ANALISIS SENTIMEN MASYARAKAT TERHADAP ISU KELANGKAAN BERAS BERDASARKAN MEDIA SOSIAL X DENGAN PENDEKATAN BIDIRECTIONAL ENCODER REPRESENTATION FROM TRANSFORMERS (BERT)

Y. A. Hafiz; Dimas Aryo Anggoro
Prodi Teknik Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta

Abstrak

Isu kelangkaan beras di Indonesia sering disebabkan oleh cuaca tidak menentu, masalah distribusi, dan aksesibilitas dari petani ke konsumen, yang berdampak besar pada sosial dan ekonomi masyarakat karena beras adalah bahan pangan utama. Diskusi tentang isu ini meningkat di media sosial seperti X, membuat analisis sentimen sangat relevan untuk memahami persepsi masyarakat. Penelitian ini menggunakan model pembelajaran mendalam BERT (Bidirectional Encoder Representations from Transformers) untuk menganalisis sentimen masyarakat terhadap isu kelangkaan beras. Tujuannya adalah untuk mengetahui persepsi masyarakat terkait isu tersebut. Data dikumpulkan dari X menggunakan alat Tweet-Harvest, dengan kata kunci "beras langka" pada periode 16-24 Februari 2024. Data tersebut diproses melalui tahapan seperti pembersihan data, case folding, tokenisasi, normalisasi, penghapusan kata umum, dan stemming. Labeling data dilakukan dengan metode kamus sentimen untuk mengkategorikan tweet menjadi sentimen positif, negatif, atau netral. Implementasi model BERT menggunakan IndoBERTweet-base-uncased dari IndoLEM, yang dioptimalkan untuk teks berbahasa Indonesia. Model dilatih dengan proporsi data 90:10 untuk training dan testing. Evaluasi model menunjukkan akurasi 80.20%, precision 71.71%, recall 73.09%, dan F1 score 80.40%. Hasil penelitian ini diharapkan dapat memberikan wawasan berharga dalam memahami persepsi masyarakat terhadap isu kelangkaan beras pada media sosial X. Visualisasi data melalui wordcloud juga memberikan gambaran yang jelas tentang sentimen masyarakat terhadap isu ini.

Kata Kunci: Analisis Sentimen, BERT, Data X, Kelangkaan Beras.

Abstract

The issue of rice scarcity in Indonesia is often caused by unpredictable weather, distribution problems, and accessibility issues for farmers to consumers, significantly impacting the social and economic aspects of society as rice is a staple food. Discussions on this issue have surged on social media platforms like X, making sentiment analysis highly relevant for understanding public perception. This research utilizes the deep learning model BERT (Bidirectional Encoder Representations from Transformers) to analyze public sentiment towards the rice scarcity issue. The aim is to gauge public perception regarding this issue. Data was collected from X using the Tweet-Harvest tool with the keyword "rice scarcity" from February 16-24, 2024. The data underwent

preprocessing steps including data cleaning, case folding, tokenization, normalization, stop word removal, and stemming. Data labeling was conducted using a sentiment lexicon method to categorize tweets into positive, negative, or neutral sentiments. BERT model implementation utilized IndoBERTweet-base-uncased from IndoLEM, optimized for Indonesian language texts. The model was trained with a 90:10 data split for training and testing. Model evaluation showed an accuracy of 80.20%, precision of 71.71%, recall of 73.09%, and an F1 score of 80.40%. This research aims to provide valuable insights into understanding public perception of the rice scarcity issue on social media platform X. Visualization through word clouds also offers a clear depiction of public sentiment towards this issue.

Keywords: BERT, Rice Scarcity, Sentiment Analysis, X Data.

