DETEKSI SPAM EMAIL

Miftahul Huda

2023-05-10

Library

```
library(dplyr)
```

```
##
## 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(skimr)
library(DataExplorer)
library(ggplot2)
library(stringr)
library(e1071)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ lubridate 1.9.2     ✓ tibble    3.1.8
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts t
o become errors
```

```
library(ROCR)
```

Dataset

```
data <- read.csv("https://raw.githubusercontent.com/Mif212/dokumen/main/spam.csv")
glimpse(data)
```

```
## Rows: 5,572
## Columns: 5
## $ v1  <chr> "ham", "ham", "spam", "ham", "ham", "spam", "ham", "ham", "spam", …
## $ v2  <chr> "Go until jurong point, crazy.. Available only in bugis n great wo…
## $ X   <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""…
## $ X.1 <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""…
## $ X.2 <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""…
```

Seleksi dan Mengubah Nama Kolom

```
data <- select(data, "v1", "v2")
colnames(data) <- c("label", "pesan")
glimpse(data)
```

```
## Rows: 5,572
## Columns: 2
## $ label <chr> "ham", "ham", "spam", "ham", "ham", "spam", "ham", "ham", "spam"…
## $ pesan <chr> "Go until jurong point, crazy.. Available only in bugis n great …
```

```
colnames(data)
```

```
## [1] "label" "pesan"
```

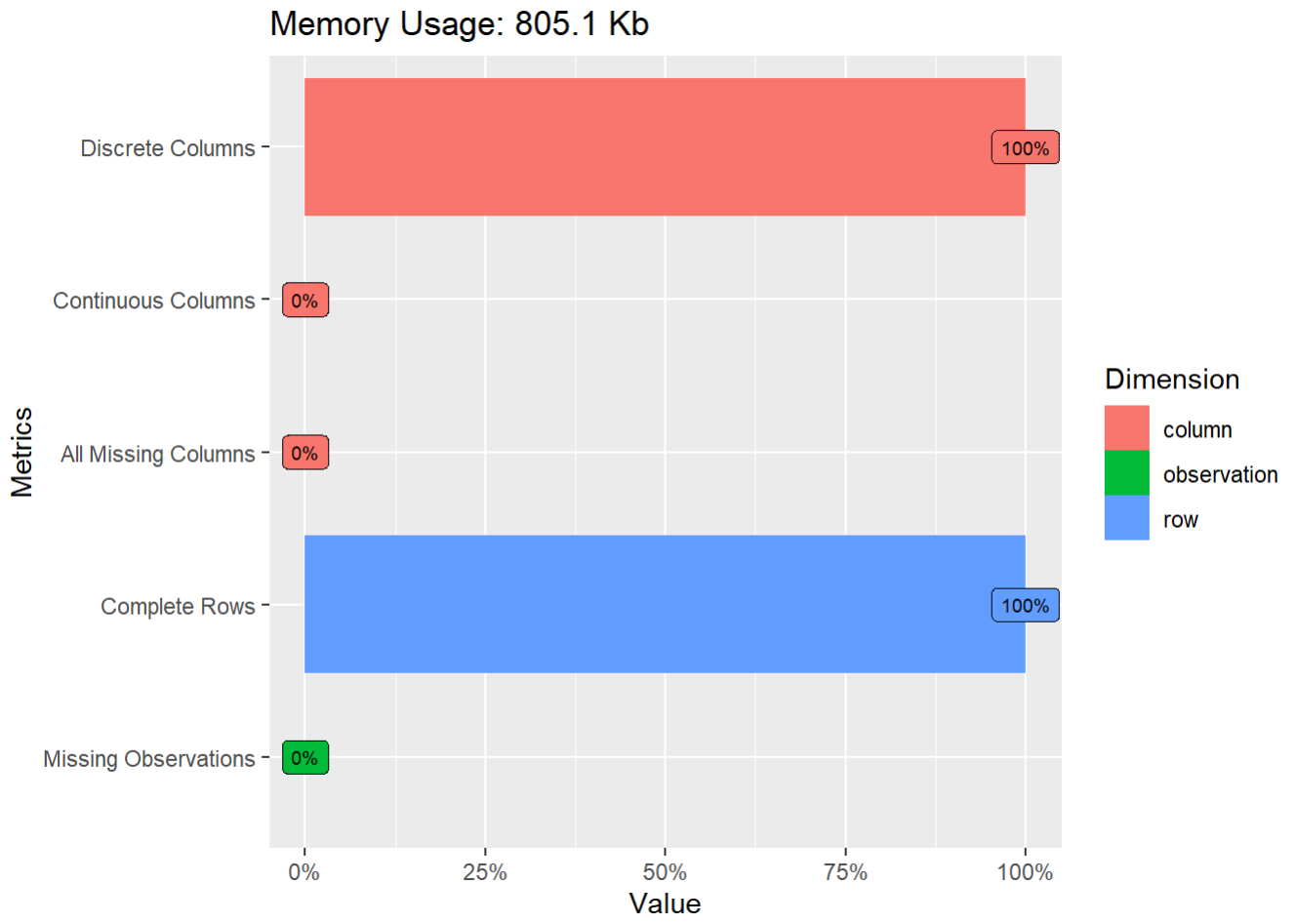
```
dim(data)
```

```
## [1] 5572    2
```

Data Wrangling

Cek Missing Value

```
plot_intro(data = data,
           geom_label_args = list(size=2.5))
```




```
skim_without_charts(data)
```

Data summary

Name	data
Number of rows	5572
Number of columns	2
_____	
Column type frequency:	
character	2
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
label	0	1	3	4	0	2	0
pesan	0	1	2	910	0	5169	0

```
sum(is.na(data))
```

```
## [1] 0
```

Menghapus Karakter Aneh

```
data$pesan <- gsub("[^a-zA-Z0-9 ]", "", data$pesan)
head(data)
```

```
##   label
## 1   ham
## 2   ham
## 3  spam
## 4   ham
## 5   ham
## 6  spam
##
pesan
## 1                                                 Go until jurong point crazy Available only i
n bugis n great world la e buffet Cine there got amore wat
## 2
Ok lar Joking wif u oni
## 3 Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to rece
ive entry questionstd txt rateTCs apply 08452810075over18s
## 4
U dun say so early hor U c already then say
## 5                                                                                          N
ah I dont think he goes to usf he lives around here though
## 6        FreeMsg Hey there darling its been 3 weeks now and no word back Id like some fun you
up for it still Tb ok XxX std chgs to send e5a3150 to rcv
```

