Progresif: Jurnal Ilmiah Komputer
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877
p-ISSN: 0216-3284

Analisis Sentimen Pada Kasus Positif *Covid-19* Berdasarkan Pemberitaan Media Di Indonesia Menggunakan *Indobert*

Widiya Nurfitri[1*], Andry Chowanda[2]
Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia
*e-mail *Corresponding Author*: widiya.fitri@binus.ac.id

Abstract

The application of sentiment analysis to news about the increase in the spread of the positive rate of Covid-19 in Indonesia using the IndoBERT model aims to find out how much influence the news of the increase in Covid-19 cases in Indonesia has on the opinion of sentiment analysis on the opinions of the Indonesian public. First, implementing Web Scraping, Labeling and Text Pre-processing techniques to collect data about the increase in Covid-19 cases in Indonesia. Second, apply the IndoBERT algorithm in sentiment analysis regarding news about the increase in positive Covid-19 cases in Indonesia. Next, evaluate the performance of the sentiment analysis model with varying batch sizes. In batch size 16, the model tends to show consistent performance with f1 scores ranging from 80.10% to 80.53%, while in batch size 32 there are variations. An increase in epochs does not necessarily mean a significant increase in performance. Although in some cases there was an increase, there was also a decrease in some cases. Overall, the model shows good performance with f1 score and accuracy above 0.80 and 0.81, while loss tends to increase with epoch. Further exploration is needed to understand the factors influencing model performance in depth.

Keywords: *Web Scraping; Labeling; Text Pre-processing; Sentiment Analysis; Model Performance*

Abstrak

Penerapan sentimen analisis pada berita kenaikan penyebaran tingkat positif *Covid-19* di Indonesia dengan menggunakan model IndoBERT bertujuan untuk mengetahui seberapa berpenagruhkan berita kenaikan kasus *Covid-19* di Indonesia terhadap opini sentimen analisis terhadap opini masyarakat Indonesia. Pertama, mengimplementasikan *Teknik Web Scraping, Labeling* dan *Pre-processing Text* untuk mengumpulkan data tentang peningkatan kasus *Covid-19* di Indonesia. Kedua, menerapkan algoritma IndoBERT dalam analisis sentimen terhadap pemberitaan peningkatan kasus positif *Covid-19* di Indonesia. Selanjutnya, mengevaluasi kinerja model sentimen analisis dengan variasi *batch size*. Pada *batch size* 16, model cenderung menunjukkan konsistensi kinerja dengan *f1 score* berkisar antara 80.10% hingga 80.53%, sedangkan pada *batch size* 32 terdapat variasi. Peningkatan *epoch* tidak selalu berarti peningkatan kinerja yang signifikan. Meskipun pada beberapa kasus terjadi peningkatan, ada juga penurunan pada beberapa kasus. Secara keseluruhan, model menunjukkan kinerja baik dengan *f1 score* dan *accuracy* di atas 0.80 dan 0.81, sementara *loss* cenderung meningkat seiring dengan *epoch*. Diperlukan eksplorasi lebih lanjut untuk memahami faktor-faktor yang mempengaruhi kinerja model secara mendalam.

Keyword: *Web Scraping; Labeling; Pre-processing Text; Sentimen Analisis; Kinerja Model*

1. Pendahuluan

Bermula dari tahun 2019 di Wuhan, China, sebuah virus tak kasat mata yang menyerang sistem pernapasan manusia yang kemudian disebut *Coronavirus disease* 2019 (*Covid-19*) menyebabkan perubahan yang sangat signifikan pada dunia. Berdasarkan data WHO pada 10 November 2020 dikonfirmasi jumlah penderita *Covid-19* di 219 negara sudah ada 50.459.886 yang terkonfirmasi dan 1.257.523 orang yang meninggal dunia. Di Indonesia terkonfirmasi Positif

444.348 orang, yang telah dinyatakan sembuh 375.741 orang dan yang meninggal 14.761 orang [1].

Pemberitaan tentang Virus *Corona* pertama kali diberitakan oleh media China, *The Lancet*; Jurnal medis yang ditulis oleh dokter China dari rumah sakit Jinyinhan dari Wuhan, yang merawat beberapa pasien paling awal. Pemberitaan meluas oleh Majalah People, kemudian secara massif dan intensif oleh berbagai platform media seiring dengan banyaknya korban dan efek Virus *Corona*. Dalam 1 hingga 2 bulan penyebarannya, berita tentang Virus *Corona* masih mendominasi [2]. Berita yang keluar masuk melalui laman media sosial bisa bercampur antara berita *valid* dan berita *hoax*, sehingga berita mengenai Virus *Corona* semakin mendominasi bahkan bisa dikatakan berita mengenai Virus *Corona* tersebut memonopoli pemberitaan di Indonesia pada kurun waktu sepanjang akhir Februari hingga Maret 2020. Berita-berita yang lain seolah tenggelam. Disinilah tampak kuatnya pengaruh media. Media menjadi kekuatan yang mampu memberi dorongan untuk melakukan sesuatu. Dampak konsumsi media menjadikan konsumen media mengkonstruksi realitas sesuai dengan konstruksi media [2].

Analisis sentimen adalah proses memahami, mengekstrak, dan memproses data tekstual secara otomatis untuk memperoleh informasi tentang sentimen dalam kalimat opini [3][4]. Analisis sentimen dilakukan untuk mengidentifikasi opini atau tren opini tentang masalah atau topik seseorang, apakah mereka memiliki opini atau opini negatif atau positif. Seperti yang telah dilakukan banyak peneliti di berbagai penjuru dunia [3], analisis sentimen di media sosial merupakah salah satu cara jitu untuk melihat reaksi publik akan suatu isu, termasuk untuk melacak dan memperkirakan kekhawatiran publik tentang pandemi.

Dengan berkembangnya teknologi di bidang NLP, proses Analisis sentimen saat ini banyak dikembangkan dengan memanfaatkan teknologi *Neural Network* [5]. IndoNLU adalah sebuah koleksi sumber untuk riset dalam topik *Natural Language Understanding* (NLU) untuk Bahasa Indonesia dengan 12 aplikasi. Kami menyediakan kode untuk mereproduksi hasil dan model besar yang sudah dilatih sebelumnya (IndoBERT and IndoBERT-lite) yang dilatih dengan kumpulan tulisan berisi sekitar 4 miliar kata (Indo4B) dan lebih dari 20 GB dalam ukuran data teks. Proyek ini awalnya dimulai dari kerjasama antara universitas dan industri, seperti Institut Teknologi Bandung, Universitas Multimedia Nusantara, *The Hong Kong University of Science and Technology*, Universitas Indonesia, Gojek, dan Prosa.AI [6].