1. PENDAHULUAN

Pada periode tertentu seringkali muncul isu kelangkaan beras yang mungkin terjadi dikarenakan berbagai alasan (Khalidin & Wahyuni, 2020), cuaca yang tidak menentu, musim kemarau yang panjang dan curah hujan yang tidak konsisten. Selain itu, permasalahan distribusi dan aksesibilitas, khususnya terkait perpindahan beras dari petani ke konsumen akhir, juga merupakan faktor lain yang berkontribusi terhadap kelangkaan beras (Sholikhah & Anjani, 2023). Dalam kehidupan masyarakat Indonesia beras merupakan bahan pangan yang paling banyak dikonsumsi (Abdummunib, 2023). Oleh karena itu, akan terdapat dampak sosial dan ekonomi yang besar akibat kelangkaan atau lonjakan harga beras. Dengan semakin populernya diskusi dan argumen mengenai isu kelangkaan beras di *platform* media sosial seperti X, analisis sentimen menjadi semakin relevan.

Media sosial, khususnya X, Dalam era digital saat ini, telah menjadi cara yang baru dan sederhana untuk mempelajari peristiwa atau berita terkini (Zukhrufillah, 2018), termasuk berita tentang masalah ketersediaan beras. X memberikan kesempatan kepada penggunanya untuk bergabung dalam percakapan dengan cepat dan luas, berbagi tujuan, menulis tentang berbagai topik dan membahas isu isu yang sedang terjadi (Afshoh & Pamungkas, 2017).

Analisis sentimen, juga dikenal sebagai *Opinion Mining*, adalah studi komputer tentang persepsi orang dan sikap terhadap hal-hal seperti barang, jasa, organisasi, orang, masalah, peristiwa, tema, dan atributnya (Zhang et al., 2018). Hal ini bisa memberikan informasi yang mendalam tentang bagaimana masyarakat merespons isu kelangkaan beras dengan menggunakan data yang tersedia di X.

Analisis sentimen dalam aplikasinya menggunakan komputasi untuk mengidentifikasi sentimen implisit dalam sebuah teks. Salah satu komputasi tersebut yaitu *Natural Language Processing* yang mampu mempelajari cara-cara memanfaatkan komputer untuk memahami dan mengubah teks atau ucapan bahasa alami (Chowdhury, 2003). Model pembelajaran mendalam yang bisa mengerjakan tugas NLP salah satunya adalah BERT (*Bidirectional Encoder Representations from Transformers*). Pendekatan menggunakan model pembelajaran mendalam tersebut telah berhasil memberikan hasil yang baik terhadap berbagai tugas *Natural Language Processing* (Mas et al., 2021), seperti *Sentiment Analysis, Question Answering, dan Text Summarization* (Sofiana, 2021).

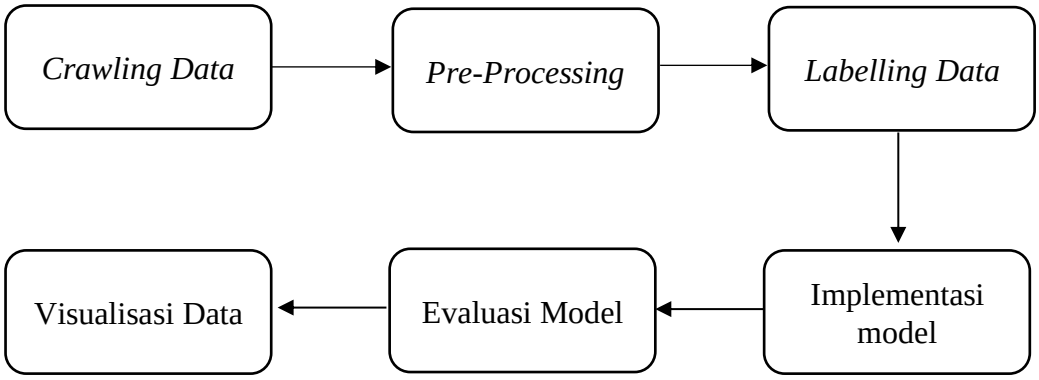
Penelitian sebelumnya yang menggunakan pendekatan BERT yaitu oleh (Mas et al., 2021) Analisis Sentimen *Customer Review* Aplikasi Ruang Guru dengan Metode BERT (*Bidirectional Encoder Representations from Transformers*), hasil yang diperoleh berdasarkan dari nilai *F1 Score* yang didapat dari presisi dan *recall* yakni 98.9% dan nilai akurasi bernilai 99%. Penelitian oleh (Putu et al., 2023) Analisis Sentimen terhadap Perundungan Siber pada Twitter menggunakan Algoritma *Bidirectional Encoder Representations from Transformer* (BERT), mendapatkan hasil akurasi sebesar 81%, presisi 80,67%, *recall* 80,33%, dan *f1 score* 80,67%.

