

Modeling Dan Analisis Menggunakan Sentiment Analysis Mengenai Konten Youtube Pada Channel Gadgetin Dengan Metode Fine-Tune Menggunakan Pre-Trained Model

Nurtriana Hidayati^{1*}, Ali²

^{1,2} Teknologi Informasi, Universitas Semarang

Kata Kunci:

Sentiment Analysis, Fine-Tune, Pre-Trained, BERT

Keywords:

Sentiment Analysis, Fine-Tune, Pre-Trained, BERT

Diterima : 04 Januari 2024
Revisi terakhir : 05 Februari 2024
Publish : 28 Februari 2024

Copyright © 2023 by Author. Published by Universitas Cipasung Tasikmalaya

ABSTRAK

Sentiment Analysis adalah proses menganalisis teks digital untuk menentukan apakah nada emosional suatu pesan positif, negatif, atau netral. Analisis akan dibuat berdasarkan hasil dari deteksi model machine learning terhadap data komentar. Model machine learning dibangun menggunakan metode fine-tune menggunakan pre-trained model. Model pre-trained Bidirectional Encoder Representations from Transformers (BERT) merupakan model yang sedang populer di kalangan peneliti. BERT merupakan Natural Language Processing (NLP) yang dikembangkan Google. Penelitian ini bertujuan untuk mengetahui nilai akurasi dari model BERT untuk analisis sentimen pada komentar Youtube. Data didapatkan dari komentar pada salah satu konten Youtube Channel "GadgetIn" yang dianalisis menggunakan pre-trained model BERT. Data yang digunakan adalah 500 komentar dimana 400 komentar digunakan sebagai data Training, 50 komentar sebagai data validation dan 50 komentar sebagai data Testing. Hasil dari penelitian ini menunjukkan nilai akurasi sebesar 80 %.

ABSTRACT

Sentiment Analysis is the process of analyzing digital text to determine whether the emotional tone of a message is positive, negative, or neutral. The analysis is based on the results of a machine learning model's detection of comments data. The machine learning model is built using the fine-tuning method with a pre-trained model. The pre-trained model used is the Bidirectional Encoder Representations from Transformers (BERT), which is a popular model among researchers. BERT is a Natural Language Processing (NLP) model developed by Google. The aim of this research is to ascertain the Accuracy value of the BERT model for Sentiment Analysis on YouTube comments. The data is obtained from comments on a YouTube Channel's content, "GadgetIn," and is analyzed using the pre-trained BERT model. The dataset used consists of 500 comments, where 400 comments are used as Training data, 50 comments for validation, and 50 comments for Testing. The results of this research show an Accuracy value of 80%.

1. PENDAHULUAN

Youtube merupakan salah satu dari *social Network* yang berkategori Multimedia sharing. Pada umumnya, file-file Multimedia yang dibagikan berupa file gambar, audio, dan video melalui internet. Fitur-fitur yang disediakan multimedia sharing antara lain mengirim dan menanggapi komentar, pemberian peringkat (*rate*), berbagi (*share*) dalam bentuk alamat (URL). Proses yang dilakukan meliputi unggah (*upload*) dan unduh (*download*). Sudah banyak penelitian dilakukan tentang pola interaksi pengguna di YouTube ini. Interaksi ini berupa komentar yang biasanya digunakan untuk menunjukkan perasaannya, memberikan dukungan emosi, kesan, atau nasehat sebagai tanggapan atas video (Sadia et al., 2018).

Diantara banyaknya jenis konten yang tersedia di Youtube, konten *review/* ulasan menjadi salah satu favorit yang banyak dicari oleh para pengguna. Konten *review/* ulasan yang tersedia di Youtube sendiri banyak sekali ragamnya mulai dari makanan/ kuliner, *tempat* wisata, otomotif, gadget, fashion, dan lain sebagainya. Konten *review* sendiri adalah konten yang menampilkan konten creator yang sedang mereview/ mengulas sebuah produk. Isi dari konten reviewnya sendiri bisa terdiri dari deskripsi produk, spesifikasi, kelebihan dan kekurangan, pengalaman selama menggunakan produk tersebut, dan lain sebagainya. Hasil dari *review/* ulasan yang disajikan oleh konten creator bisa jadi referensi dan pertimbangan kita untuk membeli atau memakai produk tersebut.

Di YouTube ada berbagai macam jenis channel review salah satunya yaitu Gadgetin, Gadgetin

adalah salah satu channel review yang ada di Youtube dengan menyediakan konten yang mereview produk Gadget / teknologi terbaru terutama Smartphone. Selain mereview smartphone Gadgetin juga mereview kamera, tv, monitor, komputer, laptop, sampai gadget – gadget unik yang di jual di jual di onlineshop. David Brendi atau orang banyak memanggilnya David Gadgetin adalah pemilik dari channel gadgetin ini, dia membuat channel ini pada tahun 2014. Selama 7 tahun di dunia Youtube, channel gadgetin menjadi salah satu channel favorit untuk review produk gadget dengan jumlah subscriber mencapai 7,11 jt dengan rata – rata view per video mencapai 1,18 jt view. Ini menunjukkan bahwa konten – konten yang disajikan di channel ini cukup menarik untuk ditonton.

