

Data Extraction

```
pip install PyPDF2
```

```
Collecting PyPDF2
```

```
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
```

```
0:00:01 0.0/232.6 kB ? eta -:-:--
174.1/232.6 kB 5.1 MB/s eta
232.6/232.6 kB 5.1
MB/s eta 0:00:00
```

```
import PyPDF2
```

```
import pandas as pd
```

```
def extract_sentences(pdf_path):
```

```
    data = {'Sentence': []}
```

```
    with open(pdf_path, 'rb') as file:
```

```
        pdf_reader = PyPDF2.PdfReader(file)
```

```
        for page_num in range(len(pdf_reader.pages)):
```

```
            page = pdf_reader.pages[page_num]
```

```
            text = page.extract_text()
```

```
            sentences = text.split('.')
```

```
            data['Sentence'].extend(sentences)
```

```
    return pd.DataFrame(data)
```

```
pdf_path = '/content/drive/MyDrive/nlp/MALIN_KUNDANG.pdf'
```

```
df = extract_sentences(pdf_path)
```

```
print(df.head(10))
```

	Sentence
0	MALIN KUNDANG \nPada suatu waktu, hiduplah seb...
1	Keluarga tersebut terdiri \ndari ayah, ibu da...
2	Karena kondisi keuangan keluarga yang \nmembr...
3	\nMaka tinggallah si Malin dan ibunya di gubug...
4	Semingg u, dua minggu, sebulan, dua \nbulan b...
5	Sehingga ibunya harus menggantikan posisi aya...
6	\nMalin termasuk anak yang cerdas tetapi sedi...
7	Ia sering mengejar ayam dan \nmemukunya den...
8	Suatu hari ketika Malin sedang me ngejar ayam...
9	Luka terse but menjadi berbekas dilengannya \...

lowercase string

```
df['Sentence'] = df['Sentence'].apply(lambda x: x.lower())
df['Sentence']
```

```
0      malin kundang \npada suatu waktu, hiduplah seb...
1      keluarga tersebut terdiri \ndari ayah, ibu da...
2      karena kondisi keuangan keluarga yang \nmempr...
3      \nmaka tinggallah si malin dan ibunya di gubug...
4      semingg u, dua minggu, sebulan, dua \nbulan b...
      ...
3734     tubuh prabu dewata cengkar dilempar aji saka ...
3735     \naji saka kemudian dinobatkan menjadi raja m...
3736           i a memboyong ayahnya ke \nistana
3737     berkat pemerintahan yang adil dan bijaksana, ...
3738
Name: Sentence, Length: 3739, dtype: object
```

```
import re
```

menghilangkan new line (\n)

```
def clean_text1(text):
    return re.sub(r'\n([a-z])', r' \1', text)
```

```
df['Sentence'] = df['Sentence'].apply(clean_text1)
df['Sentence']
```

```
0      malin kundang  pada suatu waktu, hiduplah sebu...
1      keluarga tersebut terdiri  dari ayah, ibu dan...
2      karena kondisi keuangan keluarga yang  mempri...
3      maka tinggallah si malin dan ibunya di gubug ...
4      semingg u, dua minggu, sebulan, dua  bulan ba...
      ...
3734     tubuh prabu dewata cengkar dilempar aji saka ...
3735     aji saka kemudian dinobatkan menjadi raja me...
3736           i a memboyong ayahnya ke  istana
3737     berkat pemerintahan yang adil dan bijaksana, ...
3738
Name: Sentence, Length: 3739, dtype: object
```

```
import string
```

```
def remove_digits_and_punctuation(text):
    cleaned_text = re.sub(r'[\d' + re.escape(string.punctuation) +
    ']', '', text)
    return cleaned_text
```

```
df['Sentence'] = df['Sentence'].apply(remove_digits_and_punctuation)
df['Sentence']
```

```

0      malin kundang pada suatu waktu hiduplah sebua...
1      keluarga tersebut terdiri dari ayah ibu dan ...
2      karena kondisi keuangan keluarga yang mempri...
3      maka tinggallah si malin dan ibunya di gubug ...
4      semingg u dua minggu sebulan dua bulan bahka...
      ...
3734    tubuh prabu dewata cengkar dilempar aji saka ...
3735    aji saka kemudian dinobatkan menjadi raja me...
3736    i a memboyong ayahnya ke istana
3737    berkat pemerintahan yang adil dan bijaksana a...
3738
Name: Sentence, Length: 3739, dtype: object

