Analisis Klasifikasi Sentimen Terhadap Isu Kebocoran Data Kartu Identitas Ponsel di Twitter

http://dx.doi.org/10.28932/jutisi.v8i3.5483

Riwayat Artikel
Received: 30 September 2022 | *Final Revision*: 05 Desember 2022 | *Accepted*: 05 Desember 2022

Muh Ichlasul Amal ✉ #1, Elsa Syafira Rahmasita#2, Edward Suryaputra #3, Nur Aini Rakhmawati#4

Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Kota Surabaya, Jawa Timur 60111, Indonesia
[1]muh.ichlasul.19052@student.its.ac.id
[2]elsarahmasita.19052@student.its.ac.id
[3]edwardsuryaputra.19052@student.its.ac.id
[4]nur.aini@is.its.ac.id
✉Corresponding author: muh.ichlasul.19052@student.its.ac.id

Abstrak — **Perkembangan teknologi dan internet membawa ancaman besar terkait dengan privasi dan keamanan data pribadi. Pada bulan September 2022, terdapat insiden bocornya 1,3 miliar data pendaftaran kartu identitas ponsel atau kartu SIM yang berisi data pribadi pengguna di situs gelap. Twitter sebagai salah satu media sosial terpopuler di Indonesia menjadi tempat masyarakat Indonesia menyuarakan opininya terkait isu kebocoran data tersebut. Penelitian ini bertujuan untuk mencari tahu sebaran kata dan analisis klasifikasi sentimen dari opini masyarakat di Twitter terkait dengan isu tersebut. Analisis klasifikasi sentimen dilakukan menggunakan pendekatan *machine learning* dengan empat metode, yaitu *Random Forest, Logistic Regression, Support-Vector Machine,* dan model IndoBERT. Keempat metode tersebut akan dibandingkan untuk melihat model mana yang menghasilkan performa terbaik dalam mendeteksi sentimen. Dari proses *crawling*, didapatkan 957 *tweet,* di mana 609 *tweet* diberi label dan akan dilatih menggunakan empat metode tersebut. Dari data yang didapatkan, terdapat ketidakseimbangan antar kelas, di mana sentimen positif memiliki jumlah yang jauh lebih sedikit dibandingkan sentimen negatif dan netral. Beberapa kata yang sering digunakan dalam data *tweet* yang diambil adalah *sim card, data sim, bocor data, miliar data,* dan *kominfo*. Hasil pembangunan model menunjukkan algoritma *Support-Vector Machine* memiliki performa terbaik dengan nilai *f1-score* 0.81, dilanjutkan dengan *Random Forest* sebesar 0.78, IndoBERT sebesar 0.76, dan *Logistic Regression* sebesar 0.74. Ketidakseimbangan kelas dan kurangnya data latih membuat performa IndoBERT sebagai salah satu *state-of-the-art* dalam NLP memiliki performa yang rendah dibandingkan algoritma lainnya. Hasil dari penelitian ini dapat digunakan pihak berwenang untuk mengevaluasi kebijakan dalam menangani isu keamanan data dengan mendengarkan opini dari masyarakat Indonesia.**

Kata kunci— **IndoBERT; Kebocoran Data Kartu SIM; *Logistic Regression*; *Random Forest*; *Support-Vector Machine*.**

Sentiment Classification Analysis On Phone Identity Card Data Leaks Issues On Twitter

Abstract — ***Technology developments bring great threats related to privacy and security of personal data. In September 2022, a data leak incident of 1.3 billion SIM card registration data containing user's personal data was uploaded on dark web. Indonesian people voice their opinion regarding this issue on Twitter. This study aims to find out the word distribution and sentiment classification analysis of public opinion on Twitter related to the issue. Sentiment classification analysis was carried out using a machine learning approach with four methods, namely Random Forest, Logistic Regression, Support-Vector Machine, and IndoBERT model. The four methods***

will be compared to see which model produces the best performance. From the crawling process, 957 tweets were obtained, of which 609 were labeled and trained using the four methods. From the data obtained, there is an imbalance between classes, where positive sentiment has a much smaller number than the rest. Some words that are often used in the tweet are SIM card, data SIM, bocor data, miliar data, and kominfo. The results of the model show that the Support-Vector Machine has the best performance with an f1-score of 0.81, followed by Random Forest of 0.78, IndoBERT of 0.76, and Logistic Regression of 0.74. Class imbalance and lack of training data make IndoBERT's performance lower when compared to other algorithms. The results of this study can be used by the authorities to evaluate policies in dealing with data security issues by listening to opinions from the Indonesian people.

Keywords— ***IndoBERT; Logistic Regression; Random Forest; SIM Card Data Leak, Support-Vector Machine.***

I. PENDAHULUAN

Kemajuan teknologi merevolusi bagaimana manusia saling berinteraksi satu sama lain. Aplikasi komunikasi jarak jauh seperti WhatsApp, Instagram, dan Twitter sudah menjadi bagian dari keseharian masyarakat Indonesia. Laporan terbaru mengungkap bahwa jumlah pengguna Twitter di Indonesia pada tahun 2022 mencapai 18,45 Juta. Jumlah ini naik 31,3% dari tahun sebelumnya sebanyak 14,05 Juta [1].Walaupun perkembangan teknologi dan internet menjadi aspek yang penting dalam kehidupan kita, teknologi juga membawa ancaman yang lebih tinggi perihal privasi dan keamanan [2]. Kelalaian dalam mengamankan data-data pribadi, khususnya yang bersifat sensitif dapat menimbulkan berbagai masalah utamanya terkait penyalahgunaan identitas. Peraturan Pemerintah No 71 Tahun 2019 tentang Penyelenggaraan Sistem dan Transaksi Elektronik menjelaskan peran Pemerintah dalam melindungi kepentingan umum dari segala jenis gangguan sebagai akibat penyalahgunaan Informasi Elektronik dan Transaksi Elektronik yang mengganggu ketertiban umum, sesuai dengan ketentuan peraturan perundang-undangan [3].

Salah satu insiden kebocoran data yang terjadi pada tahun 2022 adalah isu bocornya 1,3 miliar data pendaftaran kartu identitas ponsel atau kartu SIM yang dijual di situs gelap (*dark web*) pada Kamis, 1 September 2022. Di situs gelap itu, pelaku mengaku memiliki data NIK, nomor telepon, nama penyedia (*provider*), dan tanggal pendaftaran. Penjual mengatakan bahwa data ini diambil dari Kominfo RI. Bjorka, nama samara pelaku menjual data tersebut dengan harga US$50 ribu (Rp743,5 juta) [4]. Isu ini tentu menuai banyak kritik dari ahli dan masyarakat mengingat pemerintah dan penyedia layanan kartu SIM harus bertanggung jawab atas perlindungan data sensitif yang diberikan. Berbagai respons dari banyak pihak bermunculan melalui *tweet* yang ada di media sosial Twitter.

Twitter sebagai media sosial dengan pengguna aktif cukup banyak menjadi salah satu opsi utama yang digunakan masyarakat untuk menyampaikan gagasan dan pikiran mereka terhadap isu-isu yang ada, termasuk isu kebocoran data kartu SIM. Dengan alat, tujuan, dan topik yang sesuai, media sosial memiliki peran penting dalam konteks pemerintahan, khususnya untuk menggali inovasi dalam mengembangkan layanan pemerintahan berbasis elektronik [5]. Sehingga penelitian ini berfokus untuk melakukan analisis sentimen terhadap *tweet* masyarakat mengenai isu kebocoran data kartu SIM.

Penelitian ini mengambil respons masyarakat melalui Twitter terkait dengan isu kebocoran data kartu SIM pada awal September 2022. Respons-respons tersebut akan digunakan sebagai data masukan untuk membangun sebuah model *machine learning* klasifikasi sentimen, Model tersebut diharapkan dapat dengan tepat mengklasifikasikan apakah sebuah *tweet* memiliki sentimen negatif, netral, atau positif dalam konteks isu kebocoran data kartu SIM. Metode yang digunakan untuk membangun model tersebut adalah *Random Forrest* [6], [7], *Logistic Regression* [8], *Support-Vector Machine* [9], dan IndoBERT [10], [11].

