BAB V

HASIL DAN PEMBAHASAN

Bab ini terdiri dari dua subbab, yaitu hasil dan pembahasan. Subbab hasil menjelaskan hasil pengembangan sistem yang dilakukan dalam penelitian ini. Subbab pembahasan memberikan penjelasan dan analisis lebih mendalam tentang model NLP serta pengujian sistem.

5.1 Hasil

Bagian ini membahas hasil penelitian yang telah dilakukan yaitu Pra-proses dan Hasil pemodelan NLP.

5.1.1 Pra-proses data dan pembuatan model

- *Case Folding*

```
def preprocess_text(text):
    # Memeriksa jika text adalah string
    if isinstance(text, str):
        # Mengubah text menjadi huruf kecil
        lower_case = text.lower()
```

Gambar 5. 1 Kode Penggalan Case Folding

Gambar 5.1 merupakan penggalan kode untuk melakukan case folding. Kode mendefinisikan sebuah fungsi bernama preprocess_text yang menerima satu parameter text. Baris *if isinstance(text, str)* memeriksa apakah input text adalah sebuah string. Fungsi isinstance digunakan untuk memverifikasi tipe data dari variabel text. Jika text adalah string, maka kondisi ini akan benar. Jika text adalah string, maka baris *lower_case = text.lower()* akan mengubah seluruh teks menjadi huruf kecil. Metode *.lower()* adalah metode string bawaan di Python yang mengubah semua karakter huruf besar dalam string menjadi huruf kecil.

Tabel 5. 1 Hasil case Folding

Sebelum	Sesudah
selain pada kasus IKN political will yang cukup tinggi juga terjadi pada sektor transisi energi. Jokowi dengan segala cara membujuk masyarakat untuk beralih ke motor dan mobil listrik salah satunya dengan mengeluarkan kebijakan subsidi kendaraan listrik.	selain pada kasus ikn political will yang cukup tinggi juga terjadi pada sektor transisi energi jokowi dengan segala cara membujuk masyarakat untuk beralih ke motor dan mobil listrik salah satunya dengan mengeluarkan kebijakan subsidi kendaraan listrik

- *Cleaning*

```
def preprocess_text(text):
    # Memeriksa jika text adalah string
    if isinstance(text, str):
        # Mengubah text menjadi huruf kecil
        lower_case = text.lower()

        # Menghapus angka dari kalimat
        text = re.sub(r"\d+", "", lower_case)

        # Menghapus spasi pada awal dan akhir kalimat
        text = text.strip()

        # Menghapus Username Twitter
        text = re.sub(r'@[A-Za-z0-9_]+', ' ', text)

        # Menghapus https dan http
        text = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", text)

        # Menghilangkan Tanda Baca
        text = text.translate(str.maketrans('', '', string.punctuation))

        # Mengganti karakter HTML dengan tanda petik
        text = re.sub('<.*?>', ' ', text)

        # Mempertimbangkan huruf dan angka
        text = re.sub('[^a-zA-Z0-9]', ' ', text)

        # Mengganti line baru dengan spasi
        text = re.sub("\n", " ", text)
```

```
    # Menghilangkan emoji
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emotikon wajah
                               u"\U0001F300-\U0001F5FF"  # simbol & benda
                               u"\U0001F680-\U0001F6FF"  # transportasi & simbol peralatan
                               u"\U0001F1E0-\U0001F1FF"  # bendera negara
                               u"\U00002500-\U00002BEF"  # karakter CJK (Chinese, Japanese, Korean)
                               u"\U00002702-\U000027B0"  # simbol & tanda
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               u"\U0001f926-\U0001f937"
                               u"\U00010000-\U0010ffff"
                               u"\u200d"
                               u"\u2640-\u2642"
                               u"\u2600-\u2B55"
                               u"\u23cf"
                               u"\u23e9"
                               u"\u231a"
                               u"\u3030"
                               u"\ufe0f"
                               "]+", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text)
```

Gambar 5. 2 Kode Penggalan Cleaning

Gambar 5.2 menampilkan potongan kode yang digunakan untuk proses pembersihan data. Proses ini melibatkan beberapa langkah yang

memanfaatkan regex untuk menemukan pola tertentu dan menggantinya dengan string yang diinginkan. Hasil dari proses pembersihan data dapat dilihat pada Tabel 5.2.

