

BAB V

HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil dan pembahasan dari penelitian. Sub bab hasil akan membahas mengenai sistem yang telah dibuat selama penelitian. Sub bab pembahasan akan menjelaskan mengenai sistem yang telah dibuat dengan model LSTM dan aplikasi berbasis *website* yang telah dibuat.

5.1 Hasil

Hasil dari penelitian yang telah dilakukan dalam pembuatan aplikasi untuk identifikasi anomali pada kesehatan mental. Bagian ini akan membahas hasil praproses data, pembuatan model, dan penerapan antarmuka.

5.1.1 Hasil Praproses Data

Data yang telah didapatkan dari proses data collection harus dilakukan pembersihan terlebih dahulu sebelum diproses untuk latihan model. Preprocessing teks adalah membuat dokumen ke dalam format yang mudah dimengerti, diprediksi, dan dapat dianalisis oleh mesin melalui berbagai algoritma machine learning (Tabassum & Patil, 2020).

1) Case Folding

```
import re

def case_folding(text):
    text = text.lower()

    # Menghapus karakter &#39; dan &quot; pada kalimat
    text = text.replace('&#39;', '')
    text = text.replace('&quot;', ' ')
    text = text.replace(':', '')

    # Menghapus tag HTML pada kalimat menjadi spasi
    remove_tag = re.compile('<.*?>')
    text = re.sub(remove_tag, ' ', text)

    # Menghapus link url pada kalimat
    text = re.sub(r'https?://\S+', '', text)

    # Mengganti multiple spasi menjadi satu spasi
    text = re.sub(' +', ' ', text)

    # Menghapus angka
    text = re.sub(r'\d+', '', text)
```

Gambar 5.1 Kode Sumber Untuk *Case Folding* 1

```

# Menghapus emoji
emoji_pattern = re.compile("[
    u"\U0001F600-\U0001F64F"
    u"\U0001F300-\U0001F5FF"
    u"\U0001F680-\U0001F6FF"
    u"\U0001F1E0-\U0001F1FF"
    u"\U00002500-\U00002BEF"
    u"\U00002702-\U000027B0"
    u"\U00002702-\U000027B0"
    u"\U000024C2-\U0001F251"
    u"\U0001f926-\U0001f937"
    u"\U00010000-\U0010ffff"
    u"\u2640-\u2642"
    u"\u2600-\u2B55"
    u"\u200d"
    u"\u23cf"
    u"\u23e9"
    u"\u231a"
    u"\ufe0f"
    u"\u3030"
    u"\U00010000-\U0010ffff"
    u"\u0640-\u07FF"
    u"\u0900-\u097F"
    u"\u3040-\u309F\u30A0-\u30FF"
"]+", flags=re.UNICODE)

text = emoji_pattern.sub(r'', text)

return text

print('CASE FOLDING')
print('-----')
print('Success')
df['Preprocess'] = df['Comment'].apply(case_folding)

```

Gambar 5.2 Kode Sumber Untuk Case Folding 2

Gambar 5.1 dan 5.2 adalah kode sumber yang digunakan untuk melakukan case folding. Pada fungsi ini dilakukan terlebih dahulu untuk mengubah data menjadi huruf kecil. Perubahan data menjadi huruf kecil diperlukan agar ketika data diproses menjadi mudah dan menghindari kesalahan pemrosesan data yang diakibatkan oleh huruf besar dan huruf kecil. Setelah melakukan perubahan menjadi huruf kecil, data akan dibersihkan dengan menghilangkan karakter unik yang ada dalam data, *html tag*, *link url*, spasi ganda, angka, dan emoji. Penghilangan karakter tersebut diperlukan karena karakter-karakter tersebut tidak memiliki nilai untuk dilakukan pemrosesan data nantinya. Output dari fungsi ini dapat dilihat pada tabel berikut ini.

Tabel 5.1 Hasil Case Folding

Sebelum	Sesudah
whos here with eating disorder and depression and anxiety disorder in general?? anyone 😊 hey	whos here with eating disorder and depression and anxiety disorder in general?? anyone hey hey

Tabel 5.1 merupakan hasil dari perbandingan antara sebelum dan sesudah proses *case folding* dengan menggunakan *library* ReGex. Pada contoh tersebut emoji dihilangkan oleh proses *case folding*.

