**TEXT MINING**

**“Project UAS Kelompok 8”**



**Oleh:**

M. Diaz Fahlifi (2011016310019)

Muhammad Difha Wardana (2011016310014)

Rama Noor Rizki (2011016210004)

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS LAMBUNG MANGKURAT**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**PROGRAM STUDI S1 ILMU KOMPUTER**

**BANJARBARU**

**2022**

Tahapan-tahapan yang dilakukan:

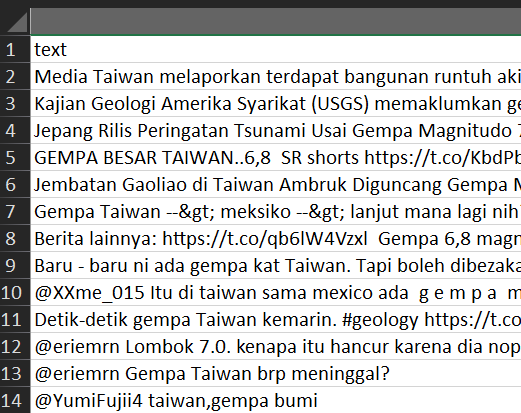
1. **Data Selection**

Hal yang pertama dilakukan adalah dengan mengambil data di **twitter** menggunakan **Library rtweet** yang telah di instal pada **RStudio**. Untuk kata kunci dalam melakukan pencarian yaitu berupa “Gempa”, “Guncangan” , “Taiwan” , “Mexico” , “Getaran”... . Dan untuk Syntax nya yaitu:

|  |
| --- |
| if (!auth\_has\_default() && interactive()) {  auth\_setup\_default()  }  data\_pariaman = search\_tweets(q = "Gempa", n = 500, lang = "id", include\_rts = FALSE)  data\_pariaman = data.frame(data\_pariaman$full\_text)  colnames(data\_pariaman) = "text" |

Untuk menyimpan obyek ke dalam file **Excel** dilakukan dengan fungsi berikut ini

|  |
| --- |
| library(xlsx)  write.xlsx(data\_gempa, "data\_gempa.xlsx", sheetName = "Sheet1", append = FALSE, row.names = FALSE) |



Gambar 1 Obyek pencarian data twitte dari data text yang telah di simpan ke dalam file Excel dengan nama file data\_gempa.xlsx

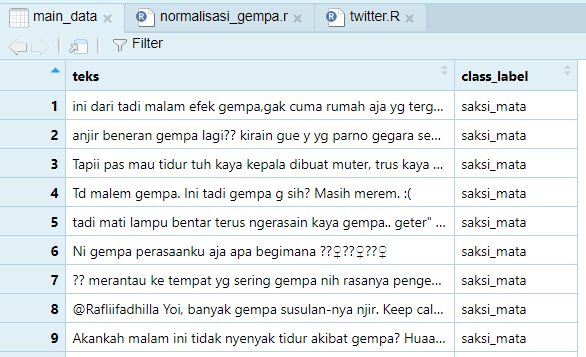
Pencarian data dilakukan secara berulang dengan kata kunci yang berbeda sehingga data yang didapat mencapai **1860** data. Setelah itu dilakukan **Pelabelan** secara manualdengan tiga *class label* yaitu “**saksi\_mata**” , “**bukan\_saksi\_mata**” , dan “**bukan\_gempa**”. Hasil pelabelan diperoleh dengan jumlah yang berbeda pada tiap labelnya, untuk label “saksi\_mata” berjumlah **763** data , “bukan\_saksi\_mata” berjumlah **978** data, dan “bukan\_gempa” berjumlah “**119**” data. Akan tetapi kami hanya mengambil **300** data pada Project Kelompok UAS ini dengan masing-masing label berjumlah **100** data.