Pengumpulan berita di dalam situs berita pada beberapa penelitian menggunakan teknik *web scraping*. *Web scraping* adalah teknik untuk mendapatkan informasi dari situs web secara otomatis tanpa harus menyalinnya secara manual. Tujuan dari *web scraping* adalah untuk mencari informasi pada bagian tertentu. Tidak seperti kegiatan *web crawling* yang mengunjungi seluruh situs yang berhubungan dengan situs utamanya, kegiatan *web scraping* hanya melakukan ekstraksi data tertentu saja dari situs yang dituju sesuai dengan kebutuhan. Hasil dari web scraping sendiri dapat dimanfaatkan kembali oleh sistem lain maupun dianalisis lebih lanjut [7].

Pada beberapa penelitian sebelumnya telah dilakukan pengambilan data sampe dengan menggunakan Teknik *web scraping* untuk menganalisis beberapa data mengenai berita seputar peningkatan kasus *Covid-19* di Indonesia dari beberapa sumber berita resmi seperti detik.com, liputan6.com dan kompas.com [7]. Hal tersebut yang melatar belakangi penelitian penerapan sentimen analysis pada berita kenaikan penyebaran tingkat positif *Covid-19* di Indonesia dengan menggunakan model IndoBERT, tujuannya untuk mengetahui seberapa berpenagruhkan berita kenaikan kasus *Covid-19* di Indonesia terhadap opini sentimen analisis terhadap opini masyarakat Indonesia.

2. Tinjauan Pustaka

Beberapa penelitian relevan disajikan berikut:

Penelitian [8] dengan judul *BERT Fine-Tuning for* Sentimen *Analysis on Indonesia Mobile Apps Reviews.* Permasalahan pada penelitian tersebut adalah analisa data review user dari *Google Play Store* terhadap aplikasi mobile android menggunakan model BERT untuk sentimen analysis dengan menggunakan dua model *pre-trained* yang berbeda. Penelitian menggunakan metode BERT *Fine-tuning, IndoBERT, dan Multilingual pre-trained*. Hasil *pre-trained* menggunakan model berbahasa Indonesia memiliki tingkat akurasi yang tinggi, yaitu 84% dengan 24 *epoch* dan di *training* dalam waktu 24 menit hasil ini lebih baik jika dibandingkan dengan multilingual *pre-trained* model.

Penelitian [9] dengan judul *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Permasalahan pada penelitian tersebut adalah Penerapan model pre-trained BERT untuk question answering. Penelitian menggunakan metode *BERT, GLUE, MultiNLU*, dan *SQuAD*. *Novelty* dari penelitian diharapkan untuk pengembangan selanjutnya dapat menggali lebih dalam lagi konsep bidirectional architectures.

Penelitian [10] dengan judul Sentimen *Analysis on the impact of Corona Virus in Social Life using the BERT model.* Permasalahan pada penelitian tersebut adalah Melakukan sentimen analisis dari opini yang dituangkan dalam media sosial *Twitter* dengan menggunakan dua dataset yaitu dataset yang dikumpulkan dari twitter yang diposting oleh pengguna secara global dan pengguna *Twitter* hanya dari India saja. Penelitian menggunakan *BERT* model, Algoritma: Sentimen Analysis, *Classification*. Dari penelitian tersebut didapatkan akurasi validasi untuk klasifikasi emosi dari *repository GitHub* dan mendapatkan hasil 94%.

Penelitian [11] dengan judul *Algoritma Multinominal Naïve Bayes untuk Klasifikasi Sentimen Pemerintah terhadap Penanganan Covid-19 menggunakan Data Twitter.* Permasalahan pada penelitian tersebut adalah Mengklasifikasi sentimen masyarakat terhadap penangan *Covid-19* dengan penggunaan 2000 dataset yang bersumber dari *twitter*. Penelitian menggunakan Algoritma: *Naïve Bayes, Text Preprocessing, Precision dan Recall*. Hasil pengujian diperoleh *weighted average* untuk *precision*, *recall* dan akurasi sebesar 74% dan akurasi metode yang diusulkan memiliki tingkatan cukup baik.

Penelitian [12] dengan judul *COVID-19 Sensing: Negative* Sentimen *Analysis on SocialMedia in China via BERT Model.* Permasalahan pada penelitian tersebut adalah Menganalisa negative public sentimen analisis dari social media selama periode *pandemic Covid 19* pada bulan 1 Januari 2020 hingga 18 Februari 2020 dengan mengklasifikasikannya menjadi Positif, Netral dan Negatif dengan meninjau empat aspek opini public. Penelitian menggunakan metode BERT dan TF-IDF. Pembaruan dari penelitian ini dengan meninjau empat aspek, diperoleh hasil: *the virus Origin* (*Gamey Food,* 3.08%; *Bat*, 2.70%; *Conspiracy Theory*, 1.43%); *Symptom* (*Fever*, 2.13%; *Cough*, 1.19%); *Production Activity* (*Go to Work*, 1.94%; *Resume Work*, 1.12%; *School New Semester Beginning*, 1.06%)

Penelitian [13] dengan judul *IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesia NLP.* Permasalahan pada penelitian tersebut adalah Kurangnya dataset dalam Bahasa indonesia dalam bidang penelitian NLP. Penelitian menggunakan metode IndoBERT, IndoLEM. Penelitian ini menghasilkan *dataset* dalam Bahasa Indonesia dengan memperhatikan morfo-sintaks, *semantic* dan wacana.

Penelitian [14] dengan judul *Implementasi Web Scraping dalam Pengumpulan Berita Kriminal pada Masa Pandemi Covid-19.* Permasalahan pada penelitian tersebut adalah Mengumpulkan berita kriminal yang terjadi pada masa pandemic *Covid-19* dari situs berita dengan menggunakan Teknik *web scraping*. Metode *web scraping* dapat dinilai lebih efektif bila dibandingkan dengan Teknik manual untuk pengmupulan data.

Penelitian [15] dengan judul *Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace.* Permasalahan pada penelitian tersebut adalah Melakukan pencarian data menggunakan metode *web scraping* untuk mendapatkan hasil terbaik. Dari hasil pengujian *White Box Testing* dan *Black Box Testing* didapatkan bahwa penggunaan teknik *web scraping* dinilai mampu memberikan hasil terbaik dalam pencarian dalam tiga situs marketplace sesuai kata kunci yang user inginkan.