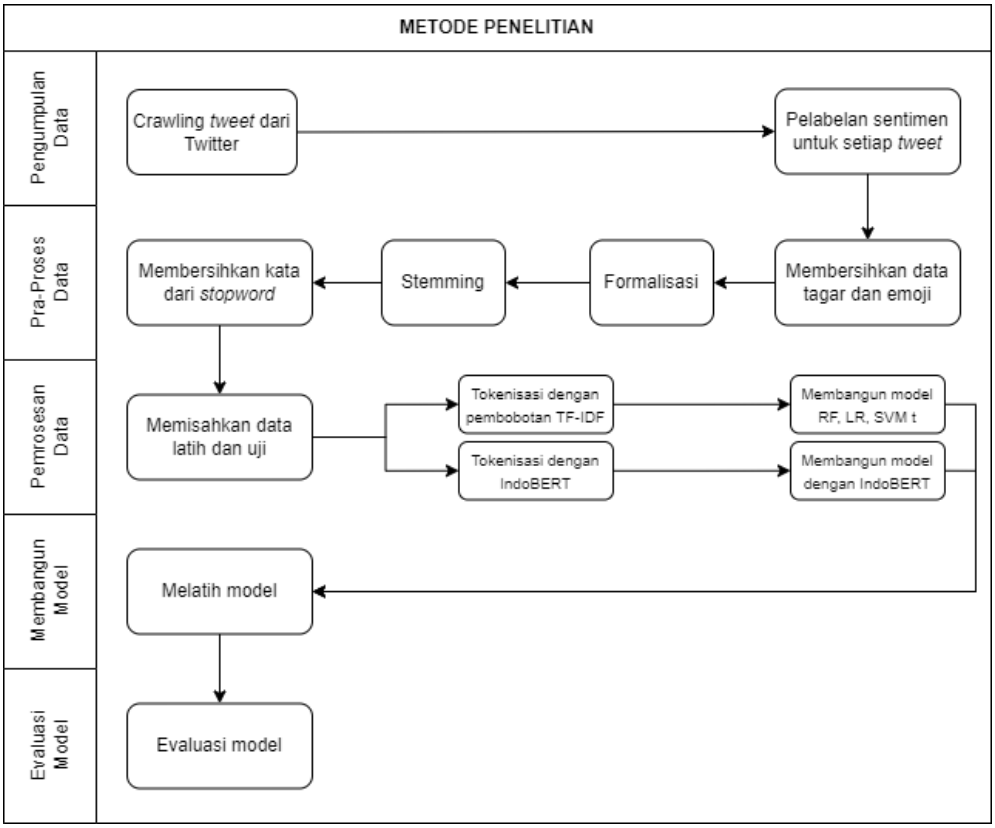
Terdapat beberapa penelitian sebelumnya yang melakukan analisis sentimen dari sumber data Twitter. Wibowo *et al* [12] melakukan analisis sentimen dalam kasus kebocoran data Tokopedia pada bulan Mei 2020 menggunakan algoritma *Random Forest, Logistic Regression*, dan *Support-Vector Machine*. Penelitian [12] membandingkan ketiga algoritma tersebut dengan melihat performa *recall, precision, F1 score,* dan melihat *confusion matrix.* Ditemukan bahwa *Support-Vector Machine* memiliki performa terbaik dengan *F1 score* sebesar 0,503583, walaupun tidak ada dari ketiga *classifier* tersebut yang mampu menebak *tweet* dengan sentimen positif [12]. Penelitian [13] melakukan analisis sentimen terhadap pengaruh akun bot mengenai sentimen masyarakat terkait pinjaman *online* di Twitter. Penelitian tersebut menggunakan model *pre-trained IndoBert* dengan hasil performa *F1 Score* dan *recall* sebesar 0,59. Dalam mengerjakan tugas analisis sentimen, model IndoBERT memiliki performa *F1 Score* terbaik dibandingkan beberapa model lain, termasuk *Naive Bayes, Logistic Regression, BiLSTM* dan lainnya [10].

Penelitian ini dilakukan untuk menjawab permasalahan bagaimana sebaran kata yang digunakan masyarakat dalam mengomentari kasus kebocoran data kartu SIM pada media sosial Twitter dan algoritma *machine learning* apa yang memiliki performa terbaik dalam menganalisis sentimen dari isu tersebut. Tujuan penelitian ini adalah untuk mendapatkan model hasil pelatihan terbaik menggunakan beberapa pendekatan algoritma dalam pemrosesan bahasa alami dan melakukan klasifikasi terhadap sentimen masyarakat mengenai isu kebocoran data kartu SIM. Sentimen yang di analisis pada penelitian ini terbatas pada sentimen negatif, netral, dan positif. Pelabelan data latih (*training data*) akan dilakukan secara manual. Hasil analisis

sentimen ini diharapkan dapat membantu pihak berwenang khususnya pihak Pemerintah dan Penyelenggara Sistem Elektronik di Indonesia untuk mengevaluasi kebijakan perlindungan data pribadi dengan mendengar tanggapan masyarakat.

II. METODE PENELITIAN

Dalam melakukan analisis sentimen dengan hasil yang optimal, terdapat beberapa tahapan yang harus dilakukan untuk mendapatkan data, membangun model, dan mengambil kesimpulan dari data set. Secara garis besar terdapat lima tahapan yang dilakukan, yaitu : Pengumpulan Data, Pra-pemrosesan data, Pemrosesan Data, Membangun Model dan Evaluasi dengan *metric* seperti yang tertera pada Gambar 1.



Gambar 1 Diagram Alur Metode Penelitian

A. Pengumpulan Data

Pada tahap ini dilakukan pengambilan data dari Twitter yang akan menjadi data masukan dalam pembangunan model analisis sentimen. Tahap pengumpulan data dilakukan menggunakan *library* Python yaitu Twitter Intelligence Tool (TWINT). TWINT merupakan alat penggalian Twitter berbasis Python untuk mendapatkan *tweet* dari pengguna Twitter tanpa menggunakan Twitter API [14]. Data yang digunakan pada tahap ini ditentukan oleh beberapa kata kunci yaitu *kebocoran data simcard, data simcard, kominfo bocor data SIM* dan *data bocor* dalam periode 1 September 2022 hingga 8 September 2022. Setelah melakukan penggalian, *tweet* diagregasi menjadi sebuah *file* Microsoft Excel untuk memudahkan filter dan pelabelan. Total *tweet* yang berhasil digali sebanyak 1414 *tweet*. Setelah dilihat lebih detail, terdapat beberapa *tweet* yang merupakan duplikat dari *tweet* yang lain, sehingga dilakukan filter *remove duplicate* dalam Microsoft Excel untuk menghilangkan duplikasi *tweet* tersebut. Total *tweet* akhir setelah dilakukan penyaringan adalah 1008 *tweet*.

Setelah penyaringan duplikasi dilakukan, tahapan selanjutnya adalah untuk memberikan label secara manual dari *tweet* yang ada. Pelabelan dilakukan dengan mengkategorikan sebuah *tweet* sebagai salah satu dari sentimen negatif, sentimen netral, dan sentimen positif. Jika sebuah *tweet* dianggap tidak sesuai dengan konteks isu kebocoran data kartu SIM, maka *tweet* akan diberi label tidak relevan, dan akan disaring sebelum pembangunan model. *Tweet* dengan sentimen negatif biasanya mengandung konotasi negatif dan menggunakan kata-kata yang bersifat keluhan, mengumpat, mencela, maupun kecewa terhadap suatu pihak terkait kebocoran data kartu SIM. *Tweet* dengan sentimen netral biasanya berupa *tweet*

informatif atau yang menyampaikan berita terkait dengan isu kebocoran data kartu SIM. *Tweet* dengan sentimen positif mengandung konotasi positif atau optimis terhadap isu kebocoran data kartu SIM, termasuk juga solusi dalam mengatasi isu tersebut.