```

menghapus stopwords

```

import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')

stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(text):
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in
stop_words]
    return ' '.join(filtered_words)

df['Sentence'] = df['Sentence'].apply(remove_stopwords)
df['Sentence']

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

0      malin kundang hiduplah keluarga nelayan pesisir...
1      keluarga ayah anak laki-laki nama malin kundang
2      kondisi keuangan keluarga memprihatinkan sang ...
3      tinggallah si malin ibunya gubug
4      semingg u minggu sebulan ayah malin kampung ha...
      ...
3734    tubuh prabu dewata cengkar dilempar aji saka j...
3735    aji saka dinobatkan raja medan kamulan
3736    i a memboyong ayahnya istana
3737    berkat pemerintahan adil bijaksana aji saka me...
3738
Name: Sentence, Length: 3739, dtype: object

```

One Hot Encoding

```
OHE = pd.get_dummies(df['Sentence'].str.split(expand=True).stack(),
drop_first=True).groupby(level=0).max()
OHE['Sentence']= df['Sentence']
```

```
OHE.head(10)
```

	aaa	aaaa	aan	aat	abad	abadi	abangnya	abdi	abdinya	abu	...
\											
0	0	0	0	0	0	0	0	0	0	0	...
1	0	0	0	0	0	0	0	0	0	0	...
2	0	0	0	0	0	0	0	0	0	0	...
3	0	0	0	0	0	0	0	0	0	0	...
4	0	0	0	0	0	0	0	0	0	0	...
5	0	0	0	0	0	0	0	0	0	0	...
6	0	0	0	0	0	0	0	0	0	0	...
7	0	0	0	0	0	0	0	0	0	0	...
8	0	0	0	0	0	0	0	0	0	0	...
9	0	0	0	0	0	0	0	0	0	0	...

	yelamatkan	yet	yik	yikan	yir	yosaku	yuk	yut	zam	\
0		0	0		0		0	0	0	
1		0	0		0		0	0	0	
2		0	0		0		0	0	0	
3		0	0		0		0	0	0	
4		0	0		0		0	0	0	
5		0	0		0		0	0	0	
6		0	0		0		0	0	0	
7		0	0		0		0	0	0	
8		0	0		0		0	0	0	
9		0	0		0		0	0	0	

	Sentence
0	malin kundang hiduplah keluarga nelayan pesisi...
1	keluarga ayah anak lakilaki nama malin kundang
2	kondisi keuangan keluarga memprihatinkan sang ...
3	tinggallah si malin ibunya gubug
4	semingg u minggu sebulan ayah malin kampung ha...
5	ibunya menggantikan posisi ayah malin mencari ...
6	malin anak cerdas nakal

```
[10 rows x 4884 columns]
```

```
OHE[OHE['malin']==1]
```

	aaa	aaaa	aan	aat	abad	abadi	abangnya	abdi	abdinya
abu	...	\							
0	0	0	0	0	0	0	0	0	0
0	...								
1	0	0	0	0	0	0	0	0	0
0	...								
3	0	0	0	0	0	0	0	0	0
0	...								
4	0	0	0	0	0	0	0	0	0
0	...								
5	0	0	0	0	0	0	0	0	0
0	...								
6	0	0	0	0	0	0	0	0	0
0	...								
8	0	0	0	0	0	0	0	0	0
0	...								
10	0	0	0	0	0	0	0	0	0
0	...								
12	0	0	0	0	0	0	0	0	0
0	...								
13	0	0	0	0	0	0	0	0	0
0	...								
14	0	0	0	0	0	0	0	0	0
0	...								
15	0	0	0	0	0	0	0	0	0
0	...								
16	0	0	0	0	0	0	0	0	0
0	...								
17	0	0	0	0	0	0	0	0	0
0	...								
18	0	0	0	0	0	0	0	0	0
0	...								

