Tugas Praktikum Natural Language Processing

Nama : Gavrilla Claudia

NIM : 21110004

Kelas : S1SD02A

Gensim Word2Vec

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

```
corpus = [["saya", "makan", "nasi"],
          ["kamu", "minum", "air"],
          ["mereka", "tidur", "nyenyak"]]
```

```
model_w2v = Word2Vec(sentences=corpus, vector_size=10, window=5, min_count=1, workers=4)
```

```
model_w2v
```

```
<gensim.models.word2vec.Word2Vec at 0x7a71fdabe9e0>
```

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

```
"""
index_to_key buat menampilkan daftar kata yang ada pada sebuah model
lakukan komperhensi untuk mengubah kata menjadi vektor dengan model_W2V
ubah menjadi numpy array
"""
```

```
'\nindex_to_key buat menampilkan daftar kata yang ada pada sebuah model\nlakukan kom
perhensi untuk mengubah kata menjadi vektor dengan model_W2V\nubah menjadi numpy arr
ay\n'
```

```
words
```

```
['nyenyak', 'tidur', 'mereka', 'air', 'minum', 'kamu', 'nasi', 'makan', 'saya']
```

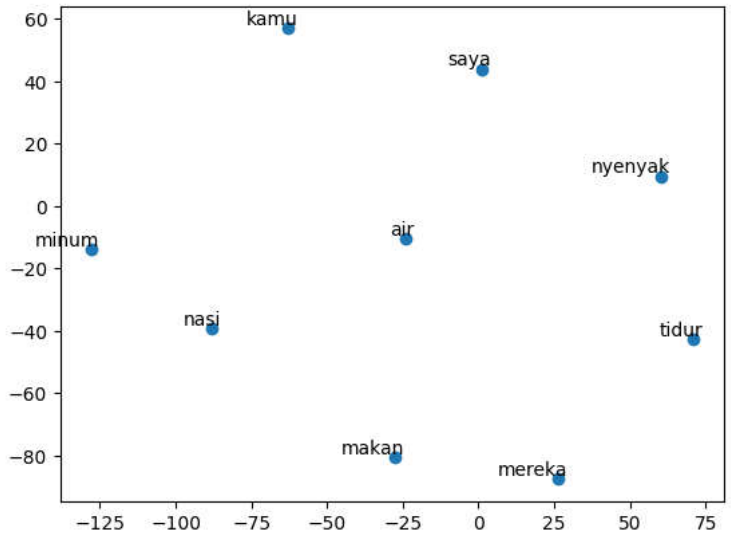
```
vector_W2V
```

```
array([[-0.00536227,  0.00236431,  0.0510335 ,  0.09009273, -0.0930295 ,
        -0.07116809,  0.06458873,  0.08972988, -0.05015428, -0.03763372],
       [ 0.07380505, -0.01533471, -0.04536613,  0.06554051, -0.0486016 ,
        -0.01816018,  0.0287658 ,  0.00991874, -0.08285215, -0.09448818],
       [ 0.07311766,  0.05070262,  0.06757693,  0.00762866,  0.06350891,
        -0.03405366, -0.00946401,  0.05768573, -0.07521638, -0.03936104],
       [-0.07511582, -0.00930042,  0.09538119, -0.07319167, -0.02333769,
        -0.01937741,  0.08077437, -0.05930896,  0.00045162, -0.04753734],
       [-0.0960355 ,  0.05007293, -0.08759586, -0.04391825, -0.000351  ,
        -0.00296181, -0.0766124 ,  0.09614743,  0.04982058,  0.09233143],
       [-0.08157917,  0.04495798, -0.04137076,  0.00824536,  0.08498619,
        -0.04462177,  0.045175  , -0.0678696 , -0.03548489,  0.09398508],
       [-0.01577653,  0.00321372, -0.0414063 , -0.07682689, -0.01508008,
         0.02469795, -0.00888027,  0.05533662, -0.02742977,  0.02260065],
       [ 0.05455794,  0.08345953, -0.01453741, -0.09208143,  0.04370552,
         0.00571785,  0.07441908, -0.00813283, -0.02638414, -0.08753009],
       [-0.00856557,  0.02826563,  0.05401429,  0.07052656, -0.05703121,
         0.0185882 ,  0.06088864, -0.04798051, -0.03107261,  0.0679763 ]],
      dtype=float32)
```

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

```
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()
```



Gensim Fasttext

```
from gensim.models import FastText

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

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