Tabel 5. 2 Hasil Cleaning

Sebelum	Sesudah
@jokowi Kelilit utang listrik listrik masuk PLN tak bisa dijual ... kebijakan kompor listrik...kendaraan listrik...mungkin jika 3 periode subsidi listrik juga dicabut...keserakahan itu membawa kesusahan !	kelilit utang listrik listrik masuk pln tak bisa dijual kebijakan kompor listrikkendaraan listrikmungkin jika periode subsidi listrik juga dicabutkeserakahan itu membawa kesusahan

Tabel 5.2 merupakan perbandingan sebelum dan sesudah cleaning data. Adapun perbedaan yang terlihat yaitu hilangnya mentions, tanda baca, dan angka.

- Normalisasi Kata

```
kamus = pd.read_csv('kamusalay.csv', header=None)
kamus.columns = ['kata_asli', 'kata_normal']
```

```
def normalisasi_teks(teks, kamus):
    kata_asli = teks.split()
    kata_normal = []

    for kata in kata_asli:
        normal = kamus[kamus['kata_asli'] == kata]['kata_normal'].values
        if len(normal) > 0:
            kata_normal.append(normal[0])
        else:
            kata_normal.append(kata)

    return ' '.join(kata_normal)
```

Gambar 5. 3 Kode Penggalan Normalisasi

Gambar 5.3 menampilkan potongan kode yang digunakan untuk melakukan normalisasi kata. Variabel kamus diisi file CSV yang berisi kamus kata asli dan kata normalisasi input berupa kamus bahasa gaul dan substitusi kata baku dalam bentuk file csv. Lalu teks dipecah menjadi kata-kata individual yang nantinya akan dilakukan iterasi melalui setiap kata dalam teks lalu mencari kata yang sesuai dalam kamus hingga mendapatkan nilai normalisasinya. Hasil normalisasi dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Hasil Normalisasi

Sebelum	Sesudah
lucu aq gk prnah pke gas meter lampu pake token non subsidi bbm aq gk prnah pake krena gk suka pake kendaraan mau kluar pesan ajh memasak pun ku pake panci listri dan kompor listrik loe ajah yg muka ngemis bansos loll jgn lupa bansos yg loe makan jg pajak dri gw	lucu saya tidak pernah pakai gas meter lampu pake token non subsidi bbm saya tidak pernah pake krena tidak suka pake kendaraan mau kluar pesan saja memasak pun ku pake panci listri dan kompor listrik kamu ajah yang muka ngemis bansos loll jangan lupa bansos yang kamu makan juga pajak dri saya

- Tokenisasi

```
def tokenize_text(kalimat):
  tokens = nltk.tokenize.word_tokenize(kalimat)
  return tokens
```

Gambar 5. 4 Kode Penggalan Tokenisasi

Menggunakan metode word_tokenize dari modul nltk.tokenize untuk memecah kalimat menjadi daftar token (kata-kata). Metode ini memisahkan kata-kata dalam kalimat berdasarkan spasi dan tanda baca, mengubahnya menjadi elemen-elemen terpisah dalam daftar.

Tabel 5. 4 Hasil Tokenisasi

Sebelum	Sesudah
kelilit utang listrik listrik masuk pln tak bisa dijual kebijakan kompor listrikkendaraan listrikmungkin jika periode subsidi listrik juga dicabutkeserakahan itu membawa kesusahan	['kelilit', 'utang', 'listrik', 'listrik', 'masuk', 'pln', 'tak', 'bisa', 'dijual', 'kebijakan', 'kompor', 'listrikkendaraan', 'listrikmungkin', 'jika', 'periode', 'subsidi', 'listrik', 'juga', 'dicabutkeserakahan', 'itu', 'membawa', 'kesusahan']

- *Stemming*

```
def stemming_text(tokens):
    hasil = []
    for token in tokens:
        # Memeriksa jika token adalah 'kendaraan', maka tidak dilakukan stemming
        if token.lower() == 'kendaraan':
            hasil.append(token)
        else:
            hasil.append(stemmer.stem(token))
    return hasil
```

Gambar 5. 5 Kode Penggalan Stemming

Fungsi stemming_text bertujuan untuk melakukan stemming pada daftar token yang diberikan, dengan pengecualian untuk token 'kendaraan' yang tidak di *stemming*. Stemming adalah proses mengubah kata-kata ke bentuk dasarnya (stem) untuk mengurangi variasi kata-kata yang serupa.

