latihan_vektorisasi (1)

November 22, 2023

Muhammad Ihsan Prawira Hutomo 2211110022

0.0.1 Latihan

[35]:
```
import pandas as pd

df = pd.read_csv('pantun.csv')
```

[36]:
```
df.head()
```

[36]:
```
                                                teks                 tipe
0  Ada motor ada sepeda \n Semuanya beroda dua \n…  Pantun Adat dan Alam
1  Ada pisang ada semangka \n Jika dimakan manis …  Pantun Adat dan Alam
2  Ada rusa ada buaya \n Sungguh hitam warna mata…  Pantun Adat dan Alam
3  Alat timbang pucuknya patah \n Beli baru henda…  Pantun Adat dan Alam
4  Anak cina makan petai \n Kakinya terikat ranta…  Pantun Adat dan Alam
```

[37]:
```
def clear(text):
  import re
  teks_bersih = re.sub(r'\\n', '.', text)
  return teks_bersih
```

[38]:
```
teks = df['teks'].values.tolist()
```

[39]:
```
teks = ''.join(teks)
```

[40]:
```
teks = clear(teks)
```

[40]:
```

```

[41]:
```
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download('punkt')

teks = sent_tokenize(teks)
```

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

[42]:

```
teks = teks[:10]
```

[43]:

```
teks
```

[43]:

```
['Ada motor ada sepeda .',
 'Semuanya beroda dua .',
 'Indonesia kaya budaya .',
 'Sepatutnya kita menjaganyaAda pisang ada semangka .',
 'Jika dimakan manis rasanya .',
 'Indonesia ragam budaya .',
 'Tugas kita tuk menjaganyaAda rusa ada buaya .',
 'Sungguh hitam warna matanya .',
 'Ada adat seribu bahasa .',
 'Kita wajib menghormatinyaAlat timbang pucuknya patah .']
```

[44]:

```
import string
from nltk.tokenize import word_tokenize

# Function to remove punctuation from a given sentence
def remove_punctuation(sentence):
    return sentence.translate(str.maketrans("", "", string.punctuation))

# Tokenize and remove punctuation for each sentence
tokenized_text = [word_tokenize(remove_punctuation(sentence)) for sentence in␣
 ↪teks]

# Print the tokenized sentences without punctuation
for tokens in tokenized_sentences:
    print(tokens)
```

```
['Ada', 'motor', 'ada', 'sepeda']
['Semuanya', 'beroda', 'dua']
['Indonesia', 'kaya', 'budaya']
['Sepatutnya', 'kita', 'menjaganyaAda', 'pisang', 'ada', 'semangka']
['Jika', 'dimakan', 'manis', 'rasanya']
['Indonesia', 'ragam', 'budaya']
['Tugas', 'kita', 'tuk', 'menjaganyaAda', 'rusa', 'ada', 'buaya']
['Sungguh', 'hitam', 'warna', 'matanya']
['Ada', 'adat', 'seribu', 'bahasa']
['Kita', 'wajib', 'menghormatinyaAlat', 'timbang', 'pucuknya', 'patah']
```

0.0.2 Ubah variabel teks(diatas), menjadi vektor word2vec dan fastext, serta tampilkan 4 kata yang similar

[45]:

```
tokenized_text
```

```
[45]: [['Ada', 'motor', 'ada', 'sepeda'],
       ['Semuanya', 'beroda', 'dua'],
       ['Indonesia', 'kaya', 'budaya'],
       ['Sepatutnya', 'kita', 'menjaganyaAda', 'pisang', 'ada', 'semangka'],
       ['Jika', 'dimakan', 'manis', 'rasanya'],
       ['Indonesia', 'ragam', 'budaya'],
       ['Tugas', 'kita', 'tuk', 'menjaganyaAda', 'rusa', 'ada', 'buaya'],
       ['Sungguh', 'hitam', 'warna', 'matanya'],
       ['Ada', 'adat', 'seribu', 'bahasa'],
       ['Kita', 'wajib', 'menghormatinyaAlat', 'timbang', 'pucuknya', 'patah']]
```

```
[23]: from gensim.models import Word2Vec
      from sklearn.manifold import TSNE
      import matplotlib.pyplot as plt
      import numpy as np
```

```
[81]: model_w2v = Word2Vec(sentences=tokenized_text, vector_size=10, window=5,␣
        ↪min_count=1, workers=4)
```