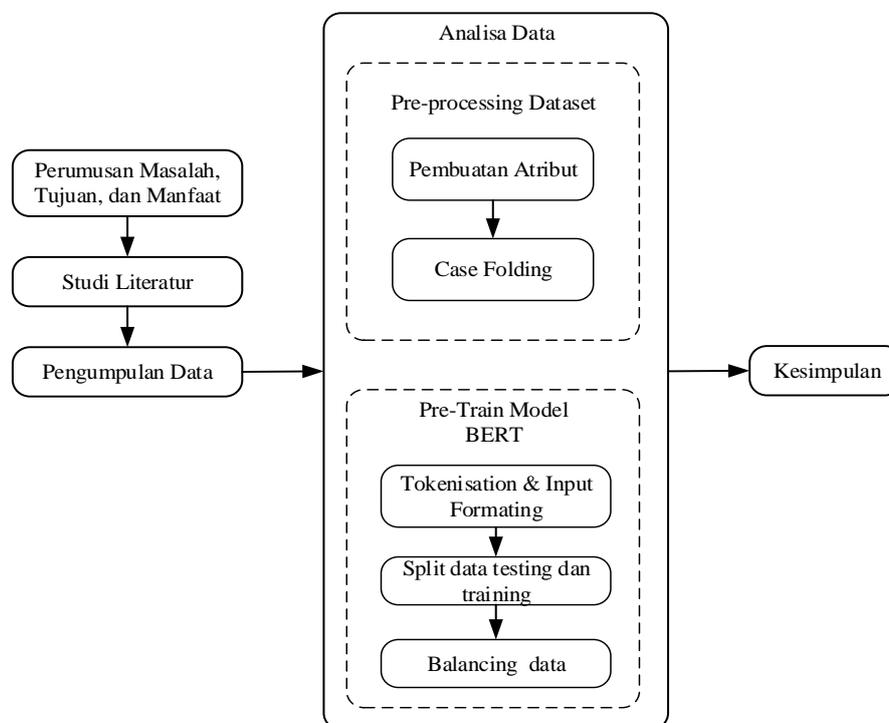
“Spek bengis, harga manis - Review POCO X5 Pro 5G Indonesia!” merupakan video yang baru seminggu di upload oleh channel Youtube gadgetin namun sudah mendapatkan respon komentar sebanyak 3000 lebih komentar. Komentar atau bisa disebut juga Opini public terhadap tayangan dapat diketahui dengan analisis sentimen. Analisis sentimen dapat menentukan, mengenali atau mengategorikan emosi atau opini pengguna pada sebuah komentar positif, negatif atau netral.

Penelitian dengan topik Sentiment Analysis pada data komentar YouTube pernah dilakukan oleh menggunakan Gaussian Naïve Bayes dan Multinomial Gaussian Naïve Bayes. Namun pada penelitian ini akan menggunakan metode fine-tune menggunakan pre-trained model BERT. BERT dilatih pada teks tak berlabel dalam jumlah besar dengan model bahasa bertopeng untuk tugas prediksi kalimat selanjutnya. Ini mengungguli model tradisional dalam sebelas jenis tugas NLP, termasuk analisis sentimen.

Penelitian ini bertujuan untuk menghasilkan informasi akurasi model yang dibuat dan juga berupa analisis perbandingan nada emosional setiap konten berupa banyaknya komentar positif, negatif atau netral di Channel Gadgetin.

2. METODE PENELITIAN

Langkah awal penelitian ini adalah menentukan permasalahan yang akan di selesaikan, tujuan dan manfaat, kemudian mencari literatur sebagai pendukung penyelesaian permasalahan yang sudah di tentukan. Hasil literatur disusun untuk merumuskan hipotesis terhadap sumber masalah dan data yang akan diolah. Dari hasil hipotesis, lalu dilakukan analisis dengan *Bidirectional Encoder Representations from Transformers* (BERT), dan dilanjutkan membuat sentimen analisis konten Review Poco X5 dan performansi dari BERT. Hasil sentimen analisis tersebut, peneliti membuat kesimpulan. Tahapan metode penelitian diatas tertuang pada gambar 1, sebagai berikut :



Gambar 1. Metode Penelitian Sentimen Analisis dengan *Bidirectional Encoder Representations from Transformers* (BERT)

a. Akuisisi Data

Pada perumusan masalah dilakukan dengan melihat berita mengenai berita terkait HP Poco X5, yang ada pada Youtube “GadgetIn” sebanyak 500 komentar. Selanjutnya menentukan tujuan yaitu Mengukur efisiensi Metode Fine-Tune Menggunakan Pre-Trained Model dalam pengklasifikasian Sentiment Analysis pada komentar Youtube “GadgetIn”, mengetahui respon masyarakat (viewer) terhadap konten Youtube “GadgetIn”.