Adapun contoh *tweet* untuk masing-masing label bisa dilihat pada Tabel 1.

TABEL 1
CONTOH DATA *TWEET* DENGAN LABEL

NO	TWEET	LABEL
1	"@detikcom Mau kata apalagi untuk menggambarkan menteri yg ga bisa kerja ...SIM card di suruh aktif pake no nik ktp, bocor data nya tpie ga mau tanggung jawab...klo di luar negeri sdh mundur"	NEGATIF
2	"@hyang_wisnu @kemkominfo @PlateJohnny @BSSN_RI langkah tepat yang diambil @kemkominfo dalam menangani kasus kebocoran data SIM Card."	POSITIF
3	"1,3 Miliar Data Registrasi SIM Prabayar Diduga Bocor, Kominfo Bantah Kecolongan https://t.co/B8T3sp8Tnl"	NETRAL
4	"@diary_gojek Mohon maaf atas kendala yg Bapak/Ibu pelanggan alami, kami sarankan restart handphone dan SIM card lepas lalu pasang kembali dan pastikan SIM card yg terhubung dengan m-BCA berada pada slot 1, gunakan akses internet paket data dari SIM card m-BCA dgn kondisi (1/2) ^Bram"	TIDAK RELEVAN

Anotasi dilakukan oleh tiga orang dengan kesepakatan sebelumnya terkait dengan ciri-ciri tiap kategori. Setelah melakukan pelabelan dan menghilangkan *tweet* yang tidak relevan dengan kasus kebocoran data kartu SIM, didapatkan total *tweet* akhir sebanyak 957. Dari 957 *tweet*, 609 *tweet* akan diberikan label sesuai dengan tiga kategori tersebut. *Tweet* dengan label akan menjadi data masukan untuk membangun model, sedangkan *tweet* yang tidak diberi label akan menjadi *test set* untuk diujikan dari model yang dibuat. Total *tweet* pada setiap label dapat dilihat pada Tabel 2.

TABEL 2
TOTAL TWEET TIAP LABEL

NO	LABEL TWEET	TOTAL TWEET
1	NEGATIF	242
2	NETRAL	324
3	POSITIF	43

B. Pra-Proses Data

Setelah data awal sudah didapat, masing-masing *tweet* akan dibersihkan dari *username*, tagar, alamat tautan dan karakter lainnya. Pembersihan data merupakan tahapan yang penting dalam masalah *text mining* dikarenakan teks yang akan dianalisis biasanya penuh dengan *noise* yang bisa mempengaruhi kinerja analisis [15]. Untuk melakukan pembersihan data, terdapat beberapa langkah yang dilakukan sebagai berikut.

1) *Initial Filtering*: Pembersihan *noise* pada *tweet* yang pertama dilakukan adalah menghilangkan karakter-karakter yang tidak dibutuhkan, seperti *username*, *hashtag*, *tautan*, maupun tanda baca. *Initial Filtering* dilakukan menggunakan *regular expression*. Dilakukan pula *case folding*, yaitu mengubah semua kata yang ada dalam data menjadi huruf kecil.

```
1. #remove unnecessary text (links, etc)
2. import re
3. def remove_unused_char(texts) :
     4. data = texts.map(lambda x:x.lower())
     5. data = data.map(lambda x: re.sub(r'[^a-zA-Z0-
        9 ]', r'', str(x))) # Remove unused character
     6. data = data.map(lambda x: re.sub('[!”#$%&’()*+,-
        ./:;<=>?@[\]^_`{|}~]', '', str(x))) # Remove punctuation
     7. data = data.map(lambda x: re.sub('[0123456789]', '', str(x))) #Remove number
     8. data = data.map(lambda x: x.lstrip())
     9. data = data.map(lambda x: re.sub(r'@\S+', '', x)) # Remove mention
     10. data = data.map(lambda x: re.sub(r'#\S+', '', x)) # Remove hashtag
     11. data = data.map(lambda x: re.sub(r'https\S+', '', x)) # Remove URL
     12. return data
```

Kode Program 1 *Intial Filtering* menggunakan Regex

Kode program 1 menunjukkan baris kode untuk *initial filtering*. Baris 2 menunjukkan *import library regex* yaitu *re.* Baris 3 merupakan fungsi yang akan digunakan untuk menghilangkan karakter tidak terpakai. Baris 4 mengubah *tweet* menjadi huruf kecil. Baris 5 sampai 8 menghilangkan karakter yang tidak dipakai termasuk angka, tanda baca, dan lainnya. Baris 9 hingga 11 menghilangkan *mention*, *hashtag*, dan URL yang biasanya ada dalam sebuah *tweet* karena tidak memberikan wawasan terkait sentimen pada *tweet.*

2) *Formalization:* Data yang diambil merupakan *tweet* berbahasa Indonesia. Penggunaan bahasa Indonesia dalam kehidupan sehari-hari biasanya penuh dengan istilah logat populer (*slang*) *[16]*. *Tweet* yang diambil dari Twitter juga penuh dengan penggunaan logat populer dan juga bahasa yang tidak formal. Agar model klasifikasi bisa dibangun dengan baik, perlu memformalkan dari kata-kata tidak baku menjadi kata bahasa Indonesia yang baku. Membuat formal kata tidak baku dilakukan menggunakan *dictionary* yang tersedia pada repositori GitHub louisowen6/NLP_bahasa_resouces *[17]*. Selain memformalkan kata tidak baku, diformalkan akronim juga dengan menggunakan *dictionary* pada repositori GitHub ramaprakoso/analisis-sentimen *[18]*.

3) *Stemming: Stemming* merupakan proses untuk mengembalikan akar kata dari sebuah kata atau frasa *[19]*. Misal, kata ‘menyukai’ dan ‘disukai’ akan dikembalikan ke kata dasarnya yaitu suka. Proses *stemming* dilakukan menggunakan *library* Python Sastrawi *[20]*.

```
1. # import StemmerFactory class
2. from Sastrawi.Stemmer.StemmerFactory import StemmerFactory


3. # create stemmer
4. factory = StemmerFactory()
5. stemmer = factory.create_stemmer()
6. unclean['stemmed']= unclean['tweet_formalakron'].apply(stemmer.stem)
```

Kode Program 2 *Stemming* Menggunakan Sastrawi

Kode program 2 menunjukkan baris kode untuk melakukan *stemming*. Baris 2 menunjukkan *import library stemming* menggunakan Sastrawi. Baris 4 dan 5 akan menginisiasi sebuah *instance stemming* dengan memanggil *function* StemmerFactory() dan create_Stemmer(). Baris 6 akan melakukan *stemming* pada *dataframe unclean* dengan fungsi apply(stemmer.stem).

4) *Remove Stopwords: Stopwords* merupakan kosa kata bahasa Indonesia yang sering digunakan sebagai kata penghubung dan bukan kata unik dari sebuah data *[21]*. *Stopwords* biasanya merupakan kata tidak penting yang tidak memberikan makna tambahan dalam sebuah konteks dan tidak memberikan kontribusi pada sentimen. Penghilangan *stopword* merupakan salah satu langkah yang penting untuk meningkatkan akurasi dan performa algoritma klasifikasi *[22]*. *Stopwords* bahasa Indonesia juga didapatkan dari *library* Sastrawi *[20]*.

```
1. from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
2. factory = StopWordRemoverFactory()
3. stopword = factory.create_stop_word_remover()


4. unclean['stop_word_removed'] = unclean['stemmed'].apply(stopword.remove)
5. unclean['stop_word_removed'].sample(5, random_state=123)
```

Kode Program 3 Penghapusan *Stopwords* dengan Sastrawi

Kode program 3 menunjukkan baris kode untuk menghilangkan kosa kata yang tidak dibutuhkan. Baris 1 akan memanggil *library stopwords* dari Sastrawi. Baris 2 dan 3 akan menginisiasi *instance stopwords remover* pada variabel *factory* dan *stopword*. Baris 4 dan 5 akan melakukan *stopwords* pada *dataframe unclean* dan menampilkan 5 contoh data yang telah dibersihkan.