0 ...

	yelamatkan	yet	yik	yikan	yir	yosaku	yuk	yut	zam	\
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	
31	0	0	0	0	0	0	0	0	0	
32	0	0	0	0	0	0	0	0	0	
33	0	0	0	0	0	0	0	0	0	
34	0	0	0	0	0	0	0	0	0	
35	0	0	0	0	0	0	0	0	0	
37	0	0	0	0	0	0	0	0	0	
38	0	0	0	0	0	0	0	0	0	
40	0	0	0	0	0	0	0	0	0	
41	0	0	0	0	0	0	0	0	0	
42	0	0	0	0	0	0	0	0	0	
43	0	0	0	0	0	0	0	0	0	
44	0	0	0	0	0	0	0	0	0	
45	0	0	0	0	0	0	0	0	0	
47	0	0	0	0	0	0	0	0	0	
49	0	0	0	0	0	0	0	0	0	
50	0	0	0	0	0	0	0	0	0	
51	0	0	0	0	0	0	0	0	0	

Sentence

0 malin kundang hiduplah keluarga nelayan pesisir...
1 keluarga ayah anak laki-laki nama malin kundang
3 tinggallah si malin ibunya gubug
4 semingg u minggu sebulan ayah malin kampung ha...

```

5  ibunya menggantikan posisi ayah malin mencari ...
6                                malin anak cerdas nakal
8  malin me ngejar ayam tersandung batu lengan ka...
10 beranjak dewasa malin kundang kasihan ibunya b...
12 malin tertarik ajakan nakhoda kapal dagang dul...
13     malin kundang mengutarakan maksudnya ibunya
14 ibunya s emula setuju maksud malin kundang mal...
15     bekal perlengkapan malin dermaga diantar ibunya
16 anakku engkau berhasil orang berkecukupan kau ...
17 kapal dinaiki malin diiringi lambaian tangan m...
18 kapal malin kundang bany ak belajar ilmu pelay...
19 perjalanan kapal dinaiki malin kundang serang ...
22 malin kundang beruntung dibunuh ba jak laut pe...
23 malin kundang terkatungkatung ditengah laut ak...
24 sisa tenaga ad a malin kundang berjalan desa t...
25 sesampainya desa te rsebut malin kundang ditolong
27     desa malin terdampar desa ya ng subur
28 keuletan kegigihannya malin kelamaa n berhasil...
30 kaya raya malin kundang m empersunting gadis i...
31 berita malin kundang kaya raya te menikah mali...
32 malin kundang bersyukur gembira anaknya berhasil
33 malin kundang perg i dermaga anaknya pulang ka...
34 menikah malin istrinya pelayaran kapal indah d...
35 malin kundang menunggui anaknya kapal indah ma...
37     berdiri anaknya malin kundang beserta istrinya
38                                malin kundang turun kapal
40 ibunya belas luka dilengan kanan orang sema ki...
41 malin kundang anakku meng kau pergi mengirimka...
42 malin kundang melepaskan pelukan ibunya mendor...
43 wanita sembarangan mengaku ibuku malin kundang...
44 malin kundang purapura mengenali ibunya malu i...
45     wanita ibumu istri malin kundang
47 mendengar pernyataan diperlakukan semenamena o...
49 kemarahannya memuncak malin menengadahkan tang...
50 angin bergemuruh kencang badai dahsyat menghan...
51 tubuh malin kundang perlahan kaku la makelamaa...

```

[40 rows x 4884 columns]

Hash Vectorization

```

import hashlib

def hash_vectoring(text, vector_size):
    vector = [0] * vector_size
    hashed_text = hashlib.sha256(text.encode()).hexdigest()
    hash_subset = hashed_text[:vector_size]
    hash_integer = int(hash_subset, 16)

```



```

    index = hash_integer % vector_size
    vector[index] = 1
    return vector

vector_size = 10

HASH = df['Sentence'].apply(lambda x: hash_vectoring(x, vector_size))

