# **LAPORAN PROYEK AKHIR**

# **PRAKTIKUM DATA SCIENCE**

# **Analisis Sentimen Pada Data Twitter Mengenai Opini Terhadap Aplikasi**

# **Peduli Lindungi Dengan Metode Naive Bayes**

|  |  |
| --- | --- |
|  |  |
|  |  |

**DISUSUN OLEH:**

FARRAS ALAM MAJID 123200044

HUGO VALE POETRATAMA 123200116

**PROGRAM STUDI INFORMATIKA**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” YOGYAKARTA**

**2022**

**HALAMAN PENGESAHAN LAPORAN PROYEK AKHIR**



Disusun oleh :

|  |  |
| --- | --- |
| *Farras Alam Majid* | *123200044* |
| *Hugo Vale Poetratama* | *123200116* |

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Data Science

Pada Tanggal : ………………………….

|  |  |  |
| --- | --- | --- |
| **Asisten Praktikum**  **Dhea Anggita**  123190046 |  | **Asisten Praktikum**  **Rico Aminanda**  123190079 |

**DAFTAR ISI**

COVER 1

HALAMAN PENGESAHAN 2

DAFTAR ISI 3

KATA PENGANTAR 5

1. Pendahuluan 6

2. Metode 6

2.1. Analisis Sentimen 6

2.2. Naïve Bayes 6

2.3. Data Collection 7

2.4. Data Preprocessing 8

2.4.1. Case Folding 8

2.4.2. Data cleaning 8

2.4.3. Filtering 8

2.5. Data Classification 8

2.4.1. Labeling 9

2.4.2. Transformasi 9

2.4.3. Modeling 9

3. Hasil dan Pembahasan 10

3.1. Data Collection 10

3.2. Data Preprocessing 10

3.3. Data Classification 11

3.4. Uji Naive Bayes dari Data 11

3.5. Visualisasi dengan Shiny 12

3.6. Listing Program 13

4. Kesimpulan…………………………………...………………………… 20

**KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat proyek akhir menyelesaikan praktikum Praktikum Data Science serta laporan proyek akhir praktikum data science yang berjudul “Analisis Sentimen Pada Data Twitter Mengenai Opini Terhadap Aplikasi Peduli Lindungi Dengan Metode Naive Bayes”. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten laboratorium yang selalu  membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam  menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan  akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan  terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 30 November 2022

Penyusun

1. **PENDAHULUAN**

Saat ini dunia sedang menghadapi pandemi Corona Virus atau yang disebut juga COVID-19. Penyebarannya sangat cepat dan sulit di deteksi. Setelah terjadi di kota Wuhan, China pada awal tahun lalu, kini COVID-19 sudah menyebar ke berbagai negara, salah satunya adalah Indonesia. Sejak awal terdeteksinya COVID-19 di Indonesia pada bulan Februari 2020 di Depok, pasien positif COVID-19 semakin bertambah dan meluas daerah penyebarannya.

Indonesia sebagai negara berkembang dengan bentuk kepulauan menjadikan tidak dapatnya dilakukan *lockdown* seperti yang sudah dilakukan di Wuhan dan beberapa negara lain yang telah terpapar. Dengan ini pemerintah bersama seluruh rakyat Indonesia gotong-royong saling membantu dan mencari solusi untuk mengurangi bahkan menghentikan angka pasien positif COVID-19, salah satunya ialah lewat program aplikasi “PeduliLindungi” yang di *publish* oleh Pusdatin Kementrian Kesehatan Republik Indonesia.

PeduliLindungi adalah aplikasi dari Kementerian Komunikasi dan Informatika yang dikembangkan untuk menghentikan penyebaran COVID-19 dengan mengandalkan partisipasi masyarakat untuk saling membagikan data lokasinya saat bepergian agar penelurusan riwayat kontak dengan penderita COVID-19 dapat dilakukan. PeduliLindungi menggunakan data yang diproduksi oleh gadget pengguna dengan bluetooth aktif untuk merekam informasi yang dibutuhkan. Ketika ada gadget lain dalam radius bluetooth yang juga terdaftar di PeduliLindungi, maka akan terjadi pertukaran id anonim yang akan direkam oleh gadget masing-masing. Oleh karena itu, dengan adanya penelitian ini, diharapkan dapat memberikan gambaran mengenai dampak positif dan negatif, serta netral berdasarkan hasil *review* atau opini mahasiswa.

1. **METODE**

Dalam penelitian proyek ini kami menggunakan teknik *data mining* Naïve Bayes. Penelitian yang dilakukan pada kali ini menghasilkan analisis sentiment dengan menerapkan Algoritma Naïve Bayes untuk menentukan kategori dari komentar positif dan negatif berdasarkan data yang telah didapatkan.

2.1 Analisis Sentiment

Analisis sentiment atau yang biasa dikenal sebagai *opinion mining*  atau *emotion artificial intelligence* merupakan teknik pengolahan bahasa alami, komputasi linguistik, dan text mining yang berguna untuk mendeteksi suatu subyek seperti pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu.

2.2 Naïve Bayes

Naive Bayes adalah metode yang cocok untuk klasifikasi biner dan multiclass. Metode yang juga dikenal sebagai Naive Bayes Classifier ini menerapkan teknik supervised klasifikasi objek di masa depan dengan menetapkan label kelas ke *instance* atau catatan menggunakan probabilitas bersyarat. Probabilitas bersyarat adalah ukuran peluang suatu peristiwa yang terjadi berdasarkan peristiwa lain yang telah (dengan asumsi, praduga, pernyataan, atau terbukti) terjadi.