1. **Preprocessing Data Teks**

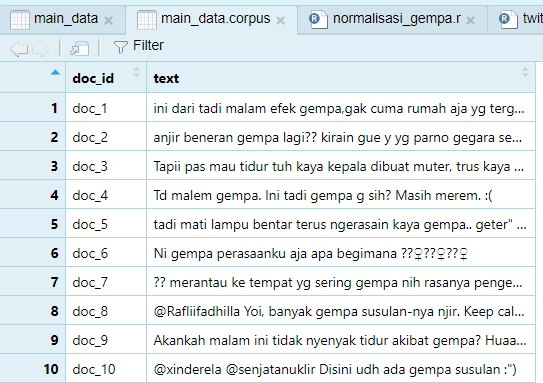
Tahap selanjutnya yaitu dengan melakukan normalisasi teks. Normalisasi teks adalah proses yang berisi tahap-tahap untuk membuat data teks menjadi bersih dan standar agar dapat digunakan oleh tahap text mining selanjutnya. Tahapan-tahapan tersebut berupa menghapus user pada teks, menghapus URL, menghapus hastag, menghapus emoji, mengubah kata menjadi huruf kecil, menghapus karakter spesial, menghapus angka, menghapus stop words, menghapus teks yang mengandung slangword, mengubah menjadi kata dasar, membersihkan spasi berlebih, menghilangkan spasi di awal, dan menghilangkan spasi di awal. Lalu data yang telah dibershikan akan disimpan ke dalam bentuk file **CSV**.

1. Langkah pertama adalah membaca data dari file gempa.xlsx kemudian memuat data teks ke dalam obyek corpus serta mengimport beberapa library yang diperlukan. Berikut adalah kode yang digunakan untuk keperluan itu.

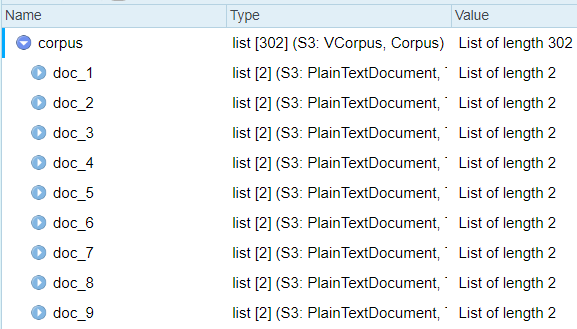
|  |
| --- |
| library(xlsx)  library(tm)  library(katadasaR)  rm(list = ls())  main\_data = read.xlsx("data/gempa.xlsx", sheetName = "Sheet1")  main\_data.corpus = cbind.data.frame(paste0("doc\_",c(1:nrow(main\_data))), main\_data$teks)  colnames(main\_data.corpus) = c("doc\_id", "text") |



Gambar 2 main data



Gambar 3 main\_data\_corpus



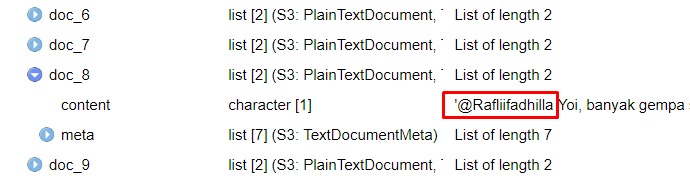
Gambar 4 corpus

1. Langkah selanjutnya yaitu melakukan normalisasi teks.
2. Membersihkan jika ada jarak spasi yang melebihi lebar 2 spasi.

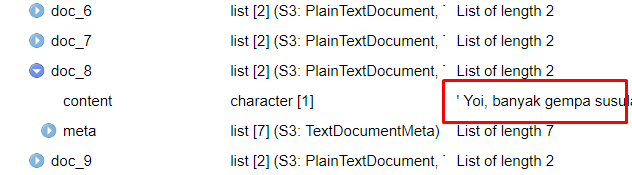
|  |
| --- |
| corpus.processed = tm\_map(corpus, stripWhitespace) |

1. Pada text mining, nama user tidak memiliki makna penting dalam sebuah kalimat sehingga dapat dihapus.

|  |
| --- |
| #menghapus user  removeUser<-function(x) {  return(gsub("@\\w+", "", x))  }  corpus.processed = tm\_map(corpus.processed,content\_transformer(removeUser)) |



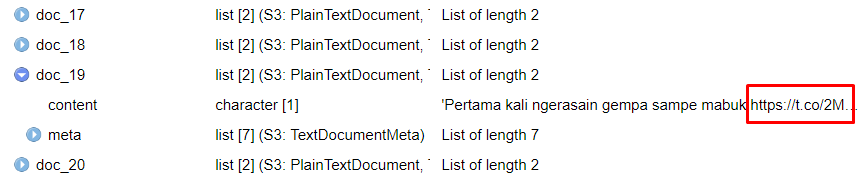
Gambar 5 Sebelum diproses menghapus user.