2) Tokenisasi

```
def tokenization(text):
    tokens = re.findall("[\w']+)", text)
    return tokens

print('TOKENIZATION')
print('-----')
print('Success')

df['Preprocess'] = df['Preprocess'].apply(tokenization)
```

Gambar 5.3 Kode Sumber Tokenisasi

Gambar 5.3 merupakan kode sumber yang digunakan proses tokenisasi. Tokenisasi dilakukan dengan menggunakan *library* ReGex. Tokenisasi dilakukan dengan cara memisahkan setiap kata dari spasi setiap kata. Output dari fungsi ini dapat dilihat pada tabel berikut ini.

Tabel 5.2 Hasil Tokenisasi

Sebelum	Sesudah
whos here with eating disorder and depression and anxiety disorder in general?? anyone hey	[whos, here, with, eating, disorder, and, depression, and, anxiety, disorder, in, general, anyone, hey]

Tabel 5.2 merupakan hasil dari perbandingan antara sebelum dan sesudah proses tokenisasi dengan menggunakan *library* ReGex. Pada contoh tersebut kalimat dipecah menjadi potongan kata-kata.

3) Stop Words Removal

```
def stopwording(tokens):
    stop_words = set(stopwords.words('english'))
    custom_stop_words = {"youre", "youve", "youll", "youd", "shes", "thatll", "theres",
                        "dont", "shouldve", "arent", "couldnt", "didnt", "doesnt", "hadnt", "hasnt",
                        "havent", "isnt", "neednt", "shant", "shouldnt", "wasnt", "werent", "wont",
                        "wouldnt", "ur", "oh", "hi", "hello",
                        "hewwo", "hey", "video", "im", "idc", "lol", "idk", "yuh", "yes", "thet",
                        "wow", "must", "wow", "ah", "fyp", "wtf", "d", "lt", "video", "hmmmmmm", "whos", "would",
                        "wouldve"}
    stop_words.update(custom_stop_words)
    tokens = [token for token in tokens if token not in stop_words]
    return tokens
```

Gambar 5.4 Kode Sumber Stop Words Removal

Gambar 5.4 merupakan kode sumber yang dilakukan untuk *stop words removal*. Proses ini menghilangkan kata yang tidak memiliki makna atau tidak memiliki bobot. Gambar x juga menunjukkan ada nya kamus *stop words removal* yang ditambahkan secara manual. Kata-kata tersebut adalah kata-kata yang tidak ada pada kamus *library* NLTK dan menyesuaikan pada kasus yang ada pada penelitian. Output dari fungsi ini dapat dilihat pada tabel berikut ini.

Tabel 5.3 Hasil *Stop Wods Removal*

Sebelum	Sesudah
[whos, here, with, eating, disorder, and, depression, and, anxiety, disorder, in, general, anyone, hey]	[eating, disorder, depression, anxiety, disorder, general, anyone]

Tabel 5.3 merupakan hasil dari perbandingan antara sebelum dan sesudah proses *stop words removal* dengan menggunakan *library* NLTK. Pada contoh tersebut, beberapa kata telah dihilangkan berdasarkan kamus NLTK.

4) Lematisasi

```

nlp = spacy.load('en_core_web_sm')

def lemmatize_spacy(text):
    doc = nlp(text)
    lemmatized_tokens = [token.lemma_ for token in doc]
    return lemmatized_tokens

print('LEMMATIZATION')
print('-----')
print('Success')

lemmatized_comments = []
for i, item in enumerate(df['Preprocess']):
    lemmatized_text = lemmatize_spacy(" ".join(item))
    lemmatized_comments.append(lemmatized_text)
    df.at[i, 'Preprocess'] = lemmatized_text

```

Gambar 5.5 Kode Sumber Lematisasi

Gambar 5.5 merupakan kode sumber yang dilakukan untuk lematisasi. Proses ini melakukan lematisasi pada data teks di kolom praproses *dataframe* menggunakan *library SpaCy*. Proses ini mengubah kata menjadi bentuk dasar atau kamusnya, yang dapat berguna untuk tugas-tugas seperti pengambilan informasi, analisis teks, dan model pembelajaran mesin yang bekerja dengan data teks. Output dari fungsi ini dapat dilihat pada tabel berikut ini.

