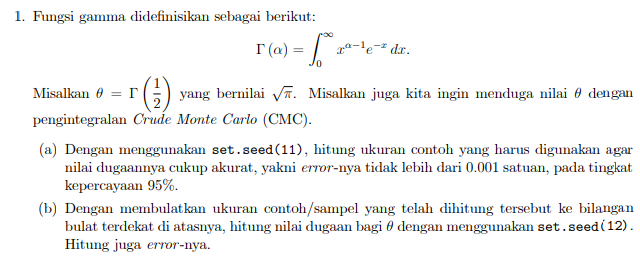
Prak7SMA\_Haidar\_G5402221039

Adhan Haidar Hakim

2025-10-01



#1.Estimasi fungsi Gammadgn pengintegralan Crude Monte Carlo.   
#1.a. ngitung ukuran contoh (n) dengan error gak lebih dari 0.001 satuan, dengan selang kepercayaan 95%.   
  
set.seed(11)  
# Parameter yang diketahui  
c\_val <- 1.96  
E\_val <- 0.001  
m\_initial <- 1000 # Ukuran sampel awal  
  
# Membangkitkan sampel awal dari sebaran Seragam(0,1)  
y\_initial <- runif(m\_initial)  
  
# Menghitung g(y)  
g\_y\_initial <- 1 / sqrt(-log(y\_initial))  
  
# Menghitung dugaan ragam (sigma^2)  
sigma2\_duga <- var(g\_y\_initial)  
print(paste("Dugaan Ragam (sigma^2):", sigma2\_duga))

## [1] "Dugaan Ragam (sigma^2): 3.74608585124726"

#ngitung ukuran sample   
n\_required <- (c\_val^2 \* sigma2\_duga) / (E\_val^2)  
print(paste("Ukuran Sampel yang Dibutuhkan (n):", n\_required))

## [1] "Ukuran Sampel yang Dibutuhkan (n): 14390963.4061515"

#1.b itung nilai pendugaan dengan set.seed dan juga errornya. jgn lupda dibulatin contoh sampkle yg diitung ke bil bulat terdekat diatasnya.   
  
set.seed(12)  
  
# Ukuran sampel final (pembulatan ke atas)  
n\_final <- 1938019  
# Membangkitkan sampel final dari sebaran Seragam(0,1)  
y\_final <- runif(n\_final)  
# Menghitung g(y) untuk sampel final  
g\_y\_final <- 1 / sqrt(-log(y\_final))  
# Menghitung nilai dugaan theta  
theta\_duga <- mean(g\_y\_final)  
print(paste("Nilai Dugaan (theta\_duga):", theta\_duga))

## [1] "Nilai Dugaan (theta\_duga): 1.77323151617891"

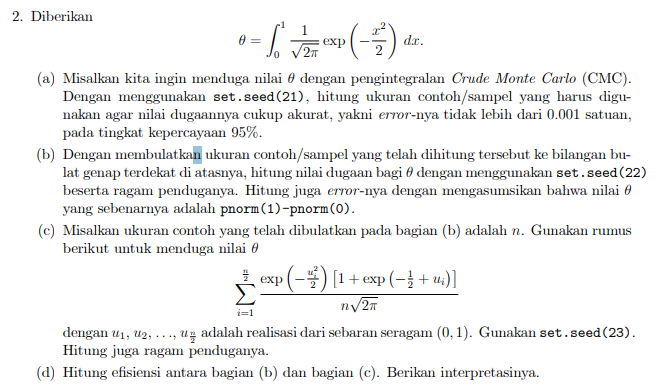
# Nilai theta sebenarnya adalah sqrt(pi)  
theta\_true <- sqrt(pi)  
print(paste("Nilai Sebenarnya (sqrt(pi)):", theta\_true))

## [1] "Nilai Sebenarnya (sqrt(pi)): 1.77245385090552"

# Menghitung error absolut  
error\_abs <- abs(theta\_duga - theta\_true)  
print(paste("Error Absolut:", error\_abs))

## [1] "Error Absolut: 0.000777665273395511"

2.



