

## Import Data

```
In [19]: !pip install -r '../requirements.txt'

import nltk

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

path_to_csv = '/Users/macbookair/Documents/SEM 3/NLP/data/dataset_m
```

ERROR: Could not open requirements file: [Errno 2] No such file or directory: '../requirements.txt'

```
[nltk_data] Downloading package punkt to
[nltk_data]   /Users/macbookair/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /Users/macbookair/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]   date!
```

```
In [20]: import re
import random
import pandas as pd
import numpy as np
from tensorflow.keras.utils import plot_model
from keras.models import load_model
import matplotlib.pyplot as plt

pd.set_option('mode.chained_assignment', None)
```

```
In [21]: data=pd.read_csv(path_to_csv, nrows=20)
```

In [22]: data.head()

Out [22]:

	Question_ID	Questions	Jawaban
0	1590140	Apa yang dimaksud dengan penyakit mental?	Penyakit mental adalah kondisi kesehatan yang ...
1	2110618	Siapa yang terpengaruh oleh penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...
2	6361820	Apa penyebab penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...
3	9434130	Apa sajakah tanda-tanda peringatan penyakit me...	Gejala gangguan kesehatan mental bervariasi te...
4	7657263	Apakah penderita penyakit jiwa bisa sembuh?	Ketika penyembuhan dari penyakit mental, ident...

## Data preprocessing

```
In [23]: def cleaning(text):
text = re.sub(r'@[A-Za-a0-9]+', ' ', text)
text = re.sub(r'#[A-Za-z0-9]+', ' ', text)
text = re.sub(r"http\S+", ' ', text)
text = re.sub(r'[0-9]+', ' ', text)
text = re.sub(r"[-()\"#/@;:<>{}'+=~|.,_]", ' ', text)
text = re.sub(r"(?:\@|https?\:\/\/)\S+", ' ', text)
text = re.sub(r'^\x00-\x7f', ' ', text)
text = re.sub(r'\n', ' ', text)
text = text.strip(' ')
text = text.lower()
return text

data['Questions'] = data['Questions'].apply(cleaning)
data['Jawaban'] = data['Jawaban'].apply(cleaning)
```

```
In [24]: import nltk
nltk.download('stopwords')
```

```
import nltk
nltk.download('punkt')
```

```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]   /Users/macbookair/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]   /Users/macbookair/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]   /Users/macbookair/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

Out[24]: True

```
In [25]: import nltk
from nltk.corpus import stopwords
stopwords.words('indonesian')

def remove_stopwords(text):
    stop_words = stopwords.words('indonesian')

    return ' '.join(c for c in nltk.word_tokenize(text) if c not in
```

```
In [26]: from nltk.stem import WordNetLemmatizer

wordnet_lemma = WordNetLemmatizer()

def lemma(text):
    lemmatize_words = [wordnet_lemma.lemmatize(word) for sent in nl
    return ' '.join(lemmatize_words)
```

```
In [27]: data['Questions'] = data['Questions'].apply(remove_stopwords)
data['Jawaban'] = data['Jawaban'].apply(remove_stopwords)
data['Questions'] = data['Questions'].apply(lemma)
data['Jawaban'] = data['Jawaban'].apply(lemma)
data.head()
```

Out [27]:

	Question_ID	Questions	Jawaban
0	1590140	penyakit mental ?	penyakit mental kondisi kesehatan mengganggu p...
1	2110618	terpengaruh penyakit mental ?	penyakit mental mempengaruhi orang dewasa amer...
2	6361820	penyebab penyakit mental ?	penyakit mental mempengaruhi orang dewasa amer...
3	9434130	sajakah tandatanda peringatan penyakit mental ?	gejala gangguan kesehatan mental bervariasi te...
4	7657263	penderita penyakit jiwa sembuh ?	penyembuhan penyakit mental identifikasi pengo...