Tabel 5.4 Hasil Lematisasi

Sebelum	Setelah
[eating, disorder, depression, anxiety, disorder, general, anyone]	[eat, disorder, depression, anxiety, disorder, general, anyone]

Tabel 5.4 merupakan hasil dari perbandingan antara sebelum dan sesudah proses lematisasi dengan menggunakan *library SpaCy*. Pada contoh tersebut, kata *eating* diubah menjadi kata dasar, yaitu *eat*.

Setelah proses praproses sudah selesai, data hasil praproses akan disimpan dalam bentuk file berformat *json* dan dilakukan proses penyaringan data dengan syarat dalam satu data memiliki minimal 4 indeks *array*. Proses penyaringan selesai, file akan disimpan dalam file berformat *json* dan sekaligus dibuat sebagai file tokenisasi untuk latihan model nantinya dengan file berformat *pickle* dari *library pickle*. Berikut adalah kode sumber untuk menyimpan file hasil praproses dan pembuatan tokenisasi.

```
# Mengonversi DataFrame ke dalam format dictionary
comments_dict = final_df.to_dict(orient='records')

# Membuat dictionary untuk menyimpan data yang akan disimpan dalam JSON
preprocessed_comments = {'data': comments_dict}

# Menyimpan dictionary ke dalam file JSON
with open('dataset.json', 'w') as json_file:
    json.dump(preprocessed_comments, json_file, indent=2)
```

Gambar 5.6 Kode Sumber Menyimpan *Json*

Gambar 5.6 merupakan kode sumber untuk menyimpan hasil praproses ke dalam file berformat *json dictionary*. Data *json* ini nantinya akan digunakan ke tahap selanjutnya, yaitu proses penyaringan data.

```

with open('/content/dataset.json', 'r') as json_file:
    data = json.load(json_file)

# Print the initial length of the data array
print(f"Initial number of objects: {len(data['data'])}")

# Print length of each Preprocess array before filtering
for idx, obj in enumerate(data['data']):
    print(f"Object {idx} - Preprocess length: {len(obj['Preprocess'])}")

# Filter out objects where "Preprocess" array has less than 4 elements
filtered_data = [obj for obj in data['data'] if len(obj['Preprocess']) >= 4]

# Print the length of the data array after filtering
print(f"Number of objects after filtering: {len(filtered_data)}")

# Save the filtered data back to a JSON file
output_path = '/content/final_dataset.json'
with open(output_path, 'w') as file:
    json.dump({"data": filtered_data}, file, indent=2)

print("Filtered data has been saved to", output_path)

```

Gambar 5.7 Kode Sumber Penyaringan Data

Gambar 5.7 merupakan kode sumber untuk proses penyaringan data. Kode ini akan menyaring data yang tidak memiliki indeks *array* sebanyak 4 indeks. Hal ini bertujuan menghapus data yang memiliki *array* kosong dan kalimat yang tidak memiliki makna. Setelah disaring, data akan disimpan dalam format file *json* kembali.

```

data_file = "/content/final_dataset.json"

with open(data_file, 'r') as json_file:
    data = json.load(json_file)

input_texts = [comment['Preprocess'] for comment in data['data']]

# Training tokenizer
max_words = 5000
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(input_texts)

# Save tokenizer
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

```

Gambar 5.8 Kode Sumber Menyimpan Tokenisasi

Gambar 5.8 merupakan kode sumber untuk menyimpan hasil praproses menjadi tokenisasi yang akan digunakan untuk melatih model. Tokenisasi ini disimpan dengan menggunakan *library pickle* yang tersedia pada *python*.

5.1.2 Hasil Pembuatan Model

Pembuatan model dilakukan dengan proses labeling dataset dan transformasi data terlebih dahulu. Berikut merupakan penjelasan langkah-langkah pembuatan model.

1) Pelabelan Dataset

```
# LABELING DATA
comments = [comment['Preprocess'] for comment in data['data']]

mental_health_keywords = [
# ...daftar keyword
]

labels = []
for comment in comments:
    if any(keyword in comment for keyword in mental_health_keywords):
        labels.append(1)
    else:
        labels.append(0)

dataset = pd.DataFrame({'Preprocess': comments, 'Label': labels})
```

Gambar 5.9 Kode Sumber Pelabelan Data

Gambar 5.9 merupakan kode sumber untuk pelabelan data secara otomatis berdasarkan daftar kata kunci yang ada di dalam *array*. Terdapat 111 kata kunci yang digunakan dalam proses pelabelan. Kata kunci dapat dilihat pada tabel berikut ini.