```

Vectorisasi Hash adalah proses mengubah data teks menjadi representasi numerik, dalam hal ini, berupa bilangan biner. Hasil dari proses Hash ini adalah bahwa setiap baris kalimat akan lebih dekat dengan satu fitur khusus. Jumlah fitur yang digunakan dalam vektor ini adalah 10, sehingga menghasilkan output sebagai berikut.

```

HASH.head(10)

0    [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
1    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
2    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
3    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
4    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
5    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
6    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
7    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
8    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
9    [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
Name: Sentence, dtype: object

```

Co-occurrence Matrix

```

import numpy as np
import nltk
from nltk import bigrams
import itertools

def co_occurrence_matrix(corpus):
    vocab = set(corpus)
    vocab = list(vocab)
    vocab_to_index = {word: i for i, word in enumerate(vocab)}

    bi_grams = list(bigrams(corpus))

    bigram_freq = nltk.FreqDist(bi_grams).most_common(len(bi_grams))

    co_occurrence_matrix = np.zeros((len(vocab), len(vocab)))

    for bigram in bigram_freq:
        current = bigram[0][1]
        previous = bigram[0][0]

```

```

        count = bigram[1]
        pos_current = vocab_to_index[current]
        pos_previous = vocab_to_index[previous]
        co_occurrence_matrix[pos_current][pos_previous] = count

    co_occurrence_matrix = np.matrix(co_occurrence_matrix)

    return co_occurrence_matrix, vocab_to_index

merged =
list(itertools.chain.from_iterable(df['Sentence'].str.split()))
matrix, vocab_to_index = co_occurrence_matrix(merged)
CoMatrixFinal = pd.DataFrame(matrix, index=vocab_to_index,
columns=vocab_to_index)

```

CoMatrixFinal

	cepatlah	kendi	dibeliorang	w	melempar	berbicar
seret \						
cepatlah	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
kendi	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
dibeliorang	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
w	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
melempar	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
...
...						
lelap	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
kepanasan	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
memikirkannya	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
ris	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
criing	0.0	0.0	0.0	0.0	0.0	0.0
0.0						

	dala	nama	nyamar	...	tukarkan	kecerdikanmu
belajar \						
cepatlah	0.0	0.0	0.0	...	0.0	0.0
0.0						
kendi	0.0	0.0	0.0	...	0.0	0.0
0.0						
dibeliorang	0.0	0.0	0.0	...	0.0	0.0
0.0						

w	0.0	0.0	0.0	...	0.0	0.0
0.0						
melempar	0.0	0.0	0.0	...	0.0	0.0
0.0						
...
.						
lelap	0.0	0.0	0.0	...	0.0	0.0
0.0						
kepanasan	0.0	0.0	0.0	...	0.0	0.0
0.0						
memikirkannya	0.0	0.0	0.0	...	0.0	0.0
0.0						
ris	0.0	0.0	0.0	...	0.0	0.0
0.0						
criing	0.0	0.0	0.0	...	0.0	0.0
0.0						
	petani	antera	lelap	kepanasan	memikirkannya	ris
criing						
cepatlah	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
kendi	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
dibeliorang	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
w	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
melempar	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
...
...						
lelap	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
kepanasan	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
memikirkannya	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
ris	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
criing	0.0	0.0	0.0	0.0	0.0	0.0
0.0						
[4884 rows x 4884 columns]						

Co-occurrence memiliki manfaat untuk menemukan kata-kata yang sering muncul bersama-sama dalam suatu konteks. Sebagai contoh, kata 'malin' sering muncul berdekatan (baik sebelum maupun sesudah) dengan beberapa kata berikut:

```
CoMatrixFinal['malin'][CoMatrixFinal['malin']==1]
```

```

terdampar      1.0
me              1.0
dermaga        1.0
menengadahkan  1.0
anak           1.0
mendesak       1.0
kelamaa        1.0
kun           1.0
tertarik       1.0
ibunya         1.0
kampung        1.0
bersembunyi   1.0
diiringi       1.0
istrinya       1.0
mencari        1.0
Name: malin, dtype: float64