```
In [28]: pasangan=[]

for i in range(data.shape[0]):
    pasangan.append(((data['Questions'][i]),data['Jawaban'][i]))
```

```
In [29]: pasangan

n membantu diagnosis ? program melibatkan layanan spesialis ? bert
anggung mengoordinasikan layanan ? lihat peran keluarga program pe
rawatan ? ak dimiliki keluarga orang orang menyediakan perawatan ?
obat diagnosis ? pengalaman miliki memperlakukan orang penyakit ?
lakukan membantu perawatan ?'),
('perbedaan profesional kesehatan mental ?',
'jenis profesional kesehatan mental variasi penyedia layanan mem
bingungkan memiliki tingkat pendidikan pelatihan memiliki bidang k
eahlian berbeda menemukan profesional sesuai kebutuhan penelitia
n'),
('menemukan ahli kesehatan mental anak ?',
'nyaman profesional anak keberhasilan perawatan menemukan profes
ional sesuai kebutuhan penelitian'),
('terlibat pengobatan ketahui ?',
'perawatan langkah individu keluarga melanjutkan keterlibatan pr
ose perawatan penyebab penyakit khusus ? diagnosis gejala gejala ?
memasukkan pemeriksaan fisik neurologis ? te ujian tambahan rekome
ndasikan ? menyarankan pendapat independen psikiater ? program pen
gobatan membantu diagnosis ? program melibatkan layanan spesialis
? bertanggung mengoordinasikan layanan ? lihat peran keluarga prog
```

```
In [30]: pertanyaan = []
         jawaban = []
         input_tokens = set()
         target_tokens = set()

         for line in pasangan:
             tanya, jawab = line[0], line[1]
             pertanyaan.append(tanya)

             jawab = " ".join(re.findall(r"[\w']+|[\^\s\w]", jawab))
             jawab = '<START> ' + jawab + ' <END>'
             jawaban.append(jawab)

             for token in re.findall(r"[\w']+|[\^\s\w]", tanya):
                 if token not in input_tokens:
                     input_tokens.add(token)
             for token in jawab.split():
                 if token not in target_tokens:
                     target_tokens.add(token)

         input_tokens = sorted(list(input_tokens))
         target_tokens = sorted(list(target_tokens))
         num_encoder_tokens = len(input_tokens)
         num_decoder_tokens = len(target_tokens)
```

```
In [31]: pertanyaan
```

```
Out[31]: ['penyakit mental ?',
          'terpengaruh penyakit mental ?',
          'penyebab penyakit mental ?',
          'sajakah tandatanda peringatan penyakit mental ?',
          'penderita penyakit jiwa sembuh ?',
          'lakukan mengenal memiliki gejala gangguan jiwa ?',
          'menemukan ahli kesehatan mental anak ?',
          'pilihan pengobatan tersedia ?',
          'terlibat pengobatan ketahui ?',
          'perbedaan profesional kesehatan mental ?',
          'menemukan ahli kesehatan mental anak ?',
          'terlibat pengobatan ketahui ?',
          'bantuan ?',
          'ketahui pengobatan ?',
          'minum obat sembuh berhenti meminumnya ?',
          'bantuan membayar pengobatan ?',
          'mencari terapi',
          'mempelajari jenisjenis perawatan kesehatan mental ?',
          'jenisjenis profesional kesehatan mental ?',
          'mencari kelompok pendukung ?']
```