#2 Estimasi Integral fungsi normal  
  
#2.a ngitung ukuran sample dgn error gak lebih dari 0.001 satuan, pad tingkat kepercayaan.   
set.seed(21)  
  
# Parameter  
c\_val <- 1.96  
E\_val <- 0.001  
m\_initial <- 1000  
  
# Membangkitkan sampel awal U(0,1)  
x\_initial <- runif(m\_initial)  
  
# Menghitung g(x) menggunakan fungsi dnorm (FKP Normal Standar)  
g\_x\_initial <- dnorm(x\_initial, mean = 0, sd = 1)  
  
# Menghitung dugaan ragam (sigma^2)  
sigma2\_hat\_2a <- var(g\_x\_initial)  
print(paste("Dugaan Ragam (sigma^2):", sigma2\_hat\_2a))

## [1] "Dugaan Ragam (sigma^2): 0.00235091198406682"

# Menghitung ukuran sampel yang dibutuhkan  
n\_required\_2a <- (c\_val^2 \* sigma2\_hat\_2a) / (E\_val^2)  
print(paste("Ukuran Sampel yang Dibutuhkan (n):", n\_required\_2a))

## [1] "Ukuran Sampel yang Dibutuhkan (n): 9031.2634779911"

#2.b estimasi dengan CMC   
# 2.b Estimasi dengan CMC  
  
set.seed(22)  
  
# Ukuran sampel final (dibulatkan ke atas genap)  
n\_final\_2b <- 20632  
  
# Membangkitkan sampel final U(0,1)  
x\_final\_2b <- runif(n\_final\_2b)  
  
# Menghitung g(x)  
g\_x\_final\_2b <- dnorm(x\_final\_2b)  
  
# Menghitung nilai dugaan theta  
theta\_hat\_2b <- mean(g\_x\_final\_2b)  
print(paste("Nilai Dugaan CMC (theta\_hat):", theta\_hat\_2b))

## [1] "Nilai Dugaan CMC (theta\_hat): 0.341389659427526"

# Menghitung ragam PENDUGA  
var\_estimator\_2b <- var(g\_x\_final\_2b) / n\_final\_2b  
print(paste("Ragam Penduga CMC:", var\_estimator\_2b))

## [1] "Ragam Penduga CMC: 1.1361901401738e-07"

# Nilai sebenarnya adalah pnorm(1) - pnorm(0)  
theta\_true\_2b <- pnorm(1) - pnorm(0)  
print(paste("Nilai Sebenarnya:", theta\_true\_2b))

## [1] "Nilai Sebenarnya: 0.341344746068543"

# Menghitung error  
error\_2b <- abs(theta\_hat\_2b - theta\_true\_2b)  
print(paste("Error Absolut:", error\_2b))

## [1] "Error Absolut: 4.49133589827122e-05"

#2.c Estimasi dgn metode kedua (rumus lain)  
# 2.c Estimasi dengan metode kedua  
  
set.seed(23)  
  
# Ukuran n/2  
n\_half\_2c <- 10316  
n\_final\_2c <- 20632  
  
# Membangkitkan sampel u sebanyak n/2  
u\_2c <- runif(n\_half\_2c)  
  
# Menghitung H(u) sesuai rumus di soal  
H\_u <- (exp(-u\_2c^2 / 2) \* (1 + exp(-0.5 + u\_2c))) / sqrt(2 \* pi)  
  
# Menghitung nilai dugaan theta  
# Estimatornya adalah (1/n) \* sum(H(u)) = (1/2) \* mean(H(u))  
theta\_hat\_2c <- sum(H\_u) / n\_final\_2c  
print(paste("Nilai Dugaan Metode Kedua (theta\_hat):", theta\_hat\_2c))

## [1] "Nilai Dugaan Metode Kedua (theta\_hat): 0.341456447837151"

# Menghitung ragam PENDUGA  
# Var(theta\_hat) = Var( (1/2) \* mean(H(u)) ) = (1/4) \* Var(H(u)) / (n/2)  
var\_estimator\_2c <- (1/4) \* var(H\_u) / n\_half\_2c  
print(paste("Ragam Penduga Metode Kedua:", var\_estimator\_2c))

## [1] "Ragam Penduga Metode Kedua: 8.67655276516125e-09"

#2.d  
# 2.d Menghitung Efisiensi  
  
efisiensi <- var\_estimator\_2b / var\_estimator\_2c  
print(paste("Efisiensi:", efisiensi))