Dalam mencari konten dalam Youtube, peneliti mengunjungi channel gadgetin pada tayangan terbaru dengan judul “Spek bengis, harga manis - Review POCO X5 Pro 5G Indonesia!” pada channel gadgetin cukup banyak mendapat respon dari viewer dalam waktu seminggu setelah diunggah, kemudian dilakukan *Crawling Comment* dengan *Youtube Data API V3* untuk menghasilkan dataset dengan format CSV yang terlihat pada tabel 1 d bawah ini.

Tabel 1. Data Akuisisi

User	Komentar	Waktu
BHIRAMA	Z,ðÿ...°i, Nyesel banget bang beli poco x5. Padahal di store resmi MI nya langsung. Cepet panas buat main pubg dan batreinya ngga awet. Kecewa sumpah	2023-07-09 09:11:43
Nur Farida	Baru beli x5 ðÿ~€, keluar x5 proðÿ™	2023-07-15 10:20:48
Rey	Bah duit duit dia sih bukan pake duit mak luðÿ¤“ðÿ¤“ðÿ¤£ðÿ¤£	2023-07-21 12:35:22
tgb 30	<ahref=""https://www.youtube.com/watch?v=mrDGb4jj32Y&=9m54s"">9:54 ada yang salfok sm gantungan tas yang kecilnya juga gak? Heheðÿ~...ðÿ~"	2023-07-09 23:41:32

b. Pre - Processing

Pre-processing data dilakukan untuk menyiapkan data mentah menjadi data yang lebih bersih agar pemrosesan data lebih efektif dan efisien, melalui tahapan

1. Pembersihan

Pembersihan pada dataset penelitian masih terdapat simbol, karakter, hashtag, dan emoji yang tidak mengandung opini. Maka dilakukan proses filterasi dengan menghapus karakter atau kata yang tidak diperlukan agar data tidak terlalu besar dan proses Training dapat berjalan lebih efektif. Pada tahap ini peneliti menggunakan microsoft excel dan Kutools For Excel. Tahapannya meliputi Mengubah representasi unicode kedalam bentuk yang mudah dibaca, Menghapus hashtag, Menghapus simbol yang tidak perlu, Menghapus emoji, Menghapus URL, Mengubah semua kata menjadi huruf kecil (*case folding*). Dengan hasil ditampilkan pada tabel 2 berikut di bawah ini.

Tabel 2. Pembersihan dataset

Kondisi	Hasil
Komentar Awal	“Bangg davidd konten mystery box dong, Yg setuju bantu up donggðÿ~ #up”
Decoding	Bangg davidd konten mystery box dong, Yg setuju bantu up donggðÿ~ #up
Menghapus hastag	Bangg davidd konten mystery box dong, Yg setuju bantu up donggðÿ~
Menghapus simbol yang tidak perlu	Bang david konten mystery box dong, Yg setuju bantu up dong
Menghapus Emoji	Bang david konten mystery box dong, Yg setuju bantu up dong
Menghapus URL	Bang david konten mystery box dong, Yg setuju bantu up dong
Case Folding	bang david konten mystery box dong, yg setuju bantu up dong

kemudian dilakukan penghapusan duplikasi data dan data yang kosong. Dari 3486 data, didapatkan 3345 data komentar.