Vector Size = 10, yang berarti jumlah dimensi vektor yang tercipta adalah 10, window = 5 berarti kata yang sebelum dan sesudah kata target dipilih adalah 5 kata, min_count = 1 berarti menghitung kemunculan angka jika pernah dalam 1 window yang sama

```
[82]: model_w2v
```

```
[82]: <gensim.models.word2vec.Word2Vec at 0x79e341bb3fd0>
```

```
[83]: words = list(model_w2v.wv.index_to_key)
      vector_W2V = [model_w2v.wv[word] for word in words]
      vector_W2V = np.array(vector_W2V)
```

```
[83]: '\nindex_to_key buat menampilkan daftar kata yang ada pada sebuah model\nlakukan
      komperhensi untuk mengubah kata menjadi vektor dengan model_W2V\nubah menjadi
      numpy array\n'
```

```
[84]: tsne = TSNE(n_components=2, perplexity=min(5, len(vector_W2V)-1),␣
        ↪random_state=42)
      vectors_tsne = tsne.fit_transform(vector_W2V)
```

```
[85]: words
```

```
[85]: ['ada',
       'Ada',
       'budaya',
       'menjaganyaAda',
       'kita',
       'Indonesia',
       'Jika',
       'semangka',
       'pisang',
       'Sepatutnya',
       'kaya',
       'manis',
       'dua',
       'beroda',
       'Semuanya',
       'sepeda',
       'motor',
       'dimakan',
       'patah',
       'pucuknya',
       'ragam',
       'timbang',
       'menghormatinyaAlat',
       'wajib',
       'Kita',
       'bahasa',
       'seribu',
       'adat',
       'matanya',
       'warna',
       'hitam',
       'Sungguh',
       'buaya',
       'rusa',
       'tuk',
       'Tugas',
       'rasanya']
```

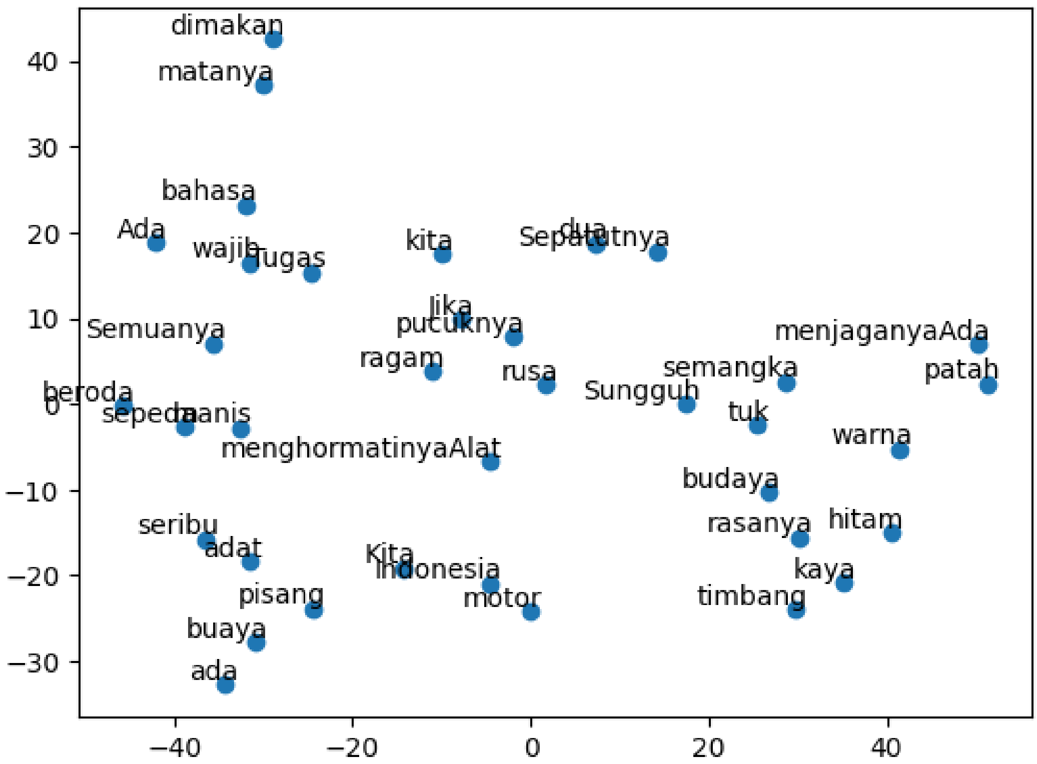
[86]:

```
tsne = TSNE(n_components=2, perplexity=min(5, len(vector_W2V)-1),␣
 ↪random_state=42)
vectors_tsne = tsne.fit_transform(vector_W2V)
```