Nah, model *machine learning* yang diterapkan pada program tersebut menggunakan teorema Bayes yang dirumuskan sebagai berikut:

**P(A│B) = P(B│A)P(A)P(B)**

Keterangan:

P(A│B) : Probabilitas A terjadi dengan bukti bahwa B telah terjadi (probabilitas superior)

P(B│A) : Probabilitas B terjadi dengan bukti bahwa A telah terjadi

P(A) : Peluang terjadinya A

P(B) : Peluang terjadinya B

2.3 Data Collection

Data Collection merupakan proses mengumpulkan, mengukur, dan menganalisis data yang akurat dari berbagai sumber yang relevan untuk menemukan jawaban atas masalah penelitian, menjawab pertanyaan, mengevaluasi hasil, dan memperkirakan tren dan probabilitas. Data Collection yang dilakukan pada project akhir ini dilakukan melalui proses scrapping yang diambil sosial media Twitter. Tahap pertama yang di lakukan pada proses data collection adalah mengambil tweet menggunakan tweet API yang telah disediakan oleh Twitter Developer. Untuk mendapatkan tweet tersebut, diperlukannya user membuat App pada Twitter Developer untuk mendapatkan *key*, *secret*, a*cces token*, dan *access secret* sehingga user dapat membuat token API. Tahap selanjutnya adalah pengambilan tweet berdasarkan kata kunci yang dicari. Karena dalam projek ini yang akan dibahas adalah Peduli Lindungi. Maka, kata kunci yang digunakan adalah “Peduli Lindungi” sebagai kata kunci, dan menambahkan lang = “id” untuk mendapatkan tweet berbahasa Indonesia.

2.4 Data Preprocessing

Data Preprocessing merupakan proses untuk mendapatkan data text atau ulasan yang bersih yang melakukan beberapa tahapan didalamnya, yaitu :

2.4.1 Case Folding merupakan proses yang diterapkan pada urutan karakter teks pada suatu dokumen dengan merubah bentuk standar huruf besar-kecil dengan mengurangi semua huruf menjadi huruf kecil atau *lowercase.*

2.4.2 Data cleaning merupakan proses menyiapkan data untuk menganalisis data lebih lanjut yang berguna untuk menghasilkan data dengan kualitas yang tinggi. Data cleaning dilakukan dengan cara menghapus atau memodifikasi data yang salah, tidak relevan, duplikat, dan tidak berformat dengan melakukan proses penghilangan komponen tertentu yang terdapat dalam data tweet yang telah didapatkan yakni *Uniform Resource Locator* (URL), *username*, RT (*Retweet*), karakter HTML, *emoticon*, hashtag, serta tanda baca dan kata-kata yang tidak berguna.

2.4.3 Filtering merupakan tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma stoplist (*stopword*) untuk membuang kata kurang penting atau wordlist (menyimpan kata penting) dengan dilakukannya penghapusan kata (*removewords*).

2.5 Data Classification

Data *Classification* merupakan salah satu contoh dari metode analisis data. *Classification* adalah teknik di mana kita bisa mengkategorikan data ke dalam sejumlah kelas yang telah ditentukan sebelumnya. Tujuan utama dari klasifikasi adalah untuk membantu praktisi data dalam menentukan kelas atau kategori dari data baru berdasarkan karakteristik data yang telah ada sebelumnya. Klasifikasi tidak hanya bisa dilakukan di data yang terstruktur, namun juga bisa digunakan untuk data yang tidak terstruktur.

Classification juga merupakan salah satu algoritma Machine Learning yang bersifat Supervised Learning, dimana data akan dikelompokkan berdasarkan label yang ada dengan melalui beberapa proses, diantaranya adalah :

2.5.1 Labeling merupakan pemberian label pada data. Proses labeling dilakukan menggunakan metode Lexical Based yang merupakan proses labeling berdasarkan kata positif, kata negatif, maupun kata netral yang terdapat pada data tweets yang telah dilakukan cleaning. Klasifikasi labeling ini dilakukan dengan pencocokkan dengan kata-kata yang terdapat dalam kamus Lexicon. Dimana, jika data tweets memiliki kata positif, maka akan digolongkan pada sentimen tweet positif. Sedangkan, jika terdapat data pada tweets yang memiliki kata negatif, maka data tersebut akan digolongkan pada sentimen tweet negatif. Namun pada kasus lain, jika kedua kata ini bernilai sama, maka digolongkan dalam tweet netral.

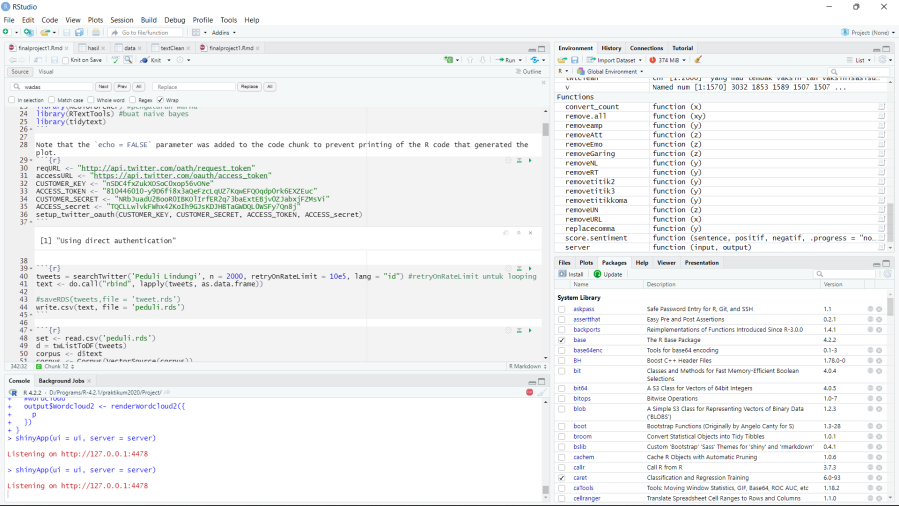
2.5.2 Transformasi merupakan proses pelabelan pada teks tweets yang akan diubah menjadi bentuk yang dapat digunakan oleh komputer. Pengubahan bentuk tersebut dikenal dengan vektorisasi. Vektorisasi merupakan proses membuat vektor dengan nilainya berupa angka-angka kemunculan term (kata unik) dalam dokumen. Teks ini akan diubah perkata. Pemisahan teks tweets menjadi kata-kata disebut dengan tokenisasi. Tokenisasi merupakan proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token/bagian-bagian tertentu. Dalam hal ini komponen individual tersebut yaitu token adalah kata dari tiap teks tweets. *Project* ini token-token dibuat dalam bentuk dokumen term matrix.

