

SENTIMEN ANALISIS KEPUASAN PELAYANAN *CUSTOMER* ISP MYREPUBLIC PADA PLATFORM TWITTER DENGAN METODE *WEB SCRAPING* DAN *BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)*

Alif M S^{1*}, Puput Irfansyah² dan Budi Santoso³

^{1,2,3}Teknik Informatika, Universitas Indraprasta PGRI, Jakarta

Corresponding author: alifmind217@gmail.com

ABSTRAK: Dalam Penelitian ini merancang sistem sentimen analisis menggunakan *Bidirectional Encoder Representations from Transformers (BERT)* sebagai algoritma sentimen analisis dan teknologi *Web Scraping* untuk mengumpulkan data sentimen. Sistem yang telah dirancang menghasilkan sebuah sentimen analisis yang dapat memperoleh opini positif, netral dan negatif berdasarkan *tweets customer provider* MyRepublic. Sistem ini didesain dengan menggunakan Bahasa pemrograman Python, Google Colab untuk merancang sistem dan model *IndoBERT_{BASE}* untuk klasifikasi sentimen yang digunakan.

Kata Kunci: BERT, *scraping*, sentimen, *dataset*

ABSTRACT: *This research designs a sentiment analysis system using Bidirectional Encoder Representations from Transformers (BERT) as a sentiment analysis algorithm and Web Scraping technology to collect sentiment data. The system that has been designed produces a sentiment analysis that can obtain positive, neutral and negative opinions based on customer of provider MyRepublic tweets. This system is designed using the Python programming language, Google Colab to design the system and model pre-trained IndoBERT_{BASE} for sentiment classification used.*

Keyword: BERT, *scraping*, sentiment, *dataset*

PENDAHULUAN

Perkembangan teknologi informasi begitu pesat saat ini, banyak memberikan dampak positif maupun negatif. Hal ini secara tidak langsung, membuat masyarakat menjadi pintar karena mudah dalam mengakses informasi di seluruh dunia tanpa kendala jarak dengan menggunakan teknologi yang disebut Internet. Tetapi berdampak negatif banyak informasi atau content yang tidak baik, berbahaya dan tidak berguna tersebar di Internet, jadi masyarakat perlu menyaring informasi-informasinya dengan bijak.

Ditambah saat ini sudah ada wadah yang bisa digunakan untuk menjadi perpanjangan “lidah manusia” dalam menyampaikan suatu pendapat/ opini atau pandangan yang disebut dengan media sosial. Beberapa contoh media sosial yang sedang populer di Indonesia yaitu Facebook, Twitter, Instagram dsb. Khususnya Twitter, sekarang banyak digunakan oleh generasi muda Indonesia karena layanannya mudah digunakan terutama dalam menyampaikan pesan atau tweets. Penulis pesan tersebut biasanya menulis kehidupan penulis, berbagi opini tentang berbagai topik dan membahas isu-isu yang sedang terjadi. Kemudian yang menjadi kelebihan

Twitter, yaitu tidak membatasi pertemanan yang biasa disebut followers.

Di zaman modern ini, sentimen atau opini masyarakat semakin bertambah luas dan bebas diungkapkan di berbagai media. Sentimen dapat menjadi potensi besar bagi perusahaan yang ingin mengetahui umpan balik (feedback) dari pelanggan terhadap produk maupun layanan yang mereka perdagangkan.

Maka solusi bagi pemanfaatan dari analisis sentimen berikut adalah dibuatkan sebuah aplikasi yang dapat melakukan analisis sentimen dengan cara mengumpulkan tweets, mengklasifikasi tweets, dan melihat kecenderungan sentimen dari tweets yang ditulis oleh pelanggan provider internet MyRepublic mengenai kepuasan produk dan layanan yang diberikan. Selain itu dalam aplikasi akan disediakan visualisasi data dalam bentuk diagram, sehingga pengguna aplikasi dapat lebih mudah melihat dan memahami data.

