RESPONSI\_DS-(B)

\_<123180119>

## Intro

1. Cuci tangan dengan sabun hingga benar-benar bersih dengan durasi mencuci tangan kurang lebih 20 dtk
2. Kerjakan soal-soal yang ada! Jangan lupa AUTHOR diberi nama (pada bagian atas soal ini)
3. Responsi terdiri dari 2 bagian yaitu bagian pertama dan bagian kedua
4. Jawab dengan membuat chunk dibawah soal!
5. Durasi pengerjaan sesuai kesepakatan yaitu 2 jam mulai pukul 20.00 hingga 22.15 tanggal 22 Januari 2021, 15 menit diberikan untuk pengumpulan hasil responsi
6. No toleransi pengumpulan telat. Telat tiap 3 menit akan ada pengurangan nilai 5 point dengan maksimal pengurangan 25 point. Telat lebih dari 15 menit atau melebihi pukul 22.30 dianggap **GUGUR**.
7. Soal yang rancu bisa menghubungi asisten terkait.
8. Pengumpulan hanya dalam bentuk **WORD Document atau PDF**. Jika pengumpulan dalam bentuk **Rmd** akan dianggap tidak mengumpulkan jawaban. Pastikan jawaban dapat dijalankan dengan baik.
9. Tenang, untuk responsi kali ini nilai akan diobral, nilai maksimal adalah 350 dari 100. Jadi, kemungkinan dapat nilai bagus besar kok.
10. Isi juga review/feedback/kritik/saran/masukan yang sudah disediakan di bagian paling bawah soal. ***WAJIB***
11. Jawaban dikumpulkan dengan format file JAWABAN\_responsi\_ dalam bentuk PDF atau DOKUMEN.

## Persiapan

Load library apa saja yang kira-kira digunakan! Lalu load dataset ‘googleplay.csv’ dan ‘googleplay\_user\_review.csv’!

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom 0.7.3 v recipes 0.1.15  
## v dials 0.0.9 v rsample 0.0.8   
## v infer 0.5.4 v tune 0.1.2   
## v modeldata 0.1.0 v workflows 0.2.1   
## v parsnip 0.1.4 v yardstick 0.0.7

## -- Conflicts ----------------------------------------- tidymodels\_conflicts() --  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

library(dbplyr)

##   
## Attaching package: 'dbplyr'

## The following objects are masked from 'package:dplyr':  
##   
## ident, sql

library(topicmodels)  
 library(ggplot2)  
 library(tm)

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(vroom)  
 library(here)

## here() starts at C:/Users/MSI GF63-024/Documents/GitHub/PraktikumDataScience/Praktikum\_Data\_Science2020

library(tidytext)  
 data\_googleplay = vroom(here("Responsi", "googleplaystore.csv"))

## Rows: 8,196  
## Columns: 13  
## Delimiter: ","  
## chr [11]: App, Category, Size, Installs, Type, Price, Content Rating, Genres, Last Updated...  
## dbl [ 2]: Rating, Reviews  
##   
## Use `spec()` to retrieve the guessed column specification  
## Pass a specification to the `col\_types` argument to quiet this message

data\_googleplay\_user = vroom(here("Responsi", "googleplaystore\_user\_reviews.csv"))

## Rows: 64,295  
## Columns: 5  
## Delimiter: ","  
## chr [3]: App, Translated\_Review, Sentiment  
## dbl [2]: Sentiment\_Polarity, Sentiment\_Subjectivity  
##   
## Use `spec()` to retrieve the guessed column specification  
## Pass a specification to the `col\_types` argument to quiet this message

## Bagian Pertama

1. Tampilkan TOP 10 Aplikasi berdasarkan peringkat PENILAIAN/RATING yang diberikan user! **point 10**

data\_googleplay %>% arrange(-Rating) %>% head(10)

## # A tibble: 10 x 13  
## App Category Rating Reviews Size Installs Type Price `Content Rating`  
## <chr> <chr> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr>   
## 1 Hoji~ COMICS 5 15 37M 1,000+ Free 0 Everyone   
## 2 Amer~ DATING 5 5 4.4M 1,000+ Free 0 Mature 17+   
## 3 Awak~ DATING 5 2 70M 100+ Free 0 Mature 17+   
## 4 Spin~ DATING 5 5 9.3M 500+ Free 0 Teen   
## 5 Girl~ DATING 5 6 5.0M 100+ Free 0 Mature 17+   
## 6 Onli~ DATING 5 5 5.0M 100+ Free 0 Mature 17+   
## 7 Spee~ DATING 5 3 25M 100+ Free 0 Mature 17+   
## 8 SUMM~ EVENTS 5 4 61M 500+ Free 0 Everyone   
## 9 Pros~ EVENTS 5 16 2.3M 100+ Free 0 Everyone   
## 10 Mind~ EVENTS 5 1 21M 100+ Free 0 Everyone   
## # ... with 4 more variables: Genres <chr>, `Last Updated` <chr>, `Current  
## # Ver` <chr>, `Android Ver` <chr>