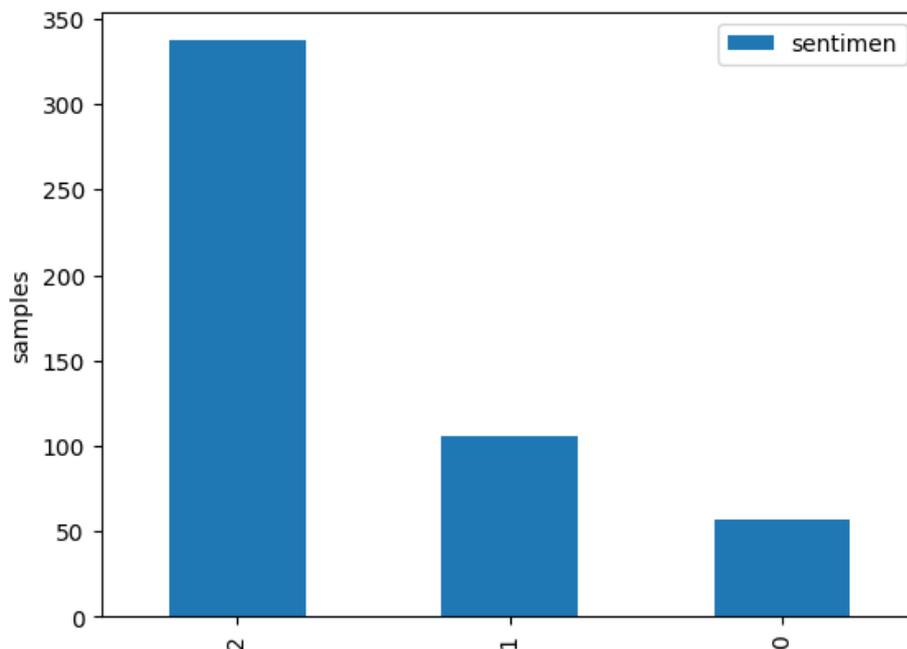
c. Pelabelan data

Hasil dari pre processing komentar bahasa Indonesia bersih, kemudian dilakukan pelabelan. Untuk optimalisasi, dari 3.345 data maka hanya 500 data yang dipakai. Berikut contoh hasil pelabelan komentar. Hasil pelabelan terlihat pada tabel 3 berikut.

Tabel 3. Pelabelan komentar

Komentar	Sentimen	Label
Gk suka bang, ic power rawan matot	Negative	0
Karna speck nya bagus	Positive	1
Sayang fps nya sering stuck 40 fps	Negative	0
Klo udh ngumpul mau ngapain bang? Giveaway hp baru apa gmn??	Netral	2
Rekomenin hp poco yg bagus di kamera dan prosesornya	Netral	2

Sentimen negative adalah komentar yang berisi kata kata hasutan, hinaan terhadap video yang diunggah. Komentar tersebut akan diberikan label 0. Sentimen positif adalah komentar yang memuji, memberikan dukungan, atau memberikan umpan balik positif video yang diunggah. Komentar tersebut akan diberi label 1. Komentar dengan sentimen netral dimana tidak terlihat pendapat positif ataupun negatif akan diberi label 2 seperti pada gambar 2 berikut.



Gambar 2. Pelabelan Data

d. Split Data

Sebelum dilakukan klasifikasi, data yang telah dilabeli dibagi menjadi 3 yaitu data train, data test, dan data validation dengan menggunakan metode sklearn train test split. Data train digunakan pengembangan model, data test digunakan untuk menguji dan melihat keakuratan model, sedangkan data validation digunakan untuk memvalidasi kinerja model dan meminimalisir overfitting yang sering terjadi. Scikit learn merupakan library Python yang sering digunakan untuk split data. Pada penelitian ini sebanyak 80% data berlabel digunakan untuk membangun model, sedangkan data lainnya akan digunakan untuk menerapkan model melakukan klasifikasi sentimen.

Pembagian data ke dalam data Training dan data Testing adalah sebanyak 80% data Training, 10% data validation, dan 10% data Testing. Hasil dari pembagian label terlihat pada gambar 3 di bawah ini.

```

== Train ==
Input: (405, 275)
Label: (405,)
Mask: (405, 275)

== Validation ==
Input: (45, 275)
Label: (45,)
Mask: (45, 275)

== Test ==
Input: (50, 275)
Label: (50,)
Mask: (50, 275)
    
```

Gambar 3. Pebagian Data yang telah dilabeli

3. HASIL DAN PEMBAHASAN

Data yang telah di bagi menjadi 3 bagian, akan dilakukan proses klasifikasi dengan tahapan sebagai berikut :

a. Pre-Train Model BERT

Dataset yang digunakan harus disesuaikan dengan representasi input yang diterima oleh IndoBERT. Setelah didapatkan *dataset* yang bersih hasil preprocessing selanjutnya dilakukan penambahan token khusus yaitu [CLS] pada awal kalimat, dan token [SEP] pada tiap akhir kalimat sebagai token pemisah antar kalimat. Kemudian dilakukan pengkodean (*encoding*) menggunakan *tokenizer* sesuai indeks *vocabulary* dari *pre-trained* IndoBERT. IndoBERT telah memiliki *vocabulary* yang nantinya akan digunakan sebagai proses transfer learning.

Pada penelitian ini, jumlah kata paling banyak dari semua komentar adalah 272 kata. Agar kalimat tidak terjadi pemisahan kata, maka ditetapkan nilai maksimal untuk kata yang digunakan dalam 1 data yaitu 275 kata. Jika kalimat lebih pendek akan dilakukan *padding* dengan menambahkan token [PAD]. Untuk kata yang tidak ada pada *vocabulary* (*out-of vocabulary*) dilakukan pemisahan menjadi sub kata (*subword*) dengan symbol ##. Sebagai contoh proses penyesuaian input yang dapat diterima IndoBERT disajikan pada tabel berikut.