METODE PENELITIAN

Metode penelitian yang penulis lakukan adalah dengan tahapan berikut:

a. Pengumpulan *Dataset*

Pengambilan *dataset* terjadi melalui proses *scraping* dimana tahapannya yaitu melakukan *request get* data ke Aplikasi Twitter. Proses *scraping* dilakukan secara online melalui internet. Untuk hasil proses *scraping* akan tersimpan dengan format *.csv*. *Dataset* yang sudah diberi label dilanjutkan tahap *pre-processing* seperti berikut:

1. Normalisasi fitur

Proses menghilangkan komponen khas yang tidak berguna seperti tanda baca, URL, tanda *retweet* dan *username* (Ronny Julianto et al., 2017).

2. *Stemming*

Proses untuk mengubah setiap kalimat pada teks untuk memisahkan kata dasar dan imbuhan (Ronny Julianto et al., 2017).

3. *Tokenizing*

Proses untuk mengubah setiap kalimat pada teks menjadi sebuah angka (*token*) yang sesuai dengan kosa kata sudah terdapat pada *pre-trained* model BERT. *Tokenizing* menggunakan *library* BERT yaitu BERT-Tokenizer.

b. Perancangan Model

Pada penelitian ini model yang digunakan adalah model *BERT Classification*.

c. Pengujian Data

Penelitian ini menguji 4 parameter yaitu sebagai berikut:

1. Akurasi

Ukuran kinerja yang paling intuitif dan ini hanyalah rasio observasi yang diprediksi dengan benar terhadap total observasi.

$$\text{Akurasi} = \frac{TP+TN}{TN+FP+FN+TP} \quad (1)$$

2. *Precision*

Rasio observasi positif (TP) yang diprediksi dengan benar terhadap total observasi positif yang diprediksi. Nilai ideal dari parameter *precision* yaitu mendekati 100% dengan melihat False Positive (FP), maka dapat terlihat seberapa besar pengaruh nilai FP terhadap hasil klasifikasi.

$$\text{Precision} = \frac{TP}{TN+FP} \quad (2)$$

3. *Recall*

Rasio observasi positif (TP) yang diprediksi dengan benar terhadap semua observasi di kelas aktual termasuk data FN. Dengan melihat *False Negative* (FN), maka dapat terlihat seberapa besar pengaruh nilai FN terhadap hasil klasifikasi.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

4. *F1-Score*

Rata-rata tertimbang dari *precision* dan *recall*. Nilai terbaik *F1-Score* adalah 1 dan nilai terburuk adalah 0. Nilai *F1-Score* juga menunjukkan distribusi yang merata antara parameter *precision* dan *recall* pada *confusion matrix*.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

d. Evaluasi

Tahap terakhir pada penelitian ini adalah evaluasi. Tahap ini menjelaskan tentang proses komputasi (*Machine Learning*) terhadap model, data, algoritma dan fungsi yang telah didefinisikan sebelumnya. Pada penelitian ini hanya hasil luaran atau *output* yang divisualisasikan. Jika hasil prediksi dari sentimen analisis masih jauh dari yang diharapkan, maka akan dilakukan evaluasi kembali dan mencari kesalahan algoritma serta melakukan *training* ulang (Sari et al., 2019).

		Predicted Class		
		Positive	Neutral	Negative
True Class	Positive	True Positive (TP)	False Neutral (FNt)	False Negative (FN)
	Neutral	False Positive (FP)	True Neutral (TNt)	False Negative (FN)
	Negative	False Positive (FP)	False Neutral (FNt)	True Negative (TN)

Gambar 1. *Confusion Matrix* Pengujian
Sumber: Jurnal (Saputro et al., 2018)

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 1,087 *tweets* untuk *train dataset* dan untuk *test dataset* sebanyak 154 *tweets* berlabel sesuai yang telah tertera pada Tabel 1 Berdasarkan jurnal aslinya (Devlin et al., 2018), BERT dapat di *fine-tuning* dengan menyesuaikan *hyperparameter*-nya.