2.5.3 Modeling merupakan tahap pengklasifikasian untuk menentukan apakah data yang akan diuji termasuk kedalam sentimen positif atau negatif. Pada tahap ini menggunakan metode Naïve Bayes Classifier dengan pelabelan terlebih dahulu. Metode ini terdiri dari 2 proses, yaitu setelah data latih dan data uji dibuat maka dilanjutkan ke proses latih. Pada proses latih diimplementasikan algoritma pengklasifikasi Naïve Bayes. Algoritma pengklasifikasi Naïve Bayes akan menggunakan kemunculan tiap kata unik dalam teks tweets untuk menghitung pembuatan set data latih dan set data uji, melakukan data latih yaitu dengan membangun model pengklasifikasi menggunakan algoritma Naïve Bayes, dan diakhiri dengan klasifikasi yang mengimplementasikan model pengklasifikasi pada set data uji teks tweets.

1. **HASIL DAN PEMBAHASAN**

**3.1 Data Collection**

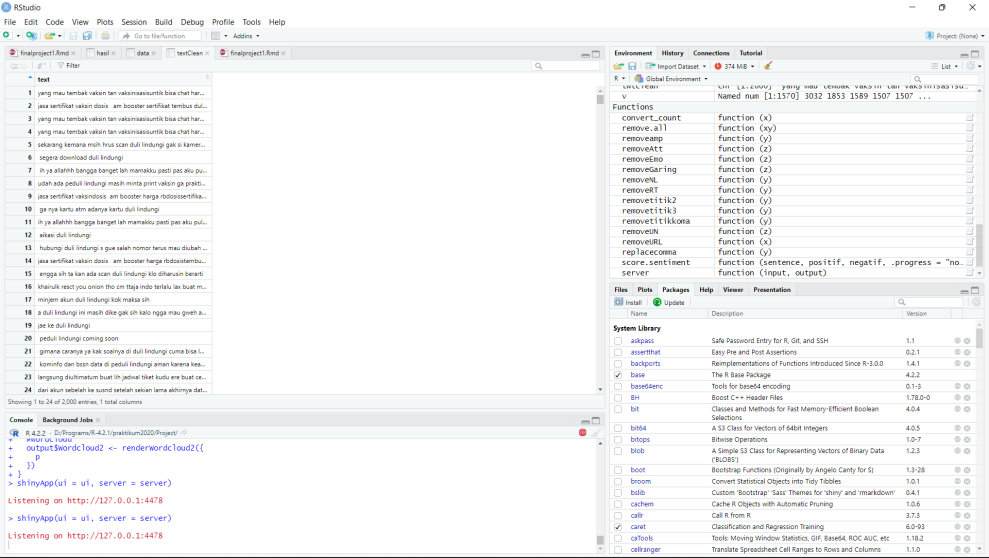
Gambar dibawah ini adalah proses *scrapping* data mengenai Peduli Lindungi menggunakan sosial media Twitter. Proses *scrapping* ini didapatkan data sejumlah 2000 data.



**Gambar 3.1** Proses *Scrapping.*

**3.2 Data Preprocessing**

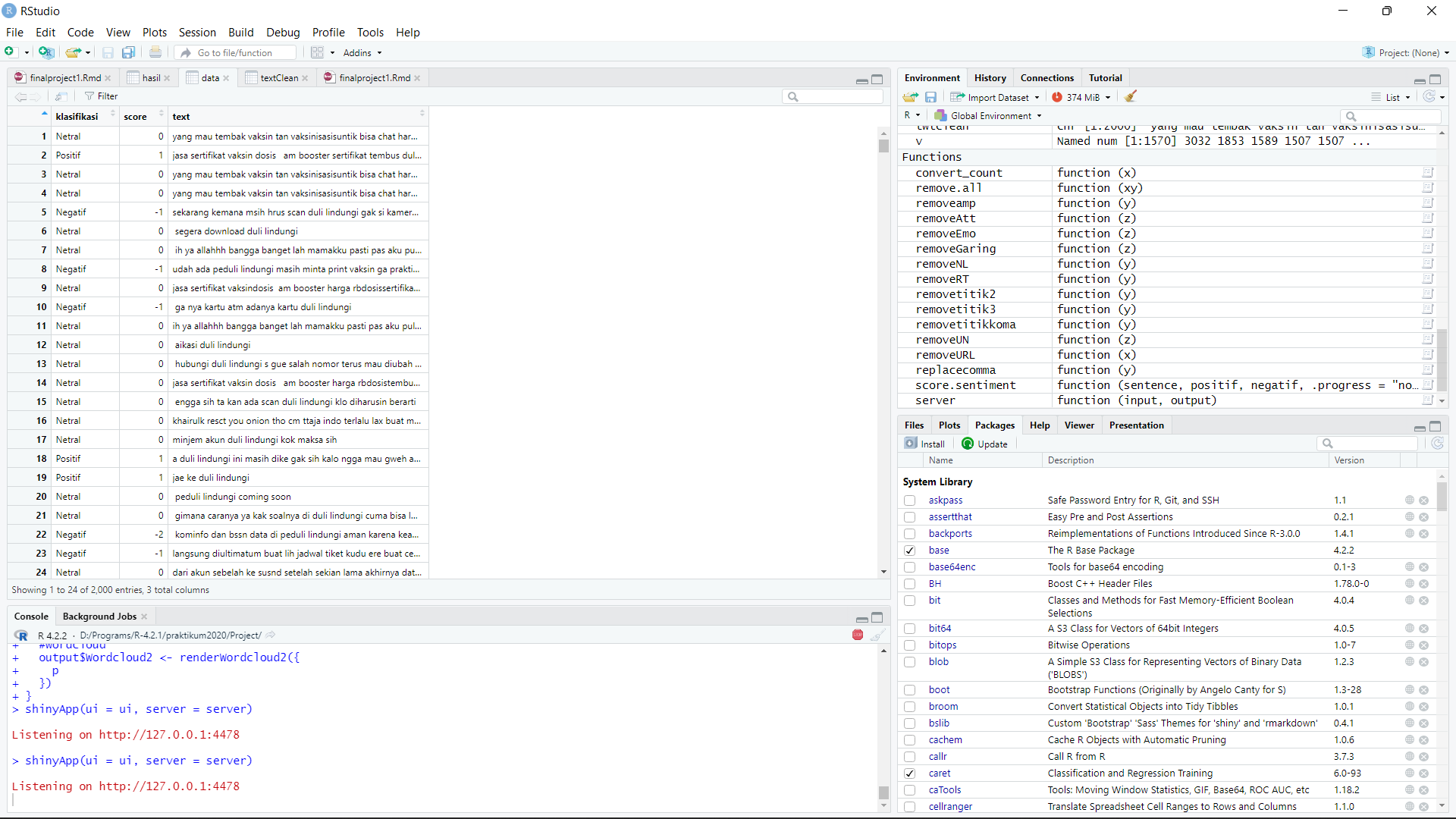
Dalam proses preprocessing ini, dilakukan *case folding* untuk mengurutkan karakter, lalu dilanjutkan dengan cleaning dengan menghilangkan komponen yang tidak dipakai dan difiltering untuk mengambil kata-kata penting.