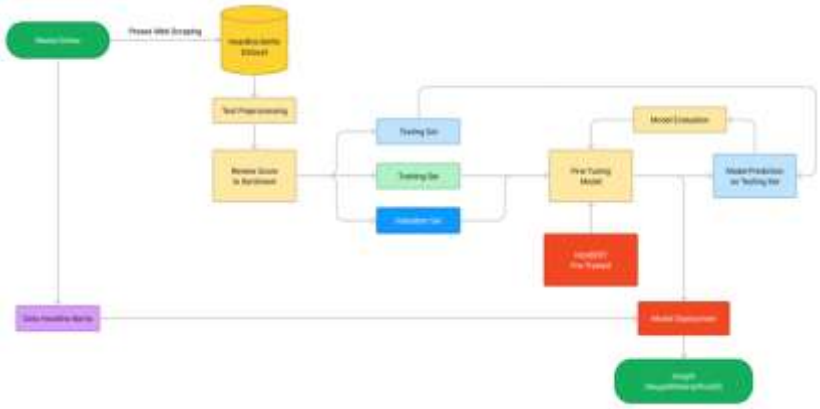
State of the art pada penelitian kami adalah mengimplementasikan *Teknik Web Scraping, Labeling* dan *Pre-processing Text* dalam mengumpulkan data, kemudian menerapkan algoritma IndoBERT dalam analisis sentimen terhadap pemberitaan peningkatan kasus positif *Covid-19*, serta mengevaluasi kinerja model sentimen analisis dengan variasi *batch size.*

3. Metodologi

Tahapan penelitian berupa: *Literature review, Web Scraping, Text Pre-processing, Model Evaluation*, Analisis dan Kesimpulan, sepeti disajikan pada gambar 1.

1) *Literature Review*

Tahan pertama yang di lakukan dalam hal ini ada *literature review* dengan menganalisis jurnal pada bidang sentimen analysis dengan menggunakan model yang sama. Beberapa jurnal literature review di ambil dari penelitian minimal lima tahun terakhir kebelakang agar informasi dan hasil penelitian masih terbarukan.



Gambar 1. Tahapan penelitian menggunakan model *IndoBERT*

2) ***Web Scraping***

Gambar 2. Tahapan *web scraping* untuk mengumpulkan data

Langkah berikutnya menggunakan Teknik web scraping untuk mengumpulkan data test dari beberapa sumber berita media online. Dengan melakukan beberapa Langkah-langkah untuk megumpulkan data. Hal pertama dalam melakukan web scraping adalah memcari headline berita tentang kasus kenaikan *Covid* dari beberapa sumber berita media online seperti kompas.com, detik.com, cnn.com dan tempo.com kemudian dari halaman berita tersebut akan diambil data komentar yang dipost oleh para pembaca setelah komentar tersebut berhasil di dapatkan lagkah selanjutnya mengekstrak data kedalam bentuk tabular seperti file .csv atau .xls.

3) *Labeling Data*

Labeling sentimen merupakan pemberian nilai sentimen terhadap suatu teks berupa positif, negatif, atau netral. Metode *labeling* memiliki kelebihan dan kekurangannya masing-masing. Untuk *labeling* manual, hasilnya akan lebih akurat tetapi memakan waktu yang cukup banyak apalagi ketika data yang diolah cukup besar. Jika *labeling* dilakukan secara otomatis seperti menggunakan *library* TextBlob, hasilnya tidak akan seakurat *labeling* manual tetapi lebih efisien dikarenakan hal ini dapat menghemat waktu dan tenaga.

Terdapat dua fungsi TextBlob yang digunakan yaitu *polarity()* dan *subjectivity()*. *Polarity* memiliki rentang nilai dari -1 hingga 1. Ketika nilai ini lebih kecil dari 0, maka sentimen termasuk ke dalam negatif, dan ketika nilai lebih besar dari 0 maka sentimen termasuk ke dalam positif. Kemudian untuk *subjectivity* memiliki rentang nilai dari 0 hingga 1. nilai ini mengacu pada pendapat dan penilaian pribadi.

4) *Text Preprocessing*

Text preprocessing dilakukan setelah data dalam bentuk tabular atau dokumen *excel* didapatkan dari proses *web scraping* dengan menggunakan metode *masic* untuk *text preprocessing*. Adapaun proses yang dilakukan dalam *text preprocessing* dengan cara mengubah *text* kedalam *Lower casing*, menghapus alamat URL dari sumber berita dan menghapus kata-kata yang memiliki frekuensi kemunculan lebih besar dari 5000 dan kurang dari sama dengan 1 pada seluruh dokumen.

4. Hasil dan Pembahasan

4.1 Persiapan Data

1) Web Scraping

Penelitian ini menggunakan data dari beberapa sumber berita media online dengan menyimpan hasil tersebut kedalam bentuk tabular dengan nama file covid-sentiment.csv.

2) Labeling Data

Dalam analisis sentimen dengan metode supervised learning, diperlukan dataset yang sudah memiliki label atau dianotasi. Labelisasi ini perlu dilakukan karena metode supervised learning membutuhkan contoh. Anotasi menggunakan dua fungsi *TextBlob* yang digunakan yaitu *polarity()* yang *subjectivity()*. *Polarity* memiliki rentang nilai dari -1 hingga 1. Ketika nilai ini lebih kecil dari 0, maka sentimen termasuk ke dalam negatif, dan ketika nilai lebih besar dari 0 maka sentimen termasuk ke dalam positif.

```
[2] #import pandas
    import pandas as pd

    #import googletrans
    !pip install googletrans==3.1.0a0
    import googletrans
    from googletrans import Translator

    #import textblob
    import textblob
    from textblob import TextBlob
```

Gambar 3 *Import library python* yang akan di gunakan

Gambar 4 Pemanggilan Dataset yang akan di olah

Gambar 5 *Output* hasil *labeling*

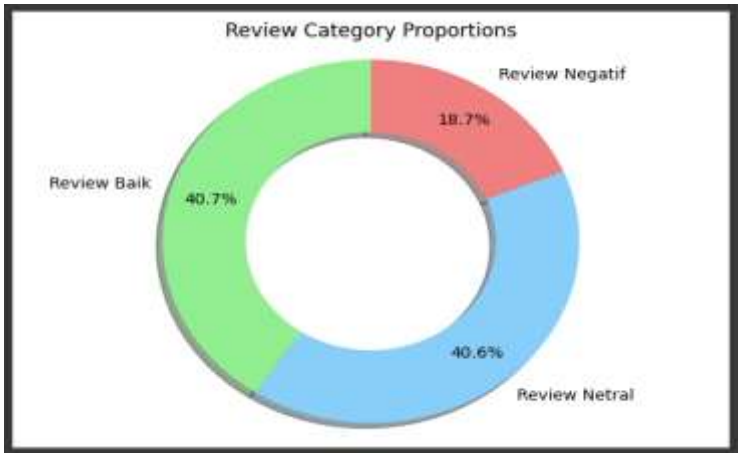
Jumlah dari masing-masing sentimen yang di-*generate* dapat diketahui dengan menggunakan fungsi *value_counts()*.

```
dataset['category'].value_counts()

positive    15012
neutral     15007
negative     6910
Name: category, dtype: int64
```

Gambar 6 Jumlah hasil dari labeling berdasarkan kategori

Dilakukan visualisasi terhadap jumlah sentimen yang didapat guna mempermudah kita dalam memperoleh informasi. Visualisasi yang dilakukan menggunakan *donut chart*.



