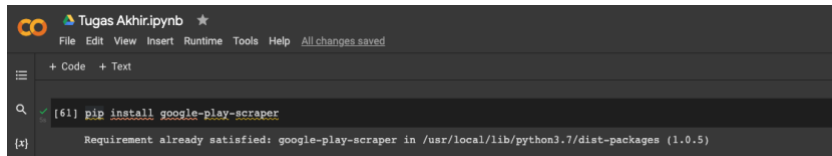


## LAMPIRAN

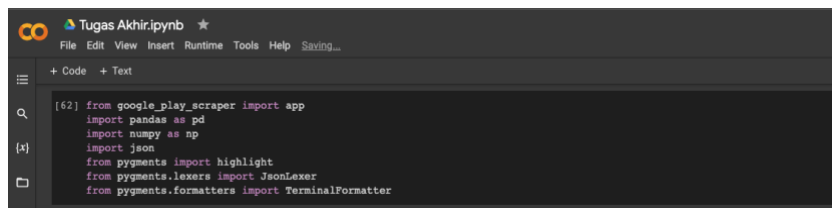
Hal pertama yang dilakukan adalah *download* dan menginstal *Google Play Scraper Package*. Tujuannya untuk mengakses API yang sudah disediakan



```
Tugas Akhir.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[61] pip install google-play-scraper
Requirement already satisfied: google-play-scraper in /usr/local/lib/python3.7/dist-packages (1.0.5)
```

Lampiran 1. *Install Scraper Package*

Dilanjut dengan melakukan proses *import library* dari *pandas*, *numpy*, *json* dan beberapa *library json* lainnya seperti *highlight*, *JsonLaser*, dan *TerminalFormatter*. JSON di sini berguna untuk menampilkan hasil dari ulasan yang diambil ke dalam bentuk JSON. Tujuannya agar lebih mudah dibaca.



```
Tugas Akhir.ipynb
File Edit View Insert Runtime Tools Help Saving...
+ Code + Text
[62] from google_play_scraper import app
import pandas as pd
import numpy as np
import json
from pygments import highlight
from pygments.lexers import JsonLexer
from pygments.formatters import TerminalFormatter
```

Lampiran 2. *Import Library*

Tahapan ketiga yaitu melakukan *scrap* semua ulasan dari aplikasi ternaknesia dengan menggunakan *function reviews\_all*. Sehingga akan mengambil semua ulasan tanpa batas. Karena aplikasi ternaknesia itu berasal dari indonesia dan ulasan yang ingin diambil yang berbahasa indonesia, maka untuk parameter *lang* dan *country* diisi dengan *id* untuk *lang*, dan ID untuk *country* dengan kode yang bergantung pada standar ISO 3166 dan ISO 639-1.

Setelah mengambil semua ulasan dari aplikasi, selanjutnya adalah mengubahnya menjadi huruf kecil dengan mengambil isi dari *key content*. Karenan nantinya yang akan difilter adalah *value* dari *content*, dimana *content* adalah komentar yang diberikan oleh pengguna untuk aplikasi di playstore.



```

Tugas Akhir.ipynb
File Edit View Insert Runtime Tools Help @dibungkus

+ Code + Text
[131]: all = df[result]
print("Total Review : " + str(all))

# length = len(str_muncul)
# print("Total Review yang @FIS1000 : " + str(length))

Total Review : 264

```

Lampiran 5. Jumlah Review Keseluruhan

Mengubah tampilan ulasan dari yang semula JSON ke dalam bentuk tabel dengan mengambil *key* dari *User Name*, *Content*, *Score*, dan *Thumb Up Count* karena ingin mencari tahu siapa yang memberikan komentar, berapa skor yang diberikan dan seberapa banyak pengguna lain yang merasa terbantu atau sependapat dengan ulasan tersebut.

```

Tugas Akhir.ipynb
File Edit View Insert Runtime Tools Help @dibungkus

+ Code + Text
[132]: data = pd.DataFrame()
# data['User Name'] = [a['username'] for a in str_muncul]
# data['Content'] = [a['comment'] for a in str_muncul]
# data['Score'] = [a['score'] for a in str_muncul]
# data['Thumb Up Count'] = [a['thumbsUpCount'] for a in str_muncul]

data['User Name'] = [a['username'] for a in result]
data['Content'] = [a['comment'] for a in result]
data['Score'] = [a['score'] for a in result]
data['Thumb Up Count'] = [a['thumbsUpCount'] for a in result]
display(data.head(5))

```

User Name	Content	Score	Thumb Up Count
0	rengas pradana salah satu aplikasi fintech (perbankan) yg rec...	5	9
1	MOCHAMMAD SOLEH suatu aplikasi yg tidak bisa terfiks dioperasikan...	1	4
2	Salfu m.safuhul aplikasiya enak banget min,, kalo bisa buat ya...	3	6
3	April Lia dari pertama instal tidak pernah bisa login p...	2	5
4	Dryza Investor bingung di koreksi update yg baru no rekening s...	3	9
...	...	...	...
261	Pengguna Google	bagus	5
262	Bahay Mufid	bagus	5
263	Muhammad Mubtahir Al-Fauzi	bagus	5
264	Tu an	mantap	5
265	Bahay Mufid	bagus	5