[87]:

```
plt.scatter(vectors_tsne[:, 0], vectors_tsne[:, 1])
for i, word in enumerate(words):
    plt.annotate(word, xy=(vectors_tsne[i, 0], vectors_tsne[i, 1]), xytext=(5,␣
 ↪2), textcoords='offset points', ha='right')

plt.show()
```



0.0.3 Gensim Fasttext

```
[88]: from gensim.models import FastText

      model_fasttext = FastText(sentences=tokenized_text, vector_size=10, window=5,␣
        ↪min_count=1, workers=4)
```

Vector Size = 10, yang berarti jumlah dimensi vektor yang tercipta adalah 10, window = 5 berarti kata yang sebelum dan sesudah kata target dipilih adalah 5 kata, min_count = 1 berarti menghitung kemunculan angka jika pernah dalam 1 window yang sama

```
[89]: words = list(model_fasttext.wv.index_to_key)
      vector_fasttext = [model_fasttext.wv[word] for word in words]
      vector_fasttext = np.array(vector_fasttext)
```

```
[90]: words
```

```
[90]: ['ada',
       'Ada',
       'budaya',
       'menjaganyaAda',
       'kita',
       'Indonesia',
       'Jika',
       'semangka',
       'pisang',
       'Sepatutnya',
       'kaya',
       'manis',
       'dua',
       'beroda',
       'Semuanya',
       'sepeda',
       'motor',
       'dimakan',
       'patah',
       'pucuknya',
       'ragam',
       'timbang',
       'menghormatinyaAlat',
       'wajib',
       'Kita',
       'bahasa',
       'seribu',
       'adat',
       'matanya',
       'warna',
       'hitam',
       'Sungguh',
       'buaya',
       'rusa',
       'tuk',
       'Tugas',
       'rasanya']
```

```
[91]: vector_fasttext
```

```
[91]: array([[-1.25113665e-03, -8.09163903e-04,  1.75516326e-02,
          1.31290602e-02,  3.47573683e-03, -1.26387766e-02,
          2.04694271e-02,  1.21637539e-03, -3.49255987e-02,
          2.57294290e-02],
        [ 1.48618435e-02, -2.40702927e-03,  7.52956094e-03,
          2.57575810e-02,  2.39870325e-03,  1.72808636e-02,
          1.98359992e-02,  8.95017851e-03, -1.20136049e-02,
         -7.72224413e-03],
        [-8.22256377e-04,  9.93958022e-03,  2.63124947e-02,
         -1.07634102e-03, -5.74612571e-03, -9.14274075e-04,
          2.10393090e-02,  1.52405407e-02, -2.65046395e-02,
         -4.04378865e-03],
        [-1.09174950e-02,  7.69271096e-03,  2.97038537e-03,
          1.52244112e-02,  9.40200221e-03, -4.22893651e-03,
         -3.27407615e-03, -3.81246064e-04, -3.15309200e-03,
          8.83153267e-03],
        [ 9.56547819e-03,  1.06474468e-02, -1.64442509e-02,
         -2.14710776e-02, -1.31667107e-02,  2.18622130e-03,
          8.05088901e-04,  3.26556936e-02,  8.56247637e-03,
          2.71646143e-03],
        [ 3.49326129e-03,  1.23286918e-02, -4.38162033e-03,
          6.58719381e-03, -4.72532306e-03,  1.61616094e-02,
         -1.79558725e-03,  1.63602605e-02, -1.18681369e-03,
          9.95372329e-03],
        [ 4.03211778e-03, -1.08528659e-02, -1.60141867e-02,
         -9.30025801e-03, -1.38010690e-02, -5.14535420e-03,
         -1.69215687e-02,  1.77991632e-02,  1.61163807e-02,
          3.87283950e-03],
        [ 3.69653944e-03,  8.78876355e-03, -2.98325196e-02,
         -6.03486830e-03, -2.44263829e-05, -1.66414566e-02,
          7.81284703e-04,  1.47665795e-02, -7.44142744e-04,
         -1.07585704e-02],
        [ 3.24002071e-03,  8.78071506e-03, -6.21060655e-03,
          2.92813871e-02, -9.07777529e-03,  3.52265127e-03,
         -1.13805458e-02,  2.78433715e-03,  1.98864639e-02,
          5.91545925e-03],
        [-4.49058972e-03, -3.25808465e-03, -3.88778583e-03,
          2.72000674e-03,  7.49632483e-03,  9.68157593e-03,
         -1.95903431e-05, -1.28001226e-02,  1.11779468e-02,
          1.60584915e-02],
        [-9.50592291e-03,  1.62764695e-02,  1.31005878e-02,
          1.23470910e-02, -1.73002314e-02, -1.09297242e-02,
          1.62182271e-03,  1.75409894e-02, -3.18265483e-02,
         -2.00219220e-03],
        [ 3.76114243e-04,  1.25257717e-02, -3.50066274e-02,
         -4.90217889e-03, -2.76620556e-02,  8.54299311e-03,
          1.11321742e-02,  3.90928332e-03,  1.03704678e-02,
         -2.29065940e-02],
        [ 4.98481328e-03, -3.13614905e-02,  4.82083485e-02,
         -1.59553085e-02,  1.32966926e-02,  8.34394619e-03,
         -3.94655354e-02, -4.07275977e-03,  9.84537136e-03,
         -4.95939702e-03],
        [ 1.29381716e-02,  1.07926792e-02,  5.22777950e-03,
          1.82438232e-02,  1.11746211e-02, -1.29754900e-03,
          2.03358177e-02, -5.61056845e-03,  2.43983772e-02,
          4.82169120e-03],
        [-5.61148534e-03,  5.53717930e-03, -2.88886819e-02,
         -3.56929586e-03,  1.01888431e-02,  2.31418777e-02,
          2.30681058e-03,  1.40880859e-02, -1.72564164e-02,