Tabel 5. 5 Hasil Stemming

Sebelum	Sesudah
['ditambah', 'subsidi', 'kendaraan', 'listrik', 'pak']	['tambah', 'subsidi', 'kendaraan', 'listrik', 'pak']

- Pelabelan Data

```
# Pembuatan Kamus Lexicon
lexicon_negative = {}
with open('negative.tsv', 'r') as tsvfile:
    reader = csv.reader(tsvfile, delimiter='\t')
    for row in reader:
        if len(row) > 1 and row[1].lstrip('-').isdigit():
            lexicon_negative[row[0]] = int(row[1])


lexicon_positive = {}
with open('positive.tsv', 'r') as tsvfile:
    reader = csv.reader(tsvfile, delimiter='\t')
    for row in reader:
        if len(row) > 1 and row[1].lstrip('-').isdigit():
            lexicon_positive[row[0]] = int(row[1])
```

```
def sentiment_analysis_lexicon_indonesia(text):
    score = 0
    words = text.split()
    for word in words:
        if word in lexicon_positive:
            score += lexicon_positive[word]
        if word in lexicon_negative:
            score -= abs(lexicon_negative[word])
    return score
```

Gambar 5. 6 Kode Penggalan Pelabelan data

Fungsi sentiment_analysis_lexicon_indonesia bertujuan untuk menghitung skor sentimen dari sebuah teks berdasarkan kamus sentimen (lexicon) bahasa Indonesia. Fungsi ini menambahkan nilai positif untuk kata-kata yang terdapat dalam kamus positif dan mengurangi nilai negatif untuk kata-kata yang terdapat dalam kamus negatif. Hasil akhirnya adalah skor sentimen yang menunjukkan seberapa positif atau negatif teks tersebut.

5.1.2 Pra-proses data dan pembuatan model

Tabel 5. 6 Hasil Pemodelan

Batch Size	Epoch	Rasio Data	Akurasi
32	5	80:20	77,7%
64	5	80:20	70,5%

128	5	80:20	71,3%

Tabel 5.6 menggambarkan hasil uji pengaruh hyperparameter terhadap akurasi yang diperoleh. Hyperparameter yang diuji yaitu batch size dan rasio pembagian data. Epoch yang digunakan pada pengujian adalah 5. Epoch menggambarkan setiap sampel dalam dataset pelatihan yang memiliki kesempatan untuk memperbarui parameter model internal satu kali selama satu epoch. Seluruh batch data yang telah melewati model dihitung sebagai satu epoch. Jumlah epoch dapat berupa satu atau lebih. Batch size menggambarkan data pelatihan yang dipotong menjadi bagian-bagian kecil. Batch yang lebih kecil ini dapat dengan mudah dimasukkan ke dalam model pembelajaran mesin untuk melatihnya. Ukuran batch size memiliki batas maksimal sesuai jumlah data pelatihan dan minimal satu.

Pada pengujian hyperparameter batch size dengan ukuran 32 memiliki akurasi rata-rata 77%. Batch size dengan ukuran 64 memiliki akurasi rata-rata 70,5%. Batch size dengan ukuran 128 memiliki akurasi rata-rata sebesar 71,3%.