264 rows x 4 columns

Lampiran 6. JSON ke Bentuk Tabel

Proses selanjutnya adalah menyimpan tabel dari ulasan tadi ke dalam bentuk file .csv karena selain akan diolah datanya ke dalam bentuk *word cloud* nantinya data tersebut diberikan ke bagian tim data di ternaknesia. File tersebut nantinya tersimpan ke dalam Google Drive dan dihubungkan dengan cara mount ke Google Colab agar file dapat diproses dan dibaca.

```

Tugas Akhir.ipynb
File Edit View Insert Runtime Tools Help @dibungkus

+ Code + Text
[133]: # url review_of = pd.DataFrame(result)
from google.colab import files
data_to_csv('review.csv', index=False, header=True)
# data_to_csv('result.json', index=False, header=True)

[134]: from google.colab import drive
drive.mount('content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount('/content/drive', force_remount=True).

readFile = pd.read_csv('/content/review.csv')
label = readFile['Content']
label[:100]

0 salah satu aplikasi fintech (perbankan) yg rec...
1 suatu aplikasi yg tidak bisa terfiks dioperasikan...
2 aplikasiya enak banget min,, kalo bisa buat ya...
3 dari pertama instal tidak pernah bisa login p...
4 bingung di koreksi update yg baru no rekening s...
...
261
262
263
264
265
Header: Content, Length: 264, dtype: object

```

Lampiran 7. Mount Google Drive dan Preview File

Proses mapping selanjutnya adalah melakukan penghapusan karakter, huruf yang berdiri sendiri di awal ataupun diseluruh kalimat, mengganti spasi yang double menjadi satu spasi, dan penghapusan prefix 'b'.

```

import re

processed_features = []

for sentence in range(0, len(labels)):
    # Remove all the special characters
    processed_feature = re.sub(r'@', '', str(labels[sentence]))

    # Remove all single characters
    processed_feature = re.sub(r'([a-z])', '', processed_feature)

    # Remove single characters from the start
    processed_feature = re.sub(r'^([a-z])', '', processed_feature)

    # Substituting multiple spaces with single space
    processed_feature = re.sub(r' ', ' ', processed_feature, flags=re.I)

    # Removing prefixes 'b'
    processed_feature = re.sub(r'b', '', processed_feature)

    # Converting to lowercase
    processed_feature = processed_feature.lower()

    processed_features.append(processed_feature)
    
```

Lampiran 8. Hapus Karakter

Hasil yang didapatkan dari proses pembersihan kalimat adalah sebagai berikut:

```

[139] labels[300]

0 salah satu aplikasi finansial (investasi) yg recom...
1 aplikasi yg dipertaya, dan salah satu data...
2 aplikasi untuk bisnis, baik untuk bisnis...
3 inskri login, dan daftar, untuk apa? ya email...
4 bahwa, update, update yg not, sehingga...

Name: Count, Length: 144, dtype: object
    
```

Lampiran 9. Hasil Hapus Karakter

Kalimat dari ulasan yang diberikan nantinya akan dipecah menjadi beberapa kata yang disebut dengan token. Proses ini dapat disebut *Tokenization* yang bisa dilakukan dengan melakukan *import library NLTK* dan mendownload 'punkt'.

```

[140] import nltk
nltk.download('punkt') # Download Punkt
nltk.download('stopwords') # Download

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True
    
```

Lampiran 10. Download Punkt dan Stopwords

Proses selanjutnya adalah *stopwords* yang dimana hanya mengambil kata yang dianggap penting dan hanya menampilkan lima dataset yang hanya digunakan untuk kalimat yang berbahasa Indonesia.

```

[141] from nltk.corpus import stopwords
stopwords = stopwords.words('indonesian')
label = label.apply(lambda x: '-'.join(w for w in x.split() if w not in stop))
label.head()

0 salah aplikasi finansial (investasi) yg recom...
1 aplikasi yg dipertaya, dan salah satu data...
2 aplikasi untuk bisnis, baik untuk bisnis...
3 inskri login, dan daftar, untuk apa? ya email...
4 bahwa, update, update yg not, sehingga...

Name: Count, dtype: object
    
```

## Lampiran 11. Proses *Stopwords*

Setelah pengolahan *stopwords* dan *tokenization*, dilanjut dengan penghapusan tanda baca di setiap kalimat dari ulasan.



```
[[14]] label = label_str.replace(' ','')
label = label

~/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: The default value of regex will change from True to False in a future version.
**Happy path! For launching an IPython kernel,
1. make sure that Emacs (or other) is pre-installed...
2. make sure you have installed the correct version of...
3. install ipynb with the correct version of...
4. follow the instructions to get the correct version...
Name: Console, type: object
```