Tabel 4. Proses *Pre-Train Model*

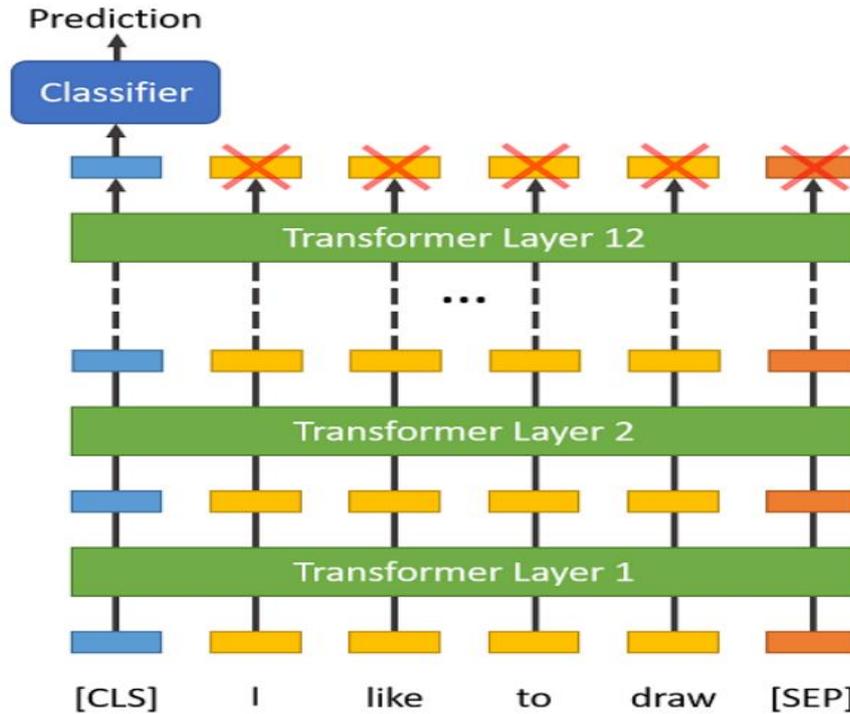
Kalimat	Bagus tuh hape nya bang kerenn sumpah
Hasil tokenisasi	'bag', '##us', 'tu', '##h', 'hap', '##e', 'nya', 'bang', 'keren', '##n', 'sum', '##pah'
Penambahan token khusus	'[CLS]', 'bag', '##us', 'tu', '##h', 'hap', '##e', 'nya', 'bang', 'keren', '##n', 'sum', '##pah', '[SEP]'
Encoding	101, 35126, 10258, 10689, 10243, 55558, 10111, 19483, 12221, 77283, 10115, 25049, 52593, 102
<i>Attention mask</i>	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Hasil dari proses tokenisasi kemudian ditambahkan token khusus yaitu token [CLS] di awal kalimat dan token [SEP] di akhir kalimat. Selanjutnya pada tiap-tiap token dikodekan sesuai dengan indeks *vocabulary*. Token [CLS] memiliki id 101, token 'bag' memiliki id 35126, dan seterusnya.

Attention mask digunakan untuk membedakan nilai token kata dengan nilai *padding*. Token *padding* [PAD] memiliki nilai 0 sedangkan token kata diberi nilai 1. Untuk memudahkan proses tokenisasi dibuat suatu data loader untuk masing-masing *dataset* yaitu data loader untuk *Training*, data loader untuk *Testing*, dan data loader untuk validasi.

b. *Fine-tuning BERT*

Selanjutnya input yang telah disesuaikan kemudian diteruskan ke dalam jaringan BERT yaitu tumpukan 12 *layer Transformers Encoder*. Tiap *layer encoder* memiliki dua sub *layer*, yang pertama adalah *multi-head self-attention mechanism*, dan yang kedua *fully connected feed forward network*. Setelah melewati semua *encoder*, didapatkan vector output dari tiap token. Namun hanya vector output dari token [CLS] yang akan digunakan sebagai vector input untuk *classifier*.