1. Tampilkan rata-rata RATING yang dihitung menggunakan fungsi buatan untuk setiap kategori aplikasi! **point 15**

data\_googleplay %>% group\_by(Category) %>% summarize(average = mean(Rating)) -> Rata\_rata\_rating

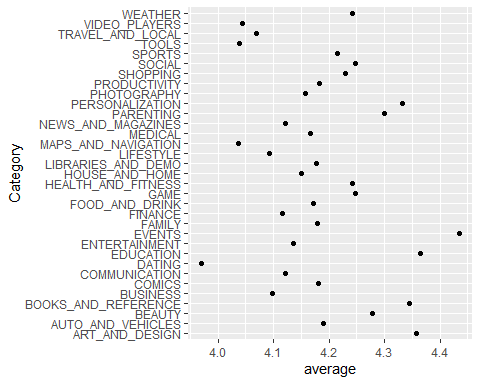
## `summarise()` ungrouping output (override with `.groups` argument)

Rata\_rata\_rating

## # A tibble: 33 x 2  
## Category average  
## <chr> <dbl>  
## 1 ART\_AND\_DESIGN 4.36  
## 2 AUTO\_AND\_VEHICLES 4.19  
## 3 BEAUTY 4.28  
## 4 BOOKS\_AND\_REFERENCE 4.34  
## 5 BUSINESS 4.10  
## 6 COMICS 4.18  
## 7 COMMUNICATION 4.12  
## 8 DATING 3.97  
## 9 EDUCATION 4.36  
## 10 ENTERTAINMENT 4.14  
## # ... with 23 more rows

1. Berdasarkan soal nomor 2, buat plot untuk memvisualisasikan hasilnya! (Bentuk plot bebas) **point 15**

Rata\_rata\_rating %>% ggplot(aes(x = average, y = Category) ) + geom\_point()



Info untuk 2 soal 4-5:Terdapat dua dataset yang digunakan. Satu dataset untuk info aplikasi dan satu dataset lagi untuk kumpulan reviewnya.

1. Dari kedua dataset tersebut, buat satu variable data baru yang isinya NAMA APLIKASI, RATING, dan JUMLAH REVIEW Positif dan/atau Negatif dan/atau Neutral (boleh semua, boleh pilih salah satu)lalu tampilkan isi data tabel tersebut! **point 20**

data\_baru <- inner\_join(data\_googleplay, data\_googleplay\_user, by = "App") %>%  
 filter(Sentiment == "Positive") %>%  
 select(App, Rating, Translated\_Review) %>%  
 group\_by(App)   
 data\_baru

## # A tibble: 21,739 x 3  
## # Groups: App [755]  
## App Rating Translated\_Review   
## <chr> <dbl> <chr>   
## 1 Coloring book moana 3.9 I love colors inspyering  
## 2 Coloring book moana 3.9 It cute.   
## 3 Coloring book moana 3.9 I love   
## 4 Coloring book moana 3.9 Love   
## 5 Coloring book moana 3.9 I love enjoyable fun   
## 6 Coloring book moana 3.9 I sure   
## 7 Coloring book moana 3.9 Too many ads   
## 8 Coloring book moana 3.9 Good great   
## 9 Coloring book moana 3.9 It's ok   
## 10 Coloring book moana 3.9 This beautiful   
## # ... with 21,729 more rows

1. Dalam dunia data scientist, sebelum melakukan pemodelan ada baiknya data dilakukan preprocessing terlebih dahulu. Dengan dataset review yang sudah dimasukkan oleh user, lakukan sebuah preprocessing data SEDERHANA yang menurut kalian dapat dilakukan untuk dataset tersebut agar dataset bisa siap untuk dimodelkan (simpan hasil preprocessing dalam variabel baru)!