5) Word Cloud: Word cloud merupakan visualisasi kata-kata dengan frekuensi kemunculan terbanyak dari sebuah teks. *Word cloud* bisa menjadi sebuah ringkasan dari teks mengenai sebuah topik tertentu. Visualisasi dari *word cloud* biasanya dilakukan dengan mengkorelasikan besar *font* sebuah kata dengan frekuensi kemunculan kata tersebut. *Word cloud* bisa menjadi awal untuk melakukan analisis lebih dalam mengenai suatu topik *[23]*. Pada penelitian ini, *word cloud* dibangun dengan *library* wordcloud di Python.

C. Pemrosesan Data

Pada tahap proses data, data *tweet* dilakukan proses tokenisasi. Tokenisasi merupakan proses normalisasi data tekstual dari sebuah kalimat yang akan dipecah menjadi kata [21]. Tokenisasi dilakukan untuk memudahkan model memberikan makna dari kata-kata yang ada dalam data latih. Tokenisasi akan mengubah kalimat menjadi sebuah token yang dapat berupa kata maupun karakter. Proses tokenisasi dilakukan dengan metode TF-IDF dan metode BERT.

1) *TF-IDF:* Metode TF-IDF merupakan metode pembobotan kata yang memberikan bobot yang berbeda pada setiap istilah dalam dokumen berdasarkan dengan frekuensi istilah per dokumen dan frekuensi istilah dalam semua dokumen *[24]*. Hasil tokenisasi dengan metode TF – IDF akan digunakan untuk pemodelan dengan algoritma *Random Forest, Logistic Regression, Support-Vector Machine*.

2) *Bert Tokenization and Encoding:* Untuk membangun model dengan model *pre-trained IndoBERTt*, maka diperlukan teknik tokenisasi khusus. BERT memiliki token khusus yaitu [CLS] untuk menentukan kategori klasifikasi dari kalimat yang menjadi masukan, dan [SEP] untuk menandai akhir dari kalimat. Dibutuhkan pula token [PAD] atau *padding* agar panjang kalimat dari semua data bisa sesuai. Setiap token akan diubah sesuai dengan id nya masing-masing *[25]*.

Pemrosesan data dilanjutkan dengan tahap memisahkan data latih dengan data tes. Data set yang telah diberikan label akan dipisahkan dengan komposisi 80:20 menjadi data latih dengan data tes. Perbandingan yang dimaksud ialah 80% dari data set akan dipahami oleh mesin dan 20% dari data set lainnya digunakan untuk memprediksi data menggunakan model yang sudah dipahami oleh mesin sebelumnya.

D. Membangun Model

Setelah pemrosesan data, tahap selanjutnya adalah pembangunan model *machine learning*. Algoritma *machine learning* akan melatih data latih untuk memahami fitur-fitur yang ada dalam tiap token, agar mesin paham terhadap fitur-fitur dalam setiap sentimen. Model *machine learning* yang digunakan adalah model *supervised learning* karena tiap token sudah memiliki label masing-masing [26]. Terdapat empat macam algoritma yang akan dilakukan, yaitu : *Random Forest* (RF), *Logistic Regression* (LR), *Suport-Vector Machine* (SVM), dan IndoBERT..

1) *Random Forest: Random Forest* merupakan salah satu tipe dari algoritma *supervised machine learning* berdasarkan *ensemble learning*. *Ensemble learning* merupakan tipe pembelajaran di mana beberapa tipe algoritma digabung untuk membuat model prediksi yang lebih baik. *Random Forest* menggabungkan beberapa algoritma dengan tipe yang sama, seperti *multiple decision tree*, sehingga dinamai *Random Forest*. *Random Forest* bisa digunakan untuk tugas klasifikasi dan regresi *[6], [7]*.

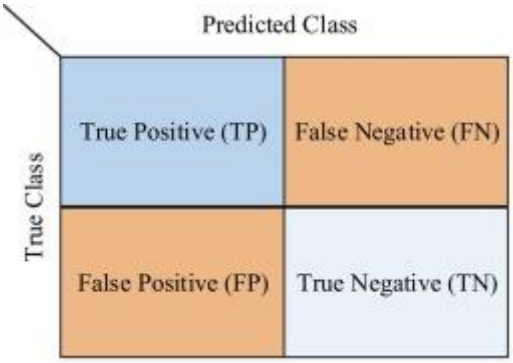
2) *Logistic Regression: Logistic Regression* merupakan metode statistik yang mirip dengan *Linear Regression* karena metode ini menemukan persamaan bersifat logistik yang bertujuan untuk analisis prediktif atau kategorial yang digunakan untuk data biner, misalnya kelangsungan hidup atau kematian. Untuk mengubah nilai Y bervariasi dalam rentang 0 hingga 1, menggunakan transformasi logis *[8]*.

3) *Support Vector Machine: Support Vector Machine* merupakan metode *machine learning* yang sering digunakan untuk analisis *neuroimaging*. Karena kesederhanaan dan fleksibilitasnya yang relatif untuk mengatasi berbagai masalah klasifikasi, SVM secara khusus memberikan kinerja prediksi yang seimbang, bahkan dalam studi di mana ukuran sampel mungkin terbatas *[9]*.

E. Evaluasi Model

Hasil dari algoritma pembelajaran perlu dinilai dan dianalisis dengan benar, sehingga dapat mengevaluasi performa algoritma pembelajaran yang berbeda. Performa klasifikasi diwakili oleh nilai skalar dalam metrik yang berbeda seperti *accuracy*, *sensitivity*, dan *specifity*. Beberapa ukuran yang diturunkan dari *confusion matrix* yaitu *precision, Recall, F1-score, ROC, Informedness, Markedness* dan metode penilaian *Correlation* [27].

Confusion Matrix adalah tabel yang berfungsi untuk mendefinisikan performa sebuah algoritma klasifikasi [28]. Matriks pada *confusion matrix* mewakili nilai TP yang terklasifikasi benar, nilai FP di kelas yang relevan saat seharusnya berada di kelas lain, dan nilai FN di kelas lain saat seharusnya berada di kelas yang relevan dan nilai TN yang diklasifikasikan dengan benar di kelas lain [29].



Gambar 2 *Confusional Matrix*

Gambar 2 menunjukkan metrik kinerja suatu algoritma, yaitu *accuracy*, *precision*, *recall*, dan *F1 score*, yang dihitung berdasarkan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [28].

Accuracy sebuah algoritma dilihat dari perbandingan data yang telah terklasifikasi (TP + TN) dengan total data (TP + TN + FP + FN). Rumus dari metrik *accuracy* ditunjukkan pada Rumus 1.

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

Precision adalah sebuah algoritma dilihat dari perbandingan data yang terklasifikasi dengan benar (TP) dengan total data yang telah terprediksi benar (TP + FP). Rumus dari metrik *precision* ditunjukkan pada Rumus 2.

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

Recall didefinisikan sebagai perbandingan antara data yang terklasifikasi dengan benar (TP) dengan total data yang sebenarnya benar (TP + FN). Rumus dari metrik *recall* ditunjukkan pada Rumus 3.

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

F1 *Score* juga merupakan pengukur F. *F1 score* menyatakan keseimbangan antara *presicion* dan *recall*. Rumus dari metrik F1 *Score* ditunjukkan pada Rumus 4.

$$F1Score = \frac{2 \cdot precision \cdot recall}{precision+recall} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Bagian ini akan membahas mengenai hasil analisis sentimen dan membandingkan performa dari keempat metode yang digunakan, yaitu RF, LR, SVM, dan IndoBERT.

A. Hasil Pemrosesan Data

Setelah data yang melewati beberapa tahapan pembersihan data, maka didapatkan 609 *tweet* yang diberi label, dan 348 *tweet* yang tidak diberi label. Berikut merupakan contoh *tweet* yang telah melalui tahapan pembersihan data.