****

**Gambar 3.2** *Preprocessing* data.

**3.3 Data Classification**

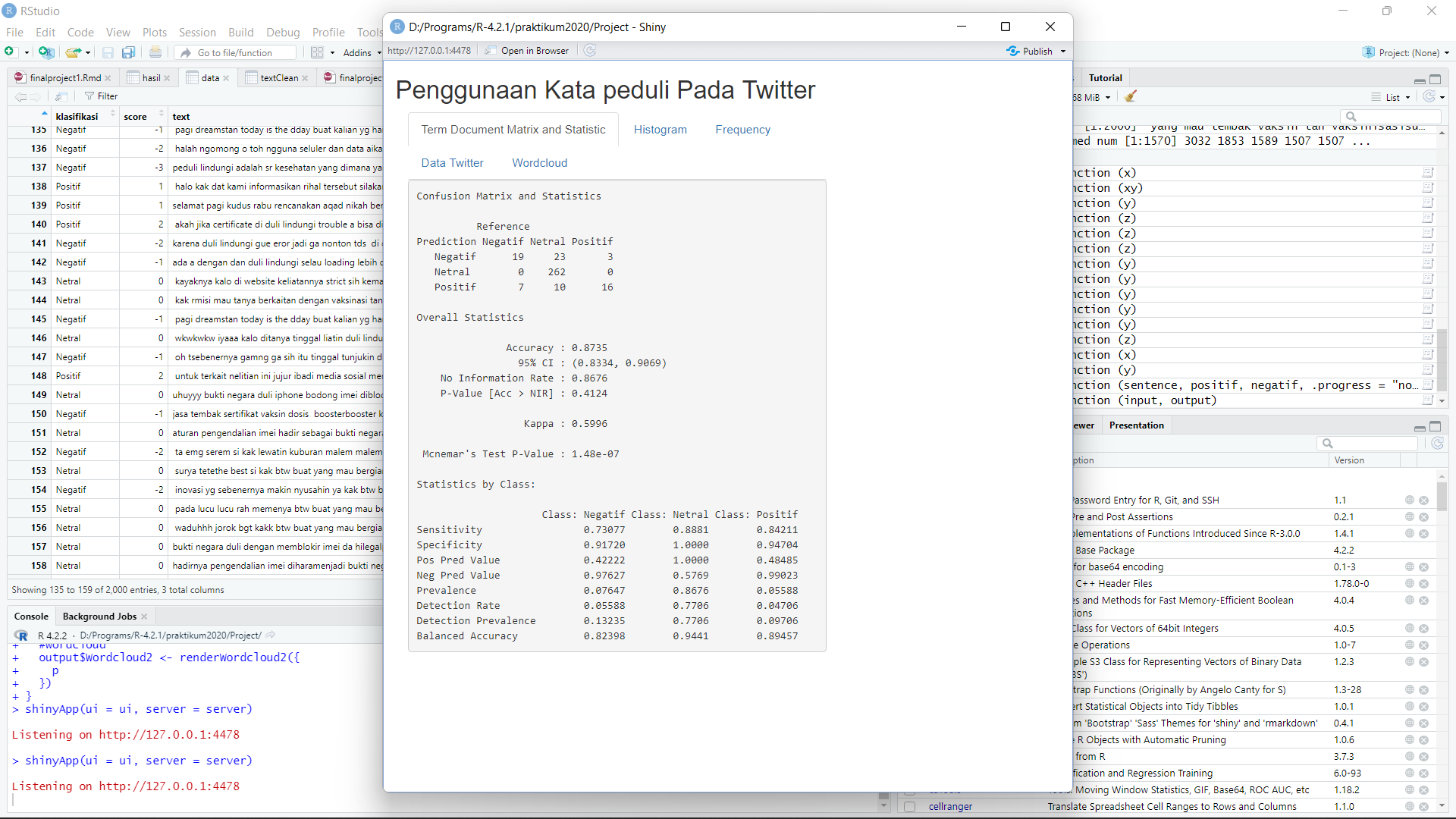
Dalam tahap ini ada labeling untuk pemberina label data, tranformasi untuk mengubah data menjadi vektor dan modelling untuk menguji kedalam metode Naive Bayes.