Gambar 4. Ilustrasi *Fine-tuning* IndoBERT untuk Klasifikasi

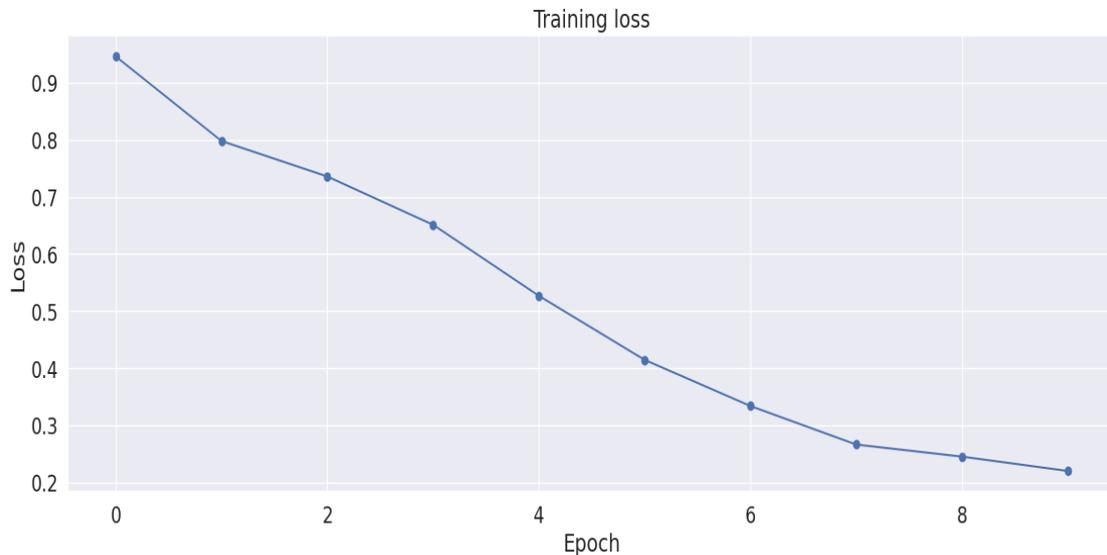
Proses *fine-tuning* IndoBERT untuk klasifikasi dilakukan dengan menambahkan layer klasifikasi. *Library Transformers* memiliki kelas *BertForSequenceClassification* yang didesain untuk tugas klasifikasi. Kelas *BertForSequenceClassification* bekerja dengan cara memasukkan output dari pooler untuk menghitung logits. Nilai logika yang dihasilkan kemudian digunakan untuk mendapatkan nilai prediksi menggunakan *softmax*. Pada penelitian ini, penulis menggunakan *learning rate* sebesar $2e-5$ dengan *optimizer* adam dengan *epsilon* $1e-8$. Berikut hasil *Training* dengan *epoch* 10.

```

===== Epoch 1 / 10 =====
Training...
Average training loss: 0.95
Training epoch took: 0:00:19
Running Validation...
Accuracy: 0.64
Validation took: 0:00:01
===== Epoch 2 / 10 =====
Training...
Average training loss: 0.81
Training epoch took: 0:00:16
Running Validation...
Accuracy: 0.64
Validation took: 0:00:01
===== Epoch 3 / 10 =====
Training...
Average training loss: 0.72
Training epoch took: 0:00:16
Running Validation...
Accuracy: 0.67
Validation took: 0:00:01
===== Epoch 4 / 10 =====
Training...
Average training loss: 0.65
Training epoch took: 0:00:17
Running Validation...
Accuracy: 0.62
Validation took: 0:00:01
===== Epoch 5 / 10 =====
Training...
Average training loss: 0.52
Training epoch took: 0:00:17
Running Validation...
Accuracy: 0.60
Validation took: 0:00:01
===== Epoch 6 / 10 =====
Training...
Average training loss: 0.41
Training epoch took: 0:00:17
Running Validation...
Accuracy: 0.69
Validation took: 0:00:01
===== Epoch 7 / 10 =====
Training...
Average training loss: 0.34
Training epoch took: 0:00:17
Running Validation...
Accuracy: 0.69
Validation took: 0:00:01
===== Epoch 8 / 10 =====
Training...
Average training loss: 0.28
Training epoch took: 0:00:17
Running Validation...
Accuracy: 0.69
Validation took: 0:00:01
    
```

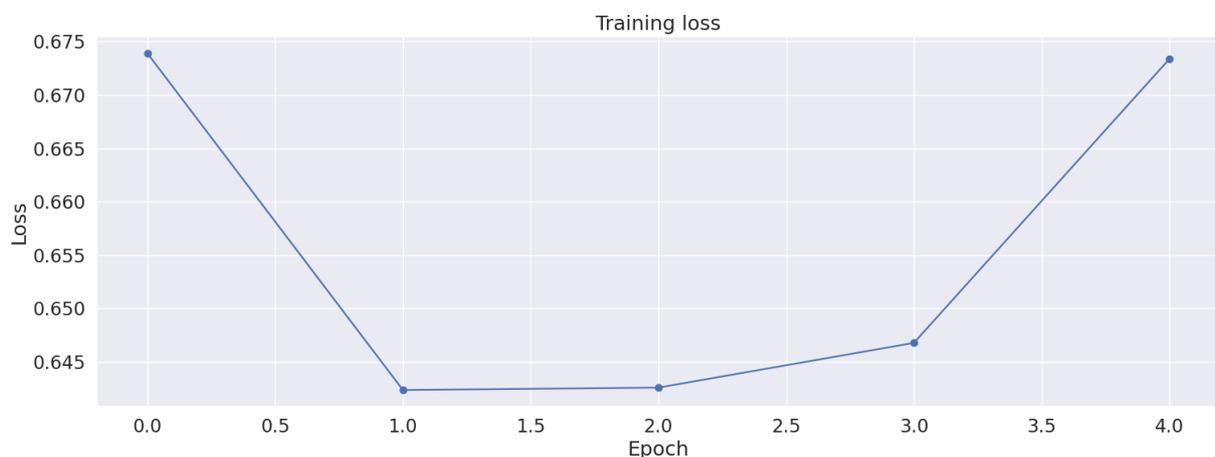
Gambar 5. Hasil *Training* dengan 10 *Epo*c