```
words
```

```
['nyenyak', 'tidur', 'mereka', 'air', 'minum', 'kamu', 'nasi', 'makan', 'saya']
```

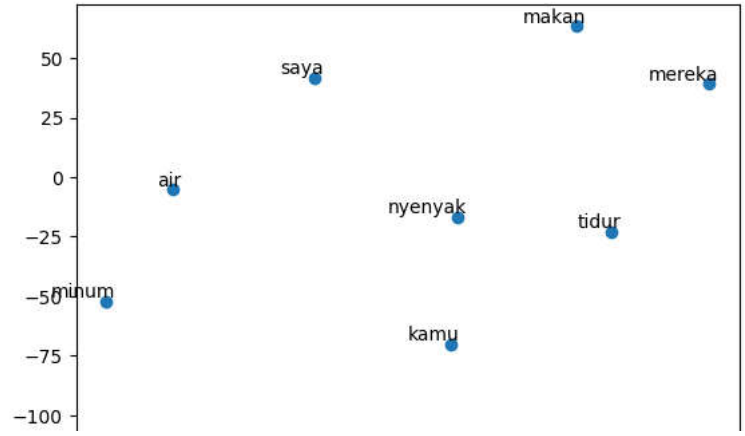
```
vector_fasttext
```

```
array([[-0.01429317,  0.01492771,  0.00758504,  0.01978461,  0.00878036,
         0.00145506,  0.01148306, -0.005627  , -0.0130773 , -0.00364854],
       [-0.00261523,  0.00424853, -0.00227577,  0.0236958 ,  0.01184708,
        -0.00236321,  0.02446533,  0.00204978, -0.00103972, -0.00385612],
       [ 0.032973  ,  0.01859976,  0.0035758 ,  0.0069081 ,  0.00761674,
        -0.00697827, -0.0112101 , -0.00180466, -0.02073616,  0.01218386],
       [-0.02763563, -0.01061732,  0.0007771 , -0.01857746, -0.01209558,
         0.02224461,  0.00321715,  0.00488243, -0.00148315, -0.01218848],
       [-0.01895148, -0.00480462, -0.00724613, -0.02728259, -0.01647825,
         0.00127699, -0.029727  , -0.00408955, -0.01080866,  0.02590004],
       [-0.00979524,  0.01303313, -0.00603292, -0.01418934,  0.023735  ,
        -0.00240588,  0.00939171, -0.00231475, -0.03419039,  0.00880968],
       [-0.013013  ,  0.01699816, -0.01976314,  0.01454479,  0.01577191,
         0.02566642, -0.03318036,  0.02793288, -0.03028855,  0.01318964],
       [ 0.02740508,  0.01998663, -0.00337412,  0.0085386 , -0.02439561,
        -0.01201405,  0.00575076,  0.00314926, -0.00707525,  0.00872171],
       [ 0.00104189,  0.01247097,  0.03406876,  0.00478015, -0.02215288,
         0.01165398, -0.00829301,  0.00749769,  0.00172518, -0.01713193]],
      dtype=float32)
```

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

```
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()
```



Perbandingan hasil Word2Vec dan Fasttext


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

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

```
Word2Vec: [ 0.07380505 -0.01533471 -0.04536613  0.06554051 -0.0486016  -0.01816018
  0.0287658   0.00991874 -0.08285215 -0.09448818]
FastText: [-0.00261523  0.00424853 -0.00227577  0.0236958   0.01184708 -0.00236321
  0.02446533  0.00204978 -0.00103972 -0.00385612]
```

Mencari kata yang similar

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

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

```
Word2Vec - Kata serupa dengan 'makan':[('mereka', 0.42731544375419617), ('air', 0.2941223382949829), ('tidur', 0.23243053257465363)
FastText - Kata serupa dengan 'makan':[('mereka', 0.6308949589729309), ('saya', 0.18492391705513), ('tidur', 0.04897456616163254),
```

Melihat struktur model

```
# Unduh model Word2Vec pre-trained
# Code for downloading pre-trained model depends on the source (e.g., Gensim's KeyedVectors or other sources)

# Inspeksi model
print(f"Ukuran vektor: {model_w2v.vector_size}")
print(f"Jumlah kata:{len(model_w2v.wv)}")
print(f"Parameter W2V: {model_w2v}")
```

```
Ukuran vektor: 10
Jumlah kata:9
Parameter W2V: Word2Vec<vocab=9, vector_size=10, alpha=0.025>
```

```
# Unduh model Word2Vec pre-trained
# Code for downloading pre-trained model depends on the source (e.g., Gensim's KeyedVectors or other sources)

# Inspeksi model
print("Ukuran vektor:", model_fasttext.vector_size)
print("Jumlah kata:", len(model_fasttext.wv))
print("Parameter W2V:", model_fasttext)
```

```
Ukuran vektor: 10
Jumlah kata: 9
Parameter W2V: FastText<vocab=9, vector_size=10, alpha=0.025>
```

Latihan

```
import pandas as pd

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

df.head()
```

	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

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


teks = df['teks'].values.tolist()


teks = ''.join(teks)


teks = clear(teks)


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]   Unzipping tokenizers/punkt.zip.
```

```
teks = teks[:10]


teks
```

```
['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 .']
```

```
teks_token = [word_tokenize(sentence.lower()) for sentence in teks]
teks_token
```

```
[['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', '.']]
```