Tabel 5.5 Daftar Kata Kunci

Kata Kunci				
1	anxiety	borderline	abandon	fear
2	disorder	bpd	forsake	mistake
3	personality	substance	reject	nothing
4	deficit	sud	desperation	underappreciate
5	hyperactivity	addiction	anguish	problem
6	adhd	drug	agony	angry

7	posttraumatic	alcoholism	sacrifice	pressure
8	stress	panic	void	overthinking
9	ptsd	gad	lost	doubt
10	depression	dissociative	tired	disappoint
11	bipolar	suicide	anxious	fail
12	obsessive	suicidal	pretending	alone
13	compulsive	threat	stigma	hate
14	ocd	die	loneliness	insult
15	autism	death	pointless	meaningless
16	spectrum	selfharm	worthlessness	suffer
17	schizophrenia	possessions	hallucinations	stupid
18	schizo	kill	confusion	struggle
19	psychotic	crisis	paranoia	abuse
20	psychosis	useless	neglect	unhelpful
21	anorexia nervosa	helplessness	distrustful	emotional
22	anorexia	guilt	unworthy	trauma
23	nervosa	blame	isolation	injury
24	bulimia nervosa	overwhelmed	difficult	inadequacy
25	bulimia	shame	afraid	dead
26	unhealthy	betray	negative	issue
27	tear	intrusive	hurt	uncomfortable
28	cry	terrible	depressive	fear
29	anxiety	borderline	abandon	mistake
30	disorder	bpd	forsake	

Tabel 5.5 merupakan 111 kata kunci yang digunakan untuk proses pelabelan. Data yang memiliki kata kunci ini akan diberikan label 1, sedangkan data yang tidak memiliki kata kunci akan diberikan label 0. Kata kunci yang digunakan dalam penelitian ini sudah divalidasi oleh Bapak Aries Yulianto, S.Psi., M.Si. selaku Kepala Program Studi dan Dosen Psikologi, Fakultas Humaniora dan Bisnis, Universitas Pembangunan Jaya. Menurut Bapak Aries, kata kunci yang kurang

memenuhi sebagai kata kunci negatif adalah *personality* dan *emotional*, hal ini karena kata tersebut memiliki arti yang netral dibandingkan kearah negatif.

```
label_counts = dataset.groupby('Label').size().reset_index(name='Jumlah')
print(label_counts)
```

	Label	Jumlah
0	0	6964
1	1	8514

Gambar 5.10 Hasil Proses Pelabelan Data

Gambar 5.10 merupakan hasil dari proses pelabelan data. Gambar tersebut menampilkan data dengan label 0 memiliki jumlah sebanyak 6964 dan data dengan label 1 memiliki jumlah sebanyak 8514. Data label 1 memiliki jumlah data 22.27% lebih banyak dari label 0. Jumlah data tersebut cukup dan dengan jumlah label 1 lebih banyak akan membuat model lebih memahami data dengan label 1 lebih baik.

2) Transformasi Data

```
# TOKENISASI KOMENTAR DAN PAD SEQUENCE
max_words = 5000
sequences = tokenizer.texts_to_sequences(dataset['Preprocess'])

max_sequence_length = max(len(seq) for seq in sequences)
X = pad_sequences(sequences, maxlen=max_sequence_length)
y = np.array(dataset['Label'])

print(X)
```

[[0	0	0 ...	342	361	722]
[0	0	0 ...	208	34	1003]
[0	0	0 ...	886	487	216]
...						
[0	0	0 ...	0	718	402]
[0	0	0 ...	20	409	140]
[0	0	0 ...	718	202	842]]

Gambar 5.11 Hasil Transformasi Data

Gambar 5.11 merupakan hasil dari transformasi data dari teks menjadi numerik. Data yang sudah menjadi numerik ini akan digunakan untuk latihan sebagai input lapisan *word embedding* pada model. Fungsi *texts_to_sequences()* digunakan untuk melakukan tokenisasi dataset menggunakan tokenisasi yang sudah dibuat dengan *pickle*. Fungsi *pad_sequences()* digunakan untuk mengubah teks

menjadi angka dan menyamaratakan panjang setiap data. Jika ada data yang terlalu pendek, maka fungsi tersebut akan memberikan angka 0 agar panjang *vector* data sama rata.