Eksplorasi Data Analisis

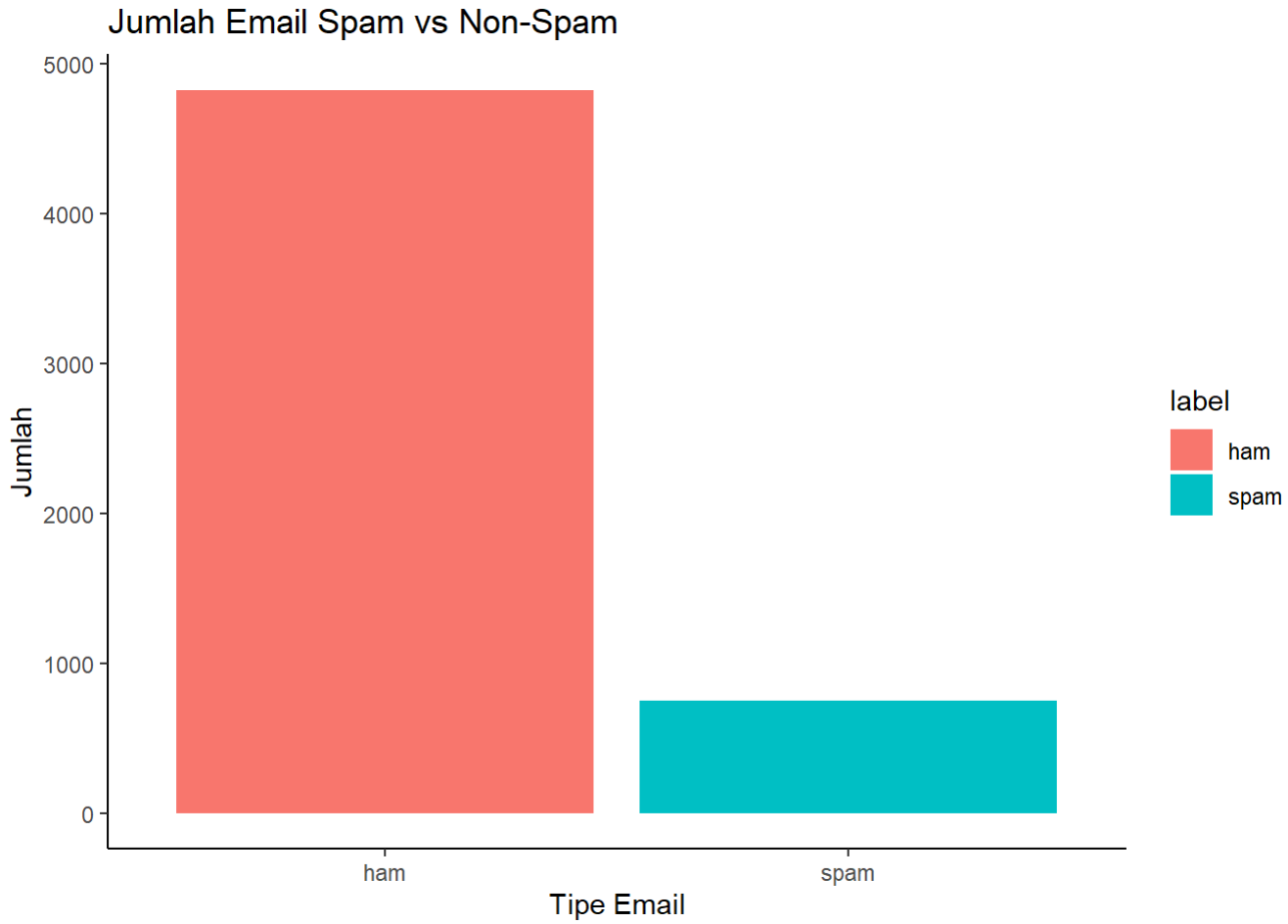
Ringkasan Data

```
summary(data)
```

```
##     label              pesan
##  Length:5572        Length:5572
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

Frekuensi Label

```
ggplot(data, aes(x = label, fill = label)) +
  geom_bar() + theme_classic() +
  labs(title = "Jumlah Email Spam vs Non-Spam",
       x = "Tipe Email",
       y = "Jumlah")
```




```
table(data$label)
```

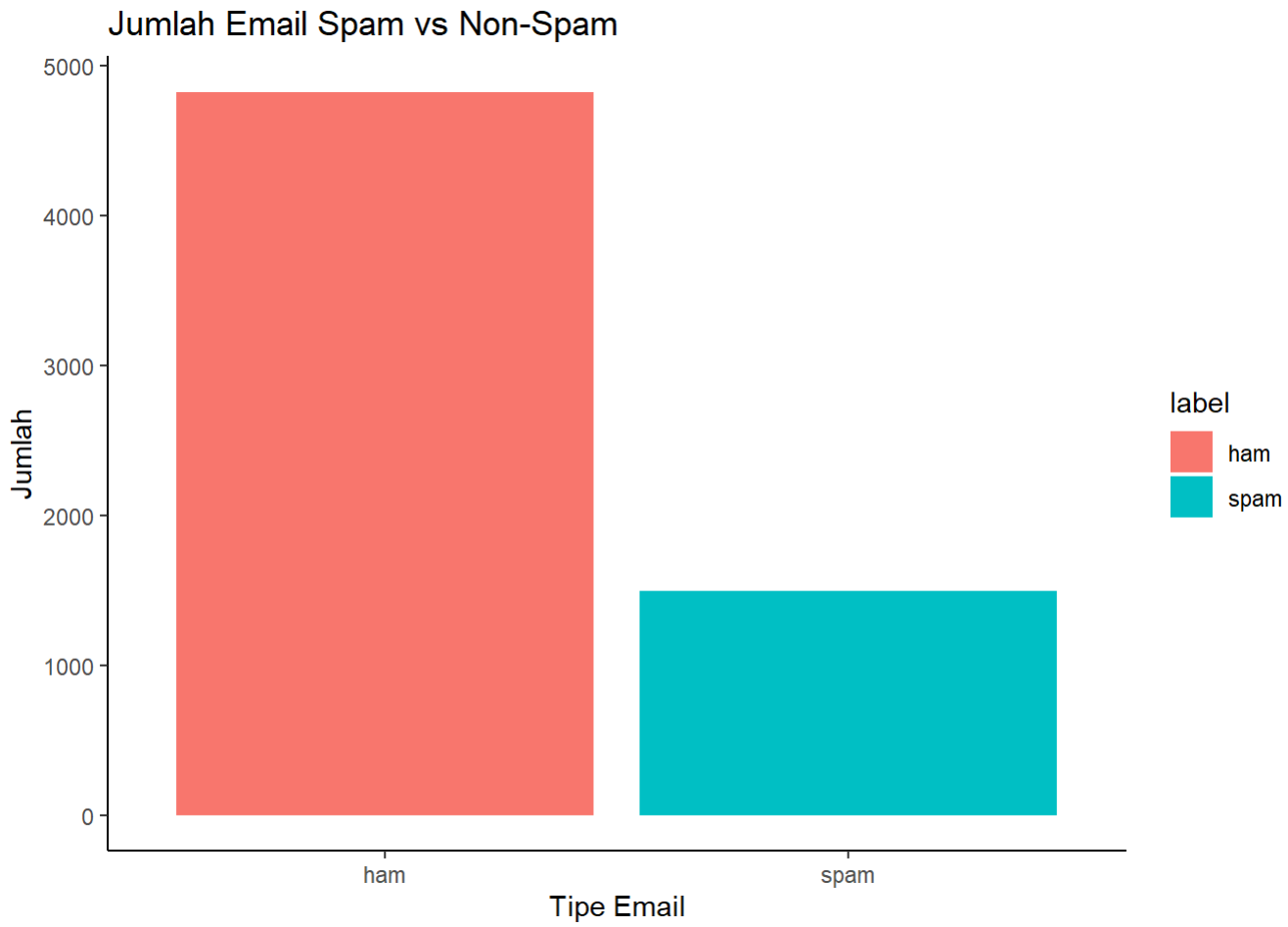
```
##
##  ham spam
## 4825  747
```

Oversampling

```
index_spam <- which((data$label == "spam") == TRUE)
over_spam <- data[index_spam, ]
data <- rbind(data, over_spam)
dim(data)
```