****

**Gambar 3.3** *Clasification* data.

**3.4 Uji Naive Bayes dari Data**

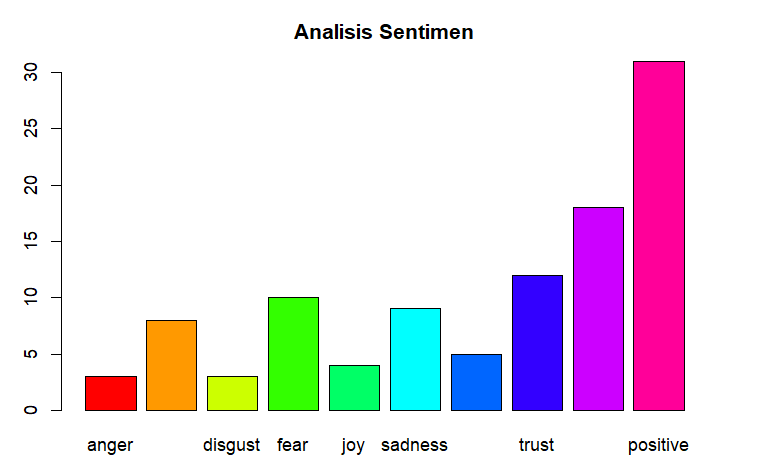
Dalam tahap ini, 2000 data yang sudah ada akan diuji menggunakan metode Naive Bayes.



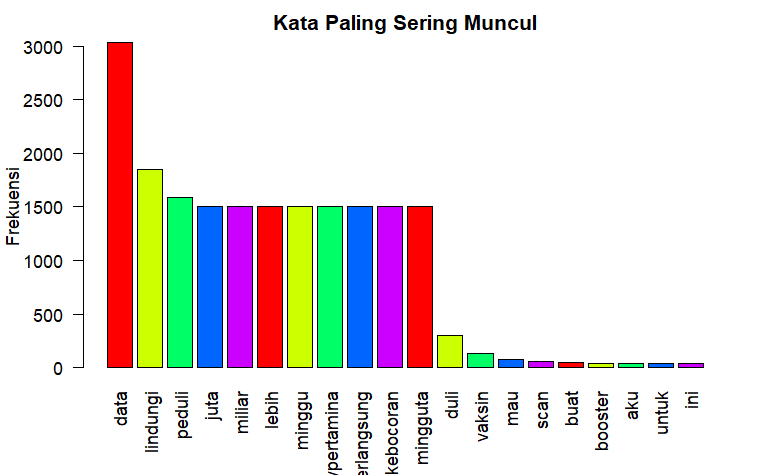
**Gambar 3.4** Pengujian dengan metode Naive Bayes data.

**3.5 Visualisasi dengan Shiny**

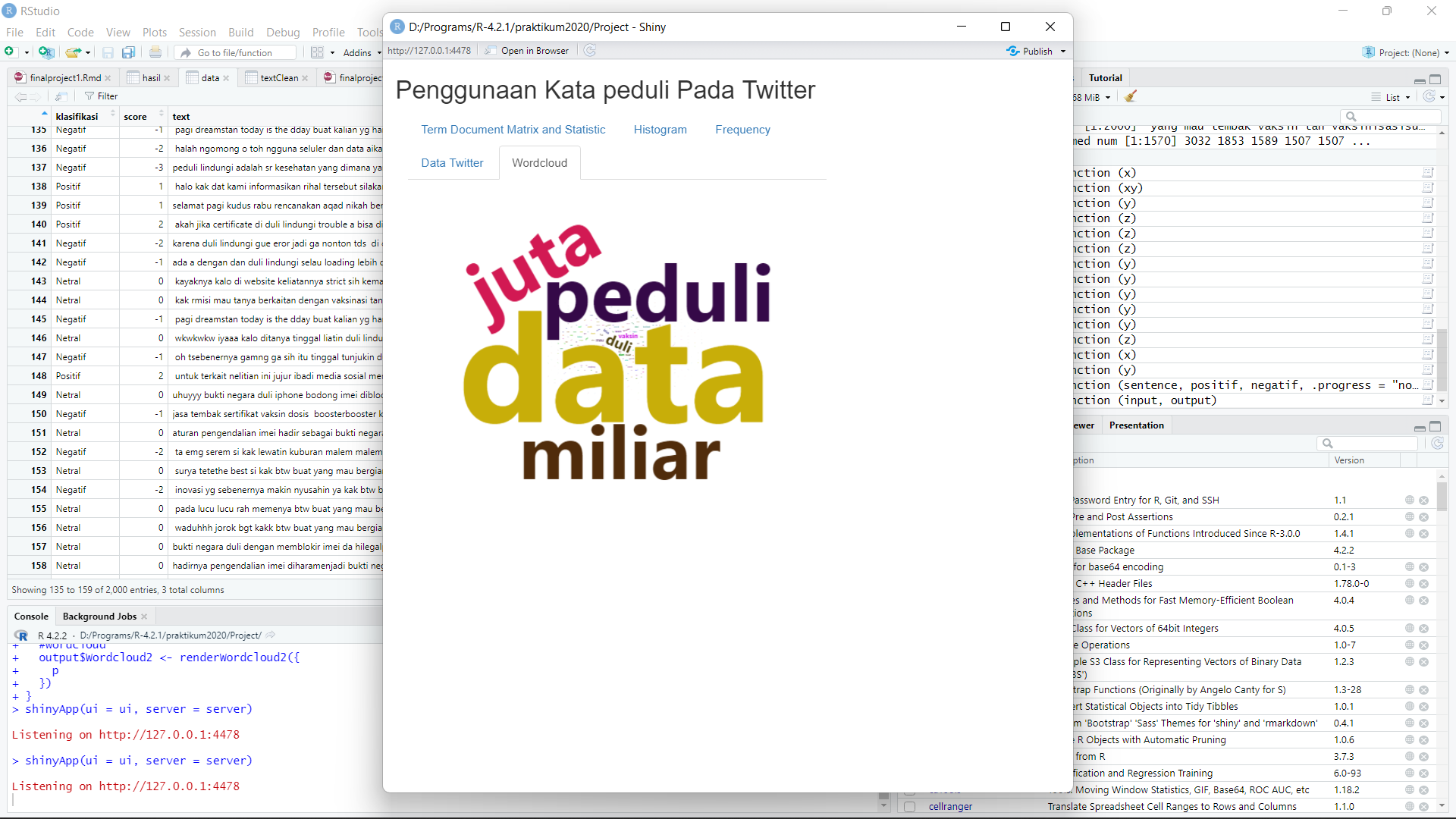
Kita menggunakan 2 jenis visualisasi yaitu dengan Barplot untuk sentimen dan kata yang paling sering muncul dan wordcloud.