3) Latihan dan Validasi Model

Proses fase pelatihan model dilakukan berdasarkan data yang sudah disiapkan dari proses sebelumnya. Proses latihan ini akan menggunakan arsitektur LSTM. Proses latihan ini akan menggunakan teknik *k-fold cross-validation*. Proses *cross-validation* ini akan memecah dataset sebanyak nilai k (dalam penelitian ini bernilai k=4) yang nantinya masing-masing fold akan digunakan sebagai latihan dan validasi secara bergantian. Semisal fold 1 digunakan sebagai data validasi, maka fold 2, 3 dan 4 akan digunakan sebagai data latihan. Proses ini akan berulang sebanyak epochs yang sudah ditentukan untuk setiap fold nya. Hal ini bertujuan untuk melakukan validasi yang lebih lanjut dan mengetahui kinerja model dapat bekerja secara konsisten atau tidak ketika data validasi dan latihan yang digunakan berubah.

```
embedding_dim = 100
model = Sequential()
model.add(Embedding(input_dim=max_words, output_dim=embedding_dim))
model.add(LSTM(16, dropout=0.5, recurrent_dropout=0.5))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])
```

Gambar 5.12 Kode Sumber Pembuatan Model

Gambar 5.12 merupakan kode sumber dari pembuatan model. Nilai input dari lapisan *embedding* adalah 5000 sesuai dengan proses transformasi data sebelumnya dan output lapisan *embedding* adalah 100. Setelah memasuki lapisan *embedding* data akan masuk ke lapisan LSTM yang mengambil input dari lapisan *embedding* dan menghasilkan output dengan dimensi yang sama (Nugroho, Akbar, Suksmawati, & Istiada, 2021). Lapisan LSTM memiliki 16 unit input dan output untuk proses data selama fase latihan. Setelah selesai melakukan proses data pada lapisan LSTM, data akan masuk ke lapisan *dropout* dengan nilai 0,5. Lapisan *dropout* akan menghentikan proses latihan sebanyak 50% dari output lapisan sebelumnya secara acak untuk mencegah adanya *overfitting* sebelum data dilempar

ke lapisan selanjutnya. Lapisan terakhir dengan fungsi aktivasi *sigmoid*. Fungsi aktivasi *sigmoid* memiliki output dengan rentang 0 sampai 1 (Firmansyah, Ilyas, & Kasyidi, 2020). Rentang nilai 0 sampai 1 ini membuat aktivasi sigmoid cocok untuk membuat sebuah probabilitas pada suatu data. *Hyperparameter* yang digunakan dalam pembuatan model adalah *epochs* 3 dan *batch size* 16.

5.1.3 Hasil Penerapan Antarmuka

Model untuk identifikasi anomali pada kesehatan mental sudah berhasil dibuat. Langkah selanjutnya adalah untuk membuat tampilan untuk menyajikan data dan pembuatan *form* input yang dapat digunakan oleh pengguna nantinya.



Gambar 5.13 Antarmuka Dataset

Gambar 5.13 merupakan hasil antarmuka untuk menampilkan dataset yang digunakan dalam penelitian. Pengguna dapat melihat seluruh dataset yang ada sekaligus mengunduhnya dengan file berformat *csv*. Fitur unduh file ini merupakan fitur yang disediakan oleh *streamlit*.

DATA TEST MODEL KOMENTAR DARI YOUTUBE

	Komentar
0	these videos always make me cry because i realise how relatable they are...
1	yes
2	your videos got me through tough times
3	if there are communities in poor countries to offer help, giving a hand to depressed
4	i watched this once and now again. but this time i have something to share. i started
5	i m depressed. i hv reasons. i wakeup with the idea i cant change my life and then i s
6	makes me wanna cry when i'm the only one caring about myself searching and learr
7	i feel so deprese and lazy, i dont know how to stop the feeling of lonliness and shan
8	i feel null.. like void. like nothing is left in me. worst part is i cant find any reason why
9	this made me realize i was deprssed i was so lazy and i was scared that i wil miss out

Gambar 5.14 Antarmuka Datatest

Gambar 5.14 merupakan hasil antarmuka untuk menampilkan datatest yang digunakan dalam penelitian sebagai contoh prediksi model. Pengguna dapat melihat seluruh datatest yang ada sekaligus mengunduhnya dengan file berformat csv.