Gambar 7 Visualisasi terhadap jumlah sentiment

Data hasil *labeling* disimpan ke dalam *file* baru dengan nama labeled_text.csv.

	A	B
1	review_text	category
2	pemerintah lampung bantuan mahasiswa derasnya donasi masuk pemerintah provinsi lampung tim gugus tugas covid 19	neutral
3	pemerintah menerapkan herd immunity kemampuan pemerintah nihil pemberantasan covid gambling kuat bertahan hidup lemah mati negara serius menangani virus	negative
4	lakukan makanan berbuka sahur 10 kepala keluarga terdampak covid hasik bantuan pemerintah masuk data ramadan rencanakan paket sembako lengkap semoga rejekinya	positive
5	warga positif corona bupati kondisi menuntut kebersamaan rakyat news bantaeng pemerintah kabupaten bantaeng tim gugus tugas penanganan covid 19 bantaeng meriis rakyatdot	positive
6	emosi banget kondisi disuruh liputan covid dosen bercanda sih pemerintah menyuruh dirumahaja disuruh liputan rumah	neutral
7	antisipa penyebaran pandemi covid 19 membutuhkan kerja pemerintah masyarakat pandemi musibah berkatagori non alam prosesnewsid psbb gorontalo	negative
8	bang pemerintah peduli republik rakyat cina urus abk meninggal 1 cina korban covid 19 ratusan meninggal pemerintah indonesia peduli bang kasihan keluarga bang sedih kali	negative
9	updates terkini covid 19 kamis 07 05 2020 sumber dinas kesehatan kabupaten magetan info grafis angka keselamatan waspada panik ikuti himbauan pemerintah kpulawancovid19	positive
10	pie mentari kouangan sri mulyani pemerintah provinsi daerah khusus kota jakarta pemerintah pusat menanggung bansos warga daerah khusus kota jakarta terdampak pandemi co	positive

Gambar 8 *Data file labeled_text.csv dari hasil labeling*

3) Pre-processing Data

Proses *case folding* digunakan untuk mengubah semua karakter menjadi huruf kecil. Tahapan ini dilakukan karena data yang diperoleh tidak selalu terstruktur dan konsisten dalam penggunaan huruf kapital, maka case folding dilakukan untuk menyamaratakan penggunaan huruf kapital. *Case folding* dilakukan dengan menggunakan fungsi *lower()* yang telah tersedia pada *library Python*.

```
[ ] character = [' ',',',':','.',...,'?','!','(',')','[',']','{','}','<','>','"','/','\','#','@',
                 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z',
                 'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

    # hapus karakter yang berulang
    def repeatcharClean(text):
      for i in range(len(character)):
        charac_long = 5
        while charac_long > 2:
          char = character[i]*charac_long
          text = text.replace(char,character[i])
          charac_long -= 1
      return text
```

Gambar 9 Proses *case folding* pada Dataset

Proses data *cleaning* ini digunakan untuk menghilangkan angka, beberapa simbol, url, *username* (@username), *hashtag* (#), spasi berlebih, tanda baca, emoji, dan pengulangan karakter yang ada pada kalimat. Tahapan ini menggunakan regular *expression* untuk menemukan karakter yang akan dihapus.

```
[ ] def clean_review(text):
    # ubah text menjadi huruf kecil
    text = text.lower()
    # ubah enter menjadi spasi
    text = re.sub(r'\n', ' ', text)
    # hapus emoji
    text = emoji.demojize(text)
    text = re.sub(':[A-Za-z_-]+:', ' ', text) # delete emoji
    # hapus emoticon
    text = re.sub(r"([xX;:]'?[dDpPvVo03)(])", ' ', text)
    # hapus link
    text = re.sub(r"(https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|www\.[a-zA-Z0-
    # hapus username
    text = re.sub(r"@[^\s]+[\s]?", ' ', text)
    # hapus hashtag
    text = re.sub(r'#(\S+)', r'\1', text)
    # hapus angka dan beberapa simbol
    text = re.sub('[^a-zA-Z,.?!]+',' ',text)
    # hapus karakter berulang
    text = repeatcharClean(text)
    # clear spasi
    text = re.sub('[ ]+',' ',text)
    return text
```

Gambar 10 Proses data *cleaning* pada Dataset

```
[ ] def preprocess_v1(df):
        df_pp = df.copy()
        df_pp.review_text = df_pp.review_text.map(clean_review)

        # delete empty row
        df_pp.review_text.replace('', np.nan, inplace=True)
        df_pp.review_text.replace(' ', np.nan, inplace=True)
        df_pp.dropna(subset=['review_text'], inplace=True)
        return df_pp
```

Gambar 11 Tahap *pre-processing* Dataset

```
[ ] # Hasil Pre-processing
    df_v1 = preprocess_v1(df)
    print(df_v1)

    df_v1.to_csv('dataset_clean.tsv', sep='\t', header=None, index=False)
```

	review_text	category
0	pemerintah lampung bantuan mahasiswa derasnya ...	neutral
1	pemerintah menerapkan herd immunity kemampuan ...	negative
2	lakukan makanan berbuka sahur kepala keluarga ...	positive
3	warga positif corona bupati kondisi menuntut k...	positive
4	emosi banget kondisi disuruh liputan covid dos...	neutral
...	...	...
36924	hoaks hoaks beredar covid pemerintah mengendal...	positive
36925	tingginya covid majelis permusyawaratan rakyat...	positive
36926	pakai masker cuci hand sanitizer namanya idiot...	negative
36927	kabupaten aceh selatan zona hijau terlibat pem...	negative
36928	keluyurann rumah pekerjaan beli sembako jalan ...	neutral

[36929 rows x 2 columns]

Gambar 12 Hasil *output pre-processing* Dataset

Hasil *pre-processing* di simpan ke dalam *file* dataset_clean.tsv. Proses tokenisasi menggunakan *regular expression* untuk menemukan karakter yang akan dihapus. Tahapan ini digunakan untuk memecah kalimat menjadi list kata. Proses ini menggunakan fungsi *word_tokenize* yang disediakan oleh library NLTK.