```
In [32]: jawaban
```

```
Out[32]: ['<START> penyakit mental kondisi kesehatan mengganggu pikiran emo
          si hubungan fungsi dikaitkan kesusahan berkurangnya kapasitas terl
```

ibat kegiatan kehidupan sehari penyakit mental jatuh rangkaian ke arah ringan mengganggu aspek kehidupan fobia ujung spektrum ter letak penyakit mental serius mengakibatkan gangguan fungsional utama gangguan kehidupan sehari gangguan depresi berat skizofrenia gangguan bipolar mengharuskan orang menerima perawatan rumah sakit penyakit mental kondisi medis hubungannya karakter kecerdasan kemauan diabetes kelainan pankreas penyakit mental kondisi medis biologi otak mengobati diabetes obat-obatan insulin penyakit mental diobati kombinasi obat dukungan sosial perawatan efektif persen orang menerima pengobatan mengalami pengurangan gejala peningkatan kualitas hidup perawatan orang penyakit mental mandiri sukses <END>',

'<START> penyakit mental mempengaruhi orang dewasa Amerika orang dewasa memiliki penyakit mental serius penyakit mental membedakan akan mempengaruhi terlepas jenis kelamin usia pendapatan status sosial etnis agama orientasi seksual latar belakang penyakit mental mempengaruhi kondisi populasi berbeda gangguan makan cenderung wanita gangguan defisit perhati gangguan hiperaktif lazim anak-anak usia rentan muda tua rentan penyakit mental mengejutkan individu hidup persen kondisi kesehatan mental berkembang usia identifikasi pengobatan gangguan mental sulit kepribadian normal perubahan perilaku remaja menutupi gejala-gejala kondisi kesehatan mental orang tua pengasuh menyadari fakta memperhatikan perubahan suasana hati kepribadian kebiasaan pribadi penarikan sosial anak-anak gangguan emosional serius sedih <END>',

'<START> penyakit mental mempengaruhi orang dewasa Amerika orang dewasa memiliki penyakit mental serius penyakit mental membedakan akan mempengaruhi terlepas jenis kelamin usia pendapatan status sosial etnis agama orientasi seksual latar belakang penyakit mental mempengaruhi kondisi populasi berbeda gangguan makan cenderung wanita gangguan defisit perhati gangguan hiperaktif lazim anak-anak usia rentan muda tua rentan penyakit mental mengejutkan individu hidup persen kondisi kesehatan mental berkembang usia identifikasi pengobatan gangguan mental sulit kepribadian normal perubahan perilaku remaja menutupi gejala-gejala kondisi kesehatan mental orang tua pengasuh menyadari fakta memperhatikan perubahan suasana hati kepribadian kebiasaan pribadi penarikan sosial anak-anak gangguan emosional serius sedih <END>',

'<START> gejala gangguan kesehatan mental bervariasi tergantung jenis tingkat keparahan kondisi daftar gejala gangguan kesehatan mental gejala berganda diekspresikan orang dewasa bingung berpikir kesedihan mudah marah tahan rendah suasana hati ketakutan mengkhawatirkan kecemasan penarikan sosial perubahan dramatis kebiasaan makan tidur perasaan marah kuat delusi halusinasi mendengar meningkatkan ketidakmampuan mengatasi kegiatan sehari pikiran bunuh penolakan fisik penyalahgunaan narkoba dan atau alkohol anak praremaja penyalahgunaan narkoba dan atau alkohol ketidakmampuan mengatasi aktivitas sehari perubahan kebiasaan tidur dan atau makan keluhan fisik menentang otoritas melewati sekolah mencuri merusak properti ketakutan intens menambah berat badan suasana hati negatif tahan seringkali nafsu makan buruk pikiran mati ledakan kemarahan anak perubahan kinerja sekolah nilai buruk upaya kuat kekhawatiran kecemasan hiperaktif mimpi buruk gigih ketidaktaatan menerus dan atau perilaku agresif tantrum temperamen <END>',

'<START> penyembuhan penyakit mental identifikasi pengobatan berdasarkan sifat penyakit perawatan efektif tersedia jenis perawatan orang terkena dampak proaktif sepenuhnya terlibat prose pemulihan orang penyakit mental didiagnosis dirawat merespons mengalami kembalinya gejala pemantauan pengelolaan gangguan cermat menjalani kehidupan terpenuhi produktif <END>',

'<START> situs web menggantikan nasihat profesional mendorong memiliki gejala berbicara teman anggota keluarga mencari nasihat profesional kesehatan mental cepat kondisi kesehatan mental diidentifikasi dirawat cepat jalan pemulihan mengenal mengalami berasumsi sembuh biarkan peduli pilihan perawatan tersedia membantu sembuh bicaralah profesional penasihat kesehatan mental pikir teman anggota keluarga mengalami gejala kondisi kesehatan mental orang terkasih mendukung cenderung mencari bantuan <END>',

'<START> nyaman profesional anak keberhasilan perawatan menemukan profesional sesuai kebutuhan penelitian mencari penyedia daerah <END>',

'<START> jenis obat penyakit fisik pilihan perawatan berbeda tersedia individu penyakit mental perawatan berbeda orang berbeda menemukan cocok anak <END>',

'<START> perawatan langkah individu keluarga terlibat terlibat prose perawatan penyebab penyakit khusus ? diagnosis gejala gejala ? memasukkan pemeriksaan fisik neurologis ? tes ujian tambahan rekomendasi ? menyarankan pendapat independen psikiater ? program pengobatan membantu diagnosis ? program melibatkan layanan spesialis ? bertanggung mengoordinasikan layanan ? lihat peran keluarga program perawatan ? ak dimiliki keluarga orang orang menyediakan perawatan ? obat diagnosis ? pengalaman miliki memperlakukan orang penyakit ? lakukan membantu perawatan ? <END>',

'<START> jenis profesional kesehatan mental variasi penyedia layanan membingungkan memiliki tingkat pendidikan pelatihan memiliki bidang keahlian berbeda menemukan profesional sesuai kebutuhan penelitian <END>',

'<START> nyaman profesional anak keberhasilan perawatan menemukan profesional sesuai kebutuhan penelitian <END>',

'<START> perawatan langkah individu keluarga melanjutkan keterlibatan prose perawatan penyebab penyakit khusus ? diagnosis gejala gejala ? memasukkan pemeriksaan fisik neurologis ? tes ujian tambahan rekomendasi ? menyarankan pendapat independen psikiater ? program pengobatan membantu diagnosis ? program melibatkan layanan spesialis ? bertanggung mengoordinasikan layanan ? lihat peran keluarga program perawatan ? ak dimiliki keluarga orang orang menyediakan perawatan ? obat diagnosis ? efek biologis obat harapkan ? risiko terkait obatnya ? cepat obatnya efektif ? pengalaman miliki memperlakukan orang penyakit ? lakukan membantu perawatan ? <END>',

'<START> mencari bantuan tergantung sifat danatau gejala cocok seringkali terbaik berbicara percayai kekhawatiran anggota keluarga teman pendeta penyedia layanan kesehatan profesional memiliki dukungan sosial penyembuhan penyakit mental rujukan rekomendasi praktisi kesehatan mental tepercaya cari sumber daya kesehatan mental daerah orang nebraska menyediakan layanan berbicara mendengarkan membantu perjalanan pemulihan ketiga orang menemukan peer mendukung alat bermanfaat membantu pemulihan organisasi menawarkan kelompok p

