Tubes Pemstok i

andini

2025-11-25

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.5.2
```

```
library(queueing)
```

```
## Warning: package 'queueing' was built under R version 4.5.2
```

```
library(fitdistrplus)
```

```
## Warning: package 'fitdistrplus' was built under R version 4.5.2
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.5.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(hms)
```

```
## Warning: package 'hms' was built under R version 4.5.2
```

```
df <-  read_excel("C:/Users/sdthi/Downloads/Galeri 2 (1).xlsx",
    sheet = "Kasir1")
df2<-  read_excel("C:/Users/sdthi/Downloads/Galeri 2 (1).xlsx",
    sheet = "Kasir2")
```

```
df2
```

```
## # A tibble: 60 × 3
##    Datang              Dilayani            Pergi
##    <dttm>              <dttm>              <dttm>
##  1 1899-12-31 09:36:49 1899-12-31 09:37:34 1899-12-31 09:37:47
##  2 1899-12-31 09:36:50 1899-12-31 09:37:49 1899-12-31 09:37:55
##  3 1899-12-31 09:36:51 1899-12-31 09:37:57 1899-12-31 09:38:07
##  4 1899-12-31 09:36:52 1899-12-31 09:38:10 1899-12-31 09:38:24
##  5 1899-12-31 09:38:03 1899-12-31 09:38:43 1899-12-31 09:39:00
##  6 1899-12-31 09:38:02 1899-12-31 09:39:13 1899-12-31 09:39:19
##  7 1899-12-31 09:38:03 1899-12-31 09:39:40 1899-12-31 09:39:46
##  8 1899-12-31 09:38:04 1899-12-31 09:39:58 1899-12-31 09:40:17
##  9 1899-12-31 09:38:05 1899-12-31 09:40:18 1899-12-31 09:40:37
## 10 1899-12-31 09:38:23 1899-12-31 09:40:34 1899-12-31 09:40:56
## # ℹ 50 more rows
```

```
# Kasir 1
df$Datang   <- as_hms(as.POSIXct(df$Datang, format="%H:%M:%S"))
df$Dilayani <- as_hms(as.POSIXct(df$Dilayani, format="%H:%M:%S"))
df$Pergi    <- as_hms(as.POSIXct(df$Pergi, format="%H:%M:%S"))

# Kasir 2
df2$Datang   <- as_hms(as.POSIXct(df2$Datang, format="%H:%M:%S"))
df2$Dilayani <- as_hms(as.POSIXct(df2$Dilayani, format="%H:%M:%S"))
df2$Pergi    <- as_hms(as.POSIXct(df2$Pergi, format="%H:%M:%S"))
```