```

Word2Vec

```

from gensim.models import Word2Vec

df['Words'] = df['Sentence'].apply(lambda x: x.split())
print(df['Words'].tolist()[:10])

[['malin', 'kundang', 'hiduplah', 'keluarga', 'nelayan', 'pesisir',
'pantai', 'wilayah', 'sumatra'], ['keluarga', 'ayah', 'anak',
'lakilaki', 'nama', 'malin', 'kundang'], ['kondisi', 'keuangan',
'keluarga', 'memprihatinkan', 'sang', 'ayah', 'memutuskan', 'mencari',
'nafkah', 'negeri', 'seberang', 'mengarungi', 'lautan', 'luas'],
['tinggallah', 'si', 'malin', 'ibunya', 'gubug'], ['semingg', 'u',
'minggu', 'sebulan', 'ayah', 'malin', 'kampung', 'halamannya'],
['ibunya', 'menggantikan', 'posisi', 'ayah', 'malin', 'mencari',
'nafkah'], ['malin', 'anak', 'cerdas', 'nakal'], ['mengejar', 'ayam',
'memukulnya', 'sapu'], ['malin', 'me', 'ngejar', 'ayam', 'tersandung',
'batu', 'lengan', 'kanannya', 'luka', 'terkena', 'batu'], ['luka',
'terse', 'but', 'berbekas', 'dilengannya', 'hilang']]

model_w2v = Word2Vec(sentences = df['Words'], vector_size = 10, window
= 10, min_count=1, workers = 4)

import numpy as np

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

```

Proses pembuatan model Word2Vec dimulai dengan membagi kata-kata dalam kalimat dan menyusunnya dalam bentuk array. Selanjutnya, model Word2Vec dibangun dengan mengatur beberapa parameter, seperti `vector_size = 10` yang menunjukkan dimensi dari hasil vektor

sebanyak 10, window = 10 yang berarti model akan mempertimbangkan 10 kata sebelum dan sesudah kata target, min_count = 1 yang berarti suatu kata akan diikutsertakan dalam model jika muncul minimal 1 kali dalam kalimat, dan worker = 4 adalah jumlah penggunaan CPU selama proses pelatihan.

```
vector_w2v[:10]
array([[ 0.251922, -0.11022416,  0.45972633,  0.20221688, -
 0.04031605,
        -0.03846484,  0.6194736,  0.315462, -0.65768456, -
 0.46572644],
       [ 0.2907108, -0.10653698,  0.26680416,  0.14920624, -
 0.00447011,
        0.01464039,  0.4689915,  0.15735221, -0.56694376, -0.4338867
],
       [ 0.32760414, -0.03444803,  0.41464064,  0.10122054,
 0.11228708,
        -0.01429018,  0.49900535,  0.22153349, -0.61829954, -0.4118636
],
       [ 0.12171634, -0.10144906,  0.42784008,  0.01395194,
 0.02455168,
        0.00229254,  0.5187373,  0.08504419, -0.4820302, -
 0.37841967],
       [ 0.07960425, -0.00793875,  0.14055945,  0.00981479,
 0.03342197,
        0.02010322,  0.27353156,  0.21993661, -0.3244387, -
 0.16718508],
       [ 0.1180898, -0.02934219,  0.25698733,  0.08026673,
 0.12705988,
        -0.03464335,  0.46846923,  0.06999288, -0.47760534, -
 0.22227265],
       [ 0.18440005, -0.08590213,  0.26909512,  0.02887606,
 0.05990948,
        0.06973686,  0.45147935,  0.21800344, -0.5407024, -
 0.34523234],
       [ 0.25331444,  0.00186541,  0.26844332, -0.02366709,
 0.08810518,
        0.0332459,  0.48320135,  0.12690769, -0.4578632, -
 0.40079436],
       [ 0.18821248, -0.05170504,  0.337436,  0.14407213, -
 0.02220995,
        0.04362818,  0.47140488,  0.0849027, -0.45922935, -
 0.24860662],
       [ 0.22193748, -0.07652396,  0.33702427,  0.07871325,
 0.13374388,
        0.06326725,  0.32818934,  0.069916, -0.44144264, -
 0.25919676]],
      dtype=float32)
```

```
similar_words_w2v = model_w2v.wv.most_similar('ayah',
                                              topn = 4)
print(f"Word2Vec = Kata serupa dengan 'ayah':{similar_words_w2v}")

Word2Vec = Kata serupa dengan 'ayah':[('puteri', 0.9867724180221558),
('tibatiba', 0.9850067496299744), ('pulang', 0.9842220544815063),
('kau', 0.9832701086997986)]
```

Hasilnya kata 'ayah' mirip atau dekat dengan kata 'puteri', 'tiba-tiba', 'pulang', 'kau'. Hasil yang relevan hanyalah kata 'puteri', beberapa kata lain tidak cocok dan mungkin perlu dilakukan pengolahan data atau pembangunan model yang lebih kompleks.