Tabel 1. Sistematika Pembagian *Dataset*

<i>Clean Dataset</i>	<i>Train Dataset</i>	<i>Validation Dataset</i>	<i>Test Dataset</i>
1,553 <i>Tweets</i>	1087 <i>Tweets</i>	312 <i>Tweets</i>	154 <i>Tweets</i>

Adapun *hyperparameter* untuk *fine-tuning* IndoBERT yang digunakan dalam penelitian ini, yaitu *batch size* 32, *learning rate* (Adam) $3\epsilon-6$ dan *epoch* 5.

```
(Epoch 3) TRAIN LOSS:0.4126 ACC:0.86 F1:0.59 REC:0.61 PRE:0.57 LR:0.00000300
0% | 0/10 [00:00<?, ?it/s]usr/local/lib/python3.10/dist-packages/torch/utlils/data/dataloader
warnings.warn(_create_warning_msg(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.4067 ACC:0.81 F1:0.54 REC:0.56 PRE:0.53: 10% | 1/10 [00:00:00:05, 1.51it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3356 ACC:0.86 F1:0.57 REC:0.59 PRE:0.55: 10% | 1/10 [00:00:00:05, 1.51it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3314 ACC:0.86 F1:0.57 REC:0.58 PRE:0.56: 30% | 3/10 [00:00:00:01, 4.39it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3178 ACC:0.88 F1:0.59 REC:0.60 PRE:0.57: 30% | 3/10 [00:00:00:01, 4.39it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3153 ACC:0.88 F1:0.59 REC:0.61 PRE:0.58: 50% | 5/10 [00:00:00:00, 7.12it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3293 ACC:0.86 F1:0.59 REC:0.61 PRE:0.56: 50% | 5/10 [00:00:00:00, 7.12it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3299 ACC:0.86 F1:0.59 REC:0.61 PRE:0.57: 70% | 7/10 [00:01:00:00, 9.57it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3359 ACC:0.86 F1:0.59 REC:0.62 PRE:0.56: 70% | 7/10 [00:01:00:00, 9.57it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3399 ACC:0.86 F1:0.59 REC:0.61 PRE:0.56: 90% | 9/10 [00:01:00:00, 10.55it/s]usr/
_warn_prf(average, modifier, msg_start, len(result))
VALID LOSS:0.3404 ACC:0.86 F1:0.58 REC:0.61 PRE:0.56: 100% | 10/10 [00:01:00:00, 5.48it/s]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
_warn_prf(average, modifier, msg_start, len(result))
(Epoch 3) VALID LOSS:0.3404 ACC:0.85 F1:0.58 REC:0.61 PRE:0.56
0% | 0/34 [00:00<?, ?it/s]usr/local/lib/python3.10/dist-packages/torch/utlils/data/dataloader
warnings.warn(_create_warning_msg(
(Epoch 4) TRAIN LOSS:0.3128 LR:0.00000300: 100% | 34/34 [00:00:00:00, 4.03it/s]
(Epoch 4) VALID LOSS:0.3128 ACC:0.90 F1:0.81 REC:0.78 PRE:0.93
0% | 0/10 [00:00<?, ?it/s]usr/local/lib/python3.10/dist-packages/torch/utlils/data/dataloader
warnings.warn(_create_warning_msg(
VALID LOSS:0.2649 ACC:0.92 F1:0.81 REC:0.78 PRE:0.93: 100% | 10/10 [00:01:00:00, 6.33it/s]
(Epoch 4) VALID LOSS:0.2649 ACC:0.92 F1:0.81 REC:0.78 PRE:0.93
0% | 0/34 [00:00<?, ?it/s]usr/local/lib/python3.10/dist-packages/torch/utlils/data/dataloader
warnings.warn(_create_warning_msg(
(Epoch 5) TRAIN LOSS:0.2416 LR:0.00000300: 100% | 34/34 [00:07:00:00, 4.25it/s]
(Epoch 5) VALID LOSS:0.2416 ACC:0.93 F1:0.85 REC:0.81 PRE:0.93 LR:0.00000300
0% | 0/10 [00:00<?, ?it/s]usr/local/lib/python3.10/dist-packages/torch/utlils/data/dataloader
warnings.warn(_create_warning_msg(
VALID LOSS:0.2242 ACC:0.93 F1:0.89 REC:0.86 PRE:0.94: 100% | 10/10 [00:01:00:00, 6.30it/s]
(Epoch 5) VALID LOSS:0.2242 ACC:0.93 F1:0.89 REC:0.86 PRE:0.94
```

Gambar 2. Proses Training dan Evaluasi

Setelah melalui proses perulangan per-epoch pada model, hasil training disimpan. Gambar 2 merupakan hasil perbandingan antara akurasi yang didapat saat training dan validasi.

