

Code R Tugas Besar Komputasi Statistik

Kelompok5

2025-11-18

Code R Tugas Besar Komputasi Statistik 2025

Algoritma Iteratif

IMPLEMENTASI ALGORITMA ITERATIF UNTUK SIMULASI DAN OPTIMASI SISTEM ANTRIAN KANTIN BKL INSTITUT TEKNOLOGI SUMATERA

Kelompok 5:

KHAIRUNNISA MAHARANI 123450071

LUTFIA AISYAH PUTRI 123450074

DEVI RAHAYU 123450010

RIDHO BENEDICTUS TOGI MANIK 123450060

```
# -----
# Install & Load Package
# -----  
  
library(dplyr)  
  
## Warning: package 'dplyr' was built under R version 4.4.3  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union  
  
library(lubridate)  
  
## Warning: package 'lubridate' was built under R version 4.4.3
```

```

## 
## Attaching package: 'lubridate'

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

# -----
# FASE 0.5: Data Munging / Wrangling
# (Mengubah data mentah dari gambar menjadi data durasi)
# -----
print("Fase 0.5: Membaca dan melakukan data wrangling...")

## [1] "Fase 0.5: Membaca dan melakukan data wrangling..."

data_mentah_raw <- NULL

tryCatch({
  data_mentah_raw <- read.csv("~/data_mentah1.csv")
}, error = function(e) {
  print("----!!! ERROR !!!----")
  print("File 'data_mentah1.csv' tidak ditemukan.")
  print("-----")
})

# -----
# Cek dan proses data
# -----
if (!is.null(data_mentah_raw)) {

  # Ubah kolom waktu (teks) menjadi format waktu (hms)
  data_mentah <- data_mentah_raw %>%
    mutate(
      AT_Awal = hms(AT_Awal),
      SET_Pesan = hms(SET_Pesan),
      SET_Kasir = hms(SET_Kasir),
      AT_Konfirmasi = hms(AT_Konfirmasi),
      SET_Total = hms(SET_Total)
    )

  # Hitung durasi (dalam detik + konversi ke menit)
  data_awal <- data_mentah %>%
    arrange(AT_Awal) %>%
    mutate(
      # IAT = waktu antar tiba
      IAT_raw = as.numeric(AT_Awal - lag(AT_Awal), units = "secs"),
      
      # ST Kasir = waktu pelayanan kasir
      ST_Kasir_raw = as.numeric(SET_Kasir - SET_Pesan, units = "secs"),
      
      # ST Dapur = waktu produksi makanan
      ST_Dapur_raw = as.numeric(SET_Total - AT_Konfirmasi, units = "secs")
    ) %>%
}

```

```

# Konversi detik ke menit
mutate(
  IAT = IAT_raw / 60,
  ST_Kasir = ST_Kasir_raw / 60,
  ST_Dapur = ST_Dapur_raw / 60
) %>%
# Pilih kolom penting
select(IAT, ST_Kasir, ST_Dapur)

# Baris pertama IAT = NA (karena ga punya data sebelumnya)
data_awal <- data_awal %>% filter(!is.na(IAT))

print("Data wrangling selesai")
print("Data durasi (menit):")
print(head(data_awal))

} else {
  print("Data mentah kosong. Proses dihentikan.")
}

## [1] "Data wrangling selesai"
## [1] "Data durasi (menit):"
##           IAT ST_Kasir ST_Dapur
## 1 6.816667 4.666667 10.33333
## 2 5.916667 3.416667 15.60000
## 3 5.400000 2.350000 11.26667
## 4 3.616667 4.133333 15.85000
## 5 3.633333 4.283333 13.51667
## 6 3.233333 3.683333 12.85000

# -----
# FASE 1: Pengolahan Data Awal (Observasi 30 Sampel)
# -----
print("Fase 1: Menghitung parameter data awal...")

## [1] "Fase 1: Menghitung parameter data awal..."

# Hitung parameter (Rata-rata & SD) dari data hasil wrangling
mean_iat <- mean(data_awal$IAT, na.rm = TRUE)
sd_iat <- sd(data_awal$IAT, na.rm = TRUE)
mean_kasir <- mean(data_awal$ST_Kasir, na.rm = TRUE)
sd_kasir <- sd(data_awal$ST_Kasir, na.rm = TRUE)
mean_dapur <- mean(data_awal$ST_Dapur, na.rm = TRUE)
sd_dapur <- sd(data_awal$ST_Dapur, na.rm = TRUE)

print("Parameter dari data awal (dalam menit):")

## [1] "Parameter dari data awal (dalam menit):"

print(paste("IAT: Rata-rata=", round(mean_iat, 2), ", SD=", round(sd_iat, 2)))

## [1] "IAT: Rata-rata= 6.99 , SD= 2.23"