         -8.98727309e-03],
        [ 3.41984385e-04,  6.87668333e-03,  5.99887036e-03,
          1.85879835e-04,  2.81921849e-02,  2.73036640e-02,
         -1.49489027e-02, -2.32211314e-03, -2.62589450e-03,
         -2.17543487e-02],
        [ 9.02393193e-04,  7.91450776e-03, -8.80122170e-05,
         -7.90704228e-03, -9.00337100e-03,  1.48238719e-03,
         -1.17229959e-02, -2.11363249e-02, -1.36270728e-02,
          2.44562235e-02],
        [-1.11429980e-02,  1.19633880e-02,  2.66193086e-03,
          8.52034986e-03, -2.15456188e-02, -5.40271262e-03,
          8.43463838e-03,  9.55146644e-03, -1.02997474e-04,
          4.60577244e-03],
        [-2.98542040e-03, -1.44970249e-02,  3.63697968e-02,
         -3.23853940e-02, -6.28803950e-03,  8.00817739e-03,
         -2.13621762e-02,  5.11264522e-03,  3.89692262e-02,
          1.74162220e-02],
        [ 1.03558311e-02, -1.34925060e-02, -3.56992095e-04,
          1.26246046e-02,  2.26175482e-03,  6.83069229e-05,
         -4.05886397e-03,  1.13191428e-02, -5.38536254e-03,
         -1.32668577e-02],
        [ 1.42360488e-02,  8.57265480e-03,  1.87631845e-02,
         -1.74919143e-02, -1.98790953e-02,  1.24506298e-02,
          1.39407511e-03,  1.48821017e-02, -1.55165447e-02,
          2.06305794e-02],
        [ 5.28624980e-03,  1.36254299e-02,  6.87308004e-03,
         -1.68968085e-03,  1.12079773e-02, -3.02548124e-03,
          3.49997990e-02, -4.56453394e-03,  7.37116765e-03,
         -1.70274843e-02],
        [-5.88513538e-03,  1.10818762e-02, -1.36447381e-02,
         -1.88284717e-03, -7.03947060e-03,  1.63788069e-03,
         -1.50822010e-02, -6.01579435e-03, -8.29538796e-03,
         -1.18995644e-02],
        [-1.20533556e-02, -2.00309232e-02,  1.06614335e-02,
         -3.85599234e-03, -9.20056645e-03, -1.93906855e-02,
         -3.75129585e-03,  1.03374319e-02, -2.43183337e-02,
          9.84442793e-03],
        [ 1.84623133e-02, -1.72869768e-02, -2.87325215e-02,
          5.25578391e-03, -9.13506560e-03,  3.35717946e-03,
          1.73928682e-02,  9.61204246e-03, -1.26932384e-02,
         -1.71245709e-02],
        [ 1.56393845e-03, -9.75805707e-03,  1.16993561e-02,
         -2.19308380e-02,  3.22214002e-03, -6.15113555e-03,
          1.36662284e-02,  3.61557934e-03, -1.38277130e-03,
         -9.86804441e-03],
        [ 1.16895121e-02, -8.58356617e-03,  1.29016489e-02,
          1.36875743e-02,  3.44097801e-03,  6.53153192e-03,
          5.41074853e-03,  3.85731924e-03, -3.49987261e-02,
          9.88968462e-03],
        [ 1.93132786e-03, -7.87625962e-04, -1.35385757e-03,
          2.05406547e-02,  1.67574659e-02,  3.93114351e-02,
          2.17561051e-02,  7.50822388e-03, -1.66116226e-02,
          4.06403281e-02],
        [ 9.51803941e-03,  1.12752346e-02,  4.36230330e-03,
         -4.61783307e-03, -3.90713708e-03,  6.03654468e-03,
         -2.80023692e-03, -2.68369238e-03, -3.11724935e-03,
          6.97846664e-03],
        [ 2.55088639e-02, -3.93934269e-03,  1.86985172e-02,
          4.25742328e-04,  9.97562055e-03,  1.21914847e-04,
         -1.07252626e-02,  1.75399857e-03, -5.89621533e-03,
         -3.39576080e-02],
        [-7.92319048e-03, -1.32868877e-02,  2.20983033e-03,
          9.08263959e-03, -1.32254958e-02, -1.79737173e-02,
          1.85959768e-02, -9.07976273e-03,  1.96422767e-02,
         -2.59221606e-02],
        [ 1.72145497e-02,  1.02796648e-02,  1.10267131e-02,
         -2.03832164e-02,  1.90154295e-02, -4.14850842e-03,
          3.74809396e-03,  7.47180264e-03,  4.84385731e-04,
         -9.99370310e-03],
        [ 2.18979851e-03,  2.25598887e-02,  2.79758833e-02,
          1.78549513e-02, -2.38263663e-02, -9.79087129e-03,
          7.39875436e-03,  1.17094861e-02,  9.43007227e-03,
         -2.35839467e-03],
        [ 5.39753912e-03,  7.57646805e-04,  1.15841962e-02,
          3.85397812e-03,  3.40118073e-02,  1.20816370e-02,
         -2.87718722e-03, -1.64645240e-02,  8.23854748e-03,
         -7.08077801e-03],
        [ 2.58237291e-02,  2.23857742e-02,  3.47553607e-04,
         -1.33251899e-03, -1.27763823e-02,  4.60770540e-02,
         -1.83284301e-02, -1.20509882e-02, -1.51078934e-02,
          6.43160287e-03],
        [-4.64767357e-03,  1.07335467e-02,  1.81359518e-02,
         -1.48293171e-02,  2.23765528e-04, -6.32374221e-03,
         -8.22894275e-03,  1.04549530e-04,  1.97026953e-02,
         -3.80389281e-02],
        [-1.32671222e-02,  2.61295252e-02,  8.74356367e-03,
         -1.22878971e-02,  2.97816913e-03,  3.09000234e-03,
          2.38043685e-02, -4.03687218e-03, -2.57601985e-03,
         -7.14053679e-03]], dtype=float32)
```