Proses normalisasi adalah tahap di mana dataset yang memiliki kata-kata tidak baku diubah menjadi kata yang baku atau sesuai dengan ejaan. hal ini dilakukan karena cukup banyak kalimat yang menggunakan kata gaul seperti: tdk, dmn, cpt, ga, enggak, ngga, gak. Jika kata tersebut tidak melalui tahap normalisasi, maka sistem akan menganggap kata ga, enggak, ngga, dan gak adalah kata yang berbeda, padahal kata tersebut memiliki makna yang seharusnya sama yaitu enggak.

4) Dataset *Splitting*

Dataset splitting adalah suatu teknik yang digunakan untuk melihat kinerja model dengan melakukan pembagian terhadap data yang akan kita olah menjadi beberapa bagian dalam hal ini *training*, *validation* dan *testing*. Dataset *training* digunakan untuk melatih model, dataset validasi digunakan untuk meminimalisir *overfitting* yang sering terjadi pada jaringan syaraf tiruan, sedangkan dataset *testing* sendiri digunakan sebagai test akhir untuk melihat keakuratan jaringan yang sudah dilatih dengan dataset *training*. Proporsi split dataset pada penelitian ini adalah 60% *train set*, 20% *validation set*, 20% *test set.*

5) Implementasi Indo-BERT

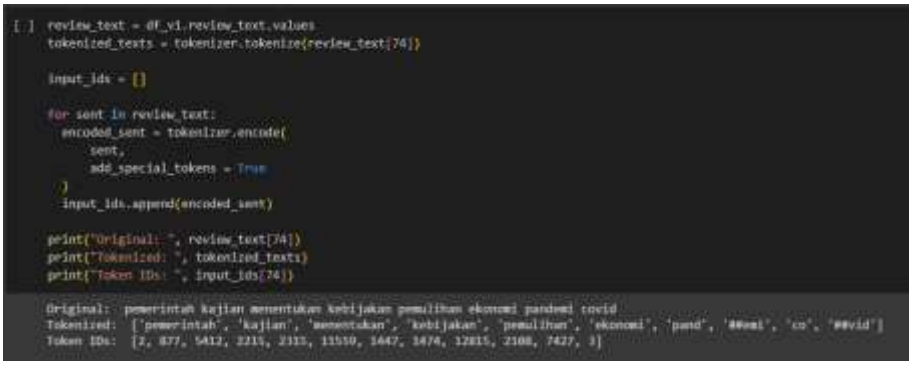
Pada penelitian ini menggunakan teknik *fine-tuning* dengan model *IndoBERT-base-p1*, salah satu model yang menggunakan arsitektur BERT-base. *Library* menggunakan Transformers yang disediakan oleh *HuggingFace*. *Library* ini menyediakan ribuan *pre-trained model* yang dapat digunakan untuk melakukan tugas-tugas klasifikasi, ekstraksi informasi, tanya jawab, *summarization*, translasi, text generation dan lain-lain dalam 100 bahasa. Transformers didukung oleh dua *library deep learning* yang terkemuka yaitu *PyTorch* dan *TensorFlow.*

Sebelum dilakukan training pada BERT, dataset harus disesuaikan dengan input yang dapat diterima oleh BERT. Oleh karena itu dibutuhkan BertTokenizer, sebuah tokenizer yang bertujuan untuk melakukan tokenisasi pada kalimat-kalimat dan menghasilkan input yang sesuai. Berikut adalah inisialisasi *common function* yang digunakan sebagai persiapan dalam menggunakan model BERT sekaligus BertTokenizer.

Pada tahap ini dilakukan inisialisasi IndoBERT-base-p1 sebagai model dan BertTokenizer yang akan digunakan untuk pelatihan.

Pada tahap ini dilakukan implementasi kelas *DocumentSentimentDataset* untuk data *loading*. *Subwords* yang dikembalikan oleh dataset dapat memiliki panjang yang berbeda-beda untuk setiap *index*. Untuk dapat diproses secara paralel oleh model, perlu dilakukan standarisasi panjang dari *subwords* dengan memotong beberapa *subwords* atau menambahkan padding token. Untuk itu perlu mengimplementasikan *DocumentSentimentDataLoader* yang memproses *list subword* dan sentiment dan mengeluarkan padded_subword, mask, dan sentiment.

Gambar 13 Proses standarisasi panjang dari subwords



Gambar 14 Contoh data yang sudah di tokenisasi

Kebanyakan *hyperparameters* ketika *fine-tuning* BERT tetap sama seperti training BERT biasanya. Sedangkan BERT dapat di-finetuning dengan menyesuaikan dengan menyesuaikan hyperparameters-nya. Hyperparameters yang digunakan dalam training BERT antara lain:

a) *Batch size* adalah jumlah sampel yang dimasukkan ke dalam *network* sebelum *weight* disesuaikan. Semakin besar *batch size*, maka semakin lama waktu yang dibutuhkan untuk menyelesaikan satu batch.
b) *Epoch* adalah jumlah berapa kali jaringan melihat seluruh dataset. Satu *epoch* terjadi ketika semua contoh sudah melewati jaringan baik *forward pass* dan *backward pass*.
c) *Learning rate* menentukan seberapa banyak *weight* pada *Neural network* yang akan diubah. Semakin tinggi rate-nya, semakin cepat gradient bergerak menuju landscapes.