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

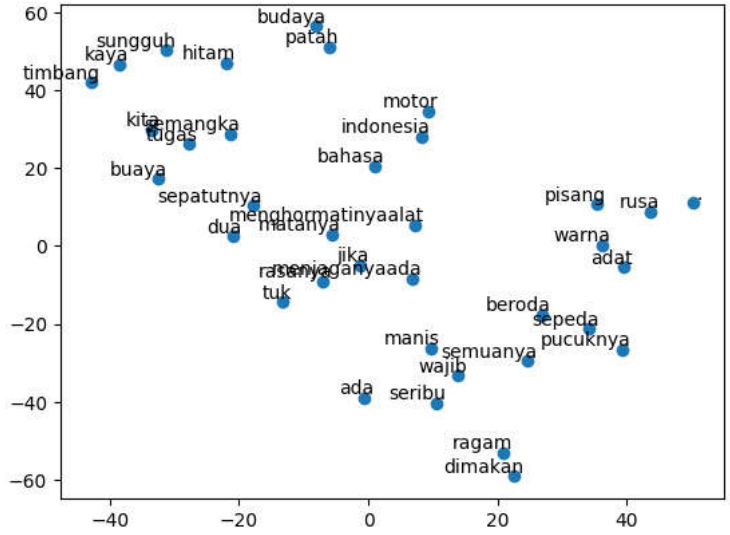
```
model_w2v = Word2Vec(sentences=teks_token, vector_size=10, window=5, min_count=1, workers=4)
```

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

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

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()
```



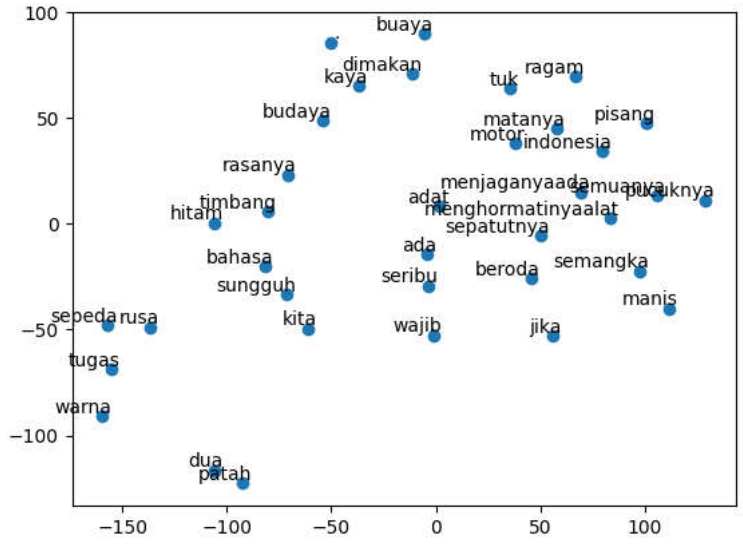

```
model_fasttext = FastText(sentences=teks_token, vector_size=10, window=5, min_count=1, workers=4)
```

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

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

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()
```




```
# 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('manis', topn=4)

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

```
:[('motor', 0.5913802981376648), ('menghormatinyaalat', 0.5115896463394165), ('tuk', 0.5002295970916748), ('bahasa', 0.4622492790222
semangka', 0.6411186456680298), ('kita', 0.43431928753852844), ('menghormatinyaalat', 0.4182778596878052), ('.', 0.41521984338760376
```

Interpretasi Hasil :

Word2Vec - Kata serupa dengan 'Indonesia'(4 kata teratas yang similar dengan kata 'Indonesia') :

Motor (0.591): Kata "motor" memiliki tingkat kemiripan sekitar 59.1%. Menghormatinyaalat (0.512): Kata yang agak unik ini memiliki tingkat kemiripan sekitar 51.2%. Tuk (0.500): Kata "tuk" memiliki tingkat kemiripan sekitar 50.0%. Bahasa (0.462): Kata "bahasa" memiliki tingkat kemiripan sekitar 46.2%. Artinya bahwa ada hubungan semantik tertentu atau penggunaan bersama antara kata "Indonesia" dan keempat kata tersebut dalam beberapa konteks.

FastText - Kata serupa dengan 'manis' (4 kata teratas yang similar dengan kata 'manis') :

Semangka (0.641): Kata "semangka" memiliki tingkat kemiripan sekitar 64.1%. Kita (0.434): Kata "kita" memiliki tingkat kemiripan sekitar 43.4%. Menghormatinyaalat (0.418): Kata yang sama dengan hasil Word2Vec, "menghormatinyaalat," memiliki tingkat kemiripan sekitar 41.8%. Tanda titik (.) memiliki tingkat kemiripan sekitar 41.5%. Artinya bahwa ada hubungan semantik tertentu atau penggunaan bersama antara kata "manis" dan keempat kata tersebut dalam beberapa konteks.