****

**Gambar 3.5** Barplot Analisis Sentimen.



**Gambar 3.6** *Frequency* Plot.



**Gambar 3.7** *Wordcloud.*

**3.6 Listing Program**

1. Mengimpor library yang diperlukan

|  |
| --- |
| ```{r}  library(tm) #data cleaning (corpus)  library(twitteR) #akses twitter APIs  library(rtweet) #collect and organize twitter data  library(shiny) #shiny  library(syuzhet) #baca fungsi get\_nrc  library(wordcloud) #wordcloud  library(vroom) #load dataset  library(here) #menyimpan dataset  library(dplyr) #manipulasi data frame  library(ggplot2) #visualisasi data (barplot, grafik)  library(RColorBrewer) #pengaturan warna  library(RTextTools) #buat naive bayes  library(tidytext)  ``` |

**Listing 3.1** *Import Library.*

2. Autentikasi API

|  |
| --- |
| ```{r}  reqURL <- "http://api.twitter.com/oath/request\_token"  accessURL <- "https://api.twitter.com/oauth/access\_token"  CUSTOMER\_KEY <- "nSDC4fxZukXOSoC0xop56vONe"  ACCESS\_TOKEN <- "810446010-y9D6fi8x3aQeFzcLqUZ7KqwEFQOqdp0rk6EXZEuc"  CUSTOMER\_SECRET <- "NRbJuadU2BooR0IBKOlIrfER2q73baExtEBjv0ZJabxjFZMsVi"  ACCESS\_secret <- "TQCLLwlvkFWhx42KoIh9GJsKDJHBTaGWDQL0WSFy7Qn8j"  setup\_twitter\_oauth(CUSTOMER\_KEY, CUSTOMER\_SECRET, ACCESS\_TOKEN, ACCESS\_secret)  ``` |

**Listing 3.2** Autentikasi API.

3. *Scrapping* Data

|  |
| --- |
| ```{r}  tweets = searchTwitter('Peduli Lindungi', n = 2000, retryOnRateLimit = 10e5, lang = "id") #retryOnRateLimit untuk looping  text <- do.call("rbind", lapply(tweets, as.data.frame))  #saveRDS(tweets,file = 'tweet.rds')  write.csv(text, file = 'peduli.rds')  ``` |

**Listing 3.3** *Scrapping* Data Twitter.

4. Data Cleaning

|  |
| --- |
| ```{r}  set <- read.csv('peduli.rds')  d = twListToDF(tweets)  corpus <- d$text  corpus <- Corpus(VectorSource(corpus))  #hapus URL  removeURL <- function(x) gsub("https\\S\*", "", x)  twtclean <- tm\_map(corpus, removeURL)  #hapus New Line  removeNL <- function(y) gsub("\n", "", y)  twtclean <- tm\_map(twtclean, removeNL)  #hapus koma  replacecomma <- function(y) gsub(",", "", y)  twtclean <- tm\_map(twtclean, replacecomma)  #hapus retweet  removeRT <- function(y) gsub("^RT:?", "", y)  twtclean <- tm\_map(twtclean, removeRT)  #hapus titik  removetitik2 <- function(y) gsub(":", "", y)  twtclean <- tm\_map(twtclean, removetitik2)  #hapus titik koma  removetitikkoma <- function(y) gsub(";", " ", y)  twtclean <- tm\_map(twtclean, removetitikkoma)  #hapus titik3  removetitik3 <- function(y) gsub("p.", "", y)  twtclean <- tm\_map(twtclean, removetitik3)  #hapus &amp  removeamp <- function(y) gsub("&amp;", "", y)  twtclean <- tm\_map(twtclean, removeamp)  #hapus Mention  removeUN <- function(z) gsub("@[A-Za-z0-9]+", "", z)  twtclean <- tm\_map(twtclean, removeUN)  #hapus Emoji  removeEmo <- function(z) gsub("[^\x01-\x7F]", "", z)  twtclean <- tm\_map(twtclean, removeEmo)  #hapus garing  removeGaring <- function(z) gsub("/", "", z)  twtclean <- tm\_map(twtclean, removeGaring)  #hapus att  removeAtt <- function(z) gsub("@", "", z)  twtclean <- tm\_map(twtclean, removeGaring)  #hapus space dll  remove.all <- function(xy) gsub("[^[:alpha:][:space:]]\*", "", xy)  twtclean <-tm\_map(twtclean,stripWhitespace)  inspect(twtclean[1:10])  twtclean <- tm\_map(twtclean,remove.all)  twtclean <- tm\_map(twtclean, removePunctuation) #tanda baca  twtclean <- tm\_map(twtclean, tolower) #mengubah huruf kecil  #hapus data yang kosong  try.error = function(x)  {  # create missing value  y = NA  # tryCatch error  try\_error = tryCatch(tolower(x), error=function(e) e)  # if not an error  if (!inherits(try\_error, "error"))  y = tolower(x)  # result  return(y)  }  # lower case using try and error with sapply  twtclean = sapply(twtclean, try.error)  # remove NAs in some\_txt  twtclean = twtclean[!is.na(twtclean)]  names(twtclean) = NULL  write.csv(twtclean, file = "peduliClean.csv")  ``` |