```
# Waktu pelayanan adalah gabungan kemampuan Kasir 1 dan Kasir 2
service_times1 <- as.numeric(df$Pergi - df$Dilayani)/60 # dalam menit
service_times2 <- as.numeric(df2$Pergi - df2$Dilayani)/60 # dalam menit
all_service_times <- c(service_times1, service_times2)
```

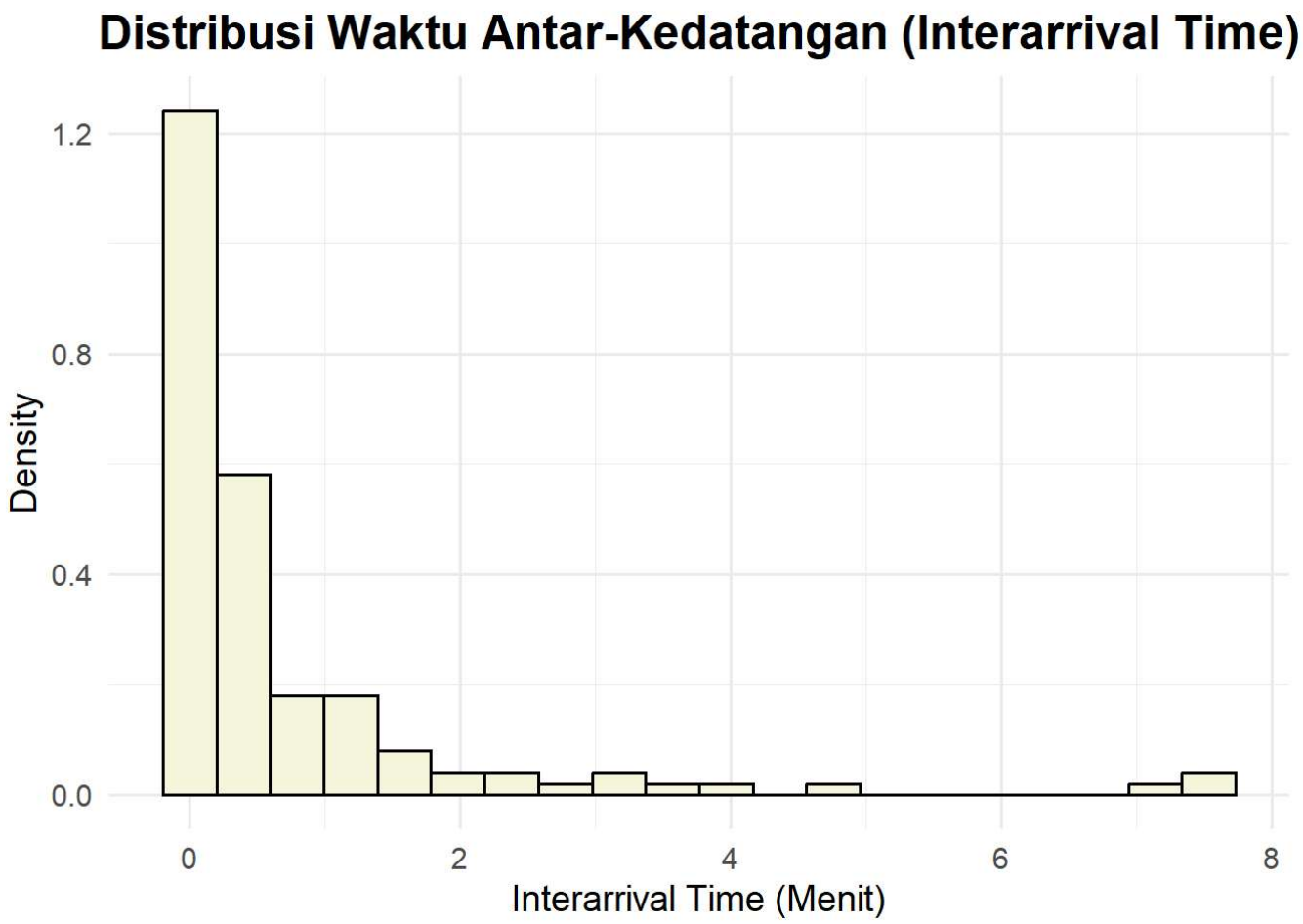
```
# Gabungkan data untuk menghitung Inter-arrival Time (Antar Kedatangan)
# Kita asumsikan antrian adalah satu kesatuan, jadi kita urutkan semua kedatangan
all_arrivals <- c(df$Datang, df2$Datang)
all_arrivals <- sort(all_arrivals) # Urutkan dari pagi ke sore
inter_arrivals <- diff(all_arrivals)/60 # Hitung selisih
head(inter_arrivals)
```