## Lampiran 12. Hapus *Punctuation*

Proses terakhir yang dilakukan adalah pengolahan teks ke dalam bentuk visual (*word cloud*) dimana metode ini populer di bidang text mining karena mudah dipahami. Dengan metode ini, gambaran frekuensi tiap kata dapat ditampilkan dalam bentuk yang menarik dan informatif. Disini dapat juga melakukan kustom warna sesuai dengan keinginan.



```
[[13]] def multi_color_wordcloud(text, size_words,
                             max_words, min_words,
                             colors = [(255, 0, 0),
                                       (0, 255, 0),
                                       (0, 0, 255),
                                       (255, 255, 0)],
                             read = random_state.RandomState(1)):
    return "hallo", [N, 100] - format(read[0], '%d'), colors[read[1]]
```

## Lampiran 13. Membuat Warna Kustom

Untuk mengimplementasikan warna yang dikustom secara manual maka perlu menambahkan function baru yang kemudian akan dipanggil sebagai argumen atau parameter.

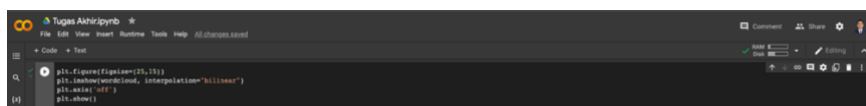


```
[[14]] from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from PIL import Image

mask = np.array(Image.open('~/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py'))
word_string = " ".join(label_str_lower)
wordcloud = WordCloud(mask=mask, background_color="black", max_words=100,
                    colormap=multi_color_func).generate(word_string)
```

## Lampiran 14. Colormap

Terkahir adalah menentukan ukuran dari *word cloud* beserta kualitas dari Gambar 4.50, yang ditampilkan.



```
[[15]] plt.figure(figsize=(25,15))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

## Lampiran 15. Ukuran Gambar



- b. Hasil
- c. Peternak
- d. Bermanfaat
- e. Membantu

```

from google_play_scraper import Sort, reviews_all

# from google_play_scraper import Sort, reviews_all
result = reviews_all(
    "com.kayakobanla.app",
    sleep_milliseconds=0, # default is 0
    lang="id", # default is "en"
    country="ID", # default is "us"
    sort="MOST_RELEVANT", # default is SORT_BY_RELEVANCE
    # filter_keywords # default is None/empty list
)

# new_result = json.dumps(result)
# final_result = [{"key": value[key] for key, value in result.items()} for value in result]
# final_result = result.items()

# HUBUNGAN DENGAN MENYALIN DINDING DINDING
for idx, review in enumerate(result):
    content_text = review["text"]
    content_title = review["title"]
    review_metadata["text"] = content_text
    result[idx] = review

# print("\n\n after : {}".format(result), result[idx])

# FILTER BERDASARKAN DATA/INFORMASI YANG DINDING DINDING
str_match = [x for x in result if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match2 = [x for x in str_match if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match3 = [x for x in str_match2 if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match4 = [x for x in str_match3 if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]

# print(str_match)
# print(str_match4)

```

Lampiran 18. Filter Ketiga

Filter keempat diperlukan dengan menghilangkan beberapa kata, diantaranya :

- a. Top
- b. Nice
- c. Super
- d. Sold

```

from google_play_scraper import Sort, reviews_all

# from google_play_scraper import Sort, reviews_all
result = reviews_all(
    "com.kayakobanla.app",
    sleep_milliseconds=0, # default is 0
    lang="id", # default is "en"
    country="ID", # default is "us"
    sort="MOST_RELEVANT", # default is SORT_BY_RELEVANCE
    # filter_keywords # default is None/empty list
)

# new_result = json.dumps(result)
# final_result = [{"key": value[key] for key, value in result.items()} for value in result]
# final_result = result.items()

# HUBUNGAN DENGAN MENYALIN DINDING DINDING
for idx, review in enumerate(result):
    content_text = review["text"]
    content_title = review["title"]
    review_metadata["text"] = content_text
    result[idx] = review

# print("\n\n after : {}".format(result), result[idx])

# FILTER BERDASARKAN DATA/INFORMASI YANG DINDING DINDING
str_match = [x for x in result if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match2 = [x for x in str_match if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match3 = [x for x in str_match2 if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]
str_match4 = [x for x in str_match3 if "top" not in x["text"] and "nice" not in x["text"] and "peternak" not in x["text"] and "hasil" not in x["text"] and "bermanfaat" not in x["text"] and "membantu" not in x["text"]]

# print(str_match)
# print(str_match4)

```

Lampiran 19. Filter Keempat

Hasil dari *filter* keempat sudah dapat memperlihatkan *pain point* ketika menggunakan aplikasi. Dari beberapa kata negatif tersebut akan dilakukan *filter* untuk mencari tahu detail ulasan dari kata tersebut tetapi yang tidak memiliki hubungan dengan ternakinvest. Cara yang akan dilakukan adalah menambahkan *filter* yang berfungsi untuk meniadakan kalimat yang terdapat kata invest, saham, lembar, dan proposal.