KATA KUNCI YANG PALING SERING MUNCUL DARI DATA TEST

	Kata	Frekuensi
0	depression	4,723
1	nothing	1,003
2	cry	734
3	tired	671
4	lost	644
5	hate	589
6	anxiety	552
7	alone	504
8	die	328
9	problem	288

Gambar 5.15 Antarmuka Kata Kunci

Gambar 5.15 merupakan antarmuka yang digunakan untuk menampilkan data kata kunci yang muncul pada datatest yang digunakan. Tampilan ini dibutuhkan untuk melihat dari kata kunci yang disediakan, kata kunci manakah yang sering muncul dalam datatest. Kata kunci yang tidak memiliki data tidak akan ditampilkan pada tabel tersebut. Pengguna dapat melihat seluruh kata kunci yang muncul pada tabel dan pengguna dapat mengunduh data kata kunci yang muncul dengan format file *csv*.

HASIL IDENTIFIKASI

	Komentar	Label
0	these videos always make me cry because i realise how relatable they are...	Membutuhkan pertolongan
1	yes	Baik-baik saja
2	your videos got me through tough times	Baik-baik saja
3	if there are communities in poor countries to offer help, giving a hand to depressed p	Baik-baik saja
4	i watched this once and now again. but this time i have something to share. i started	Membutuhkan pertolongan
5	i m depressed. i hv reasons. i wakeup with the idea i cant change my life and then i slk	Membutuhkan pertolongan
6	makes me wanna cry when i'm the only one caring about myself searching and learni	Membutuhkan pertolongan
7	i feel so depressive and lazy, i dont know how to stop the feeling of loneliness and sham	Membutuhkan pertolongan
8	i feel null.. like void. like nothing is left in me. worst part is i cant find any reason why	Membutuhkan pertolongan
9	this made me realize i was depressed i was so lazy and i was scared that i wil miss out	Membutuhkan pertolongan

Gambar 5.16 Antarmuka Hasil Identifikasi Datatest Pratinjau

	Komentar	Label	Prediksi	Prediksi (%)
0	these videos always make me cry because i realise how relatable they are...	Membutuhkan pertolongan	1	99.9964
1	yes	Baik-baik saja	0.0046	0.4554
2	your videos got me through tough times	Baik-baik saja	0.0206	2.0613
3	if there are communities in poor countries to offer help, giving a hand to depressed p	Membutuhkan pertolongan	1	99.9969
4	i watched this once and now again. but this time i have something to share. i started	Membutuhkan pertolongan	1	99.9993
5	i m depressed. i hv reasons. i wakeup with the idea i cant change my life and then i slk	Membutuhkan pertolongan	1	99.9966
6	makes me wanna cry when i'm the only one caring about myself searching and learni	Membutuhkan pertolongan	1	99.9964
7	i feel so depressive and lazy, i dont know how to stop the feeling of loneliness and sham	Membutuhkan pertolongan	1	99.9994
8	i feel null.. like void. like nothing is left in me. worst part is i cant find any reason why	Membutuhkan pertolongan	1	99.9967
9	this made me realize i was depressed i was so lazy and i was scared that i wil miss out	Membutuhkan pertolongan	1	99.9966
10	i went through horrible depression lately, like nightmare, crying, weeping, stressing,	Membutuhkan pertolongan	1	99.9971
11	stressed out worrying do beat myself up once in awhile or sometimes alot	Baik-baik saja	0.0955	9.5452
12	procrastinate	Baik-baik saja	0.0072	0.7224
13	procrastination	Baik-baik saja	0.0051	0.5124
14	once in great while dont think that depressed overly stressed	Baik-baik saja	0.012	1.1973

Gambar 5.17 Antarmuka Hasil Identifikasi Lengkap

Gambar 5.16 merupakan antarmuka untuk menampilkan hasil prediksi model. Gambar 5.17 merupakan tabel apabila tampilan diperluas hanya untuk menampilkan tabel. Tabel akan menampilkan data komentar, label, dan hasil prediksi dari komentar. Seperti halnya tabel lain, pengguna dapat mengunduh data yang ada di tabel dengan format *csv*.



Gambar 5.18 Antarmuka Bagan

Gambar 5.18 adalah antarmuka untuk menampilkan bagan data dengan tipe bagan batang. Antarmuka ini juga akan menampilkan ringkasan data dalam bentuk tabel untuk penyampaian informasi yang lebih mudah.