```
## [1] 6319    2
```

```
ggplot(data, aes(x = label, fill = label)) +
  geom_bar() + theme_classic() +
  labs(title = "Jumlah Email Spam vs Non-Spam",
       x = "Tipe Email",
       y = "Jumlah")
```

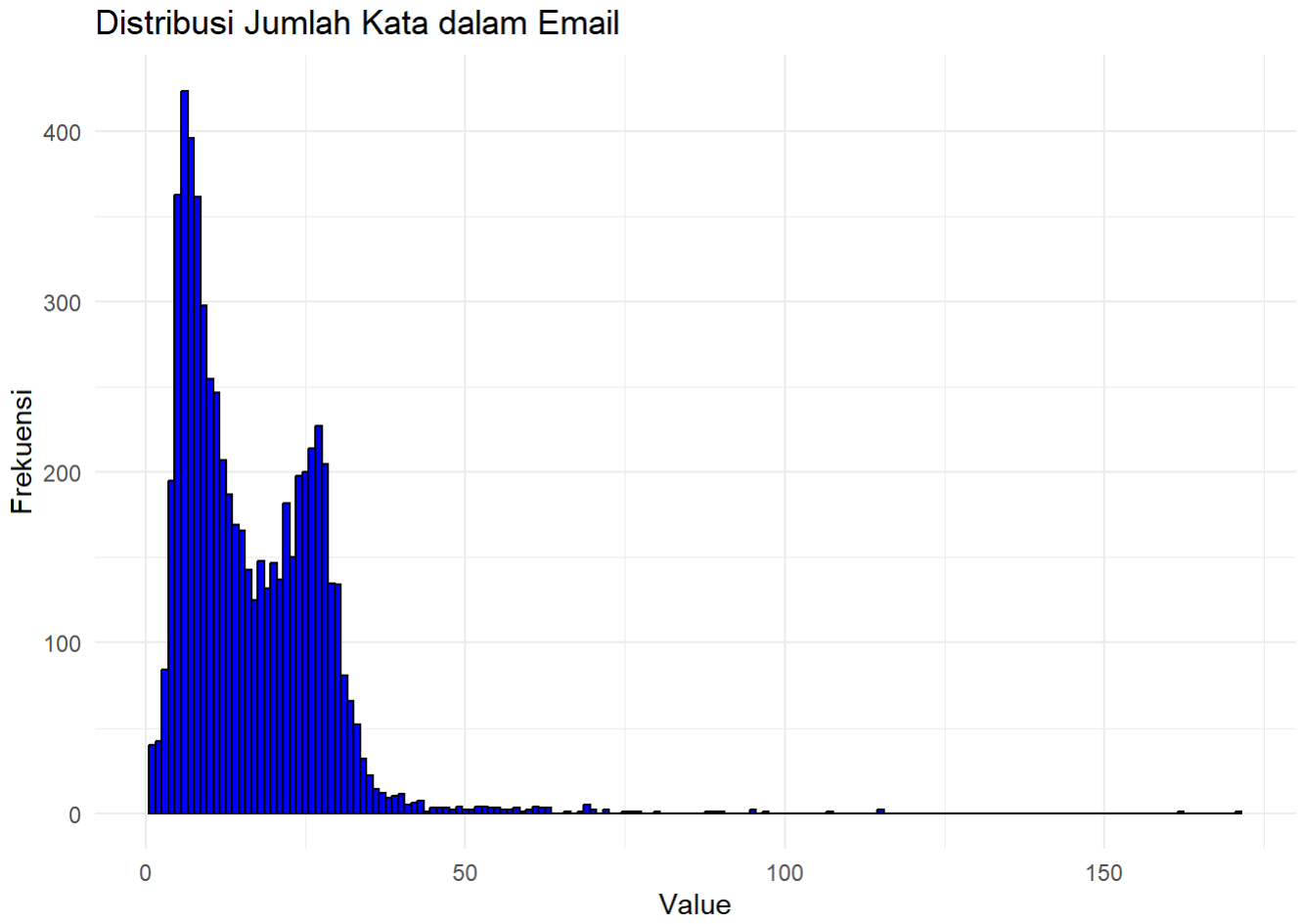



```
table(data$label)
```

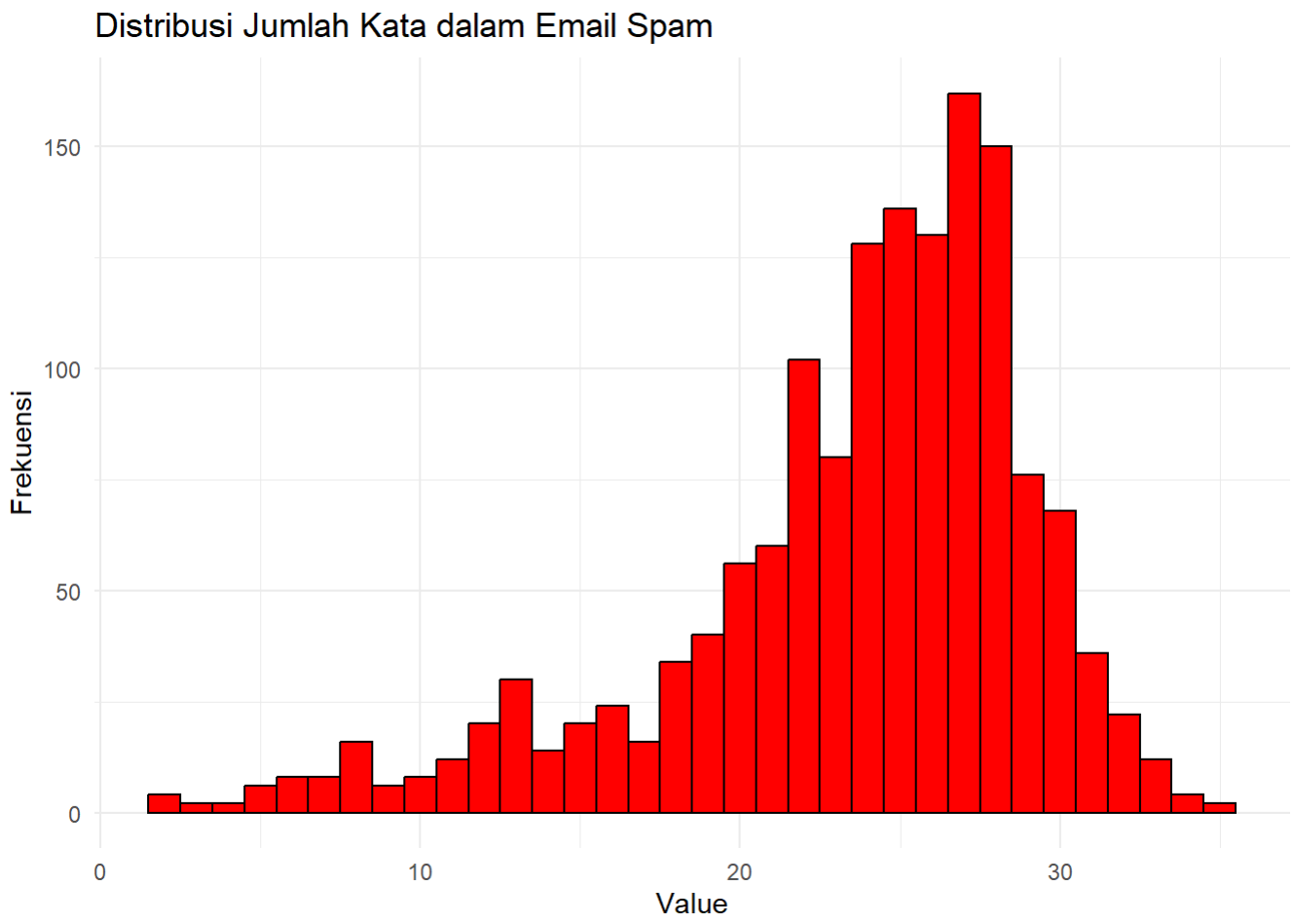
```
##
##  ham spam
## 4825 1494
```

Kata dalam Pesan

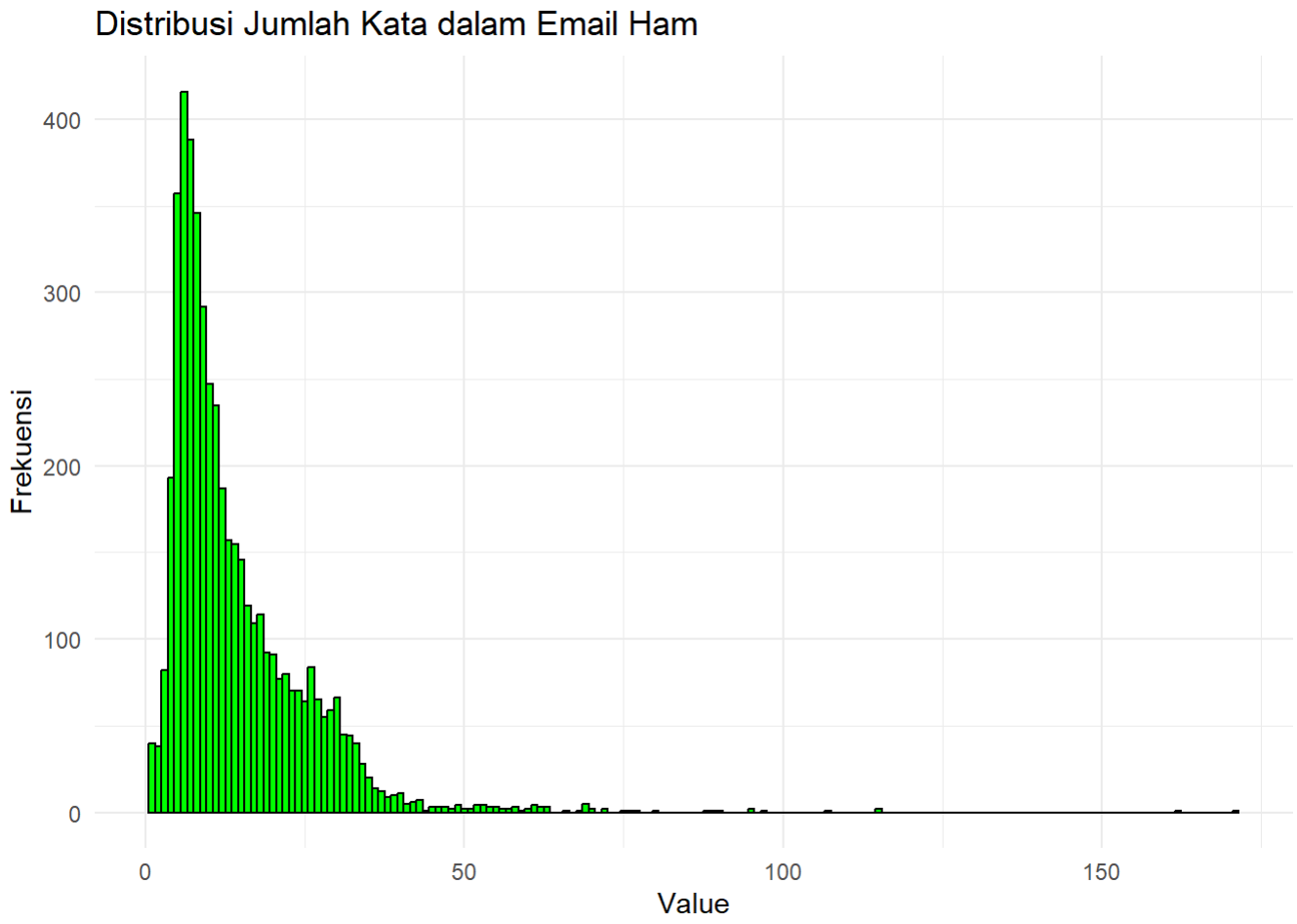
```
count_words <- function(text) {
  words <- str_split(text, "\\s+")
  return(length(words[[1]]))
}