Adapun *hyperparameters* untuk *fine-tuning* IndoBERT yang digunakan dalam penelitian ini, antara lain: *Batch size* 16, *learning rate* (Adam) 3e-6, dengan *epoch* 5, *epoch* 10 dan *epoch* 15 (akan disajikan sampel untuk *epoch* 5 dan *epoch* 15)

4.2 Hasil Evaluasi *IndoBERT*

Hasil dari proses *training* dan evaluasi dari *Batch size* 16 dan *learning rate* (Adam) 3e-6 dapat dilihat berikut:

Tabel 1 Hasil Presisi, *Recall*, dan *F1 score* dari Training dan Validasi dengan total *epoch* 5

Training				Validasi			
Epoch	***Recall***	**Presisi**	***f1 Score***	**Epoch**	***Recall***	**Presisi**	***f1 Score***
1	65%	68%	66%	**1**	78 %	77%	77%
2	79%	80%	79%	**2**	77%	78%	77%
3	83%	84%	83%	**3**	81%	81%	81%
4	86%	87%	86%	**4**	81%	82%	82%
5	89%	89%	89%	**5**	81%	81%	81%

Tabel 2 Hasil Presisi, *Recall*, dan *F1 score* dari Training dan Validasi dengan total *epoch* 15

Training				Validasi			
Epoch	***Recall***	**Presisi**	***f1 Score***	**Epoch**	***Recall***	**Presisi**	***f1 Score***
1	65%	68%	66%	**1**	78 %	77%	77%
2	79%	80%	79%	**2**	77%	78%	77%
3	83%	84%	83%	**3**	81%	81%	81%
4	86%	87%	86%	**4**	81%	82%	82%
5	89%	89%	89%	**5**	81%	81%	81%
6	91%	92%	92%	**6**	81%	81%	81%
7	93%	93%	93%	**7**	81%	82%	81%
8	95%	95%	95%	**8**	81%	79%	80%
9	96%	96%	96%	**9**	80%	81%	81%
10	97%	97%	97%	**10**	81%	80%	80%
11	97%	98%	97%	**11**	80%	80%	80%
12	98%	98%	98%	**12**	79%	80%	79%
13	98%	98%	98%	**13**	80%	80%	80%
14	98%	98%	98%	**14**	80%	79%	80%
15	98%	98%	98%	**15**	81%	80%	80%

Hasil dari proses *training* dan evaluasi dari *Batch size* 32 dan *learning rate* (Adam) 3e-6 dapat dilihat berikut:

Tabel 3 Hasil Presisi, *Recall*, dan *F1 score* dari Training dan Validasi dengan total *epoch* 5

Training				Validasi			
Epoch	***Recall***	**Presisi**	***f1 Score***	**Epoch**	***Recall***	**Presisi**	***f1 Score***
1	62%	65%	62%	**1**	75%	76%	75%
2	77%	79%	78%	**2**	79%	78%	78%

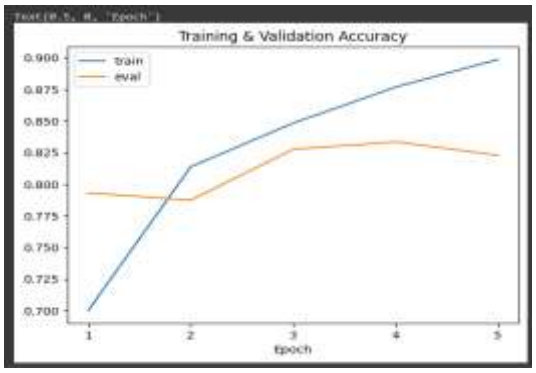
Epoch	*Recall*	Presisi	*f1 Score*
3	81%	82%	82%
4	84%	85%	84%
5	86%	87%	87%

Epoch	*Recall*	Presisi	*f1 Score*
3	81%	80%	81%
4	81%	81%	81%
5	81%	81%	81%

Tabel 4 Hasil Presisi, *Recall*, dan *F1 score* dari Training dan Validasi dengan total *epoch* 15

Training				Validasi			
Epoch	***Recall***	**Presisi**	***f1 Score***	**Epoch**	***Recall***	**Presisi**	***f1 Score***
1	62%	65%	62%	**1**	75%	76%	75%
2	77%	79%	78%	**2**	79%	78%	78%
3	81%	82%	82%	**3**	81%	80%	81%
4	84%	85%	84%	**4**	81%	81%	81%
5	86%	87%	87%	**5**	81%	81%	81%
6	89%	89%	89%	**6**	81%	80%	80%
7	91%	91%	91%	**7**	81%	81%	81%
8	92%	93%	93%	**8**	81%	80%	80%
9	94%	94%	94%	**9**	81%	81%	81%
10	95%	95%	95%	**10**	81%	81%	81%
11	96%	96%	96%	**11**	80%	80%	80%
12	97%	97%	97%	**12**	80%	80%	80%
13	97%	97%	97%	**13**	80%	80%	80%
14	97%	98%	98%	**14**	80%	80%	80%
15	98%	98%	98%	**15**	81%	81%	81%

Setelah melalui proses perulangan per *epoch* pada model, hasil pelatihan disimpan. Gambar 15 menyajikan contoh hasil akurasi pelatihan dan evaluasi selama beberapa *epoch* untuk memantau kinerja model selama proses pelatihan.



Gambar 15 Grafik Akurasi *Training* dan Validasi dengan *Batch Size* 16 dan Epoch 5

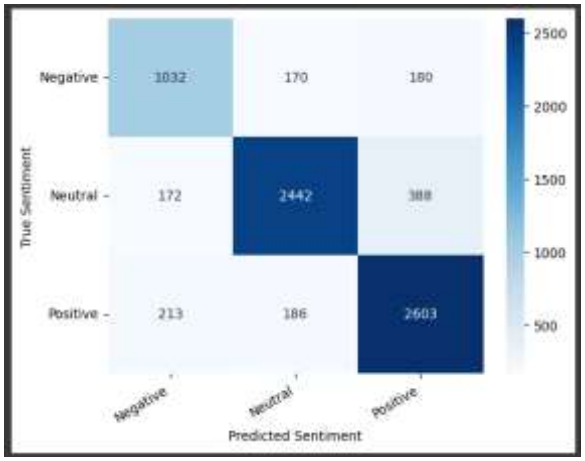
4.3 Hasil Evaluasi Pengukuran Performa

Setelah dilakukan proses training maka selanjutnya dilakukan uji coba untuk mengetahui performa model terhadap data baru. Berdasakan hasil implementasi model terhadap data *testing* didapatkan akurasi sebagai berikut:

Tabel 5 Hasil nilai *Loss* pada data setelah peroses testing

Arsitektur	Epoch	*F1 Score*	*Accuracy*	*Recall*	Presisi	*Loss*
IndoBERT Batch_Size 16	5	80.10%	81.60%	80.24%	80.10%	55.23%
	10	80.51%	82.19%	80.37%	80.66%	76.79%
	15	80.53%	82.07%	80.82%	80.27%	93.53%
IndoBERT Batch_Size 32	5	80.51%	82.19%	80.37%	80.66%	51.53%
	10	79.94%	81.66%	79.74%	80.20%	67.77%
	15	80.16%	81.91%	80.14%	80.19%	87.41%

Diagram *confusion matrix* dan performansi model pada tiap-tiap hasil percobaan data validasi sebagai berikut.