TABEL 3
CONTOH TWEET YANG TELAH DIBERSIHKAN

NO	TWEET	LABEL TWEET
1	gatau mau bicara apa data bocor bahkan bikin narasi warga suruh jaga nik baik padahal tiap register sim card setor data kamu lah sistem kamu kocak gimane coba mana bilang hacker jangan serang njirr stop being an idiot lah ente kadang kadang ente	NEGATIF
2	langkah tepat ambil tangan kasus bocor data sim card	POSITIF
3	dpr ri minta tanggung jawab pihak kait bocor data sim card jumlah milyar	NETRAL

Dapat dilihat dari *tweet* yang ada pada Tabel 3, kata-kata dalam kalimat tersebut sudah diformalkan dan dikembalikan ke kata dasarnya. Namun, perlu diketahui bahwa masih terdapat beberapa kata-kata yang bentuknya tidak formal dikarenakan kata-kata tersebut tidak terdaftar dalam *dictionary* yang digunakan. Untuk memudahkan keberlangsungan penelitian ini, kata-kata tidak resmi yang tidak berhasil diubah menggunakan *dictionary* akan dibiarkan.

B. Word Cloud

Untuk menjawab permasalahan sebaran kata untuk setiap sentimen, maka akan dilakukan pembuatan *word cloud* untuk masing-masing sentimen. *Word cloud* akan menampilkan kata-kata yang memiliki frekuensi kemunculan terbesar dengan *font* yang besar pula.

Gambar 3 *Word Cloud* untuk keseluruhan *Tweet*

Gambar 3 menunjukkan *word cloud* untuk keseluruhan *tweet* yang digali. Dapat dilihat bahwa kata-kata yang sering muncul dalam keseluruhan *tweet* adalah kata *SIM card, data SIM, bocor data, miliar data, kominfo,* dan *card bocor*. Kata-kata tersebut merupakan kata-kata yang menerangkan isu kebocoran data kartu SIM yang terjadi. Kata *kominfo* juga sering

muncul dalam keseluruhan data. Kominfo merupakan pihak yang bertanggung jawab dalam menetapkan kebijakan di bidang Komunikasi dan Informatika, sehingga masyarakat Indonesia ingin tahu bagaimana tanggapan pihak Kominfo mengenai isu kebocoran data kartu SIM ini.

Gambar 4 *Word Cloud* untuk Sentimen Negatif

Gambar 4 menunjukkan *word cloud* untuk *tweet* dengan sentimen negatif. Dapat dilihat bahwa kata-kata yang sering muncul dalam *tweet* negatif adalah kata *sim card, data sim, kominfo,* dan *bocor data*. Kata-kata tersebut sama dengan kata-kata yang ada pada *word cloud* dari keseluruhan data. Dapat dilihat bawah kata *kominfo* lebih sering muncul pada data dengan sentimen negatif. Pada data yang dikumpulkan, dapat dilihat bahwa kebanyakan *tweet* dengan sentimen negatif merupakan *tweet* yang ditujukan kepada Kominfo. *Tweet* tersebut pada umumnya membahas amarah masyarakat mengenai pertanggung jawaban Kominfo sebagai lembaga yang seharusnya menjaga data pribadi seluruh masyarakat Indonesia.

Gambar 5 *Word Cloud untuk sentimen netral*

Gambar 5 menunjukkan *word cloud* untuk *tweet* dengan sentimen netral. Dapat dilihat bahwa kata-kata yang sering muncul dalam *tweet* positif adalah kata *data sim, kominfo, miliar data, sim card,* dan *kartu sim.* Kata-kata tersebut juga sama dengan kata-kata yang ada pada *word cloud* dari keseluruhan data. Dapat dilihat bawah kata *kominfo* dan *miliar data* juga sering muncul pada data dengan sentimen netral. Kata *miliar data* khususnya sering muncul karena sentimen netral biasanya dipenuhi dengan *tweet* dari media Indonesia yang melaporkan berita kasus bocornya 1,3 miliar data kartu SIM. Kata *kominfo* juga tetap sering muncul, karena banyak dari media memberitakan tanggapan Kominfo mengenai kasus kebocoran ini.

Gambar 6 *Word Cloud* untuk sentimen positif

Gambar 6 menunjukkan *word cloud* untuk *tweet* dengan sentimen positif. Dapat dilihat bahwa kata-kata yang sering muncul dalam *tweet* positif adalah kata *sim card, data sim, bocor data,* dan *card bocor*. Kata-kata tersebut juga sama dengan kata-kata yang ada pada *word cloud* lain. Salah satu kata yang unik muncul di *word cloud* positif adalah kata *bukan* dan *bukan kominfo*. Kata-kata tersebut biasanya muncul dalam *tweet* yang menjelaskan bahwa data yang bocor bukan dari sistem *database* Kominfo. Kata *badan siber* juga muncul pada *tweet* yang menjelaskan bahwa Kominfo akan bekerja sama dengan Badan Siber dan Sandi Negara (BSSN) untuk menyelesaikan kasus ini. Secara umum, kata-kata pada *word cloud* positif jauh lebih sedikit dibandingkan dua *word cloud* sebelumnya, dikarenakan sedikitnya *tweet* yang memiliki sentimen positif.

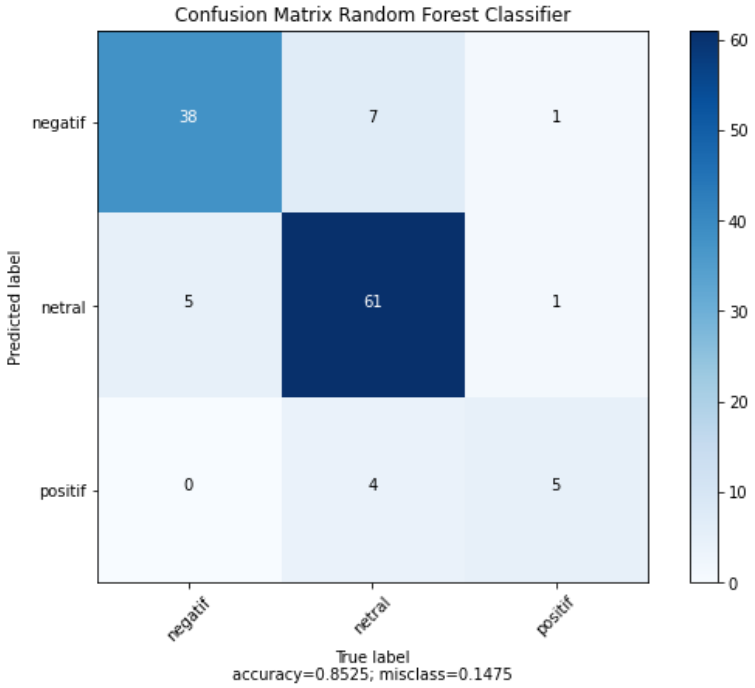
C. Hasil Algoritma RF, LR, dan SVM

Model pertama yang akan dibangun adalah model menggunakan algoritma *machine learning Random Forests, Logistic Regression*, dan *Support-Vector Machine*. Pemodelan tersebut akan dilakukan bersamaan menggunakan *library* Scikit-learn untuk klasifikasi [30]. Sebelum pemodelan dilakukan, maka akan dilakukan *oversampling* terlebih dahulu, khususnya untuk data dengan sentimen positif dikarenakan jumlahnya yang terlalu sedikit. *Oversampling* merupakan kegiatan menambahkan data dari kelas minoritas. Terdapat banyak cara untuk melakukan *oversampling*, salah satunya adalah SMOTE (*Synthetic Minority Oversampling Technique*). Dengan membuat data kelas minoritas sintetik sehingga jumlahnya sama dengan jumlah data pada kelas mayoritas, pendekatan SMOTE melakukan *oversampling* kelas minoritas. Studi sebelumnya telah menunjukkan bahwa pendekatan SMOTE telah berhasil meningkatkan kinerja akurasi model [31]. Data *oversampling* akan diisi dengan data sintesis.