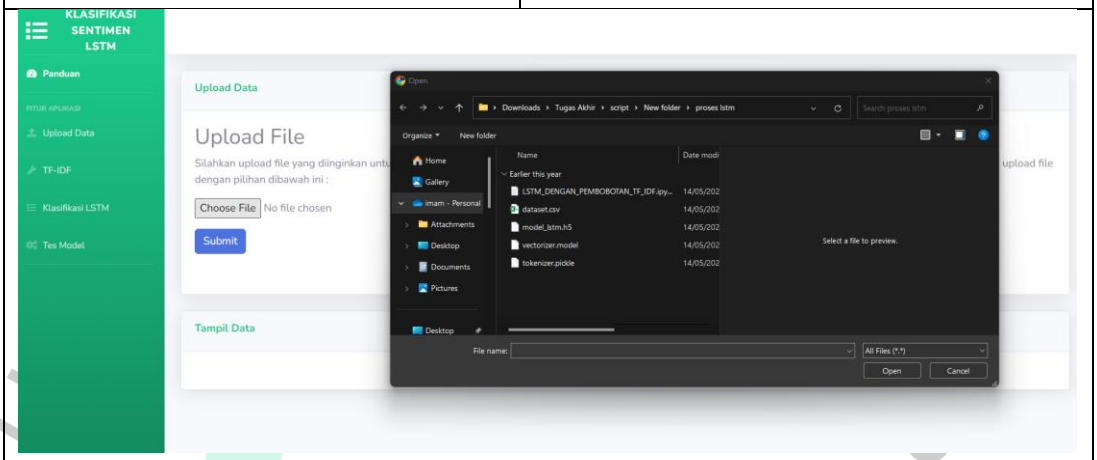
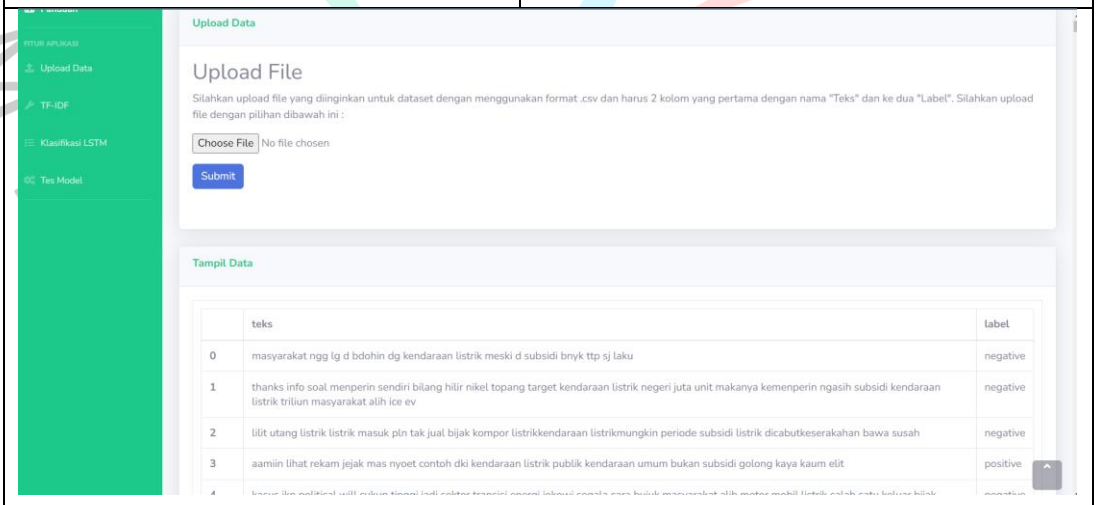
5.2 Pembahasan

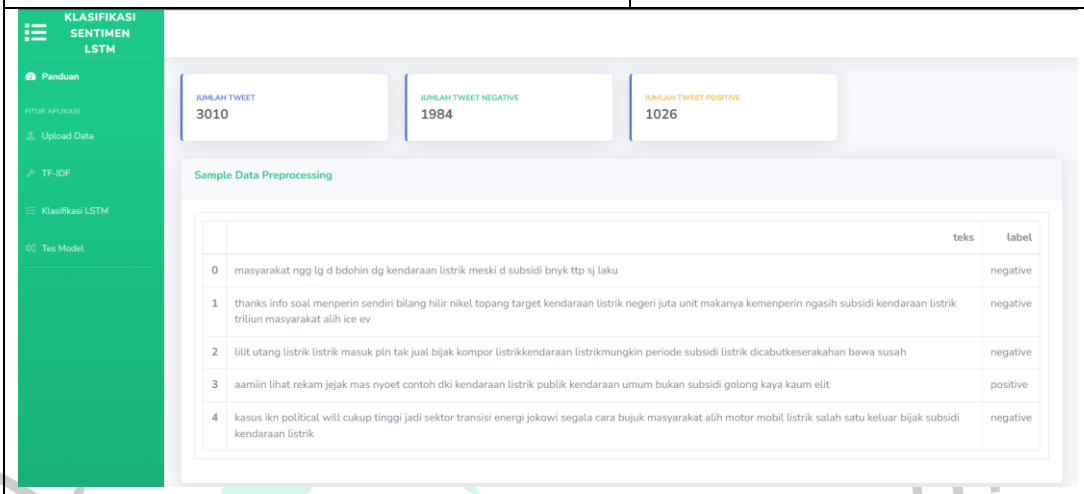
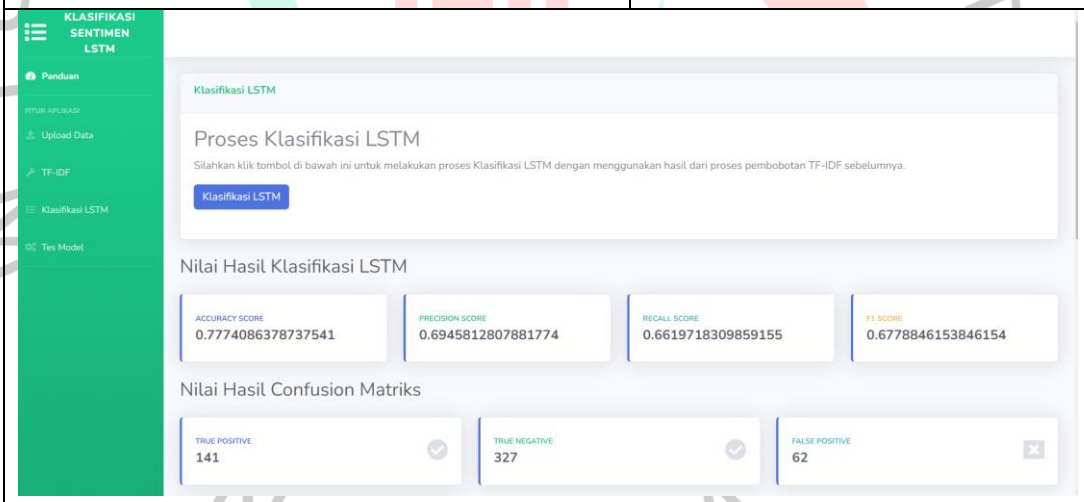
Dalam penelitian ini, dua metode pengujian yang digunakan adalah metode Black Box dan White Box. Berikut ini adalah hasil pengujian dan pembahasan perangkat lunak yang telah dibuat.