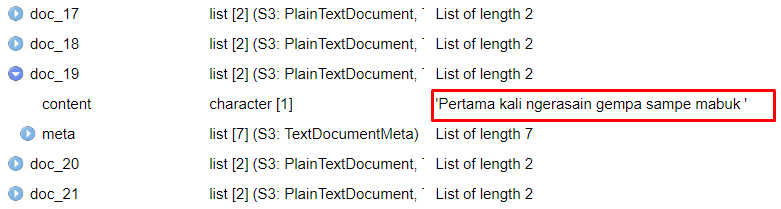
Gambar 6 Sesudah diproses menghapus user.

1. Menghapus URL

|  |
| --- |
| #menghapus URL  cleanURL<-function(x) {  return(gsub("(f|ht)tp(s?)://\\S+", "", x, perl=T))  }  corpus.processed = tm\_map(corpus.processed,content\_transformer(cleanURL)) |



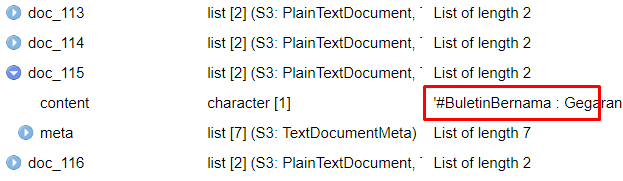
Gambar 7 Sebelum diproses menghapus URL.



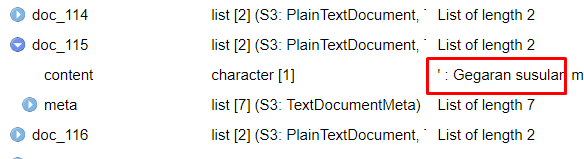
Gambar 8 Sesudah diproses menghapus URL.

1. Menghapus Hashtag

|  |
| --- |
| #menghapus hashtag  cleanHashtag<-function(x) {  return(gsub('#\\S+', '', x))  }  corpus.processed = tm\_map(corpus.processed,content\_transformer(cleanHashtag)) |



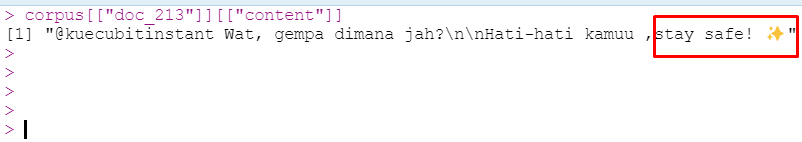
Gambar 9 Sebelum diproses menghapus Hashtag.



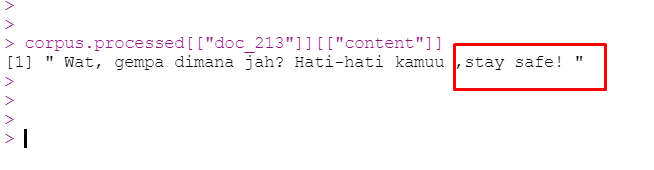
Gambar 10 Sesudah diproses menghapus Hashtag.

1. Menghapus emoji

|  |
| --- |
| #menghapus emoji  corpus.processed = tm\_map(corpus.processed, content\_transformer(cleanEmoji)) |



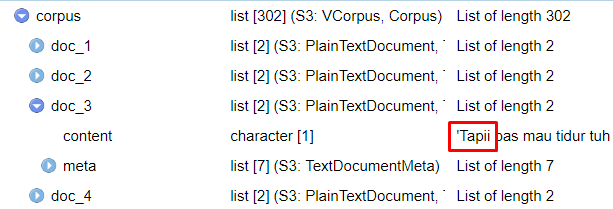
Gambar 11 Sebelum diproses menghapus emoji.