TABEL 4
TOTAL DATA SETELAH OVERSAMPLING

NO	LABEL TWEET	TOTAL *TWEET* SEBELUM *OVERASMPLING*	TOTAL *TWEET* SETELAH *OVERASMPLING*
1	NEGATIF	199	252
2	NETRAL	252	252
3	POSITIF	36	252

Setelah dilakukan *upsampling*, maka pemodelan dimulai. Berikut merupakan *confusion matrix* dari algoritma *Random Forest*.



Gambar 7 *Confusion Matrix Random Forest*

Dari Gambar 7, dapat dilihat *confusion matrix* dari algoritma RF yang digunakan untuk pemodelan klasifikasi sentimen. Total akurasi dari RF adalah 85,2%. Dapat dilihat bahwa algoritma RF dapat mendeteksi sentimen negatif dan netral cukup baik, sedangkan sentimen positif memiliki performa yang lebih rendah. Hal ini dikarenakan sebaran kelas sentimen yang tidak seimbang, walaupun telah dilakukan *oversampling* sebelumnya.

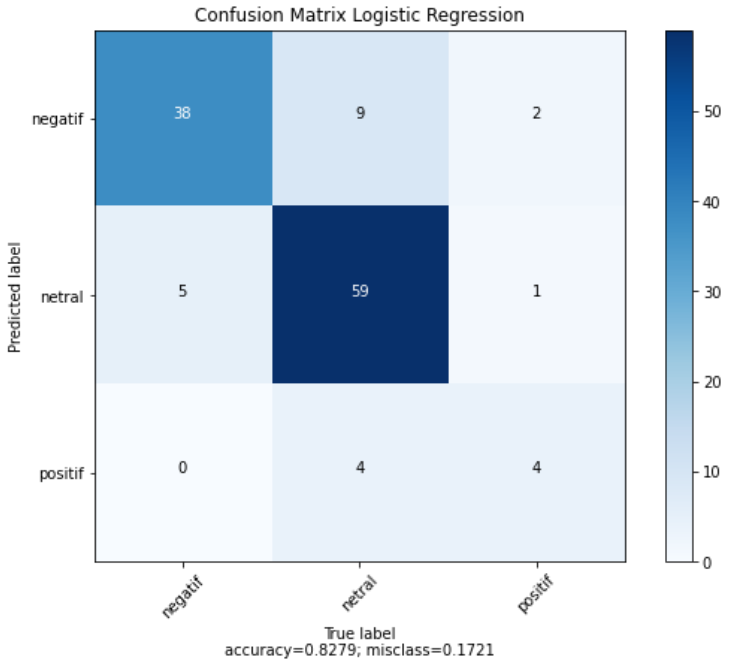
Berikut merupakan hasil dari *precision*, *recall*, dan *f1-score* dari algoritma RF.

TABEL 5
PERFORMA ALGORITMA *RANDOM FOREST*

LABEL	*PRECISION*	*RECALL*	*F1-SCORE*	*SUPPORT*
NEGATIF	0.83	0.88	0.85	43
NETRAL	0.91	0.85	0.88	72
POSITIF	0.56	0.71	0.63	7

Dari Tabel 5, dapat dilihat performa untuk label positif cukup rendah dibandingkan dengan performa untuk label negatif dan netral. Hal ini juga dipengaruhi oleh total data validasi label positif yang hanya berjumlah 7. Untuk *weighted average* dari RF akan ditampilkan di akhir. Untuk algoritma LR, berikut merupakan *confusion matrix*-nya.

p-ISSN: 2443-2210
e-ISSN: 2443-2229



Gambar 8 *Confusion Matrix Logistic Regression*

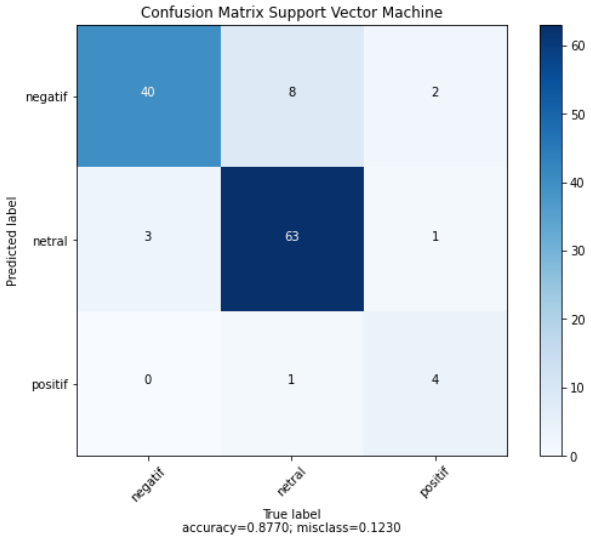
Dari Gambar 8, dapat dilihat *confusion matrix* dari algoritma LR. Total akurasi dari LR adalah 82,7%. Akurasi dari LR lebih rendah dibandingkan dengan RF. Dari *confusion matrix* pada Gambar 8, dapat dilihat pula bahwa algoritma LR lebih sulit dalam menentukan *true netral* dan *true positive* dibandingkan dengan RF, walaupun tidak signifikan. Berikut merupakan hasil dari *precision, recall,* dan *f1-score* dari algoritma LR.

TABEL 6
PERFORMA ALGORITMA *LOGISTIC REGRESSION*

LABEL	*PRECISION*	*RECALL*	*F1-SCORE*	*SUPPORT*
NEGATIF	0.78	0.88	0.83	43
NETRAL	0.91	0.82	0.86	72
POSITIF	0.50	0.57	0.53	7

Dari Tabel 6, dapat dilihat kembali untuk performa pada label positif lebih rendah dibandingkan dengan netral dan negatif dengan alasan yang sama seperti pada algoritma RF. Secara umum, performa dari algoritma LR lebih rendah dibandingkan dengan RF. Semua nilai *f1-score* dari ketiga label lebih rendah dibandingkan dengan pada algoritma RF, walaupun tidak secara signifikan.

Selanjutnya, berikut merupakan *confusion matrix* untuk algoritma SVM.



Gambar 9 *Confusion Matrix Support Vector Machine*

Dari Gambar 9, dapat dilihat *confusion matrix* dari algoritma SVM Total akurasi dari LR adalah 87,7%. Akurasi dari SVM merupakan akurasi yang tertinggi dibandingkan dengan LR dan RF. Dari *confusion matrix* pada Gambar 9 dapat dilihat pula bahwa algoritma SVM mampu mendeteksi *true negative* dan *true netral* lebih baik dibandingkan dengan RF dan LR, namun performanya sedikit lebih rendah dalam mendeteksi *true positive* dibandingkan dengan RF. Berikut merupakan hasil dari *precision, recall,* dan *f1-score* dari algoritma SVM.

TABEL 7
PERFORMA ALGORITMA *SUPPORT-VECTOR MACHINE*

LABEL	*PRECISION*	*RECALL*	*F1-SCORE*	*SUPPORT*
NEGATIF	0.80	0.93	0.86	43
NETRAL	0.94	0.88	0.91	72
POSITIF	0.80	0.57	0.67	7

Dari Tabel 7, dapat dilihat bahwa secara umum, performa SVM lebih baik dibandingkan dengan RF dan LR. Semua nilai *f1-score* untuk setiap label pada SVM lebih tinggi dibandingkan pada RF dan LR. Secara umum, performa untuk sentimen negatif dan netral dari ketiga algoritma tersebut hampir serupa, sedangkan pada sentimen positif, terdapat penurunan performa yang cukup signifikan untuk semua algoritma.
Berikut merupakan *macro average* untuk *precision, recall,* dan *f1-score* pada ketiga algoritma tersebut. *Macro average* dipilih karena memperlakukan semua kelas secara setara terlepas dari ketidakseimbangan kelas klasifikasi.