```
In [33]: input_features_dict = dict([(token, i) for i, token in enumerate(input_features)])
target_features_dict = dict([(token, i) for i, token in enumerate(target_features)])

reverse_input_features_dict = dict([(i, token) for token, i in input_features_dict.items()])
reverse_target_features_dict = dict([(i, token) for token, i in target_features_dict.items()])
```



```
In [34]: input_features_dict
```

```
Out[34]: {'?': 0,  
          'ahli': 1,  
          'anak': 2,  
          'bantuan': 3,  
          'berhenti': 4,  
          'gangguan': 5,  
          'gejala': 6,  
          'jenisjenis': 7,  
          'jiwa': 8,  
          'kelompok': 9,  
          'kesehatan': 10,  
          'ketahui': 11,  
          'lakukan': 12,  
          'membayar': 13,  
          'memiliki': 14,  
          'meminumnya': 15,  
          'mempelajari': 16,  
          'mencari': 17,  
          'menemukan': 18,  
          'mengenal': 19,  
          'mental': 20,  
          'minum': 21,  
          'obat': 22,  
          'penderita': 23,  
          'pendukung': 24,  
          'pengobatan': 25,  
          'penyakit': 26,  
          'penyebab': 27,  
          'perawatan': 28,  
          'perbedaan': 29,  
          'peringatan': 30,  
          'pilihan': 31,  
          'profesional': 32,  
          'sajakah': 33,  
          'sembuh': 34,  
          'tandatanda': 35,  
          'terapi': 36,  
          'terlibat': 37,  
          'terpengaruh': 38,  
          'tersedia': 39}
```

## Vektorisasi (Encoder Decoder)