data$word_count <- sapply(data$pesan, count_words)

ggplot(data, aes(word_count)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Kata dalam Email") +
  theme_minimal()
```



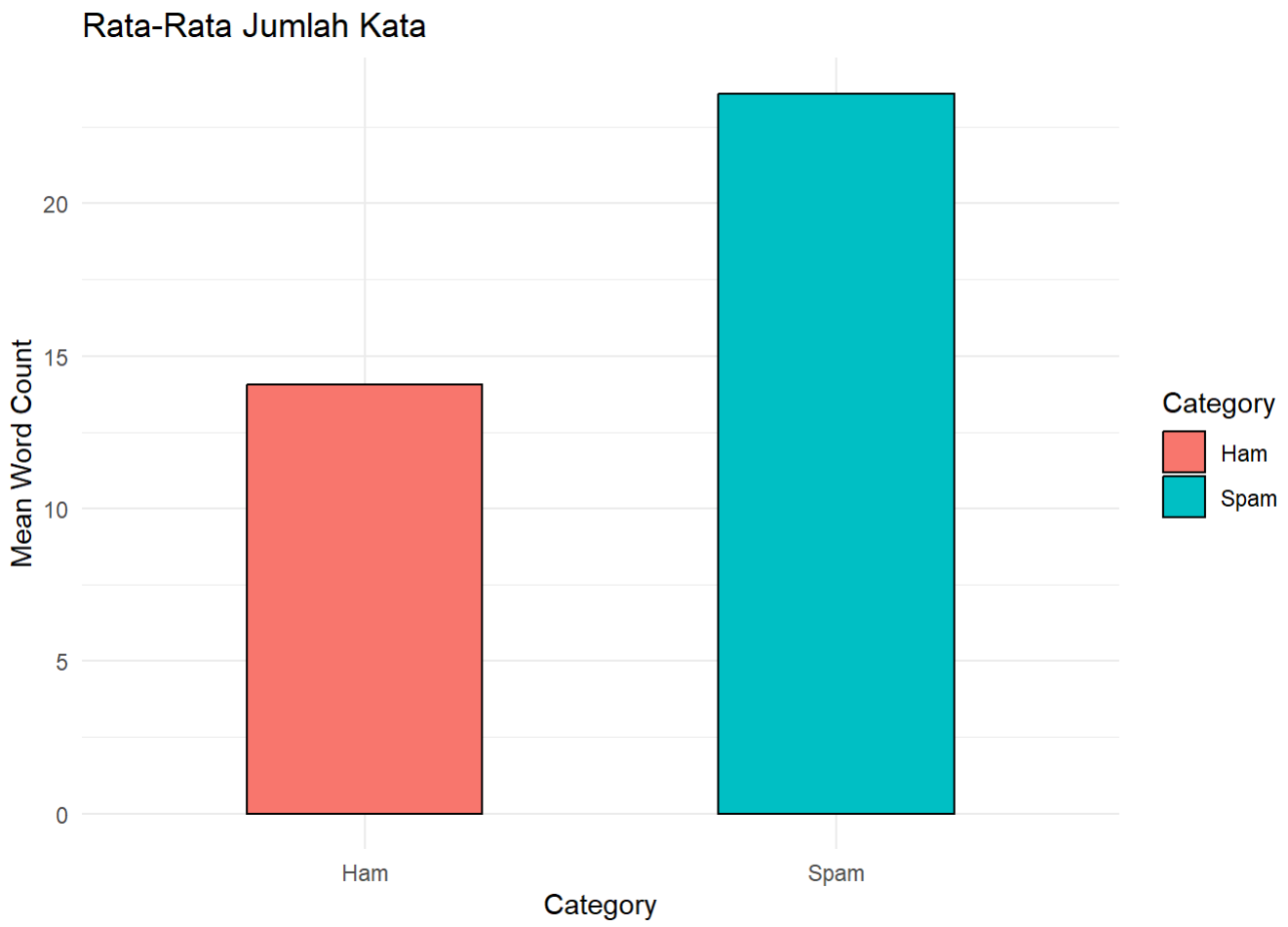