TABEL 8
MACRO AVEGRAGE RF, LR, DAN SVM

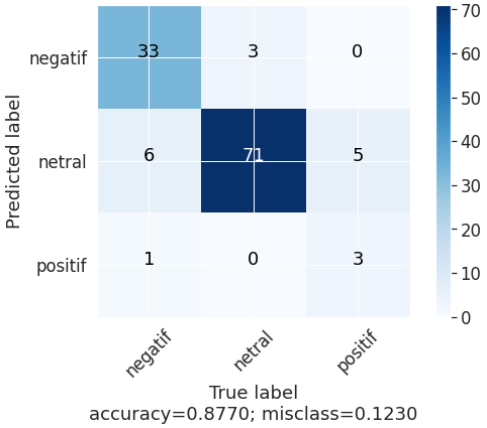
ALGORITMA	*PRECISION*	*RECALL*	*F1-SCORE*
RANDOM FOREST	0.76	**0.81**	0.78
LOGISTIC REGRESSION	0.72	0.75	0.74
SUPPORT-VECTOR MACHINE	**0.85**	0.79	**0.81**

Dari Tabel 8, dapat dilihat performa ketiga algoritma tersebut dalam *metric marco-average*. Secara umum, *support-vector machine* memiliki performa terbaik kecuali untuk aspek *recall*, dimana *random forest* memiliki performa yang terbaik. Nilai *precision* dari dapat diartikan bahwa dari semua *tweet* yang diprediksi dalam suatu kategori sentimen, *support-vector machine* dapat menebaknya dengan benar dengan performa terbaik. Perbedaan nilai *precision* antara *support-vector machine* dan dua algoritma lain cukup signifikan, yaitu 0.82 untuk SVM, 0.76 untuk *random forest* dan 0.72 untuk *logistic regression.* Nilai *recall* pada Tabel 8 menunjukkan bahwa dari semua *tweet* dalam suatu kategori sentimen sebenarnya, *random forest* dapat

menebak *tweet* tersebut dengan benar paling banyak. Perbedaan nilai *recall* antara ketiga algoritma tersebut tidak sesignifikan pada *precision,* dengan 0.81 untuk *random forest,* 0.75 untuk *logistic regression,* dan 0.79 untuk *support-vector machine.* Sedangkan nilai *F1-Score* memperhitungkan kombinasi antara hasil *precision* dan *recall*. *F1-Score* terbaik juga dimiliki oleh algoritma SVM. *Logistic Regression* memiliki performa terburuk dibandingkan dengan ketiga algoritma lainnya. Nilai *F1-Score* antara *support-vector machine* dan *random forest* juga tidak berbeda secara signifikan.

D. Hasil dengan Model IndoBERT

Selanjutnya, model akan dibangun menggunakan model IndoBERT. Model IndoBERT yang dipilih adalah versi IndoBERT-large-p2 dari Hugging Face. Proses tokenisasi juga dilakukan menggunakan model IndoBERT yang sama. *Training* dilakukan selama 5 *epochs* dan menggunakan rasio pembagian data latih dan data validasi yang sama seperti sebelumnya, yaitu 80:20. Pelatihan dilakukan selama 5 *epochs* karena model BERT biasanya cukup dilakukan selama 3-5 *epcohs.* Berikut merupakan *confusion matrix* dari pembangunan model menggunakan IndoBERT.



Gambar 10 *Confusion Matrix Model IndoBERT*

Dari Gambar 10, dapat dilihat *confusion matrix* dari pembangunan model klasifikasi menggunakan IndoBERT, model tersebut dapat mengklasifikasikan *true negative* dan *true netral* dengan cukup baik. Untuk sentimen positif, model IndoBERT juga sulit untuk mendeteksi sentimen tersebut dikarenakan sedikitnya data dengan sentimen positif. Secara umum, performa dari model ini hampir mirip dengan performa algoritma SVM pada bagian sebelumnya. Berikut merupakan hasil dari *precision*, *recall*, dan *f1-score* dari model IndoBERT.

TABEL 9
PERFORMA MODEL INDOBERT

LABEL	*PRECISION*	*RECALL*	*F1-SCORE*	*SUPPORT*
NEGATIF	0.92	0.82	0.87	40
NETRAL	0.87	0.96	0.91	74
POSITIF	0.75	0.38	0.50	8

E. Perbandingan Semua Model

Setelah mendapatkan hasil dari semua model, berikut merupakan perbandingan *macro average* dari semua model yang dibangun.

TABEL 10
MACRO AVEGRAGE RF, LR, SVM, DAN INDOBERT

ALGORITMA	*PRECISION*	*RECALL*	*F1-SCORE*
RANDOM FOREST	0.76	**0.81**	0.78
LOGISTIC REGRESSION	0.72	0.75	0.74
SUPPORT-VECTOR MACHINE	**0.85**	0.79	**0.81**
INDOBERT	0.84	0.72	0.76

Dari Tabel 10, dapat dilihat bahwa performa dari SVM memberikan hasil yang terbaik dibandingkan algoritma yang lain. Performa terbaik kedua adalah RF, dilanjutkan dengan IndoBERT dan LR. Performa IndoBERT yang dianggap sebagai *state-of-the-art* dalam NLP lebih rendah dibandingkan algoritma *supervised machine learning* SVM dan RF. Salah satu alasannya adalah karena ketidakseimbangan data untuk setiap kelas yang digunakan. Kurangnya data yang digunakan juga membuat performa IndoBERT kalah dengan algoritma lainnya. Sedangkan untuk SVM, RF, dan LR, adanya *oversampling* juga membantu performa algoritma tersebut dalam mendeteksi sentimen yang ada.

IV. SIMPULAN

Dalam penelitian ini, telah dilakukan analisis klasifikasi sentimen terhadap isu kebocoran data kartu SIM menggunakan pendekatan *machine learning*. Terdapat 4 model yang digunakan dengan menggunakan algoritma *Random Forest, Logistic Regression, Support-Vector Machine* dan menggunakan model IndoBERT. Data yang digunakan didapatkan dari *tweet* masyarakat Indonesia dari media sosial Twitter. Penelitian ini bertujuan untuk menemukan sebaran kata-kata yang digunakan masyarakat Indonesia dalam menanggapi isu kebocoran data kartu SIM dan untuk mengetahui perbandingan dari keempat model algoritma tersebut dalam mengklasifikasikan sentimen.

Dari data yang diambil, ditemukan bahwa sentimen negatif dan netral mendominasi isu kebocoran data kartu SIM, dengan kata-kata seperti *sim card, data sim, bocor data, miliar data, kominfo*. Banyak dari *tweet* menyuarakan ketidakpuasannya terhadap kinerja Kominfo sebagai lembaga yang seharusnya menjaga data pribadi masyarakat Indonesia. Masyarakat Indonesia juga kecewa karena Kominfo merupakan Lembaga yang mewajibkan seluruh masyarakat Indonesia memasukkan data pribadi mereka dalam pendaftaran kartu SIM. Sentimen netral dipenuhi dengan *tweet* dari portal berita yang melaporkan kasus ini, sedangkan sangat sedikit sentimen positif yang ditemukan. Sentimen positif biasanya berisi dengan *tweet* yang menjelaskan bahwa Kominfo akan bekerja sama dengan Badan Siber dan Sandi Negara (BSSN) untuk menyelesaikan kasus ini.

Dalam membangun model sentimen, ditemukan bahwa algoritma SVM merupakan algoritma dengan performa terbaik, dilihat dari nilai *f1-score* yang paling tinggi dibandingkan dengan algoritma lainnya. Model IndoBERT yang merupakan *state-of-the-art* memiliki performa yang lebih rendah dibandingkan dengan algoritma *supervised learning* biasa lainnya. Hal ini dikarenakan terdapat ketidakseimbangan kelas yang cukup besar antara sentimen positif dengan sentimen netral dan negatif. Dapat disimpulkan bahwa dalam kasus dengan data yang sedikit dan ketidakseimbangan kelas yang cukup besar, algoritma *supervised learning* biasa dapat memiliki performa yang lebih baik dibandingkan dengan model *pre-trained state-of-the-art*. Data set dan *source code* dari penelitian ini dapat diakses melalui Zenodo [32].