```
## Time differences in secs
## [1] 0.01666667 0.01666667 0.01666667 0.16666667 1.00000000 0.01666667
```

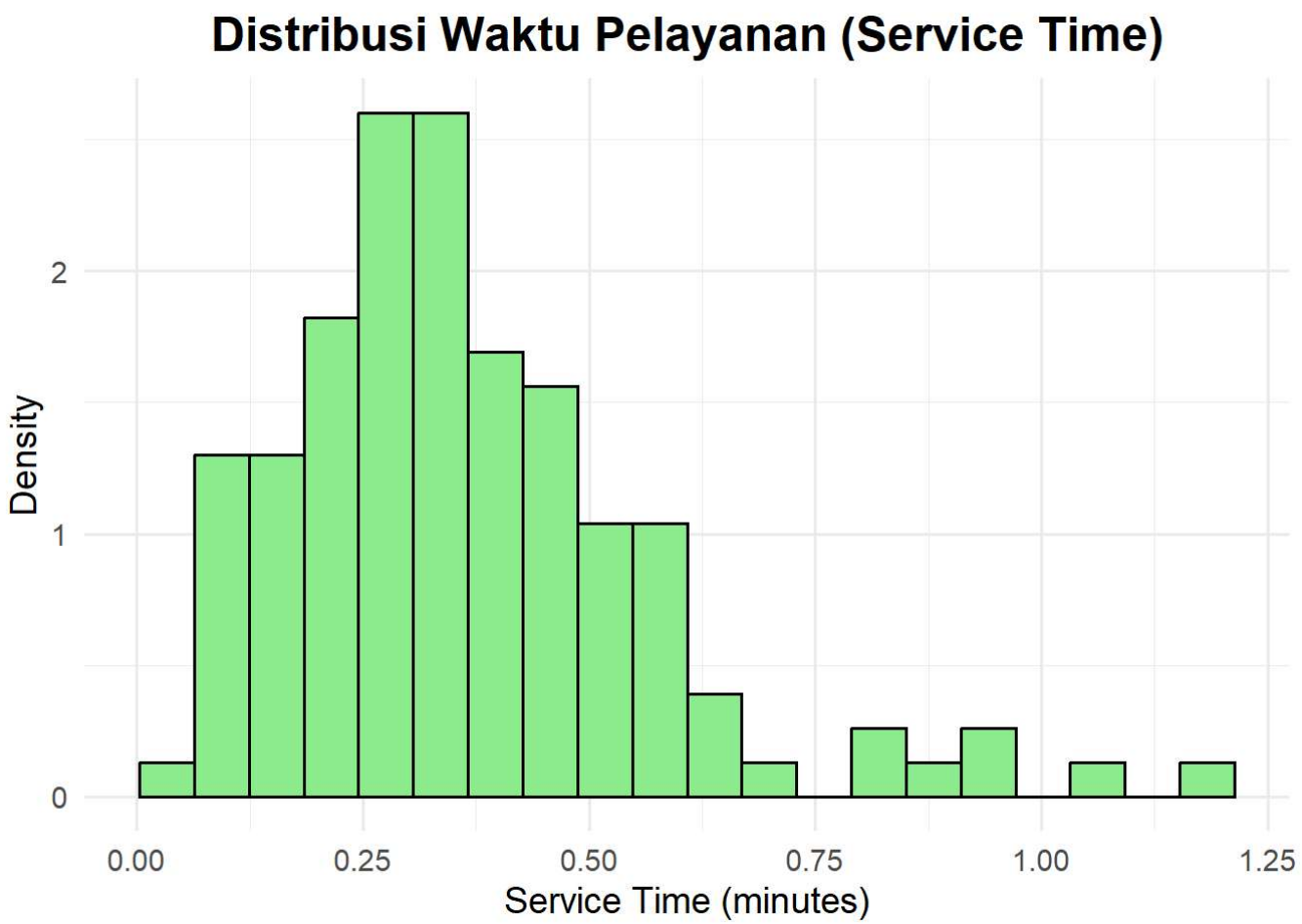
```
inter_arrivals <- as.numeric(inter_arrivals) # dalam detik
```

```
ggplot(data.frame(inter_arrivals), aes(x = inter_arrivals)) +
  # Menggunakan y = ..density.. dan bins = 20 sesuai contoh target
  geom_histogram(aes(y = ..density..), bins = 20, fill = "beige", color = "black") +

  # Menyesuaikan Label
  labs(
    title = "Distribusi Waktu Antar-Kedatangan (Interarrival Time)",
    x = "Interarrival Time (Menit)",
    y = "Density"
  ) +

  # Menyesuaikan Tema (Font besar dan Judul Tengah)
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold")
  )
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