```
spam_count <- data[which((data$label == "spam") == TRUE),]
ham_count <- data[which((data$label == "ham") == TRUE),]

ggplot(spam_count, aes(word_count)) +
  geom_histogram(binwidth = 1, fill = "red", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Kata dalam Email Spam") +
  theme_minimal()
```




```
ggplot(ham_count, aes(word_count)) +
  geom_histogram(binwidth = 1, fill = "green", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Kata dalam Email Ham") +
  theme_minimal()
```

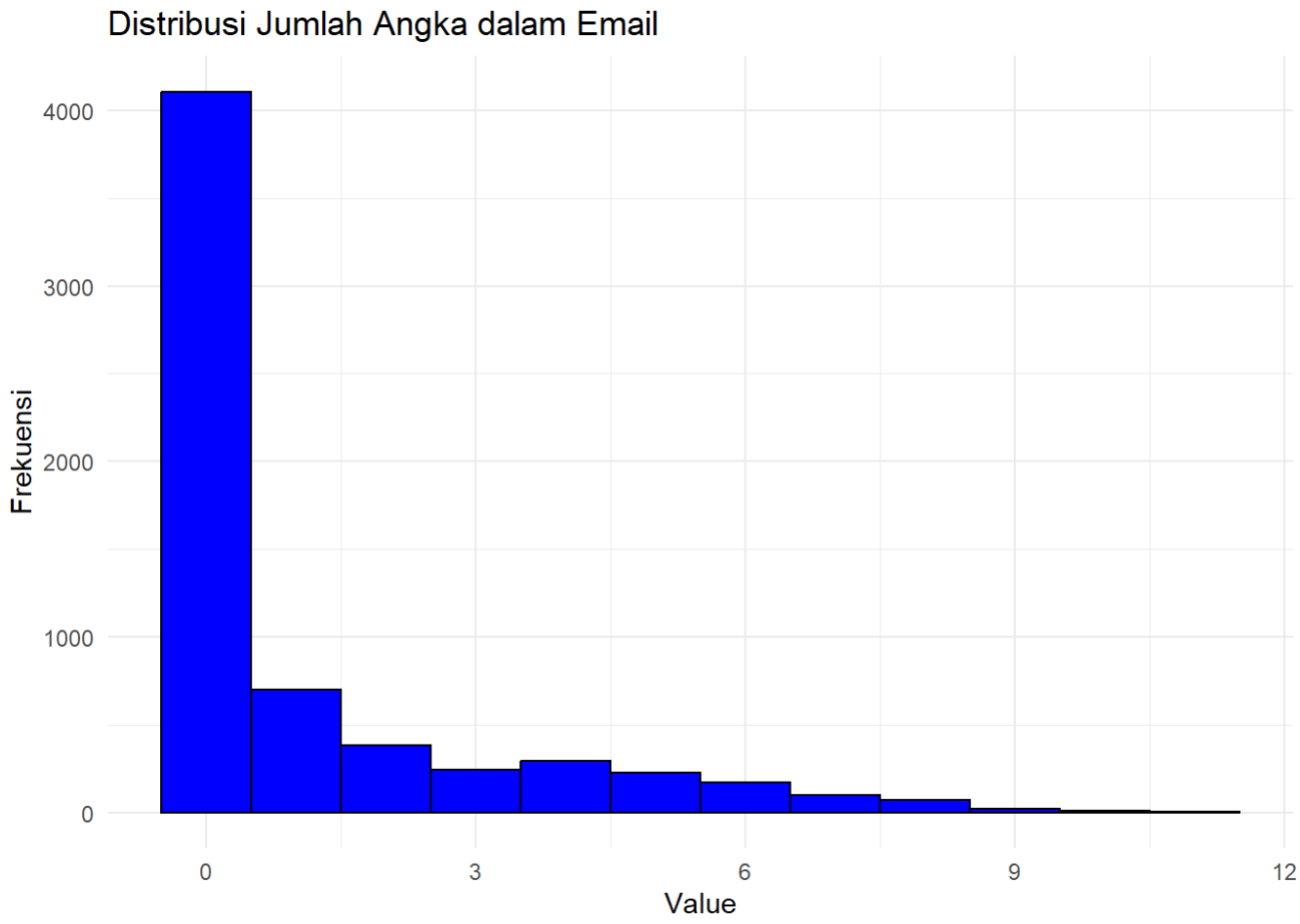