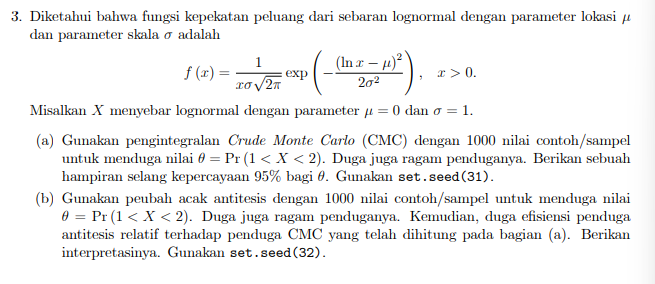
## [1] "Efisiensi: 13.0949487766145"

Interpretasi:

Nilai efisien 13.094 menunjukkan bahwa pada bagian (c) 13x lebih efisien daripada metode Crude Monte Carlo (CmC) di bagian b.

dengan tingkat akurasi yg sama, jawaban c butuh ukuran sample 13x lebih sedikit dibandingkan dengan CMC

3.



#3. Estimasi Peluang Sebaran Lognormal  
# 3.a Estimasi dengan CMC  
  
set.seed(31)  
  
n\_3a <- 1000  
  
# Membangkitkan sampel y ~ U(0,1)  
y\_3a <- runif(n\_3a)  
  
# Transformasi y ke x  
x\_3a <- 1 + y\_3a  
  
# Menghitung g(y) = f(x)  
g\_y\_3a <- dlnorm(x\_3a, meanlog = 0, sdlog = 1)  
  
# Nilai dugaan theta  
theta\_hat\_3a <- mean(g\_y\_3a)  
print(paste("Nilai Dugaan CMC:", theta\_hat\_3a))

## [1] "Nilai Dugaan CMC: 0.254388963376076"

# Ragam PENDUGA CMC  
var\_estimator\_3a <- var(g\_y\_3a) / n\_3a  
print(paste("Ragam Penduga CMC:", var\_estimator\_3a))

## [1] "Ragam Penduga CMC: 4.90955561328789e-06"

# ngitung Selang Kepercayaan 95%  
c\_val <- 1.96  
margin\_error <- c\_val \* sqrt(var\_estimator\_3a)  
lower\_bound <- theta\_hat\_3a - margin\_error  
upper\_bound <- theta\_hat\_3a + margin\_error  
print(paste("Selang Kepercayaan 95%:", paste("(", lower\_bound, ",", upper\_bound, ")")))

## [1] "Selang Kepercayaan 95%: ( 0.250046090036453 , 0.258731836715699 )"

#3.b. Estimasi dgn peubah acak Antitesis make 1000 nilai sample tuk duga thetha = Pr (1 < X < 2 ) dan juga duga ragam penduganya.   
  
  
set.seed(32)  
  
n\_pairs <- 500  
  
# Membangkitkan n/2 sampel awal  
u\_3b <- runif(n\_pairs)  
# Membuat pasangan antitesisnya  
v\_3b <- 1 - u\_3b  
  
# Transformasi u dan v ke domain x  
x1\_3b <- 1 + u\_3b  
x2\_3b <- 1 + v\_3b  
  
# Menghitung g(u) dan g(v)  
g1 <- dlnorm(x1\_3b, meanlog = 0, sdlog = 1)  
g2 <- dlnorm(x2\_3b, meanlog = 0, sdlog = 1)  
  
# Gabungkan hasil untuk estimator antitesis  
g\_anti <- (g1 + g2) / 2  
  
# Nilai dugaan theta antitesis  
theta\_hat\_3b <- mean(g\_anti)  
print(paste("Nilai Dugaan Antitesis:", theta\_hat\_3b))

## [1] "Nilai Dugaan Antitesis: 0.255921762023447"

# Ragam PENDUGA Antitesis  
var\_estimator\_3b <- var(g\_anti) / n\_pairs  
print(paste("Ragam Penduga Antitesis:", var\_estimator\_3b))

## [1] "Ragam Penduga Antitesis: 1.94497994390342e-07"

# Menghitung efisiensi  
efisiensi\_3b <- var\_estimator\_3a / var\_estimator\_3b  
print(paste("Efisiensi Antitesis vs CMC:", efisiensi\_3b))

## [1] "Efisiensi Antitesis vs CMC: 25.2421914615469"