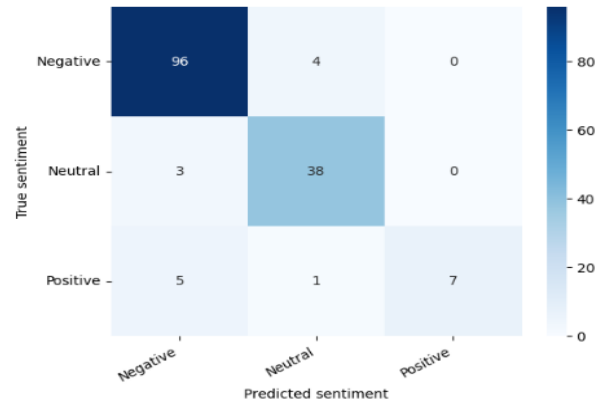
Tabel 2. Hasil Training dan Evaluasi Dataset

Epoch	Akurasi	F1-Score	Recall	Presisi
1	81%	54%	55%	54%
2	84%	57%	59%	55%
3	86%	58%	61%	56%
4	92%	81%	78%	93%
5	93%	89%	86%	94%

Berdasarkan hasil dari pengujian sistem yang telah dijalankan, nilai akurasi terbaik yang didapat yaitu sebesar 93% dengan learning rate 0.000003. Hasil dari pengujian ditampilkan dengan menggunakan Confusion Matrix untuk mengetahui hasil dari hyperparameter yang diujikan serta mengetahui nilai True Positive, True Negative, True Neutral, False Positive, False Neutral dan False Negative.

	precision	recall	f1-score	support
Negative	0.92	0.96	0.94	100
Neutral	0.88	0.93	0.90	41
Positive	1.00	0.54	0.70	13
accuracy			0.92	154
macro avg	0.94	0.81	0.85	154
weighted avg	0.92	0.92	0.91	154

Gambar 3. Classification Report



Gambar 4. Confusion Matrix Dataset

KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada sistem sentimen analisis dengan metode BERT, maka dapat diambil kesimpulan sebagai berikut:

- Analisis sentimen berhasil dilakukan menggunakan pre-trained model IndoBERT_{BASE} dengan teknik fine-tuning.
- Model yang digunakan menghasilkan akurasi sebesar 93% dengan pemilihan hyperparameter yaitu batch size 32, learning rate (Adam) 3e-6 dan epoch 5.
- Model mengalami sedikit kesulitan dalam mengklasifikasikan sentimen positif, hal ini disebabkan karena jumlah dataset positif yang tergolong sedikit dibandingkan dengan jumlah dataset netral dan dataset negatif. Terdapat inkonsistensi dalam proses labelisasi pada dataset yang bersifat ambigu dan analisis dilakukan dengan menggunakan model yang menerapkan arsitektur BERT_{BASE} yaitu versi terkecil dari BERT.

DAFTAR PUSTAKA

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Ronny Julianto, Evi Dianti Bintari, & Indrianti. (2017). Analisis Sentimen Layanan Provider Telepon Seluler pada Twitter menggunakan Metode Naïve Bayesian Classification. Journal of Big Data Analytic and Artificial Intelligence, 3(1), 23–30.

Saputro, A. E. S., Notodiputro, K. A., & A, I. (2018). Study of Sentiment of Governor’s Election Opinion in 2018. International Journal of Scientific Research in Science, Engineering and Technology, 231–238. <https://doi.org/10.32628/ijrsrset21841124>

Sari, R., Nusa, S., & Jakarta, M. (2019). Analisis Sentimen Review Restoran menggunakan Algoritma Naive Bayes berbasis Particle Swarm Optimization. JURNAL INFORMATIKA, 6(1), 23–28. <http://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/4695>