```
spam_mean <- mean(spam_count$word_count)
ham_mean <- mean(ham_count$word_count)
df <- data.frame(
  Category = c("Spam", "Ham"),
  Mean = c(spam_mean, ham_mean)
)
plot <- ggplot(df, aes(x = Category, y = Mean, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  labs(x = "Category", y = "Mean Word Count", title = "Rata-Rata Jumlah Kata") +
  theme_minimal()
print(plot)
```

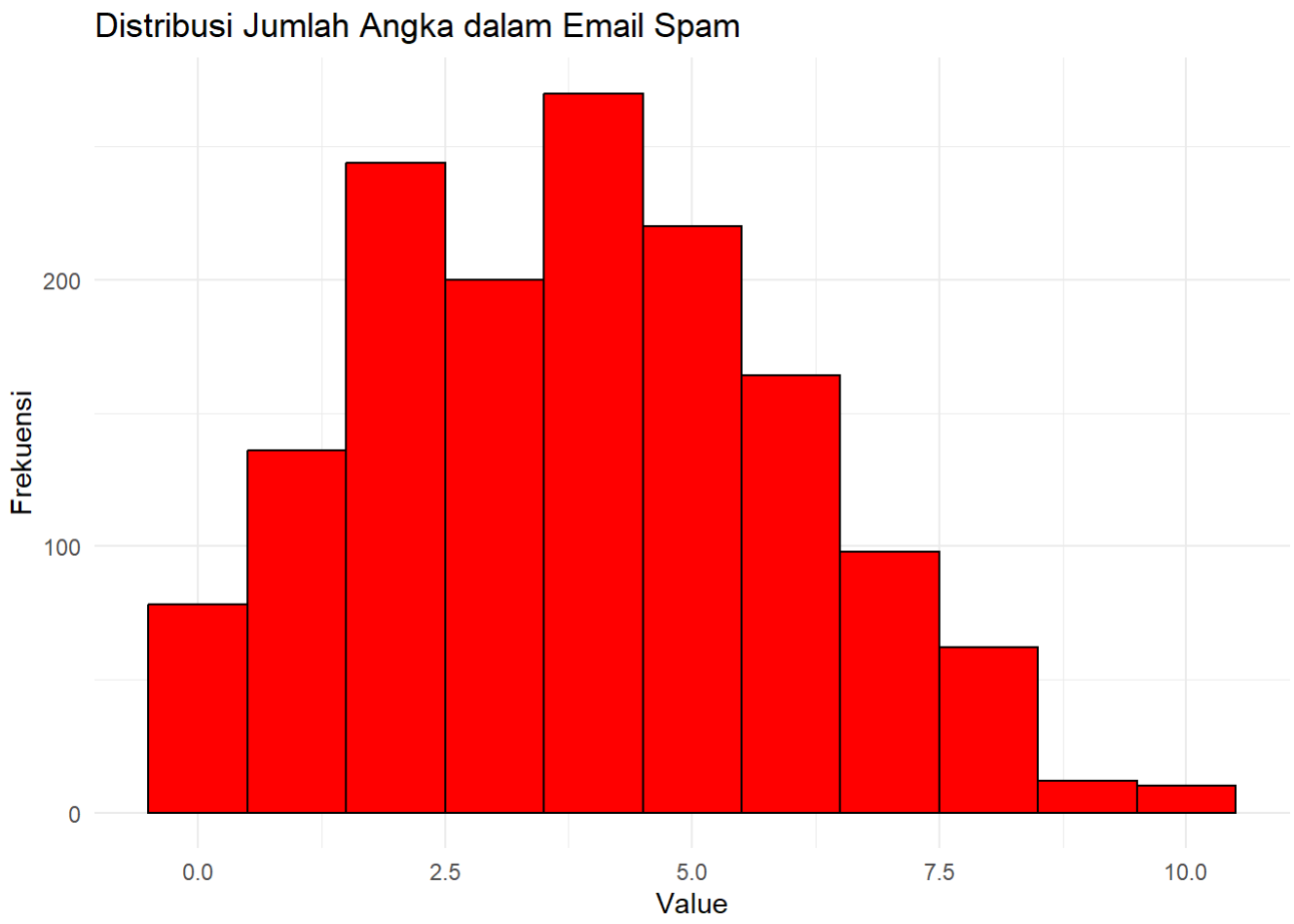


Angka dalam Pesan

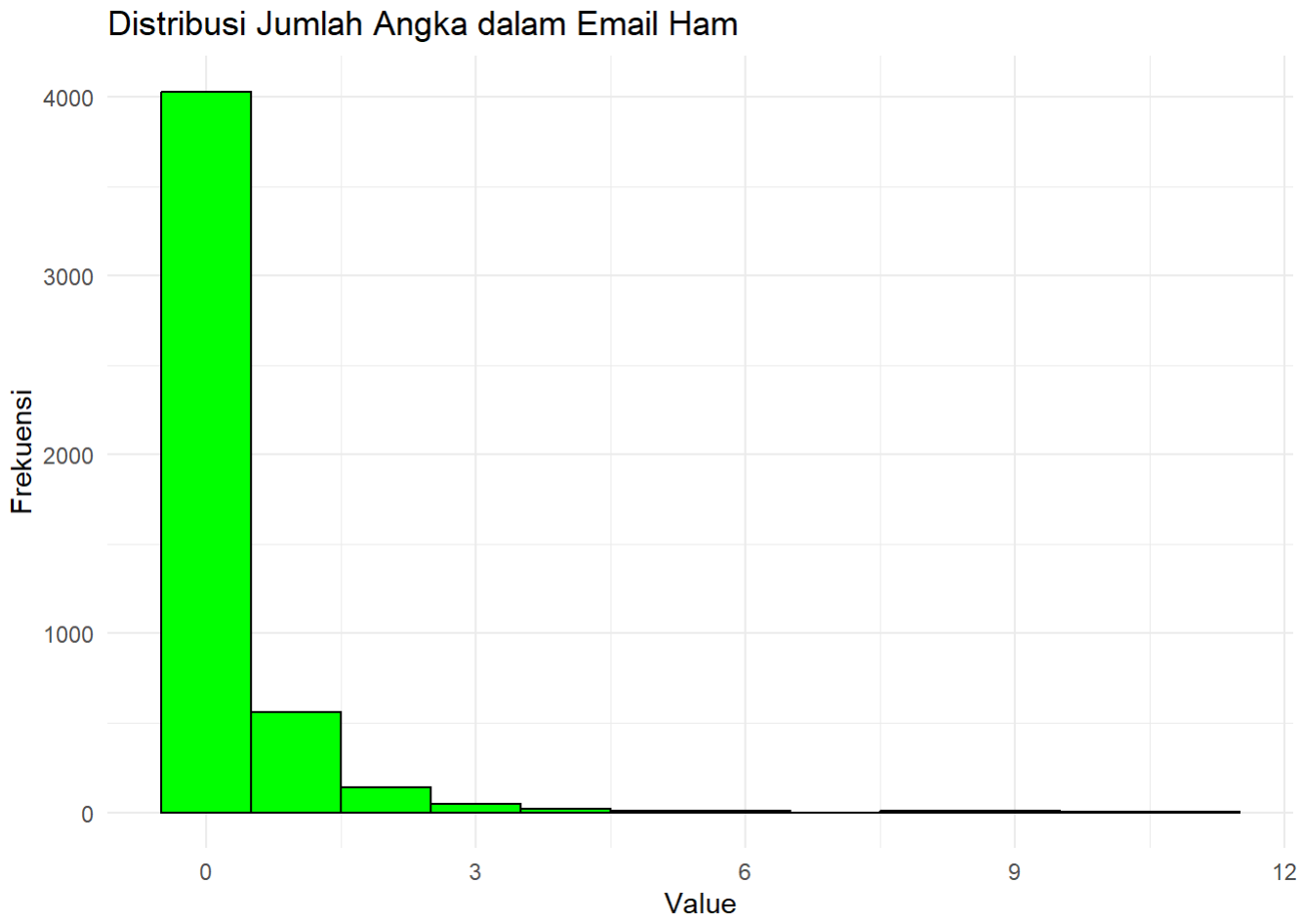
```
data$number_count <- sapply(strsplit(data$pesan, "\\s+"), function(words) sum(grepl("[0-9]", wor
ds)))

ggplot(data, aes(number_count)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Angka dalam Email") +
  theme_minimal()
```



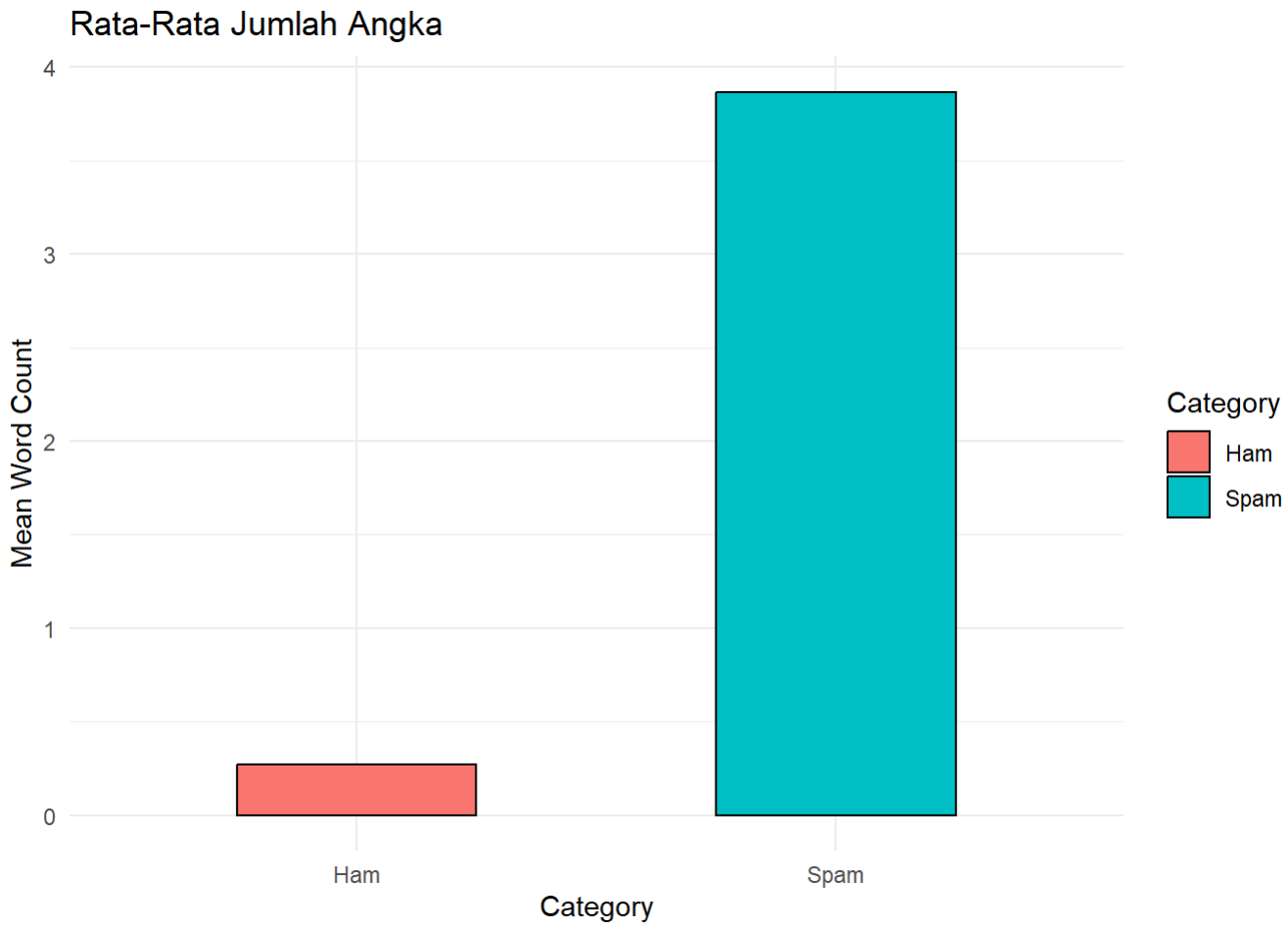