```

```

print(paste("ST Kasir: Rata-rata=", round(mean_kasir, 2), ", SD=", round(sd_kasir, 2)))

## [1] "ST Kasir: Rata-rata= 3.51 , SD= 0.89"

print(paste("ST Dapur: Rata-rata=", round(mean_dapur, 2), ", SD=", round(sd_dapur, 2)))

## [1] "ST Dapur: Rata-rata= 14.86 , SD= 3.68"

# -----
# FASE 2: Implementasi Algoritma & Skenario Basis
# -----

print("Fase 2: Menyiapkan simulasi dan fungsi...")

## [1] "Fase 2: Menyiapkan simulasi dan fungsi..."

n_simulasi <- 10000

fungsi_hitung_tis <- function(sim_st_kasir, sim_st_dapur) {
  tis_per_pelanggan <- sim_st_kasir + sim_st_dapur
  tis_per_pelanggan[tis_per_pelanggan < 0] <- 0
  return(tis_per_pelanggan)
}

# -----
# FASE 3: Optimasi dan Pencarian Minimum (Algoritma Iteratif)
# -----

print("Fase 3: Menjalankan Perulangan Skenario...")

## [1] "Fase 3: Menjalankan Perulangan Skenario..."

hasil_tis_rata2 <- c()
nama_skenario <- c()

for (i in 1:4) {
  if (i == 1) {
    nama <- "Skenario 1 (Basis)"
    mean_k_temp <- mean_kasir
    mean_d_temp <- mean_dapur
  } else if (i == 2) {
    nama <- "Skenario 2 (Tambah Kasir, Efisiensi +30%)"
    mean_k_temp <- mean_kasir * (1 - 0.30)
    mean_d_temp <- mean_dapur
  } else if (i == 3) {
    nama <- "Skenario 3 (Dapur -15%)"
    mean_k_temp <- mean_kasir
    mean_d_temp <- mean_dapur * (1 - 0.15)
  } else {
    nama <- "Skenario 4 (Dapur -30%)"
  }
}