Pada uji coba BERT, peneliti melakukan *Training* dan evaluasi pada 10 *epoch*, dimana *epoch* ini berfungsi untuk menentukan beberapa kali model dapat melihat *dataset* secara keseluruhan. Rata-rata Waktu eksekusi pada tiap *epoch* yang dibutuhkan adalah 20 detik dan menghasilkan rata-rata akurasi data 0.60 pada tiap *epoch*. Ratarata *Training loss* pada *epoch* pertama cukup tinggi yaitu 0.95. tetapi pada *epoch* kelima nilai ratarata *Training loss* turun menjadi 0.53, dan di *epoch* terakhir mendapat nilai *Training loss* sebesar 0.22 seperti pada gambar 6



Gambar 6. Grafik Training Loss dengan 10 Epoch

Pada pengujian lain, dengan model yang telah dibuat namun dengan perbedaan nilai *epoch*. Pada percobaan menggunakan nilai *epoch* 5 adalah Rata-rata Waktu eksekusi pada tiap *epoch* yang dibutuhkan adalah 20 detik dan menghasilkan rata-rata akurasi data 0.64 pada tiap *epoch*. Ratarata *Training loss* pada *epoch* pertama cukup tinggi yaitu 0.67. tetapi pada *epoch* ketiga nilai ratarata *Training loss* turun menjadi 0.64, dan di *epoch* terakhir mendapat nilai *Training loss* naik sebesar 0.67 seperti pada gambar berikut.



Gambar 7. Grafik Training Loss dengan 5 Epoch

Pada pengujian lain, dengan model yang telah dibuat namun dengan perbedaan nilai *epoch*. Pada percobaan menggunakan nilai *epoch* 15 adalah Rata-rata Waktu eksekusi pada tiap *epoch* yang dibutuhkan adalah 20 detik dan menghasilkan rata-rata akurasi data 0.63 pada tiap *epoch* dengan nilai akurasi paling tinggi yaitu 0,72 pada *epoch* 11 dan nilai paling rendah yaitu 0.44 pada *epoch* 5. Rata-rata *Training loss* pada *epoch* pertama cukup tinggi yaitu 0.91. tetapi pada *epoch* kelima nilai rata-rata *Training loss* turun menjadi 0.56, pada *epoch* kesepuluh nilai rata-rata *Training loss* turun menjadi 0.32 dan di *epoch* terakhir mendapat nilai *Training loss* naik sebesar 0.25. Berikut merupakan grafiknya



Gambar 8. Grafik Training Loss dengan 15 Epoch

a. Evaluasi

Setelah dilakukan proses *Training*, maka selanjutnya dicoba model untuk mengetahui performa terhadap data baru. Hasil percobaan implementasi model terhadap data Testing dengan menggunakan *Confusion Matrix* dengan jumlah *epoch* yang berbeda bisa dilihat dari tabel berikut

Tabel 5. Perbandingan Nilai Akurasi Menurut Jumlah Epoch

Jumlah Epoch	MCC	Akurasi
5	0.292	0.700
10	0.572	0.800
15	0.463	0.740

Berdasarkan hasil implementasi model terhadap data Testing dengan menggunakan *Confusion Matrix*, model dengan *epoch* 10 menjadi model yang paling baik nilai akurasinya yaitu sebesar 0.80. Semakin tinggi nilai akurasi, semakin baik pula model yang dihasilkan. Selain nilai, didapat juga nilai *Matthew Correlation Coefficient* (mcc) sebesar 0.572 . Semakin nilai MCC mendekati +1 maka semakin baik kinerja algoritma klasifikasinya. Sebaliknya, jika nilai nya mendekati -1, maka semakin buruk kinerja algoritma klasifikasi.

```

[ ] from sklearn.metrics import matthews_corrcoef

flat_prediction = [item for sublist in prediction for item in sublist]
flat_prediction = np.argmax(flat_prediction, axis=1).flatten()

flat_true_labels = [item for sublist in true_labels for item in sublist]

mcc = matthews_corrcoef(flat_true_labels, flat_prediction)

print("MCC: %.3f" %mcc)

MCC: 0.572

▶ from sklearn.metrics import accuracy_score

acc = accuracy_score(flat_true_labels, flat_prediction)

print("ACC: %.3f" %acc)

ACC: 0.800
    
```

Gambar 9. Nilai akurasi dan MCC dengan epoch 10

Dari 500 data uji, model mampu mengklasifikasikan 400 data sentimen dengan benar. Oleh karena itu model ini dipilih untuk diterapkan pada data yang tidak berlabel untuk mendapatkan hasil klasifikasi sentimen.