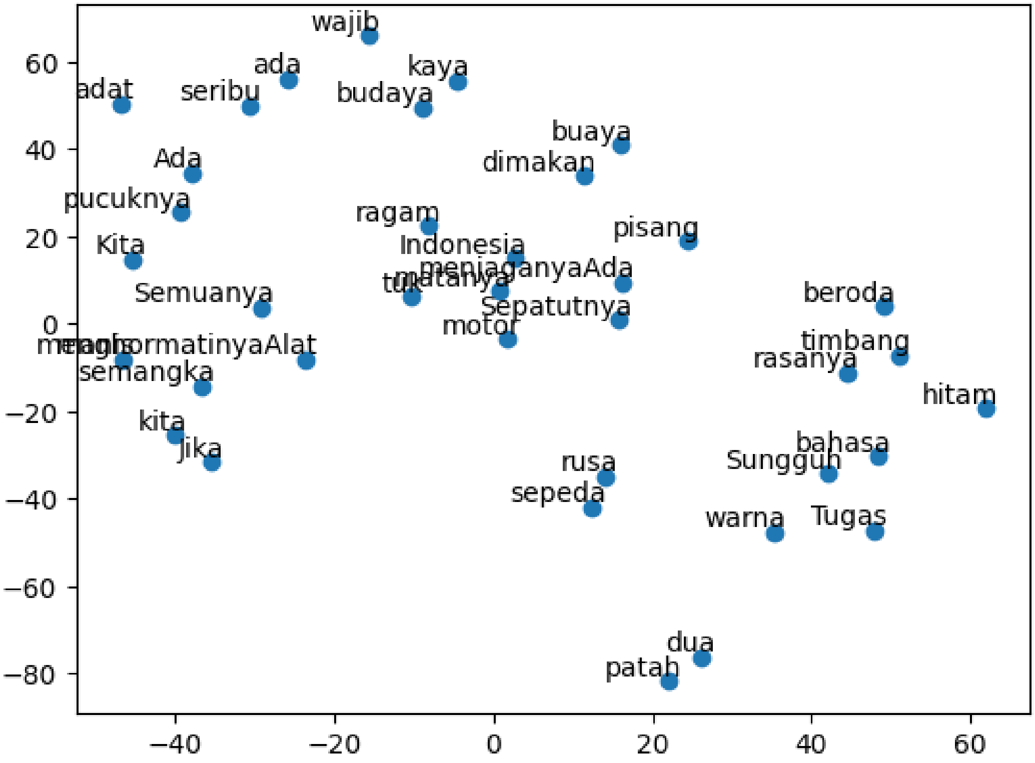
[92]:

```
tsne = TSNE(n_components=2, perplexity=min(5, len(vector_fasttext)-1),␣
 ↪random_state=42)
vectors_tsne = tsne.fit_transform(vector_fasttext)
```