**Listing 3.4** Data *Cleaning*.

5. Scoring

|  |
| --- |
| ```{r}  library(plyr)  peduliClean <- read.csv("peduliClean.csv", header=T)  #skoring  kata.positif <- scan("kata-pos.txt",what="character",comment.char=";")  kata.negatif <- scan("kata-neg.txt",what="character",comment.char=";")  score.sentiment = function(sentence, positif, negatif,  .progress='none')  {  require(plyr)  require(stringr)  scores = laply(sentence, function(kalimat, positif,  negatif) {  kalimat = gsub('[[:punct:]]', '', kalimat)  kalimat = gsub('[[:cntrl:]]', '', kalimat)  kalimat = gsub('\\d+', '', kalimat)  kalimat = tolower(kalimat)  list.kata = str\_split(kalimat, '\\s+')  kata2 = unlist(list.kata)  positif.matches = match(kata2, kata.positif)  negatif.matches = match(kata2, kata.negatif)  positif.matches = !is.na(positif.matches)  negatif.matches = !is.na(negatif.matches)  score = sum(positif.matches) - (sum(negatif.matches))  return(score)  }, kata.positif, kata.negatif, .progress=.progress )    scores.df = data.frame(score=scores, text=sentence)  return(scores.df)}  hasil = score.sentiment(peduliClean$x, kata.positif, kata.negatif)  #CONVERT SCORE TO SENTIMENT  hasil$klasifikasi<- ifelse(hasil$score<0, "Negatif",ifelse(hasil$score==0,"Netral","Positif"))  hasil$klasifikasi  View(hasil)  #EXCHANGE ROW SEQUENCE  data <- hasil[c(3,1,2)] #ubah urutan kolom  View(data)  write.csv(data, file = "peduliLabel.csv")  ``` |

**Listing 3.5** Scoring.

6. Analisis Sentiment dan perhitungan *Confusion Matrix* dan *Statistics*

|  |
| --- |
| ```{r}  require(corpus)  data.frame <- read.csv("peduliLabel.csv",stringsAsFactors = F)  data.frame$klasifikasi <- factor(data.frame$klasifikasi)  glimpse(data.frame)  set.seed(20)  data.frame<-data.frame[sample(nrow(data.frame)),]  data.frame<-data.frame[sample(nrow(data.frame)),]  glimpse(data.frame)  corpus<-Corpus(VectorSource(data.frame$text))  corpus  inspect(corpus[1:10])  #fungsinya untuk membersihkan data data yang tidak dibutuhkan  corpus.clean<-corpus %>%  tm\_map(content\_transformer(tolower)) %>% #digunakan untuk mengubah huruf besar dari string menjadi string huruf kecil  tm\_map(removePunctuation)%>% #menghapus tanda baca  tm\_map(removeNumbers)%>% #menghapus nomor  tm\_map(removeWords,stopwords(kind="en"))%>% #menghapus stopwords  tm\_map(stripWhitespace)  dtm<-DocumentTermMatrix(corpus.clean)  inspect(dtm[1:10,1:20])  df.train<-data.frame[1:1580,]  df.test<-data.frame[1601:1980,]  dtm.train<-dtm[1:1580,]  dtm.test<-dtm[1601:1980,]  corpus.clean.train<-corpus.clean[1:1580]  corpus.clean.test<-corpus.clean[1601:1980]  dim(dtm.train)  fivefreq<-findFreqTerms(dtm.train,5)  length(fivefreq)  dtm.train.nb<-DocumentTermMatrix(corpus.clean.train,control = list(dictionary=fivefreq))  dim(dtm.train.nb)  dtm.test.nb<-DocumentTermMatrix(corpus.clean.test,control = list(dictionary=fivefreq))  dim(dtm.test.nb)  #Boolan Naive Bayes  convert\_count <- function(x){  y<-ifelse(x>0,1,0)  y<-factor(y,levels=c(0,1),labels=c("no","yes"))  y  }  #Naive Bayes Model  trainNB<-apply(dtm.train.nb,2,convert\_count)  testNB<-apply(dtm.test.nb,2,convert\_count)  #Training  classifier <- naiveBayes(trainNB, df.train$klasifikasi, laplace = 1)  #Use the NB classifier we built to make predictions on the test set  pred <- predict(classifier, testNB)  #Create a truth table by tabulating the predicted class labels with the actual predicted class labels with the actual class labels  NB\_table=table("Prediction"= pred, "Actual" = df.test$klasifikasi)  NB\_table  #confussion Matrix  conf.matNB <- confusionMatrix(pred, df.test$klasifikasi)  conf.matNB  ``` |

**Listing 3.6** Analisis Sentiment dan perhitungan

*Confusion Matrix* dan *Statistics*.