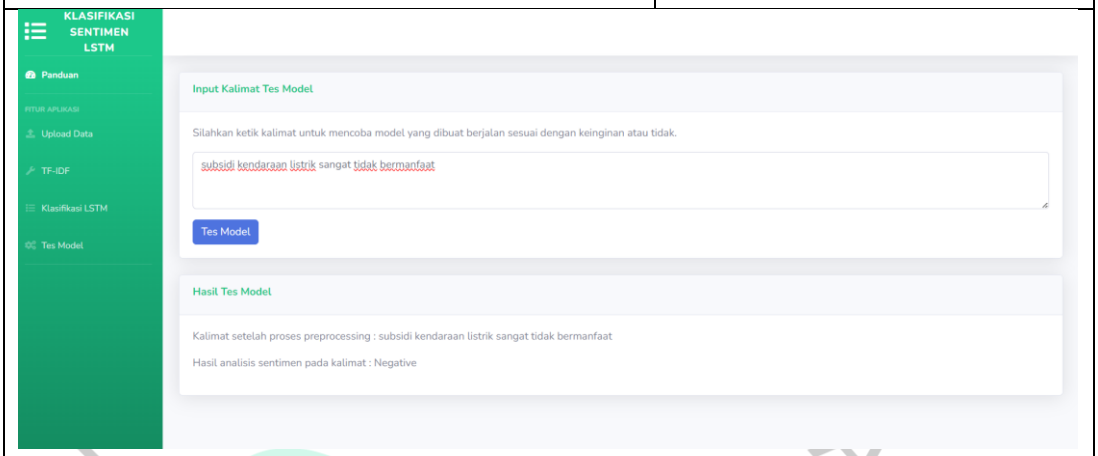
5.2.1 Hasil Pengujian *Black Box*

Black box testing adalah metode pengujian eksternal yang bertujuan memastikan bahwa suatu sistem atau aplikasi dapat memenuhi kebutuhan dan harapan pengguna. Pengujian ini berfokus pada penilaian kualitas semua fitur aplikasi, sehingga penguji dapat memastikan bahwa sistem atau aplikasi berfungsi dengan benar.

Tabel 5. 7 Hasil Pengujian Black Box

<table>
<tr><td>No</td><td>Skenario pengujian</td><td>Hasil yang diharapkan</td></tr>
<tr><td rowspan="3">1</td><td>Pengguna klik Choose file</td><td>Muncul jendela explorer untuk unggah data</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Kesimpulan : berhasil memunculkan jendela explorer</td></tr>
<tr><td rowspan="3">2</td><td>Pengguna klik tombol Submit</td><td>Menampilkan data yang sudah diberikan label positif dan negatif</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Kesimpulan : berhasil menampilkan data</td></tr>
</table>