Fasttext

```
from gensim.models import FastText
model_fasttext = FastText(sentences = df['Words'], vector_size = 10,
window = 10, min_count=1, workers = 4)

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

vector_fasttext[:10]

array([[ 8.0181968e-01,  3.0262951e-02, -4.1384715e-01,
 3.5681853e+00,
        -3.9012995e-01,  2.8208992e+00,  2.1007819e+00,  9.6046394e-
01,
        -1.7808685e-02,  1.1142895e+00],
 [ 6.0955554e-01,  2.2561384e-02, -3.6665747e-01,
 2.7334082e+00,
        -3.5184693e-01,  2.1739292e+00,  1.5914333e+00,  7.2932410e-
01,
        -2.2679012e-02,  8.0591291e-01],
 [ 6.5547514e-01,  7.2192587e-02, -3.2155257e-01,
 2.8439081e+00,
        -2.9700077e-01,  2.2026634e+00,  1.6281202e+00,  7.3201185e-
01,
        -5.0856642e-02,  8.0316681e-01],
 [ 1.7275116e+00,  8.8775724e-02, -9.5265841e-01,
 7.7173505e+00,
        -8.5131645e-01,  6.1072264e+00,  4.5953236e+00,
 2.0472546e+00,
        6.4193018e-02,  2.3552134e+00],
 [ 6.5140676e-01,  3.5441775e-02, -3.5823247e-01,
 2.9497511e+00,
        -3.2846043e-01,  2.3023543e+00,  1.7710357e+00,  7.6198530e-
01,
```

```

-1.4131800e-03, 9.0873307e-01],
[ 7.6604480e-01, 3.6848359e-02, -4.6473339e-01,
3.4592581e+00,
-3.4480992e-01, 2.7206931e+00, 2.0795324e+00, 9.0035212e-
01,
3.4333335e-04, 1.1041213e+00],
[ 8.2066810e-01, 4.6676029e-02, -4.5175508e-01,
3.6906028e+00,
-4.1664782e-01, 2.9439673e+00, 2.1939235e+00,
1.0059720e+00,
-3.7665378e-02, 1.1504297e+00],
[ 6.4113659e-01, 1.3441878e-02, -3.5103172e-01,
2.7606101e+00,
-2.8794280e-01, 2.1920357e+00, 1.6428335e+00, 7.0702410e-
01,
1.2017380e-03, 8.3670354e-01],
[ 3.0236050e-01, -2.0261794e-02, -1.8813825e-01,
1.5619526e+00,
-1.9878648e-01, 1.1755016e+00, 8.7406766e-01, 3.7485713e-
01,
-6.4420931e-02, 5.1787519e-01],
[ 1.4446962e+00, 4.7537073e-02, -7.7105522e-01,
6.2483540e+00,
-6.8891197e-01, 4.9431663e+00, 3.7183456e+00,
1.6588931e+00,
4.0685423e-02, 1.9526007e+00]], dtype=float32)

similar_words_fasttext = model_fasttext.wv.most_similar('ayah', topn =
4)
print(f"Fasttext = Kata serupa dengan 'ayah':
{similar_words_fasttext}")

Fasttext = Kata serupa dengan 'ayah':[('putriterpeso',
0.9999829530715942), ('sergah', 0.9999828338623047), ('kurak',
0.9999768137931824), ('perlengkapan', 0.9999760389328003)]

```

Seluruh hasil diatas lebih buruk ketimbang model sebelumnya, karena tidak ada satupun kata yang berkaitan dengan 'ayah'.