Clue : Clean, Tidy, no redundacy, no dupe, no null. **point 40**

data\_baru\_clean <- data\_baru %>%   
 filter(!is.na(Translated\_Review)) %>%   
 group\_by(App) %>%  
 unnest\_tokens(word, Translated\_Review) %>%  
 ungroup() %>%  
 anti\_join(stop\_words)

## Joining, by = "word"

data\_baru\_clean = unique(data\_baru\_clean)  
 data\_baru\_clean

## # A tibble: 111,551 x 2  
## App word   
## <chr> <chr>   
## 1 Coloring book moana love   
## 2 Coloring book moana colors   
## 3 Coloring book moana inspyering  
## 4 Coloring book moana cute   
## 5 Coloring book moana enjoyable   
## 6 Coloring book moana fun   
## 7 Coloring book moana ads   
## 8 Coloring book moana beautiful   
## 9 Coloring book moana perfect   
## 10 Garden Coloring Book coloring   
## # ... with 111,541 more rows

## Bagian Kedua

Referensi mengerjakan: <https://www.tidytextmining.com/>

1. Import library tidymodels, vroom, here, tidytext dan dua dataset ke dalam objek R **nilai 10**

library(tidymodels)  
library(vroom)  
library(here)  
library(tidytext)  
  
user\_reviews <- vroom(here("Responsi" , "googleplaystore\_user\_reviews.csv"))

## Rows: 64,295  
## Columns: 5  
## Delimiter: ","  
## chr [3]: App, Translated\_Review, Sentiment  
## dbl [2]: Sentiment\_Polarity, Sentiment\_Subjectivity  
##   
## Use `spec()` to retrieve the guessed column specification  
## Pass a specification to the `col\_types` argument to quiet this message

googleplaystore <- vroom(here("Responsi", "googleplaystore.csv"))

## Rows: 8,196  
## Columns: 13  
## Delimiter: ","  
## chr [11]: App, Category, Size, Installs, Type, Price, Content Rating, Genres, Last Updated...  
## dbl [ 2]: Rating, Reviews  
##   
## Use `spec()` to retrieve the guessed column specification  
## Pass a specification to the `col\_types` argument to quiet this message

1. Joining dua dataset menggunakan inner join **nilai 10**

joining\_data <- user\_reviews %>%  
 inner\_join(googleplaystore)

## Joining, by = "App"

1. Tahap pre-processing data. Ketika ingin melakukan analisis sentimen beberapa hal harus dilakukan sebelum data dapat digunakan. Bersihkan dan rapikan data dengan membuang data yang “nan” di bagian Translated\_review. Setelah itu, data juga harus dibersihkan dari kata-kata yang mengandung stop\_word (seperti: a, a’s, after, dll). Data yang siap diolah juga harus ditokenisasi yaitu proses membagi teks dari paragraf atau kalimat ke kata. Hasil dari tokenisasi adalah tiap baris data hanya mengandung 1 kata. **nilai 15**

tidy\_user\_reviews <- joining\_data %>%  
 filter(Translated\_Review != "nan") %>%  
 unnest\_tokens(word, Translated\_Review) %>%  
 anti\_join(stop\_words)

## Joining, by = "word"

1. Sentimen analisis dapat menggunakan beberapa jenis metode berdasarkan sentiment lexicon. Ada beberapa sentiment lexicon seperti bing, afinn, dan nrc. Gunakan sentiment lexicon nrc untuk mendapatkan jumlah kata untuk 10 kategori nrc (positive, negative, fear, surprise, dll). ***nilai 15***

nrc\_n <- tidy\_user\_reviews %>%   
 inner\_join(get\_sentiments("nrc")) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

nrc\_n

## # A tibble: 2,673 x 2  
## word n  
## <chr> <int>  
## 1 money 8766  
## 2 love 8646  
## 3 fun 5187  
## 4 pay 4956  
## 5 bad 4450  
## 6 time 4237  
## 7 hate 2525  
## 8 level 2396  
## 9 pretty 2320  
## 10 star 2292  
## # ... with 2,663 more rows

1. Kita dapat mengetahui banyaknya kata tiap kategori nrc untuk tiap aplikasi. Cobalah untuk mencari banyak kata tiap kategori nrc yang dikelompokkan berdasarkan nama aplikasi. **nilai 15**

user\_reviews\_nrc <- tidy\_user\_reviews %>%   
 inner\_join(get\_sentiments("nrc")) %>%  
 group\_by(App) %>%  
 count(sentiment, sort = TRUE) %>%  
 spread(sentiment, n, fill = 0) %>%  
 ungroup()