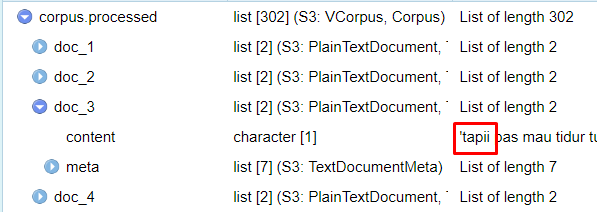
Gambar 12 Sesudah diproses menghapus emoji.

1. Mengubah menjadi huruf kecil

|  |
| --- |
| #mengubah jadi huruf kecil  corpus.processed = tm\_map(corpus.processed,content\_transformer(tolower)) |



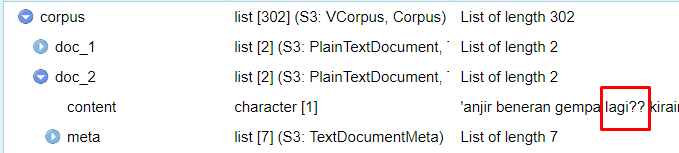
Gambar 13 Sebelum diproses mengubah jadi huruf kecil.



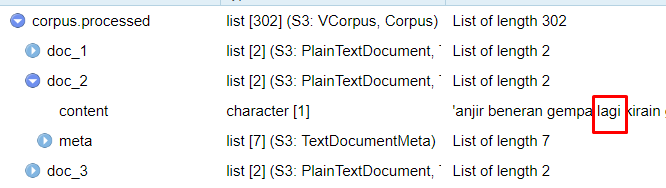
Gambar 14 Sesudah diproses mengubah jadi huruf kecil.

1. Menghapus karakter spesial

|  |
| --- |
| #menghapus tanda baca  corpus.processed = tm\_map(corpus.processed, removePunctuation) |



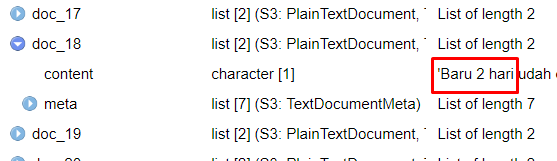
Gambar 15 Sebelum diproses menghapus karakter spesial.



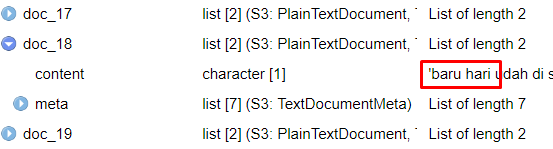
Gambar 16 Sesudah diproses menghapus karakter spesial.

1. Menghapus angka

|  |
| --- |
| #menghapus angka  corpus.processed = tm\_map(corpus.processed, removeNumbers) |



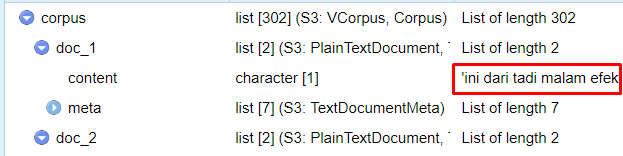
Gambar 17 Sebelum diproses menghapus angka



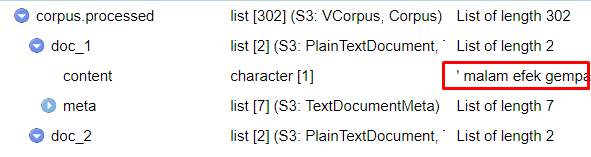
Gambar 18 Sesudah dilakukan proses menghapus angka.

1. Menghapus stopwords

|  |
| --- |
| #menghapus stop words  stopwords.id=readLines("data/stopword/tala-masdevid.txt")  corpus.processed = tm\_map(corpus.processed, removeWords, stopwords.id) |



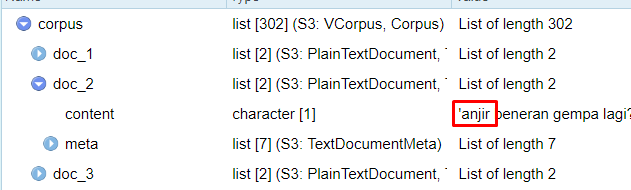
Gambar 19 Sebelum diproses menghapus stopwords.