```

```

    mean_k_temp <- mean_kasir
    mean_d_temp <- mean_dapur * (1 - 0.30)
}

sim_st_kasir_skenario <- rnorm(n_simulasi, mean_k_temp, sd_kasir)
sim_st_dapur_skenario <- rnorm(n_simulasi, mean_d_temp, sd_dapur)

tis_simulasi <- fungsi_hitung_tis(sim_st_kasir_skenario, sim_st_dapur_skenario)
rata2_tis_skenario_ini <- mean(tis_simulasi)

hasil_tis_rata2 <- c(hasil_tis_rata2, rata2_tis_skenario_ini)
nama_skenario <- c(nama_skenario, nama)

cat("Iterasi ke-", i, " Selesai. ", nama, " | Rata-rata TIS:", round(rata2_tis_skenario_ini, 2), "\\\n")
}

## Iterasi ke- 1 Selesai. Skenario 1 (Basis) | Rata-rata TIS: 18.39
## Iterasi ke- 2 Selesai. Skenario 2 (Tambah Kasir, Efisiensi +30%) | Rata-rata TIS: 17.31
## Iterasi ke- 3 Selesai. Skenario 3 (Dapur -15%) | Rata-rata TIS: 16.09
## Iterasi ke- 4 Selesai. Skenario 4 (Dapur -30%) | Rata-rata TIS: 13.87

print("...Perulangan Skenario Selesai.")

## [1] "...Perulangan Skenario Selesai.

# -----
# FASE 4: Analisis dan Pencarian Minimum (Menggunakan Kasus I)
# -----"

print("Fase 4: Analisis Hasil dan Pencarian Minimum...")

## [1] "Fase 4: Analisis Hasil dan Pencarian Minimum..."

# Gabungkan hasil simulasi ke dalam data frame
hasil_akhir <- data.frame(
  Skenario = nama_skenario,
  Rata_Rata_TIS = hasil_tis_rata2
)

print(hasil_akhir)

##           Skenario Rata_Rata_TIS
## 1      Skenario 1 (Basis)     18.38530
## 2 Skenario 2 (Tambah Kasir, Efisiensi +30%)     17.31464
## 3      Skenario 3 (Dapur -15%)     16.08881
## 4      Skenario 4 (Dapur -30%)     13.87291

# -----
# Pencarian Minimum Iteratif (Modifikasi Kasus I dari Modul)
# -----"

x_hasil <- hasil_akhir$Rata_Rata_TIS

```

```

n <- length(x_hasil)
nilai_minimum_sementara <- x_hasil[1]
indeks_minimum <- 1

for (i in 2:n) {
  if (x_hasil[i] < nilai_minimum_sementara) {
    nilai_minimum_sementara <- x_hasil[i]
    indeks_minimum <- i
  }
}

cat("\n--- Pencarian Minimum Iteratif (Modifikasi Kasus I) ---\n")

## --- Pencarian Minimum Iteratif (Modifikasi Kasus I) ---

cat("Nilai TIS minimum ditemukan:", round(nilai_minimum_sementara, 2), "\n")

## Nilai TIS minimum ditemukan: 13.87

cat("Didapat dari Skenario:", hasil_akhir$Skenario[indeks_minimum], "\n")

## Didapat dari Skenario: Skenario 4 (Dapur -30%)

# -----
# Visualisasi Perbandingan Skenario
# -----
library(ggplot2)

print("Membuat visualisasi perbandingan skenario dengan ggplot2...")

## [1] "Membuat visualisasi perbandingan skenario dengan ggplot2...""

df_plot <- hasil_akhir

# Tambahkan kolom singkatan skenario
df_plot$Kode <- paste0("S", seq_len(nrow(df_plot)))

# Warnanya
warna_penelitian <- c("#4E79A7", "#F28E2B", "#E15759", "#76B7B2")

ggplot(df_plot, aes(x = Kode, y = Rata_Rata_TIS, fill = Skenario)) +
  # Bar
  geom_col(width = 0.65) +
  # Label angka
  geom_text(aes(label = round(Rata_Rata_TIS, 2)),
            vjust = -0.5, size = 4) +

```

```

# Warna manual
scale_fill_manual(values = warna_penelitian) +
  
# Batas tinggi grafik supaya bar tidak terlalu ke atas
scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  
# Garis rata-rata
geom_hline(yintercept = mean(df_plot$Rata_Rata_TIS),
            linetype = "dashed", color = "gray30") +
  
labs(
  title = "Perbandingan Mean TIS Antar Skenario",
  x = "Skenario",
  y = "Rata-rata Waktu Total (TIS) dalam Menit",
  fill = "Nama Skenario"
) +
  
theme_minimal(base_size = 11) +
theme(
  plot.title = element_text(hjust = 0.5, face = "bold"),
  legend.position = "right"
)

```