```
In [35]: max_encoder_seq_length = max([len(re.findall(r"[\w']+|^[\s\w]", tan
max_decoder_seq_length = max([len(re.findall(r"[\w']+|^[\s\w]", jaw

encoder_input_data = np.zeros(
    (len(pertanyaan), max_encoder_seq_length, num_encoder_tokens),
    dtype='float32')
decoder_input_data = np.zeros(
    (len(pertanyaan), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')
decoder_target_data = np.zeros(
    (len(pertanyaan), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')
for line, (tanya, jawab) in enumerate(zip(pertanyaan, jawaban)):
    for timestep, token in enumerate(re.findall(r"[\w']+|^[\s\w]",
        encoder_input_data[line, timestep, input_features_dict[token]

    for timestep, token in enumerate(jawab.split()):
        decoder_input_data[line, timestep, target_features_dict[token]
        if timestep > 0:
            decoder_target_data[line, timestep - 1, target_features
```

In [36]: encoder\_input\_data

```
Out[36]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [1., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 1., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               ...,

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [1., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [1., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]]) dtype=float32)
```

In [37]: `decoder_target_data`

```
Out[37]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

                [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

                [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

                ...,

                [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

                [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

                [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]])
```

## Training Model

```
In [38]: from tensorflow import keras
from keras.layers import Input, LSTM, Dense
from keras.models import Model
dimensionality = 256
batch_size = 10
epochs = 5000

#Encoder
encoder_inputs = Input(shape=(None, num_encoder_tokens))
encoder_lstm = LSTM(dimensionality, return_state=True)
encoder_outputs, state_hidden, state_cell = encoder_lstm(encoder_inputs)
encoder_states = [state_hidden, state_cell]

#Decoder
decoder_inputs = Input(shape=(None, num_decoder_tokens))
decoder_lstm = LSTM(dimensionality, return_sequences=True, return_state=True)
decoder_outputs, decoder_state_hidden, decoder_state_cell = decoder_lstm(decoder_inputs)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)

In [39]: training_model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
```

In [40]: `training_model.summary()`

Model: "model"

Layer (type) Connected to	Output Shape	Param #
=====		
input_1 (InputLayer) []	[(None, None, 40)]	0
input_2 (InputLayer) []	[(None, None, 401)]	0
lstm (LSTM) ['input_1[0][0]']	[(None, 256), (None, 256), (None, 256)]	304128
lstm_1 (LSTM) ['input_2[0][0]', 'lstm[0][1]', 'lstm[0][2]']	[(None, None, 256), (None, 256), (None, 256)]	673792
dense (Dense) ['lstm_1[0][0]']	(None, None, 401)	103057
=====		
Total params: 1080977 (4.12 MB)		
Trainable params: 1080977 (4.12 MB)		
Non-trainable params: 0 (0.00 Byte)		

In [41]: `plot_model(training_model, to_file='model_plot.png', show_shapes=True)`

You must install pydot (`pip install pydot`) and install graphviz (see instructions at <https://graphviz.gitlab.io/download/>) ([http s://graphviz.gitlab.io/download/](http://s://graphviz.gitlab.io/download/)) for plot\_model to work.

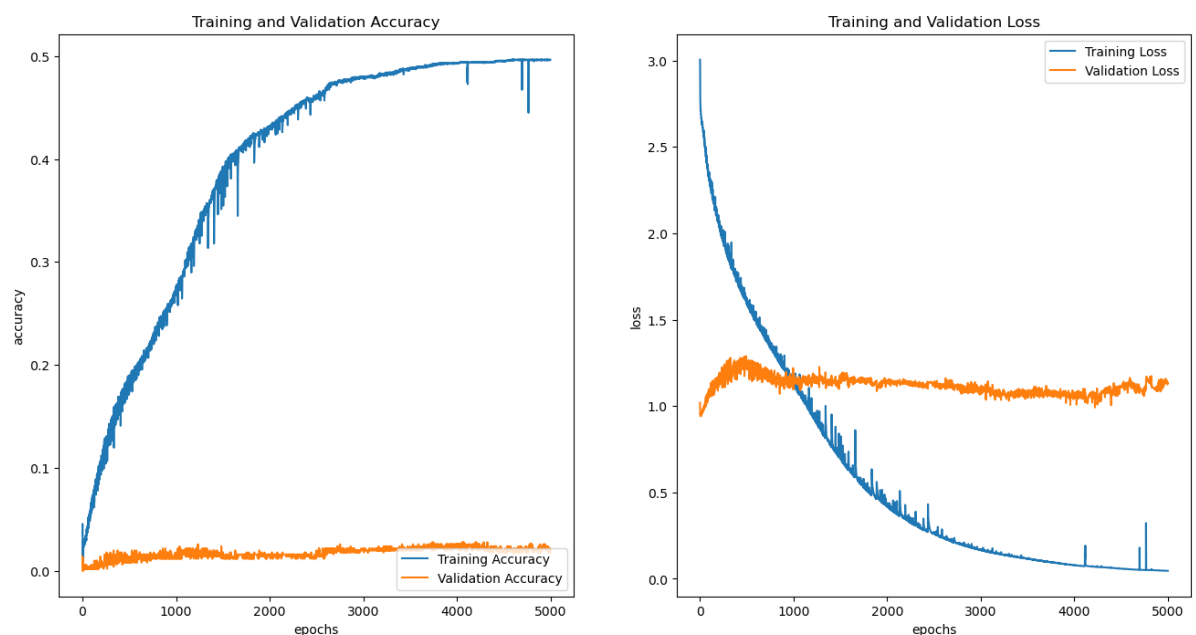
```
In [42]: training_model.compile(optimizer='rmsprop', loss='categorical_crossentropy',
history1=training_model.fit([encoder_input_data, decoder_input_data],
training_model.save('training_model.h5py'))