4. SIMPULAN

Berdasarkan penelitian yang dilakukan maka dapat disimpulkan antara lain:

1. Masyarakat Indonesia cenderung memiliki sentimen netral terhadap tayangan “Spek bengis, harga manis - Review POCO X5 Pro 5G Indonesia!” pada Channel “GadgetIn”. Kecenderungan sentimen netral di sebabkan banyak dari penonton berkomentar tentang rekomendasi gadget. Selain itu nilai sentiment positif juga cukup bagus. itu menunjukkan bahwa tayangan “Spek bengis, harga manis - Review POCO X5 Pro 5G Indonesia!” pada Channel “GadgetIn” di sukai oleh penonton.
2. Performansi model BERT dengan pembagian data 80% data *Training*, 10% data validation, dan 10% *data Testing* mendapatkan nilai akurasi sebesar 0.80 dengan jumlah *epoch* 10 mengungguli percobaan yang lain dengan jumlah *epoch* 5 dengan nilai akurasi 0.70 dan percobaan dengan jumlah *epoch* 15 dengan nilai akurasi yaitu sebesar 0.74. Hasil evaluasi menggunakan *confusion matrix* menunjukkan bahwa model BERT Cukup baik untuk melakukan klasifikasi sentiment.

5. DAFTAR PUSTAKA

- Amien, M. (2023). *Sejarah dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia: Tinjauan tentang sejarah, perkembangan teknologi, dan aplikasi NLP dalam bahasa Indonesia*. 2007, 1–7. <http://arxiv.org/abs/2304.02746>
- Ardiani, L., & Sujaini, H. (2020). *Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak* *Implementation of Sentiment Analysis of Community Responses to Development in Pontianak City*. 8(2), 183–190. <https://doi.org/10.26418/justin.v8i2.36776>
- Cahyawijaya, S., Winata, G. I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., Ruder, S., Lim, Z. Y., Bahar, S., Khodra, M. L., Purwarianti, A., & Fung, P. (2021). IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 8875–8898. <https://doi.org/10.18653/v1/2021.emnlp-main.699>
- Ervin Winardo Toepak, I. A. T. A. (2018). *Pembangunan Aplikasi Penyedia Informasi Lowongan Pekerjaan Menggunakan Youtube API Pada Smartphone Android*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(1), 168–174.
- Gelar Guntara, R. (2023). *Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7*. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55–60. <https://doi.org/10.47233/jteksis.v5i1.750>
- Hasdyna, N., & Dinata, R. K. (2020). Analisis Matthew Correlation Coefficient pada K-Nearest Neighbor dalam Klasifikasi Ikan Hias. *INFORMAL: Informatics Journal*, 5(2), 57. <https://doi.org/10.19184/isj.v5i2.18907>
- Kanakaraddi, S. G., Chikaraddi, A. K., Gull, K. C., & Hiremath, P. S. (2020). Comparison Study of *Sentiment Analysis* of Komentars using Various Machine Learning Algorithms. *Proceedings of the 5th International Conference on Inventive Computation Technologies, ICICT 2020*, 287–292. <https://doi.org/10.1109/ICICT48043.2020.9112546>
- Mehta, P., & Pandya, S. (2020). A review on *Sentiment Analysis* methodologies, practices and applications. *International Journal of Scientific and Technology Research*, 9(2), 601–609.
- Muhammad Romzi, & Kurniawan, B. (2020). Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, 03(2), 37–44.
- Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based Deep Intelligent Contextual Embedding for Twitter *Sentiment Analysis*. *Future Generation Computer Systems*, 113, 58–69. <https://doi.org/10.1016/j.future.2020.06.050>
- Negeri, U., Jl, S., Lidah Wetan, L., Wetan, K., Lakarsantri, K., Surabaya, J., & Timur, I. (2023). *Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN)* Nanang Husin. *Jurnal Esensi Infokom*, 7(1), 75.
- Pratama, F. A., & Romadhony, A. (2020). Identifikasi Komentar Toksik Dengan Bert. *EProceedings of Engineering*, 7(2), 7941–7949. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/1>

[3073](#)

- Putra, I. F., & Purwarianti, A. (2020). Improving Indonesian Text Classification Using Multilingual Language Model. *2020 7th International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA 2020*, 0–4. <https://doi.org/10.1109/ICAICTA49861.2020.9429038>
- Rao, D. (2019). NLP with Pytorch. In *Climate Change 2013 - The Physical Science Basis* (Vol. 53, Issue 9). <http://ebooks.cambridge.org/ref/id/CB09781107415324A009>
- Ren, Z., Shen, Q., Diao, X., & Xu, H. (2021). A sentiment-aware deep learning approach for personality detection from text. *Information Processing and Management*, 58(3), 102532. <https://doi.org/10.1016/j.ipm.2021.10253>