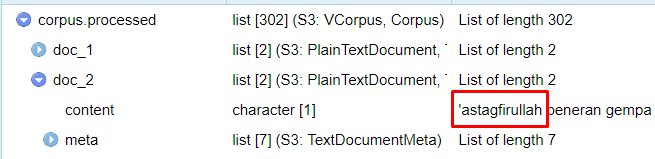
Gambar 20 Sesudah diproses menghapus stopwords..

1. Mengubah slangword

|  |
| --- |
| #read slangword  data\_slang <- read.csv("data/slangword.csv")  #menghapus text yang mengandung slangword  getSlangwords<-function(x) {  x <- replace\_internet\_slang(x, slang = paste0("\\b", data\_slang$slang, "\\b"), replacement = data\_slang$formal, ignore.case = TRUE)  return(x)  }  corpus.processed = tm\_map(corpus.processed, content\_transformer(getSlangwords)) |



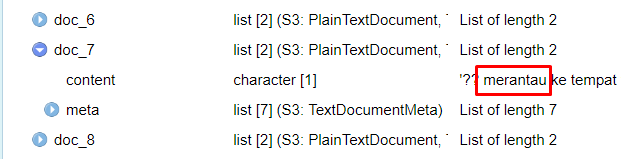
Gambar 21 Sebelum diproses mengubah slangword.



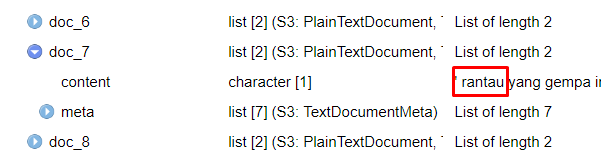
Gambar 22 Sesudah diproses mengubah slangword.

1. Mengubah jadi kata dasar

|  |
| --- |
| #mengubah jadi kata dasar  getKataDasar<-function(x) {  str = unlist(strsplit(stripWhitespace(x), " "))  str<-sapply(str,katadasaR)  str<-paste(str,collapse = " ")  return(str)  }  corpus.processed = tm\_map(corpus.processed,content\_transformer(getKataDasar)) |



Gambar 23 Sebelum diproses mengubah menjadi kata dasar.



Gambar 24 Sesudah diproses mengubah jadi kata dasar.

1. Membersihkan jarak spasi yang melebihi lebar 2 spasi lagi, karena banyak proses normalisasi pada langkah sebelumnya sehingga memunculkan spasi berlebih.

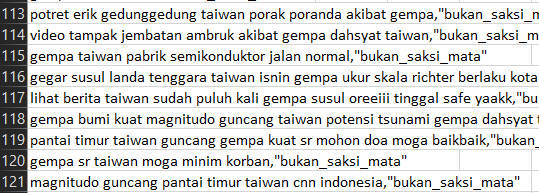
|  |
| --- |
| #membersihkan teks  corpus.processed = tm\_map(corpus.processed, stripWhitespace) |

1. Menghapus spasi di awal teks karena penghapusan user akan memunculkan spasi di awal teks.

|  |
| --- |
| #Menghilangkan Spasi diawal  cleanLeadingWhiteSpace<-function(x) {  return(trimws(x))  }  corpus.processed = tm\_map(corpus.processed, content\_transformer(cleanLeadingWhiteSpace)) |

1. Langkah terakhir pada proses ini yaitu menyimpan data yang telah dilakukan normalisasi ke dalam bentuk file  **CSV** agar dapat dilakukan pada proses selanjutnya

|  |
| --- |
| corpus.df = data.frame(text = unlist(sapply(corpus.processed,'[', "content")), stringsAsFactors = FALSE)  main\_data.processed = cbind.data.frame(corpus.df, main\_data$class\_label)  colnames(main\_data.processed) = c("teks","class\_label")  write.csv(main\_data.processed,"data/gempa\_cleaned.csv", row.names = FALSE) |



Gambar 25 File CSV.