```
spam_count_a <- data[which((data$label == "spam") == TRUE),]
ham_count_a <- data[which((data$label == "ham") == TRUE),]

ggplot(spam_count_a, aes(number_count)) +
  geom_histogram(binwidth = 1, fill = "red", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Angka dalam Email Spam") +
  theme_minimal()
```




```
ggplot(ham_count_a, aes(number_count)) +
  geom_histogram(binwidth = 1, fill = "green", color = "black") +
  labs(x = "Value", y = "Frekuensi", title = "Distribusi Jumlah Angka dalam Email Ham") +
  theme_minimal()
```




```
spam_mean_a <- mean(spam_count_a$number_count)
ham_mean_a <- mean(ham_count_a$number_count)
df <- data.frame(
  Category = c("Spam", "Ham"),
  Mean = c(spam_mean_a, ham_mean_a)
)
plot <- ggplot(df, aes(x = Category, y = Mean, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  labs(x = "Category", y = "Mean Word Count", title = "Rata-Rata Jumlah Angka") +
  theme_minimal()
print(plot)
```




```
spam_mean_a
```

```
## [1] 3.86747
```

```
ham_mean_a
```

```
## [1] 0.267772
```

Naive Bayes

Split Data

```
set.seed(125)
trainid <- sample(1:nrow(data), nrow(data)*0.8)
training <- data[trainid,]
testing <- data[-trainid,]
```

Mencari Parameter Terbaik (laplace dan threshold)

```
laplace_values <- c(0, 0.5, 1)
thresholds_values <- c(0.3, 0.5, 0.7)

best_accuracy <- 0
best_laplace <- 0
best_threshold <- 0

for (laplace in laplace_values) {
  for (threshold in thresholds_values) {
    model <- naiveBayes(label ~ ., data = training, laplace = laplace)
    predict <- predict(model, testing, threshold = threshold)
    confusion_matrix <- table(predict, testing$label)
    accuracy <- mean(predict == testing$label)
    if (accuracy > best_accuracy) {
      best_accuracy <- accuracy
      best_laplace <- laplace
      best_threshold <- threshold
    }
  }
}

print(paste("Laplace:", best_laplace))
```

```
## [1] "Laplace: 0.5"
```

```
print(paste("Threshold:", best_threshold))
```

```
## [1] "Threshold: 0.3"
```

```
print(paste("Akurasi:", best_accuracy))
```

```
## [1] "Akurasi: 0.945411392405063"
```

Fit Model

```
model <- naiveBayes(label ~ ., data = training, laplace = best_laplace)
str(model)
```

```
## List of 5
##  $ apriori  : 'table' int [1:2(1d)] 3863 1192
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ Y: chr [1:2] "ham" "spam"
##  $ tables   :List of 3
##   ..$ pesan       : 'table' num [1:2, 1:4255] 0.000388 0.000419 0.000388 0.000419 0.000388
...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ Y    : chr [1:2] "ham" "spam"
##   .. .. ..$ pesan: chr [1:4255] " " "   and  picking them up from various points  going 2 yeo
vil  and they will do the motor project 4 3 hours  and "| __truncated__ " Am on a train back fro
m northampton so im afraid not" " and  picking them up from various points" ...
##   ..$ word_count  : num [1:2, 1:2] 14.11 23.69 11.24 5.68
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ Y         : chr [1:2] "ham" "spam"
##   .. .. ..$ word_count: NULL
##   ..$ number_count: num [1:2, 1:2] 0.276 3.885 0.854 2.16
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ Y           : chr [1:2] "ham" "spam"
##   .. .. ..$ number_count: NULL
##  $ levels   : chr [1:2] "ham" "spam"
##  $ isnumeric: Named logi [1:3] FALSE TRUE TRUE
##   ..- attr(*, "names")= chr [1:3] "pesan" "word_count" "number_count"
##  $ call     : language naiveBayes.default(x = X, y = Y, laplace = laplace)
##  - attr(*, "class")= chr "naiveBayes"
```

```
predictions <- predict(model, testing, threshold = best_threshold)
```

Evaluasi Model

```
confusion_matrix <- table(predictions, testing$label)
confusion_matrix
```

```
##
## predictions ham spam
##        ham  936   43
##        spam  26  259
```

```
precision <- confusion_matrix[2, 2] / sum(confusion_matrix[, 2])
print(paste("Presisi:", precision))
```

```
## [1] "Presisi: 0.857615894039735"
```

```
specificity <- confusion_matrix[1, 1] / sum(confusion_matrix[, 1])
print(paste("Spesifisitas:", specificity))
```

```
## [1] "Spesifisitas: 0.972972972972973"
```

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Akurasi:", accuracy))
```

```
## [1] "Akurasi: 0.945411392405063"
```

```
recall <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])
print(paste("Recall:", recall))
```

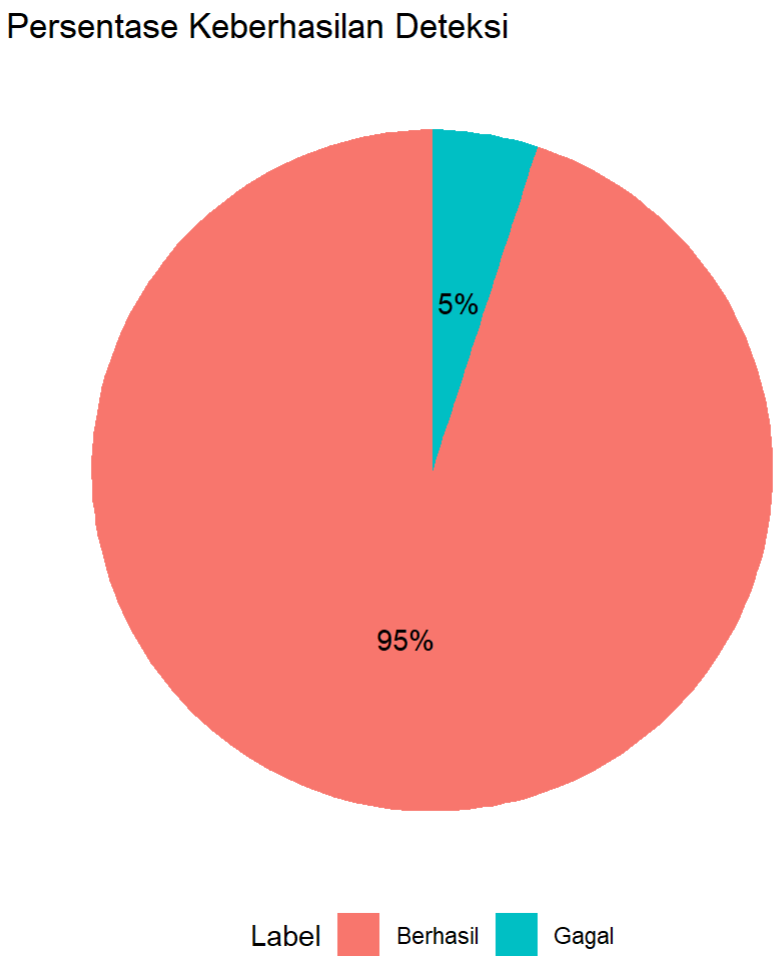
```
## [1] "Recall: 0.908771929824561"
```

```
f1_score <- 2 * (precision * recall) / (precision + recall)
print(paste("F1-Score:", f1_score))
```