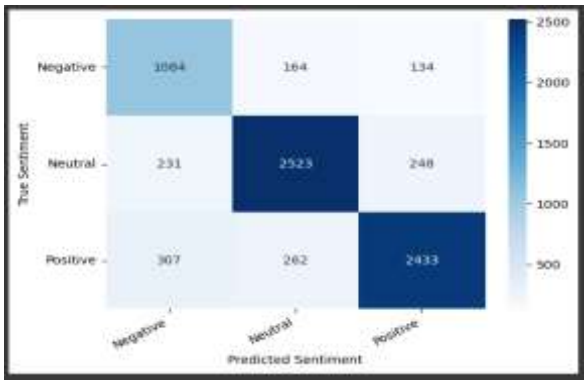


(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.73	0.75	0.74	1382
Neutral	0.87	0.81	0.84	3002
Positive	0.82	0.87	0.84	3002
accuracy			0.82	7386
macro avg	0.81	0.81	0.81	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 16 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 16 dan 5 *epoch*

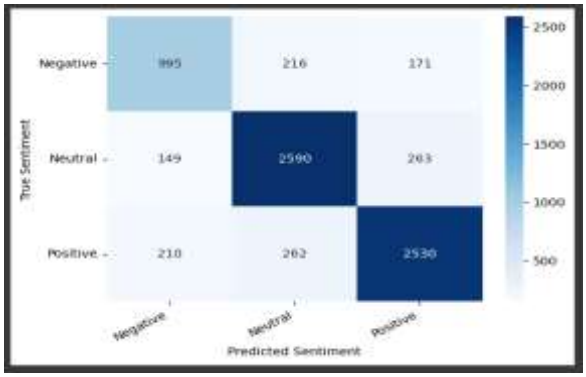


(a) Confusion Matrix

	precision	recall	f1-score	support
Negative	0.69	0.75	0.72	1382
Neutral	0.86	0.84	0.85	3002
Positive	0.85	0.84	0.85	3002
accuracy			0.82	7386
macro avg	0.80	0.81	0.80	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 17 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 16 dan 15 *epoch*

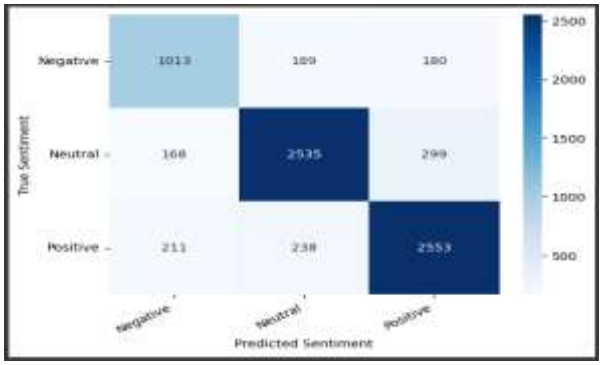


(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.73	0.72	0.73	1382
Neutral	0.84	0.86	0.85	3002
Positive	0.85	0.84	0.85	3002
accuracy			0.83	7386
macro avg	0.81	0.81	0.81	7386
weighted avg	0.83	0.83	0.83	7386

(b) Performansi Model

Gambar 18 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 32 dan 5 *epoch*



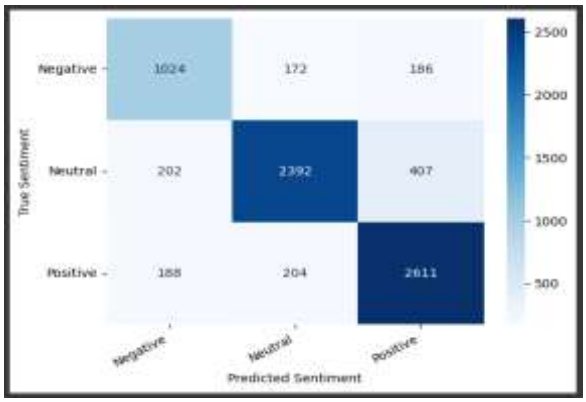
(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.73	0.73	0.73	1382
Neutral	0.86	0.84	0.85	3002
Positive	0.84	0.85	0.85	3002
accuracy			0.83	7386
macro avg	0.81	0.81	0.81	7386
weighted avg	0.83	0.83	0.83	7386

(b) Performansi Model

Gambar 19 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 32 dan 15 *epoch*

Diagram *confusion matrix* dan performansi model pada tiap-tiap hasil percobaan data test sebagai berikut:

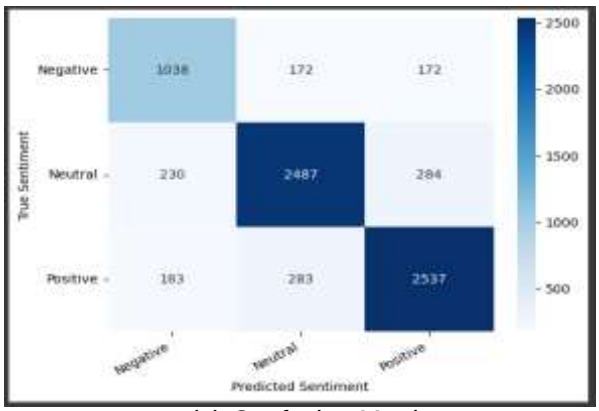


(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.72	0.74	0.73	1382
Neutral	0.86	0.80	0.83	3001
Positive	0.81	0.87	0.84	3003
accuracy			0.82	7386
macro avg	0.80	0.80	0.80	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 20 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 16 dan 5 *epoch*

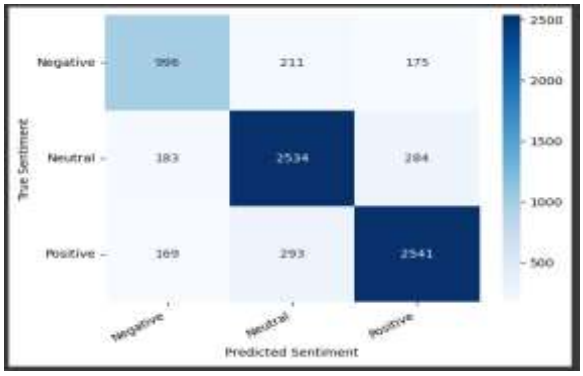


(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.72	0.75	0.73	1382
Neutral	0.85	0.83	0.84	3001
Positive	0.85	0.84	0.85	3003
accuracy			0.82	7386
macro avg	0.80	0.81	0.81	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 21 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 16 dan 15 *epoch*