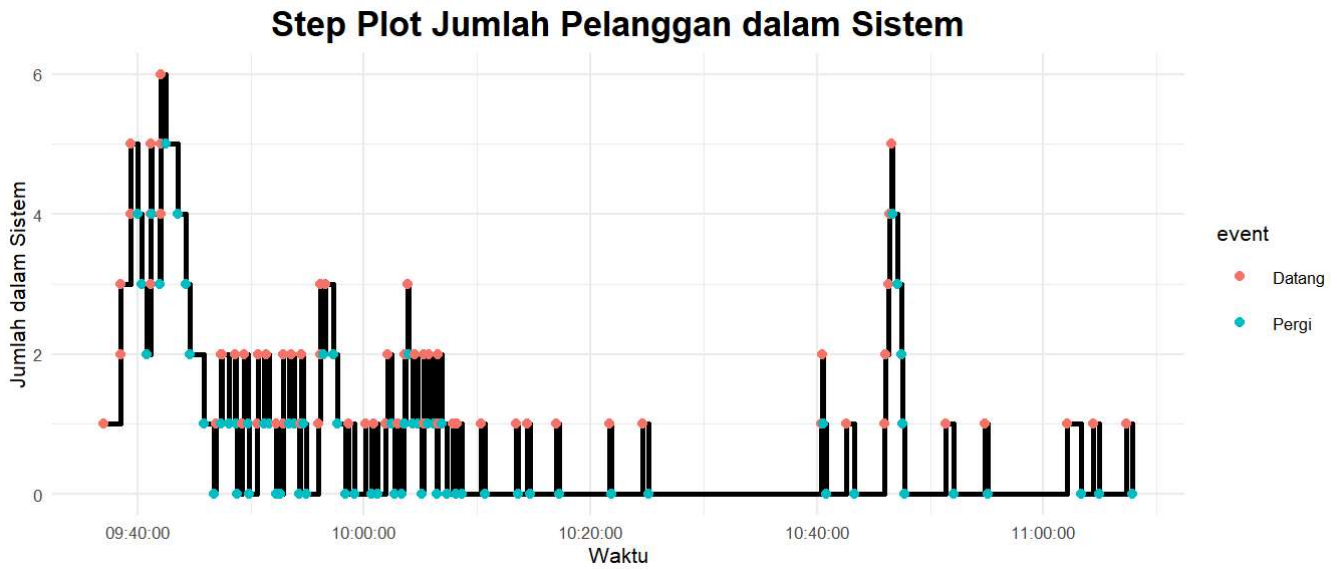



```
ggplot(data.frame(all_service_times ), aes(x = all_service_times )) +
  geom_histogram(aes(y = ..density..), bins = 20, fill = "lightgreen", color = "black") +
  labs(title = "Distribusi Waktu Pelayanan (Service Time)",
       x = "Service Time (minutes)", y = "Density")+
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold")
  )
```




```
# Bersihkan data
all_service_times <- all_service_times[all_service_times > 0 & !is.na(all_service_times)]
inter_arrivals <- inter_arrivals[inter_arrivals > 0 & !is.na(inter_arrivals)]
```

```
events <- data.frame(
  time = c(df$Datang, df$Pergi),
  change = c(rep(1, nrow(df)), rep(-1, nrow(df)))
)

# Urutkan berdasarkan waktu
events <- events %>% arrange(time)

# Tambahkan label event
events$event <- ifelse(events$change == 1, "Datang", "Pergi")

# Hitung jumlah dalam sistem N(t)
events$N <- cumsum(events$change)
ggplot(events, aes(x = time, y = N)) +
  geom_step(linewidth = 1, color = "black") +
  geom_point(aes(col = event), size = 1.5) +
  labs(
    title = "Step Plot Jumlah Pelanggan dalam Sistem",
    x = "Waktu",
    y = "Jumlah dalam Sistem"
  ) +
  theme_minimal(base_size = 8) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold")
  )
```




```
# Hitung Parameter Lambda dan Miu
lambda <- 1 / mean(inter_arrivals)
mu     <- 1 / mean(all_service_times)
c_server <- 2  # Jumlah server

cat("=== PARAMETER M/M/2 ===\n")
```

```
## === PARAMETER M/M/2 ===
```

```
cat("Tingkat Kedatangan (Lambda) :", round(lambda, 4), "pelanggan/menit\n")
```

```
## Tingkat Kedatangan (Lambda) : 1.2785 pelanggan/menit
```

```
cat("Tingkat Pelayanan (Mu)      :", round(mu, 4), "pelanggan/menit\n")
```

```
## Tingkat Pelayanan (Mu)      : 2.7589 pelanggan/menit
```

```
cat("Jumlah Server (c)           :", c_server, "\n")
```

```
## Jumlah Server (c)           : 2
```

```
a <- mean(all_service_times)
a
```

```
## [1] 0.3624672
```

```
# stady state
c_values <- c(1, 2, 3)

cat("\n--- HASIL CEK STEADY STATE ---\n")
```

```
##
## --- HASIL CEK STEADY STATE ---
```

```
for (c_server in c_values) {

  # Hitung utilisasi
  rho <- lambda / (c_server * mu)

  # Tampilkan hasil
  cat("\nJumlah Kasir:", c_server, "\n")
  cat("Nilai Utilitas (rho):", round(rho, 4), "\n")

  # Cek kondisi steady state
  if (rho < 1) {
    cat("Status: Sistem berada pada kondisi steady-state\n")
  } else {
    cat("Status: Sistem tidak stabil (tidak memenuhi steady-state)\n")
  }
}
```

```
##
## Jumlah Kasir: 1
## Nilai Utilitas (rho): 0.4634
## Status: Sistem berada pada kondisi steady-state
##
## Jumlah Kasir: 2
## Nilai Utilitas (rho): 0.2317
## Status: Sistem berada pada kondisi steady-state
##
## Jumlah Kasir: 3
## Nilai Utilitas (rho): 0.1545
## Status: Sistem berada pada kondisi steady-state
```

```
input_mmc <- NewInput.MMC(lambda = lambda, mu = mu, c = c_server)

  # Jalankan Model
  model_mmc <- QueueingModel(input_mmc)

  # Tampilkan Hasil Lengkap
  print(summary(model_mmc))