MASUKKAN KOMENTAR UNTUK IDENTIFIKASI (MENGUNAKAN BAHASA INGGRIS)

Masukkan Komentar

I, a smart and responsible final semester student, often feel burdened by the pressure to achieve su

Submit

Komentar: I, a smart and responsible final semester student, often feel burdened by the pressure to achieve success in my career. As the first child in my family, I feel a big responsibility to set an example for my younger siblings. However, my anxiety about the future often keeps me trapped in a cycle of overthinking that haunts my every step. In the midst of final assignments and preparations to enter the world of work, negative thoughts often run rampant, making me doubt my abilities and fear failure. Although I tried hard to downplay my anxiety in front of my friends, inside, I felt alone and afraid of not living up to the expectations of the people I loved.

Prediksi: 100.00%

Label: Membutuhkan pertolongan

Gambar 5.19 Antarmuka *Form*

Gambar 5.19 adalah antarmuka untuk menampilkan *form* input pengguna. Pada antarmuka ini pengguna akan memasukan data melalui *form* yang sudah disediakan. Ketika data input berhasil diproses oleh model, aplikasi akan menampilkan hasil prediksi di bawah *form* input. Data yang ditampilkan setelah proses data adalah komentar yang dimasukan, hasil prediksi komentar, dan hasil label yang diberikan oleh aplikasi.

5.2 Pembahasan

Pembahasan dari penelitian yang telah dilakukan dalam pembuatan aplikasi untuk identifikasi anomali pada kesehatan mental. Bagian ini akan membahas pengujian *black box*, *white box*, dan *confusion matrix*. Penjelasan pengujian *black box* dan *white box* akan dijelaskan secara naratif deskriptif tanpa mengulang kembali tangkapan layer atau kode sumber yang sudah dijelaskan pada sub bab hasil. Sedangkan pengujian *confusion matrix* akan dijelaskan dengan menghitung rumus dan hasil perhitungan.

5.2.1 Pengujian Black Box

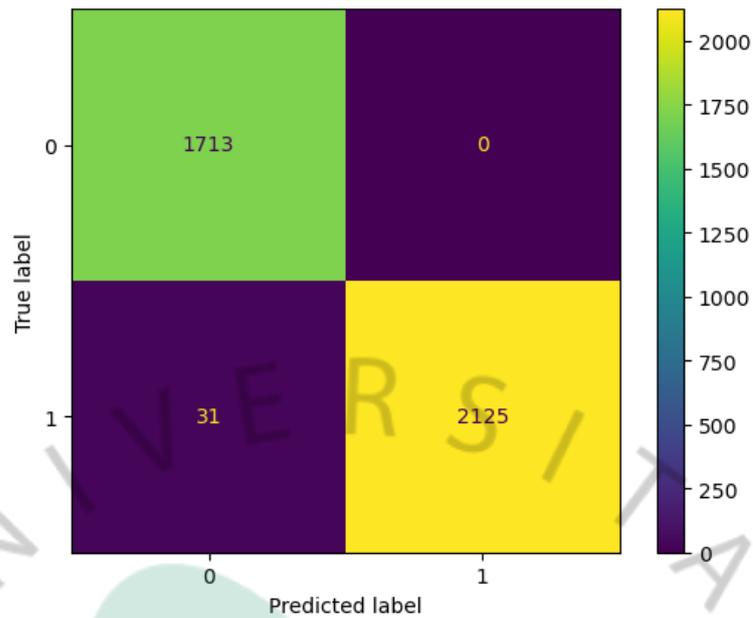
Setelah melakukan pengujian menggunakan metode *black box*, hasilnya menunjukkan bahwa aplikasi berjalan sesuai dengan yang diharapkan. Seluruh