[93]:

```
plt.scatter(vectors_tsne[:, 0], vectors_tsne[:, 1])
for i, word in enumerate(words):
    plt.annotate(word, xy=(vectors_tsne[i, 0], vectors_tsne[i, 1]), xytext=(5,␣
 ↪2), textcoords='offset points', ha='right')

plt.show()
```



0.0.4 Perbandingan hasil Word2Vec dan Fasttext

[94]:

```
# representasi kata makan
word_w2v = model_w2v.wv['Indonesia']
word_fasttext = model_fasttext.wv['Indonesia']

print("Word2Vec:", word_w2v)
print("FastText:", word_fasttext)
```

```
Word2Vec: [-0.08157917  0.04495798 -0.04137076  0.00824536  0.08498619
-0.04462177
  0.045175   -0.0678696  -0.03548489  0.09398508]
FastText: [ 0.00349326  0.01232869 -0.00438162  0.00658719 -0.00472532
0.01616161
 -0.00179559  0.01636026 -0.00118681  0.00995372]
```

0.0.5 Mencari kata yang similar

[95]:

```
# Gunakan model Word2Vec atau FastText yang telah dilatih
similar_words_w2v = model_w2v.wv.most_similar('Indonesia', topn=4)
similar_words_fasttext = model_fasttext.wv.most_similar('Indonesia', topn=4)

print(f"Word2Vec - Kata serupa dengan 'Indonesia':{similar_words_w2v}")
print(f"FastText - Kata serupa dengan 'Indonesia':{similar_words_fasttext}")
```

```
Word2Vec - Kata serupa dengan 'Indonesia':[('motor', 0.5916882157325745),
('menghormatinyaAlat', 0.5115841627120972), ('rusa', 0.5002487897872925),
('Kita', 0.46192681789398193)]
FastText - Kata serupa dengan 'Indonesia':[('adat', 0.5899585485458374), ('tuk',
0.5600650906562805), ('kita', 0.5257759690284729), ('ragam',
0.4961792230606079)]
```

Interpretasi Dari 2 model tersebut, pencarian 4 kata terbaik relatif jatuh kepada FastText. Dari kata ‘Indonesia yang diuji kata nya cukup mirip dan dekat dengan ’adat’, ‘ragam’ yang cocok dengan konteks ‘Indonesia’.