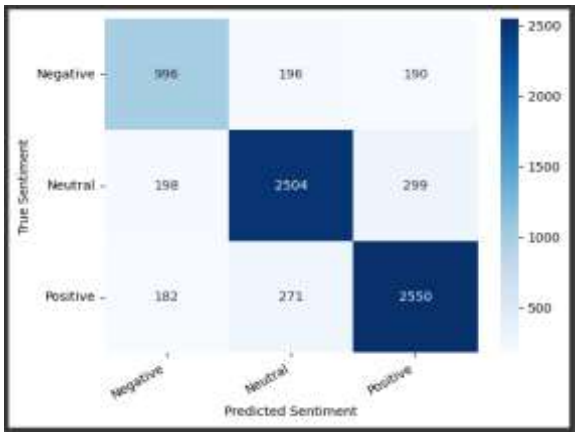


(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.74	0.72	0.73	1382
Neutral	0.83	0.84	0.84	3001
Positive	0.85	0.85	0.85	3003
accuracy			0.82	7386
macro avg	0.81	0.80	0.81	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 22 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 32 dan 5 *epoch*



(a) *Confusion Matrix*

	precision	recall	f1-score	support
Negative	0.72	0.72	0.72	1382
Neutral	0.84	0.83	0.84	3001
Positive	0.84	0.85	0.84	3003
accuracy			0.82	7386
macro avg	0.80	0.80	0.80	7386
weighted avg	0.82	0.82	0.82	7386

(b) Performansi Model

Gambar 23 Diagram *Confusion Matrix* dan Performansi Model dengan *Batch size* 32 dan 15 *epoch*

5. Simpulan

Pada *Batch Size* 16 Model cenderung memiliki kinerja yang konsisten dengan *Batch Size* 16, dengan F1 *Score* berkisar antara 80.10% hingga 80.53%. Sedangkan *Batch Size* 32 Terlihat bahwa pada *Batch Size* 32, terdapat variasi dalam F1 *Score, Accuracy*, dan *Loss*. Meskipun pada beberapa kasus hasilnya lebih baik, pada kasus tertentu, *Batch Size* 16 dapat memberikan hasil yang lebih baik.

Peningkatan jumlah epoch tidak selalu berarti peningkatan kinerja yang signifikan. Meskipun pada beberapa kasus F1 *Score* dan *Accuracy* meningkat (seperti dari Epoch 5 ke Epoch 10), terdapat juga kasus di mana terjadi penurunan (seperti dari Epoch 10 ke Epoch 15). Epoch 10 pada *Batch Size* 16 memiliki F1 *Score* dan *Accuracy* yang tinggi, sementara pada *Batch Size* 32, Epoch 5 memberikan hasil yang baik.

Secara umum, model tampaknya memiliki kinerja yang baik dengan F1 *Score* dan *Accuracy* di atas 0.80 dan 0.81, masing-masing. *Loss* cenderung meningkat seiring dengan peningkatan *epoch*, tetapi perlu diingat bahwa *Loss* bukan satu-satunya indikator kinerja model. Kesimpulannya, perlu dilakukan eksplorasi lebih lanjut dan eksperimen untuk memahami faktor-faktor yang mempengaruhi kinerja model secara lebih mendalam. Pemilihan *Batch Size* dan *Epoch* dapat tergantung pada kondisi dan kebutuhan spesifik. Perlu diingat bahwa performa model bisa berbeda-beda tergantung pada dataset dan tugas tertentu.

Evaluasi lebih lanjut, seperti menganalisis metrik lainnya dan melakukan validasi silang, dapat membantu memberikan pemahaman yang lebih komprehensif tentang kinerja model.

Daftar referensi:

[1] I.R. Ginting, M.R. Makful, M. Muhtar, & J. Pusat, "*Pola Penyebaran COVID-19 di DKI Jakarta pada Bulan Maret-Juli Tahun 2020 Secara Spasial*. Pp. 161–169, 2020.

[2] E.P. Kurniasih, *Dampak Pandemi Covid 19 Terhadap Penurunan Kesejahteraan Masyarakat Kota Pontianak*. Pp. 277–289, 2020.

[3] K.S. Nugroho, A.Y. Sukmadewa, F.A. Bachtiar, & N. Yudistira, *BERT Fine-Tuning for* Sentimen *Analysis on Indonesian Mobile Apps Reviews*. Pp. 1–10, 2020.

[4] Y.V. Wijaya, A. Erfina, & C. Warman, "Analisis Sentimen Seputar UU ITE Menggunakan Algoritma Support Vector Machine". *Progresif: Jurnal Ilmiah Komputer*, Vol. 17, no. 2, pp. 1-14, 2021.

[5] H. Juwiantho, E.I. Setiawan, J. Santoso, & M.H. Purnomo, "Sentiment analysis twitter bahasa indonesia berbasis word2vec menggunakan deep convolutional neural network". *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 1, pp. 181-188, 2020.

[6] T. Baldwin, *IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP,* 2020.

[7] S. Satriajati, S.B. Panuntun, & S. Pramana, "Implementasi web scraping dalam pengumpulan berita kriminal pada masa pandemi COVID-19. In *Seminar Nasional Official Statistics,* Vol. 2020, No. 1, pp. 300-308, 2020

[8] K.S. Nugroho, A.Y. Sukmadewa, D.W.H. Wuswilahaken, F.A. Bachtiar, & N. Yudistira, "BERT fine-tuning for sentiment analysis on Indonesian mobile apps reviews. In *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology,* pp. 258-264, 2021.

[9] J.C. Devlin, "BERT: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies- Proceedings of the Conference*, pp. 4171–4186, 2019.

[10] M. Singh, A.K. Jakhar, & S. Pandey, "Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 33-42, 2021.

[11] N. Hidayah, & S. Sahibu, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 820-826, 2021.

[12] T. Wang, K. Lu, K.P. Chow, & Q. Zhu, "COVID-19 sensing: negative sentiment analysis on social media in China via BERT model. *Ieee Access*, vol. *8*, pp. 138162-138169, 2020.

[13] F. Koto, A. Rahimi, J.H. Lau, & T. Baldwin, T. IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. *arXiv preprint arXiv:2011.00677, 2020*.

[14] S. Satriajati, S.B. Panuntun, & S. Pramana, "Implementasi web scraping dalam pengumpulan berita kriminal pada masa pandemi COVID-19. In *Seminar Nasional Official Statistics,* vol. 2020, no. 1, pp. 300-308, 2020.

[15] D.D.A. Yani, H.S. Pratiwi, & H. Muhardi, "Implementasi web scraping untuk pengambilan data pada situs marketplace. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 257-262, 2019.