proses pengujian aplikasi yang sudah dirancang pada bab sebelumnya telah berhasil berjalan dimulai dengan menampilkan data yang disediakan ketika pengguna pertama kali mengakses aplikasi, menampilkan data pada saat sedang memasukan data pada *form*, dan ketika pengguna menekan tombol submit, maka aplikasi akan memproses data komentar dengan model yang sudah dibuat dan menampilkan kembali hasilnya kepada pengguna. Semua skenario pengujian *black box* sudah berjalan sesuai yang diharapkan. Aplikasi ini juga sudah diuji oleh Bapak Aries Yulianto, S.Psi., M.Si. selaku Kepala Program Studi dan Dosen Psikologi, Fakultas Humaniora dan Bisnis, Universitas Pembangunan Jaya. Menurut Bapak Aries, hasil dari sistem sudah cukup baik dalam penentuan kondisi namun masih ada beberapa hasil identifikasi yang dirasa kurang sesuai menurutnya. Selain itu beliau juga mengatakan bahwa aplikasi ini bisa digunakan sebagai deteksi indikasi kondisi seseorang berdasarkan apa yang orang lain sampaikan melalui komentar. Secara psikologi, apa yang diungkapkan oleh seseorang itu termasuk memberikan indikasi terhadap situasi yang dihadapinya, namun untuk penilaian tidak boleh langsung dihukumi atau di-*judge*, tetap membutuhkan penilaian lebih lanjut oleh professional.

5.2.2 Pengujian White Box

Dalam melakukan pengujian *white box*, Peneliti telah membuat rancangan pengujian *white box* pada bab sebelumnya. Rancangan ini dimulai dari pengujian fungsi praproses, penyimpanan token, pemuatan model dan tokenisasi, dan pemrosesan input yang dilakukan pengguna. Seluruh rancangan pengujian *white box* telah berjalan sesuai dengan yang telah dirancang pada bab sebelumnya tanpa adanya kendala.

5.2.3 Pengujian Confusion Matrix



Gambar 5.20 Hasil *Confusion Matrix Library Sklearn*

Gambar 5.20 merupakan hasil *confusion matrix* yang dilakukan dengan *library sklearn*. Berdasarkan data tersebut, Peneliti dapat melakukan perhitungan rumus *confusion matrix* untuk *accuracy*, *precision*, *recall*, dan *f1-score*. Total data uji yang digunakan sebanyak 3869. Penguraian nilai dan perhitungan *confusion matrix* ditampilkan sebagai berikut.

- i. True Positives (TP): 2125 (kelas 1 yang diprediksi benar)
- ii. True Negatives (TN): 1713 (kelas 0 yang diprediksi benar)
- iii. False Positives (FP): 0 (kelas 0 yang diprediksi salah)
- iv. False Negatives (FN): 31 (kelas 1 yang diprediksi salah)

1) *Accuracy*

Tabel 5.6 Perhitungan *Accuracy*

Perhitungan	Hasil (%)
$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	99%
$accuracy = \frac{1713 + 2125}{1713 + 2125 + 0 + 31}$	
$accuracy = \frac{3838}{3869}$	
$accuracy = 0.99$	

2) *Precision*

Tabel 5.7 Perhitungan *Precision*

Perhitungan	Hasil (%)
$precision = \frac{TP}{TP + FP}$	Kelas 0 = 98% Kelas 1 = 100%
$precision_0 = \frac{1713}{1713 + 31}$	
$precision_0 = \frac{1713}{1744}$	
$precision_0 = 0.98$	
$precision_1 = \frac{2125}{2125 + 0}$	
$precision_1 = \frac{2125}{2125}$	
$precision_1 = 1$	

3) *Recall*

Tabel 5.8 Perhitungan *Recall*

Perhitungan	Hasil (%)
$recall = \frac{TP}{TP + FN}$	Kelas 0 = 100% Kelas 1 = 98%
$recall_0 = \frac{1713}{1713 + 0}$	
$recall_0 = \frac{1713}{1713}$	
$recall_0 = 1$	
$recall_1 = \frac{2125}{2125 + 31}$	
$recall_1 = \frac{2125}{2156}$	
$recall_1 = 0.98$	

4) *F1-score*

Tabel 5.9 Perhitungan *F1-score*

Perhitungan	Hasil (%)
$F1\ score = \frac{2 \times precision \times recall}{precision + recall}$	98%

$F1\ score = \frac{2 \times 1 \times 0.98}{1 + 0.98}$ $F1\ score = \frac{1.96}{1.98}$ $F1\ score = 0.98$	
--	--

Hasil perhitungan *classification report* berdasarkan *library sklearn* dapat dilihat pada Gambar 5.21

Confusion Matrix:

```
[[1713  0]
 [  31 2125]]
```

-----Pembatas-----

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	1713
1	1.00	0.99	0.99	2156
accuracy			0.99	3869
macro avg	0.99	0.99	0.99	3869
weighted avg	0.99	0.99	0.99	3869

Gambar 5.21 *Classification Report Library Sklearn*