```

```
##     lambda      mu c  k  m       RO        P0          Lq          Wq        X
## 1 1.278472 2.75887 3 NA NA 0.154468 0.6287761 0.002253216 0.001762429 1.278472
##           L         W       Wqq      Lqq
## 1 0.4656573 0.3642296 0.1428951 1.182687
```

```
  # Ambil nilai spesifik untuk Laporan
  Lq <- model_mmc$Lq
  Ls <- model_mmc$L
  Wq <- model_mmc$Wq
  Ws <- model_mmc$W
  P0 <- model_mmc$Pn[1] # Probabilitas kosong (n=0)

  cat("=== HASIL KINERJA M/M/2 ===\n")
```

```
## === HASIL KINERJA M/M/2 ===
```

```
  cat("P0 (Peluang sistem kosong) :", round(P0, 4), "\n")
```

```
## P0 (Peluang sistem kosong) : 0.6288
```

```
  cat("Lq (Antrian)               :", round(Lq, 4), "orang\n")
```

```
## Lq (Antrian)               : 0.0023 orang
```

```
  cat("Ls (Sistem)               :", round(Ls, 4), "orang\n")
```

```
## Ls (Sistem)               : 0.4657 orang
```

```
  cat("Wq (Waktu tunggu antrian)  :", round(Wq, 4), "menit\n")
```

```
## Wq (Waktu tunggu antrian)  : 0.0018 menit
```

```
  cat("Ws (Waktu tunggu sistem)   :", round(Ws, 4), "menit\n")
```

```
## Ws (Waktu tunggu sistem)   : 0.3642 menit
```

```
# simulasi dengan c = 1,2,3
# Parameter
c_values <- c(1, 2, 3)  # jumlah kasir yang ingin diuji

for (c_server in c_values) {

  # Buat input model M/M/c
  input_mmc <- NewInput.MMC(lambda = lambda, mu = mu, c = c_server)

  # Jalankan model
  model_mmc <- QueueingModel(input_mmc)

  # Ambil nilai spesifik
  Lq <- model_mmc$Lq
  Ls <- model_mmc$L
  Wq <- model_mmc$Wq
  Ws <- model_mmc$W
  P0 <- model_mmc$Pn[1]

  # Output
  cat("\n============================\n")
  cat("=== HASIL KINERJA M/M/", c_server, " ===\n", sep = "")
  cat("============================\n")
  cat("P0 (Peluang sistem kosong) :", round(P0, 4), "\n")
  cat("Lq (Antrian)               :", round(Lq, 4), "orang\n")
  cat("Ls (Sistem)                :", round(Ls, 4), "orang\n")
  cat("Wq (Waktu tunggu antrian)  :", round(Wq, 4), "menit\n")
  cat("Ws (Waktu tunggu sistem)   :", round(Ws, 4), "menit\n")
}
```

```
##
## ============================
## === HASIL KINERJA M/M/1 ===
## ============================
## P0 (Peluang sistem kosong) : 0.5366
## Lq (Antrian)               : 0.4002 orang
## Ls (Sistem)                : 0.8636 orang
## Wq (Waktu tunggu antrian)  : 0.313 menit
## Ws (Waktu tunggu sistem)   : 0.6755 menit
##
## ============================
## === HASIL KINERJA M/M/2 ===
## ============================
## P0 (Peluang sistem kosong) : 0.6238
## Lq (Antrian)               : 0.0263 orang
## Ls (Sistem)                : 0.4897 orang
## Wq (Waktu tunggu antrian)  : 0.0206 menit
## Ws (Waktu tunggu sistem)   : 0.383 menit
##
## ============================
## === HASIL KINERJA M/M/3 ===
## ============================
## P0 (Peluang sistem kosong) : 0.6288
## Lq (Antrian)               : 0.0023 orang
## Ls (Sistem)                : 0.4657 orang
## Wq (Waktu tunggu antrian)  : 0.0018 menit
## Ws (Waktu tunggu sistem)   : 0.3642 menit
```