6 - accuracy: 0.4965 - val_loss: 1.1070 - val_accuracy: 0.0179
Epoch 4765/5000
2/2 [=====] - 1s 820ms/step - loss: 0.050
8 - accuracy: 0.4965 - val_loss: 1.1094 - val_accuracy: 0.0198
Epoch 4766/5000
2/2 [=====] - 1s 775ms/step - loss: 0.051
3 - accuracy: 0.4965 - val_loss: 1.0973 - val_accuracy: 0.0198
Epoch 4767/5000
2/2 [=====] - 1s 911ms/step - loss: 0.322
8 - accuracy: 0.4449 - val_loss: 1.1600 - val_accuracy: 0.0238
Epoch 4768/5000
2/2 [=====] - 1s 569ms/step - loss: 0.053
0 - accuracy: 0.4965 - val_loss: 1.1636 - val_accuracy: 0.0238
Epoch 4769/5000
2/2 [=====] - 1s 549ms/step - loss: 0.051
8 - accuracy: 0.4965 - val_loss: 1.1660 - val_accuracy: 0.0238
Epoch 4770/5000
2/2 [=====] - 1s 603ms/step - loss: 0.051
5 - accuracy: 0.4965 - val_loss: 1.1698 - val_accuracy: 0.0238
Epoch 4771/5000
```

```
In [43]: acc = history1.history['accuracy']
val_acc = history1.history['val_accuracy']
loss=history1.history['loss']
val_loss=history1.history['val_loss']

plt.figure(figsize=(16,8))
plt.subplot(1, 2, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.xlabel("epochs")
plt.ylabel("accuracy")

plt.subplot(1, 2, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.xlabel("epochs")
plt.ylabel("loss")
plt.show()
```



## Predictions Model

```
In [65]: from keras.models import load_model
training_model = load_model('training_model.h5py')
encoder_inputs = training_model.input[0]
encoder_outputs, state_h_enc, state_c_enc = training_model.layers[2]
encoder_states = [state_h_enc, state_c_enc]
encoder_model = Model(encoder_inputs, encoder_states)
```



```

In [66]: latent_dim = 256
         decoder_state_input_hidden = Input(shape=(latent_dim,))
         decoder_state_input_cell = Input(shape=(latent_dim,))
         decoder_states_inputs = [decoder_state_input_hidden, decoder_state_

In [68]: decoder_outputs, state_hidden, state_cell = decoder_lstm(decoder_in
         decoder_states = [state_hidden, state_cell]
         decoder_outputs = decoder_dense(decoder_outputs)