7. Visualisasi Data

|  |
| --- |
| #naive bayes  library(e1071) #library yang terdapat sebuah algoritma naivebayes  library(caret) #library yang terdapat sebuah algoritma naivebayes  library(syuzhet) #library yang terdapat sebuah algoritma naivebayes  #digunakan untuk membaca file csv yang sudah di cleaning data  peduli\_dataset <-read.csv("peduliClean.csv",stringsAsFactors = FALSE)  #digunakan untuk mengeset variabel cloumn text menjadi char  review <- as.character(peduli\_dataset$text)  #memanggil sentimen dictionary untuk menghitung presentasi dari beberapa emotion dan mengubahnya ke dalam text file  get\_nrc\_sentiment('happy')  get\_nrc\_sentiment('excitement')  s <- get\_nrc\_sentiment(review)  review\_combine<-cbind(peduli\_dataset$text,s)  par(mar=rep(3,4))  b <- barplot(colSums(s),col=rainbow(10),ylab='count',main='Analisis Sentimen')  ```  #WordCloud  library(wordcloud2)  data1 <- read.csv('peduliLabel.csv')  text <- data1$text  docs <- Corpus(VectorSource(text))  docs <- tm\_map(docs, removeWords,"yang")  docs <- tm\_map(docs, removeWords,"lebih")  docs <- tm\_map(docs, removeWords,"dari")  docs <- tm\_map(docs, removeWords,"udah")  docs <- tm\_map(docs, removeWords,"mingguta")  dtm <- TermDocumentMatrix(docs)  matrix <- as.matrix(dtm)  words <- sort(rowSums(matrix),decreasing=TRUE)  df <- data.frame(word = names(words),freq=words)  tweets\_words <- data1 %>%  select(text) %>%  unnest\_tokens(word, text)  words <- tweets\_words %>% dplyr::count(word, sort=TRUE)  p <- wordcloud2(data=df, size=1, color='random-dark')  p  ```  #HistogramFrequency  data1 = read.csv("peduliLabel.csv")  corpus = Corpus(VectorSource(data1$text))  corpus <- tm\_map(corpus, removeWords,"kak")  corpus <- tm\_map(corpus, removeWords,"aja")  corpus <- tm\_map(corpus, removeWords,"gak")  corpus <- tm\_map(corpus, removeWords,"ðÿ")  corpus <- tm\_map(corpus, removeWords,"amp")  corpus <- tm\_map(corpus, removeWords,"yang")  corpus <- tm\_map(corpus, removeWords,"dan")  corpus <- tm\_map(corpus, removeWords,"bisa")  corpus <- tm\_map(corpus, removeWords,"udah")  corpus <- tm\_map(corpus, removeWords,"dari")  dtm <- TermDocumentMatrix(corpus)  m <- as.matrix(dtm)  v <- sort(rowSums(m),decreasing=TRUE)  d <- data.frame(word = names(v),freq=v)  barplot(d[1:20,]$freq, las = 2, names.arg = d[1:20,]$word, col=rainbow(5),  main = "Kata Paling Sering Muncul", ylab = "Frekuensi")  ``` |

**Listing 3.7** Visualisasi Data.

8. *User Interface*

|  |
| --- |
| #shiny  #membuka file csv  twitter <- read.csv(file="peduliClean.csv", header=TRUE)  #membuka text file pada data frame twitter  tweet <- twitter$text  #mengatur tampilan web  ui <- fluidPage(  titlePanel("Penggunaan Kata peduli Pada Twitter"), #judul  mainPanel( #tab  #plot output : untuk scatterplot  tabsetPanel(type = "tabs",  tabPanel("Term Document Matrix and Statistic", verbatimTextOutput("result")),  tabPanel("Histogram", plotOutput("scatterplot")), #tab berupa histogram  tabPanel("Frequency", plotOutput("freqplot")), #tab berupa frequency  tabPanel("Data Twitter", DT::dataTableOutput('tbl')), #tab berupa data cleaning twitter & skoring  tabPanel("Wordcloud", wordcloud2Output("Wordcloud2")) #tab berupa worldcloud  )  )  )  ``` |

**Listing 3.8** *User Interface.*

9. *Server* Shiny

|  |
| --- |
| #Server  #tempat data akan dianalisis dan diproses, hasilnya ditampilkan/diplotkan pada bagian mainpanel() ui  server <- function(input, output) {  #output Data  output$result <-renderPrint({  conf.matNB  })  peduliLabel <- read.csv('peduliLabel.csv')  output$tbl = DT::renderDataTable({  DT::datatable(peduliLabel, options = list(lengthChange = FALSE)) #data ditampilkan dalam beberapa halaman  })    #barplot  output$scatterplot <- renderPlot({  barplot(colSums(s), col=rainbow(10), ylab='count',main='Sentiment Analysis')  }, height = 400)    #freq Plot  output$freqplot <- renderPlot({  barplot(d[1:20,]$freq, las = 2, names.arg = d[1:20,]$word, col=rainbow(5),  main = "Kata Paling Sering Muncul", ylab = "Frekuensi")  }, height = 400)    #wordcloud  output$Wordcloud2 <- renderWordcloud2({  p  })  }  ``` |

**Listing 3.9** *Server* Shiny.

10. *Running* Shiny

|  |
| --- |
| ```{r}  shinyApp(ui = ui, server = server)  ``` |

**Listing 3.10** *Running* Shiny.

1. **KESIMPULAN**

Cara mengetahui sentimen pengguna twitter terhadap Apilkasi Peduli Lindungi adalah dengan melakukan sentimen analisis menggunakan data yang diambil dari tweet pengguna twitter dengan menggunakan metode Naive Bayes. Dari hasil project Analisis Sentimen Twitter terhadap Apilkasi Peduli Lindungi dapat diambil kesimpulan bahwa sentimen pengguna twitter terhadap pengguna twitter cenderung ke arah positif daripada negatif. Kata yang paling sering muncul adalah data dan peduli. Proses pengambilan data dapat dilakukan dengan web *scrapping*. Preprocessing dilakukan dengan menghapus *Uniform Resource Locator* (URL), *username*, RT (*Retweet*), karakter HTML, *emoticon*, hashtag, serta tanda baca dan kata-kata yang tidak berguna.

1. **PEMBAGIAN TUGAS**

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktivitas** | **Pelaksana** |
| 1. | Penentuan tema dan judul proyek | Ale, Farras |
| 2. | Perancangan konsep dasar dan konsep utama proyek | Farras |
| 3. | Pencarian dataset dan scrapping | Ale |
| 4. | Pembuatan program | Ale, Farras |
| 5. | Pembuatan laporan | Ale |
| 6. | Penyempurnaan laporan | Farras |