<table>
<tr><td rowspan="3">3</td><td>Pengguna pilih menu TF IDF dan klik tombol Proses TF IDF</td><td>Menampilkan jumlah data tweet serta jumlah tweet positif dan negatif</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Kesimpulan : Berhasil menampilkan data tweet</td></tr>
<tr><td rowspan="3">4</td><td>Pengguna pilih menu klasifikasi LSTM dan klik tombol Klasifikasi LSTM</td><td>Menampilkan hasil klasifikasi</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Kesimpulan : berhasil menampilkan nilai hasil klasifikasi</td></tr>
</table>

5	Pengguna pilih menu Tes Model, menginput teks lalu klik tombol Tes Model	Menampilkan hasil prediksi kata
		
	Kesimpulan : Berhasil menampilkan prediksi kalimat	

5.2 Hasil Pengujian *White Box*

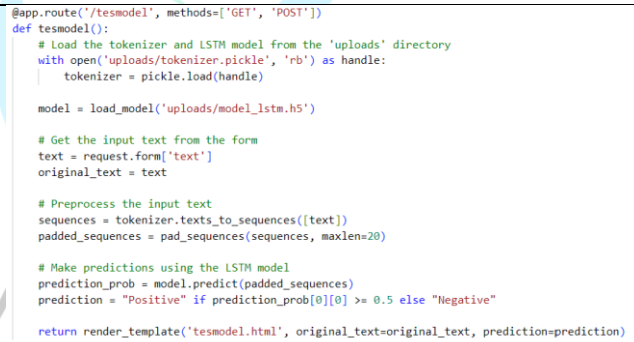
White box testing adalah metode pengujian internal yang memastikan bahwa kode sistem atau aplikasi bebas dari bug dan memenuhi standar kualitas yang ditetapkan. Selain itu, pengujian ini bertujuan untuk memeriksa bagaimana setiap bagian dari sistem atau aplikasi bekerja secara internal menunjukkan beberapa skenario pengujian dengan berisi hasil yang dirapkan, kode program dan hasil pengujian.

Tabel 5. 8 Hasil Pengujian White Box

No	Hasil yang diharapkan	Kode program	Hasil Pengujian

1	Aplikasi memuat model dan tokenizer	`@app.route('/klasifikasilstm', methods=['GET', 'POST'])`
		`def klasifikasilstm():`
		`  if request.method == 'GET':`
		`  return render_template('klasifikasilstm.html')`
		
		`  elif request.method == 'POST':`
		`  # Load the tokenizer and LSTM model`
		`  with open('uploads/tokenizer.pickle', 'rb') as handle:`
		`  tokenizer = pickle.load(handle)`
		
		`  model = load_model('uploads/model_lstm.h5')`
			Berhasil memuat model dan tokenizer

2	Menyamakan tipe huruf menjadi huruf kecil, menghapus username twitter dan hyperlinks	`def preprocess_text(text):`
		`  # Memeriksa jika text adalah string`
		`  if isinstance(text, str):`
		`  # Mengubah text menjadi huruf kecil`
		`  lower_case = text.lower()`
		
		`  # Menghapus angka dari kalimat`
		`  text = re.sub(r"\d+", "", lower_case)`
		
		`  # Menghapus spasi pada awal dan akhir kalimat`
		`  text = text.strip()`
		
		`  # Menghapus Username Twitter`
		`  text = re.sub(r'@[A-Za-z0-9_]+', ' ', text)`
		
		`  # Menghapus https dan http`
		`  text = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", text)`
			Berhasil Menyamakan tipe huruf menjadi huruf kecil, menghapus username twitter dan hyperlinks

3	Menghilangkan emoji pada kalimat		Berhasil Menghilangkan emoji pada kalimat
4	Menghilangkan stopword pada kalimat		Berhasil Menghilangkan stopword pada kalimat
5	Melakukan prediksi yang kemudian dikonversi menjadi label "Positive" atau "Negative" berdasarkan nilai probabilitas.		Berhasil Melakukan prediksi yang kemudian dikonversi menjadi label "Positive" atau "Negative" berdasarkan nilai probabilitas.

Tabel 5.8 mencakup skenario pengujian white box. Pengujian ini dimulai dengan proses aplikasi memuat model dan tokenizer ke sesi web browser. Hasilnya, aplikasi berhasil memuat model dan tokenizer dengan sukses. Selanjutnya, dilakukan pengujian kode program untuk menghilangkan emoji, menghapus stopword, dan membersihkan teks. Kode berjalan lancar dan menghasilkan output sesuai yang diharapkan. Pengujian berikutnya mencakup kode untuk melakukan word embedding, padding, prediksi, dan mengkodekan output dari bentuk numerik menjadi kata.