## Joining, by = "word"

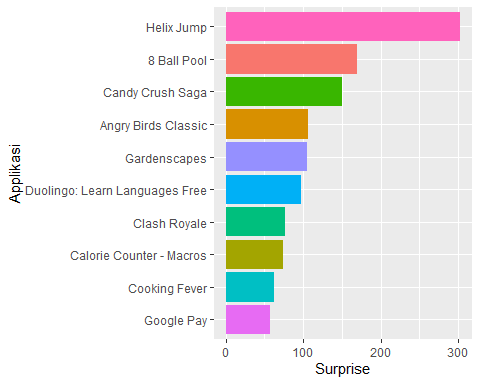
user\_reviews\_nrc

## # A tibble: 758 x 11  
## App anger anticipation disgust fear joy negative positive sadness  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 10 B~ 8 38 10 12 96 26 176 16  
## 2 11st 8 20 2 12 13 18 27 10  
## 3 1800~ 8 40 4 6 34 16 76 6  
## 4 21-D~ 20 82 16 20 68 46 149 26  
## 5 2Dat~ 6 31 7 3 31 11 45 4  
## 6 2GIS~ 5 12 5 6 8 12 24 6  
## 7 2ndL~ 3 19 3 3 14 31 19 19  
## 8 2Red~ 3 7 3 2 13 5 22 3  
## 9 30 D~ 11 16 6 12 25 19 39 15  
## 10 365S~ 1 2 0 1 3 2 3 1  
## # ... with 748 more rows, and 2 more variables: surprise <dbl>, trust <dbl>

1. Setelah mendapatkan jumlah kata tiap kategori tiap aplikasi, kita dapat mngetahui aplikasi mana yang memiliki kata dengan kategori ‘surprise’ terbanyak untuk tiap aplikasi. Kita akan memvisualisasikan dengan grafik batang 10 aplikasi dengan jumlah kata kategori ‘surprise’ terbanyak. ***nilai 20***

user\_reviews\_nrc %>%  
 arrange(desc(surprise)) %>%  
 top\_n(10) %>%  
 ggplot(aes(reorder(App, surprise), y = surprise, fill = App)) +  
 geom\_col(show.legend = FALSE) +  
 coord\_flip() +  
 labs(  
 x = "Applikasi",  
 y = "Surprise"  
 )

## Selecting by trust



1. Selain menggunakan sentiment lexicon ‘nrc’, sentimen analisis juga dapat menggunakan sentiment lexicon ‘bing’. Bing hanya akan memberikan label untuk tiap kata positif atau negatif saja. Carilah kata positif yang paling umum dan kata negatif yang paling sering digunakan saat memberikan review pada aplikasi! ***nilai 15***

bing\_word\_counts <- tidy\_user\_reviews %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE)

## Joining, by = "word"

bing\_word\_counts

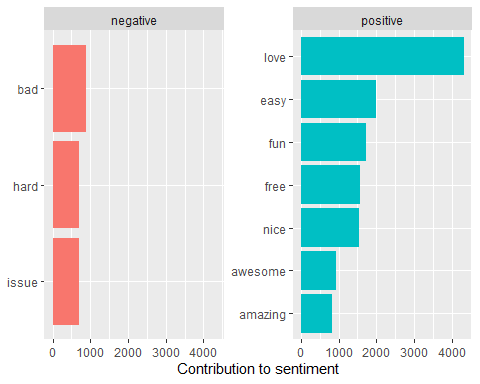
## # A tibble: 2,286 x 3  
## word sentiment n  
## <chr> <chr> <int>  
## 1 love positive 4323  
## 2 easy positive 1988  
## 3 fun positive 1729  
## 4 free positive 1574  
## 5 nice positive 1531  
## 6 awesome positive 941  
## 7 bad negative 890  
## 8 amazing positive 820  
## 9 hard negative 707  
## 10 issue negative 701  
## # ... with 2,276 more rows

1. Pembacaan data akan lebih mudah jika ditampilkan dalam bentuk grafik. Tampilkan grafik 10 kata positif dan negatif terbanyak! ***nilai 20***

bing\_word\_counts %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word, n, fill = sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~sentiment, scales = "free\_y") +  
 labs(y = "Contribution to sentiment",  
 x = NULL) +  
 coord\_flip()

## Joining, by = c("word", "sentiment")

## Selecting by n



1. Penganalisis data membutuhkan jumlah kata tiap kategori yang belum digabung dengan sentiment lexicon untuk menghitung rasio positif, ratio negatif dan net sentiment. Bantulah penganalisis tersebut untuk mendapatkan jumlah kata tiap kategori dari data yang sudah dirapikan! ***nilai 15***
2. Selanjutnya penganalisis data ingin mendapatkan jumlah kata positif, jumlah kata negatif, rasio positif (jumlah kata positif/jumlah keseluruhan kata), rasio negatif (jumlah kata negatif/jumlah keseluruhan kata), dan net sentiment (jumlah kata positif - jumlah kata negatif) dengan menggunakan sentiment lexicon bing untuk tiap kategorinya. Tabel yang diinginkan oleh analisis adalah seperti berikut ***nilai 40*** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Category | poisitive | negative | words | positive\_ratio | negative\_ratio | net\_sentiment | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kritik/saran/masukan/feedback/review/uneg-uneg: Kritik : sebenarnya penyampaian materi sudah bagus, tetapi karna corona saat ini, jadi lebih sedikit masnya dalam menyampaikan materi, jadi serasa tidak maksimal

================================================ SELESAI ================================================