In [69]: decoder_model = Model([decoder_inputs] + decoder_states_inputs, [de

In [70]: training_model = load_model('training_model.h5py')
         encoder_inputs = training_model.input[0]
         encoder_outputs, state_h_enc, state_c_enc = training_model.layers[2]
         encoder_states = [state_h_enc, state_c_enc]
         encoder_model = Model(encoder_inputs, encoder_states)

         latent_dim = 256
         decoder_state_input_hidden = Input(shape=(latent_dim,))
         decoder_state_input_cell = Input(shape=(latent_dim,))
         decoder_states_inputs = [decoder_state_input_hidden, decoder_state_
         decoder_outputs, state_hidden, state_cell = decoder_lstm(decoder_in
         decoder_states = [state_hidden, state_cell]
         decoder_outputs = decoder_dense(decoder_outputs)
         decoder_model = Model([decoder_inputs] + decoder_states_inputs, [de

def decode_response(test_input):
    states_value = encoder_model.predict(test_input)

    target_seq = np.zeros((1, 1, num_decoder_tokens))

    target_seq[0, 0, target_features_dict['<START>']] = 1.

    decoded_sentence = ''

    stop_condition = False
    while not stop_condition:
        output_tokens, hidden_state, cell_state = decoder_model.p

        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        sampled_token = reverse_target_features_dict[sampled_toke
        decoded_sentence += " " + sampled_token

        if (sampled_token == '<END>' or len(decoded_sentence) > m
            stop_condition = True

        target_seq = np.zeros((1, 1, num_decoder_tokens))
        target_seq[0, 0, sampled_token_index] = 1.

        states_value = [hidden_state, cell_state]
    return decoded_sentence

```

```

In [71]: class ChatBot:
    negative_responses = ("no", "nope", "nah", "naw", "tidak", "nggak", "tidak", "nggak", "tidak", "nggak")
    exit_commands = ("quit", "pause", "exit", "goodbye", "bye", "keluar", "stop", "end", "finish", "over", "quit", "exit", "quit", "exit")

    def start_chat(self):
        user_response = input("Haloo, aku chatbot pintar!\n")

        if user_response in self.negative_responses:
            print("Ok, bubayy lop u!")
            return
        self.chat(user_response)

    def chat(self, reply):
        while not self.make_exit(reply):
            reply = input(self.generate_response(reply)+"\n")

    def string_to_matrix(self, user_input):
        tokens = re.findall(r"[\w']+|[\^\s\w]", user_input)
        user_input_matrix = np.zeros(
            (1, max_encoder_seq_length, num_encoder_tokens),
            dtype='float32')
        for timestep, token in enumerate(tokens):
            if token in input_features_dict:
                user_input_matrix[0, timestep, input_features_dict[token]] = 1
        return user_input_matrix

    def generate_response(self, user_input):
        input_matrix = self.string_to_matrix(user_input)
        chatbot_response = decode_response(input_matrix)
        #Remove <START> and <END> tokens from chatbot_response
        chatbot_response = chatbot_response.replace("<START>", '')
        chatbot_response = chatbot_response.replace("<END>", '')
        return chatbot_response

    def make_exit(self, reply):
        for exit_command in self.exit_commands:
            if exit_command in reply:
                print("Ok, bubayy lop u!")
                return True
        return False

chatbot = ChatBot()

```

## Demo Pertanyaan

```
In [72]: chatbot.start_chat()
```

```
penyakit mental mempengaruhi orang dewasa amerika orang dewasa me  
miliki penyakit mental serius penyakit mental membeda bedakan  
mencari terapi
```

```
1/1 [=====] - 0s 77ms/step  
1/1 [=====] - 0s 213ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 82ms/step  
1/1 [=====] - 0s 59ms/step  
1/1 [=====] - 0s 47ms/step  
1/1 [=====] - 0s 55ms/step  
1/1 [=====] - 0s 51ms/step  
1/1 [=====] - 0s 53ms/step  
1/1 [=====] - 0s 107ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 52ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 63ms/step
```

```
penyembuhan penyakit mental identifikasi pengobatan berdasarkan p  
erawatan perawatan perawatan efektif tersedia perawatan penyakit  
exit
```