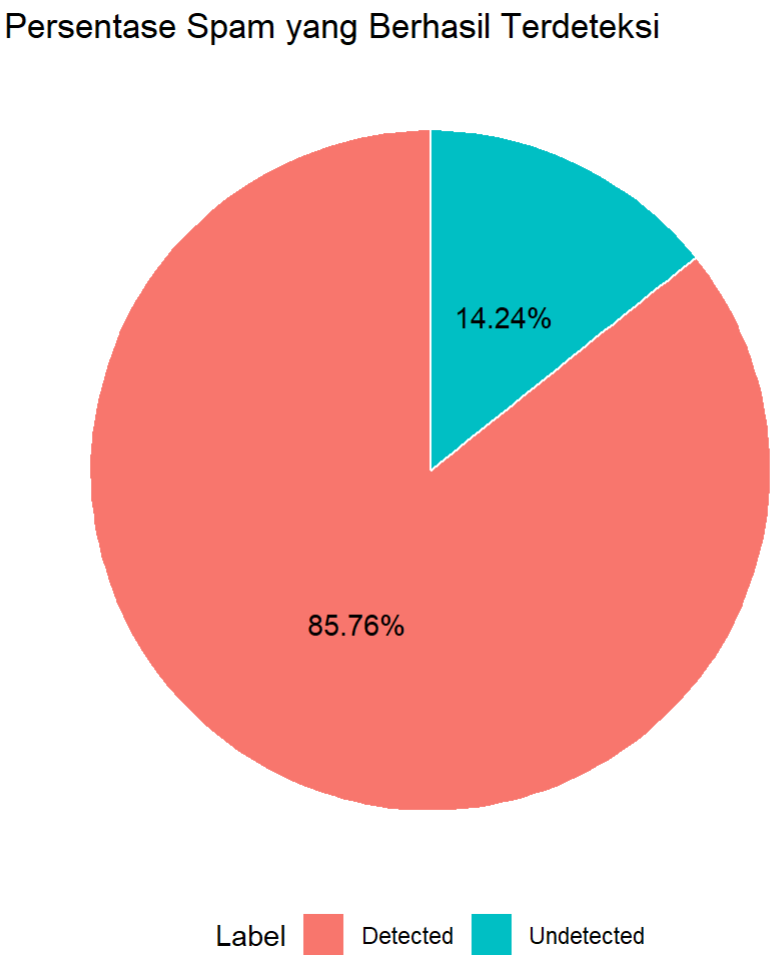
```
## [1] "F1-Score: 0.882453151618399"
```

Hasil Prediksi

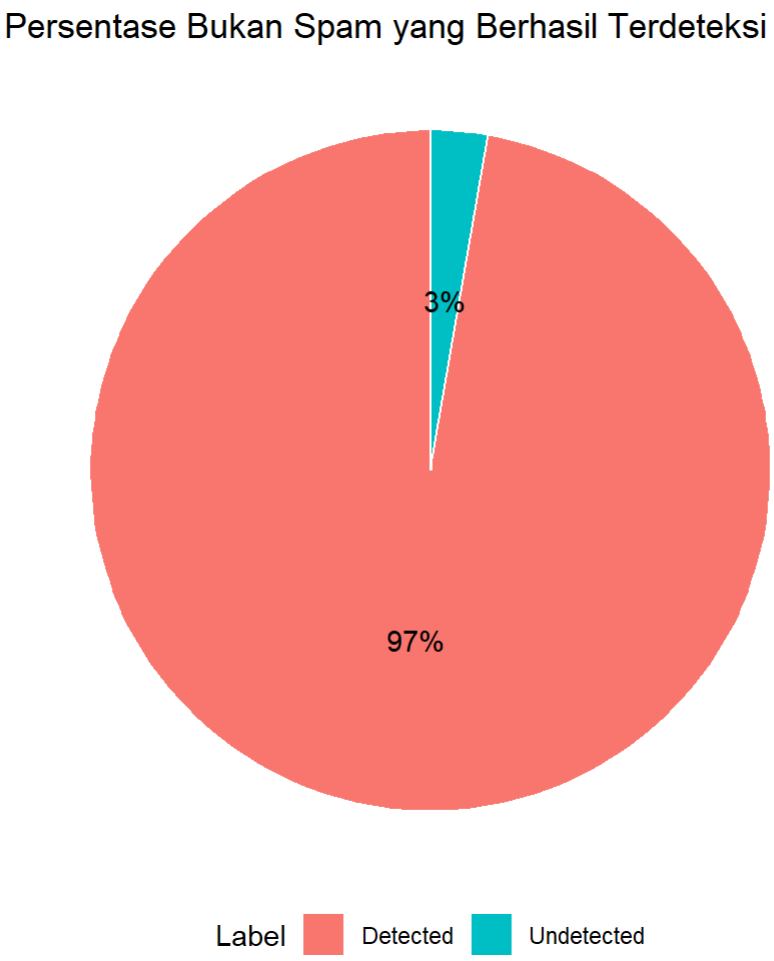
```
akurasi <- round(accuracy * 100)
error <- round(100 - akurasi)
plot_data <- data.frame(Category = c("Berhasil", "Gagal"), Percentage = c(akurasi, error))

ggplot(plot_data, aes(x = "", y = Percentage, fill = Category)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  geom_text(aes(label = paste0(Percentage, "%")), position = position_stack(vjust = 0.5)) +
  labs(title = "Persentase Keberhasilan Deteksi",
       fill = "Label",
       x = NULL,
       y = NULL) +
  theme_void() +
  theme(legend.position = "bottom")
```




```
spam_detected <- confusion_matrix["spam", "spam"]
total_spam <- sum(testing$label == "spam")
percentage_detected <- spam_detected / total_spam * 100
plot_data <- data.frame(Label = c("Detected", "Undetected"),
                        Percentage = c(percentage_detected, 100 - percentage_detected))
p <- ggplot(plot_data, aes(x = "", y = Percentage, fill = Label)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  labs(title = "Persentase Spam yang Berhasil Terdeteksi",
       x = NULL,
       y = NULL) +
  theme_void() +
  theme(legend.position = "bottom")
p + geom_text(aes(label = paste0(round(Percentage, 2), "%")), position = position_stack(vjust =
0.5))
```




```
ham_detected <- confusion_matrix["ham", "ham"]
total_ham <- sum(testing$label == "ham")
percentage_detected <- ham_detected / total_ham * 100
plot_data <- data.frame(Label = c("Detected", "Undetected"),
                        Percentage = c(percentage_detected, 100 - percentage_detected))
p <- ggplot(plot_data, aes(x = "", y = Percentage, fill = Label)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  labs(title = "Persentase Bukan Spam yang Berhasil Terdeteksi",
       x = NULL,
       y = NULL) +
  theme_void() +
  theme(legend.position = "bottom")
p + geom_text(aes(label = paste0(round(Percentage), "%")), position = position_stack(vjust = 0.
5))
```


Persentase Bukan Spam yang Berhasil Terdeteksi
3%
97%
Label
Detected
Undetected