dalam menyelesaikan aktivitasnya**

[1] "Rata-rata 8549 orang pelanggan dalam menyelesaikan aktivitas di Foodcourt adalah 28.647 menit"

BAB V

Pembahasan

Dari histogram 4.1, bisa kita lihat bahwa 25 dari 100 percobaan, *foodcourt* selesai melayani 80-90 orang pelanggan dalam 120 menit, dengan rata-rata 85.49 pelanggan. Maka dengan selang kepercayaan 95%, jumlah pelanggan yang selesai dilayani dalam 120 menit adalah 69.76 hingga 101.23 menit.

Sedangkan berdasarkan histogram 4.2, sebanyak 20 kelompok percobaan simulasi menunjukkan lama setiap pelanggan melakukan aktivitasnya adalah 47-48 menit, dengan rata-rata setiap orang selama 48.01 menit. Maka dengan selang kepercayaan 95%, jumlah pelanggan dalam antrean sistem adalah 40.31-55.71 menit.

Dapat kita lihat juga dari histogram 4.3, 20 kelompok percobaan menunjukkan pelanggan yang mengantri dalam waktu 120 menit berjumlah 39-40 orang dan 20 kelompok percobaan menunjukkan jumlah orang yang mengantre sebesar 43-44 orang. Jumlah rata-rata pelanggan yang mengantri adalah 40.95 orang. Maka dengan selang kepercayaan 95%, jumlah pelanggan yang mengantri dalam 120 menit adalah 33.51-48.39 orang.

Berdasarkan pembahasan di atas, jelas bahwa sistem kurang efisien karena dari rata-rata 85.49 pelanggan yang dapat dilayani, hampir setengahnya mengantri yaitu sebesar 40.95 orang, sehingga setiap pelanggan di dalam sistem pun cukup lama yaitu 48.01 menit. Oleh karena itu, kita perlu meningkatkan jumlah *server*.

Dari grafik utilisasi server 4.4, terlihat bahwa utilisasi meja paling banyak yaitu hampir 100%. Restoran yang tersedia memiliki tingkat utilisasi tertinggi kedua yaitu hampir 90% dan kasir memiliki tingkat utilisasi terkecil yaitu hampir 40%. Ini menunjukkan setiap meja dan restoran yang tersedia benar-benar digunakan oleh setiap pelanggan, sedangkan utilisasi dua kasir tidak maksimal.

Dari grafik *resource usage* 4.5, terlihat semakin di akhir masa waktu makan siang, penggunaan meja semakin banyak di angka mendekati 15, sedangkan tingkat pelanggan di restoran juga meningkat namun cukup beragam di antara 4-7 restoran. Jumlah kasir yang melayani pelanggan tidak maksimal yaitu di sekitar angka satu.

Dari grafik *resource usage* 4.6, terlihat jumlah antrean tertinggi ada di meja, sedangkan tingkat antrean tertinggi kedua ada di restoran dan terakhir ada di kasir. Kita bisa lihat juga dari grafik *activity time evolution* 4.7, lama aktivitas naik secara landai dan terlihat lama menunggu setiap pelanggan naik secara drastis seiring jam makan siang berjalan dari grafik *waiting time evolution* 4.8.

BAB VI

Penutup

6.1. Kesimpulan

Simulasi antrean di *foodcourt* belum efisien karena masih banyak pelanggan yang mengantre dalam sistem, sehingga jumlah server yang tersedia harus ditingkatkan lagi agar lamanya antrean bisa berkurang dan *foodcourt* lebih bisa melayani pelanggan lebih banyak lagi saat jam makan siang. Berdasarkan pembahasan, bisa disimpulkan bahwa jumlah meja harus ditingkatkan lebih banyak lagi karena antrean terlalu lama pelanggan adalah saat menunggu meja. Apabila jumlah restoran ditingkatkan juga akan menurunkan lama antrean namun jumlah yang ditambah tidak perlu terlalu banyak. Dua kasir yang disediakan tidak memiliki masalah dalam kasus antrean ini, namun karena utilitasnya tidak maksimal, jumlahnya dapat juga dikurangi menjadi satu untuk menghemat biaya.