Hasil penelitian ini bisa menjadi referensi penelitian selanjutnya dalam menganalisis klasifikasi sentimen dengan beberapa algoritma *supervised machine learning* maupun dengan model *pre-trained*. Penelitian ini juga bisa menjadi referensi bagi pemangku kebijakan untuk mengevaluasi kinerjanya terhadap perlindungan data pribadi dengan mendengarkan opini masyarakat Indonesia di media sosial.

UCAPAN TERIMA KASIH

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan anugerah-Nya. Pada kesempatan ini penulis hendak menyampaikan ucapan terimakasih kepada:

1. Bapak Dr. Mudjahidin, S.T., M.T., selaku Kepala Departemen Sistem Informasi ITS
2. Ibu Nur Aini Rakhmawati S.Kom., M.Sc., Eng, PhD selaku Dosen Mata Kuliah Etika Profesi yang telah meluangkan waktu pada pengerjaan artikel ini
3. Orang tua dan saudara penulis yang telah memberikan dukungan dan doa dalam pembuatan artikel ini
4. Teman-teman penulis yang telah saling mendukung dan memotivasi satu sama lain dalam pembuatan artikel ini

Besar harapan kami artikel ini akan memberikan manfaat bagi pembaca ke depannya.

DAFTAR PUSTAKA

[1] M. A. Rizaty, "Pengguna Twitter di Indonesia Capai 18,45 Juta pada 2022," 17 September 2022. [Online]. Available: https://dataindonesia.id/digital/detail/pengguna-twitter-di-indonesia-capai-1845-juta-pada-2022.

[2] J. Karthik, V. Tamizhazhagan dan S.Narayana, "Data Leak Identification in Social Networks using K Means Clustering & Tabu K Means Clustering," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 9, no. 2, 2019.

[3] P. Pemerintah, Peraturan Pemerintah Republik Indonesia Nomor 71 Tahun 2019 Tentang Penyelenggaraan Sistem dan Transaksi Elektronik, 2019.

[4] "Lagi-lagi Bocor Data, Kali Ini 1,3 Miliar Info Registrasi Kartu SIM," CNN Indonesia, [Online]. Available: https://www.cnnindonesia.com/teknologi/20220902062206-192-842221/lagi-lagi-bocor-data-kali-ini-13-miliar-info-registrasi-kartu-sim. [Diakses 17 Sep 2022].

[5] J. I. Criado, R. Sandoval-Almazan dan J. R. Gil-Garcia, "Government innovation through social media," *Gov Inf Q,* vol. 30, p. 319–326, 2013.

[6] S. Wager, "Comments on: A random forest guided tour," *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research,* vol. 25, p. 261–263, 2016.

[7] Bahrawi, "Sentiment Analysis using Random Forest Algorithm-Online Social Media Based," *Journal of Information Technology and Its Utilization,* vol. 2, no. 2, pp. 29-33, 2019.

[8] J. O. E. Hoffman, "Logistic Regression. Academic Press," 2019.

[9] D. A. Pisner dan D. M. Schnyer, "Support vector machine," *Academic Press,* 2020.

[10] A. R. F. Koto, J. H. Lau dan T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," [Online]. Available: https://arxiv.org/abs/2011.00677. [Diakses 2020].

[11] B. Wilie, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," 2020. [Online]. Available: https://www.semanticscholar.org/paper/IndoNLU%3A-Benchmark-and-Resources-for-Evaluating-Wilie-Vincentio/03f22e693a0c00bae8a66a64a2fecb0f11a4b034.

[12] N. I. Wibowo, T. A. Maulana, H. Muhammad dan N. A. Rakhmawati, "Perbandingan Algoritma Klasifikasi Sentimen Twitter Terhadap Insiden Kebocoran Data Tokopedia," *JISKa,* vol. 6, no. 2, pp. 120-129, 2021.

[13] A. Ikhsan, F. Arrizal, A. C. M. Wibowo dan N. A. Rakhmawati, "Pengaruh Akun BOT pada Sentiment Masyarakat terhadap Pinjaman Online di Twitter The Effect of Bot Accounts on Community Toward Online Loans on Twitter," *SISTEMASI : Jurnal Sistem Informasi,* vol. 11, no. 1, pp. 137-147, 2022.

[14] "twintproject/twint: An advanced Twitter scraping & OSINT tool written in Python.," twintproject, [Online]. Available: https://github.com/twintproject/twint . [Diakses 18 Sep 2022].

[15] J.Tang, H.Lie, Y.Cao dan Z.Tang, "Email data cleaning,," dalam *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005.

[16] N. A. Salsabila, Y. A. Winatmoko, A. A. Septiandri dan A. Jamal, "Colloquial Indonesian Lexicon," *IEEE IALP,* 2018.

[17] louisowen6, "NLP_bahasa_resources/combined_slang_words.txt at master · louisowen6/NLP_bahasa_resources," [Online]. Available: https://github.com/louisowen6/NLP_bahasa_resources/blob/master/combined_slang_words.txt. [Diakses 18 Sep 2022].

[18] ramaprakoso, "analisis-sentimen/acronym.txt at master · ramaprakoso/analisis-sentimen," [Online]. Available: https://github.com/ramaprakoso/analisis-sentimen/blob/master/kamus/acronym.txt. [Diakses 18 Sep 2022].

[19] B. Vimal, "Application of Logistic Regression in Natural Language Processing," [Online]. Available: www.ijert.org.

[20] sastrawi, "sastrawi/sastrawi: High quality stemmer library for Indonesian Language (Bahasa)," [Online]. Available: https://github.com/sastrawi/sastrawi. [Diakses 18 Sep 2022].

[21] I. S. I. R. Sholehurrohman, "Analisis Sentimen Tweet Kasus Kebocoran Data Penggunaan Facebook oleh Cambrigde Analytica," *Jurnal Pepadun,* vol. 3, no. 1, pp. 140-147, April 2022.

[22] C. S. a. B. Ribeiro, "The importance of stop word removal on recall values in text categorization," *Proceedings of the International Joint Conference on Neural Networks,* vol. 3, pp. 1661-1666, 2003.

[23] F. Heimerl, S. Lohmann, S. Lange dan T. Ertl, "Word Cloud Explorer: Text Analytics Based on Word Clouds," dalam *47th Hawaii International Conference on System Sciences*, 2014.

[24] D. E. C. a. I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics,* vol. 10, no. 5, pp. 2780-2788, 2021.

[25] J. Devlin, M.-W. Chang, K. Lee dan K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," dalam *NAACL-HLT*, 2019.

[26] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky dan N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Sekolah Daring pada Twitter Menggunakan Supervised Machine Learning," *JuTISI,* vol. 7, no. 1, 2021.

[27] S. A. Shaikh, "Measures Derived from a 2 x 2 Table for an Accuracy of a Diagnostic Test," *J Biom Biostat,* vol. 2, no. 5, 2011.

[28] P. Singh, N. Singh, K. K. Singh dan A. Singh, "Diagnosing of disease using machine learning," *Machine Learning and the Internet of Medical Things in Healthcare,* pp. 89-111, 2011.

[29] F. Demir, "Deep autoencoder-based automated brain tumor detection from MRI data," *Artificial Intelligence-Based Brain-Computer Interface,* p. 317–351, 2022.

[30] Scikit-learn, "Supervised learning — scikit-learn 1.1.2 documentation," [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning. [Diakses 19 Sep 2022].

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall dan W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *JAIR,* vol. 16, 2002.

[32] N. Wibowo, N. Rakhmawati, T. Maulana dan H. Muhammad, "Data Set Sentimen Twit Terhadap Insiden Kebocoran Data Tokopedia," 2020. [Online]. Available: https://zenodo.org/record/4230596#.Y579ghVBw2w.