Penelitian tentang analisis sentimen masyarakat terhadap isu kelangkaan beras di X menjadi penting karena beberapa alasan. Pertama, informasi yang diperoleh dari analisis sentimen dapat membantu pemerintah dan pihak berkepentingan lainnya dalam membuat kebijakan yang secara efektif mengatasi masalah pasokan beras. Kedua, Pemahaman menyeluruh terhadap perspektif masyarakat dapat memfasilitasi advokasi dan komunikasi yang lebih efektif dalam menanggapi masalah ini. Ketiga, dengan melihat pola dan tren dalam sentimen masyarakat, kita dapat mengidentifikasi perubahan persepsi dan tingkat kepuasan terhadap langkah-langkah yang diambil oleh pemerintah atau pelaku industri dalam mengatasi kelangkaan beras.

Namun, dalam konteks analisis sentimen masyarakat terhadap isu kelangkaan beras pada X, penggunaan BERT masih terbatas pada penelitian yang ada. Oleh karena itu, penelitian ini bertujuan untuk menerapkan BERT dalam menganalisis sentimen masyarakat terhadap isu kelangkaan beras dan juga bertujuan untuk mengetahui persepsi masyarakat terkait dengan isu tersebut. Diharapkan hasil penelitian ini dapat memberikan pemahaman yang lebih baik tentang dinamika sosial terkait dengan kelangkaan beras.

2. METODE PENELLITIAN

Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap isu kelangkaan beras di Indonesia dengan memanfaatkan teknologi pembelajaran mendalam. Proses penelitian ini dirancang secara sistematis untuk menghasilkan data yang akurat dan dapat diandalkan. Tahapan-tahapan penelitian meliputi pengumpulan data, pemrosesan awal, pelabelan data, implementasi model, evaluasi model, dan visualisasi data. Setiap tahapan memiliki peranan penting dalam keseluruhan proses analisis sentiment. Langkah penelitian bisa dilihat pada gambar 1.



Gambar 1. Langkah penelitian

2.1 Crawling Data

Data diambil dari hasil cuitan atau twit pengguna X dengan menggunakan tools Tweet-Harvest, data yang diambil merupakan twit dengan kata kunci ‘beras langka’ pada tanggal 16 Februari 2024 sampai dengan tanggal 24 Februari 2024. Berikut adalah kode untuk melakukan crawling data dengan Tweet-Harvest.

```
data = ‘data_beras.csv’
search_keyword = ‘beras langka’
limit = 1500
!npx --yes tweet-harvest@2.2.8 -o “{data}” -s “{search_keyword}” -1 {limit} --token “9a762107d9e14122a22e8c189499abd57fa9fd02”
```

Gambar 2. Kode *Crawling Data*

Baris pertama kode mendefinisikan variable ‘data’ yang berisi nama file CSV yang

akan digunakan untuk menyimpan data hasil pencarian dari X, baris selanjutnya mendefinisikan variabel '*search_keyword*' yang berisi string pencarian untuk mencari tweet-tweet yang mengandung kata kunci "beras langka", kemudian baris berikutnya mendefinisikan variabel '*limit*' yang menentukan jumlah maksimum tweet yang akan diambil dalam pencarian, Di baris terakhir ini adalah perintah *shell* yang menggunakan 'npx' (*npm package runner*) untuk menjalankan utilitas 'tweet-harvest'. Opsi yang digunakan adalah -o "{data}" Menentukan nama *file output* untuk menyimpan hasil pencarian tweet, yang sudah didefinisikan sebelumnya dalam variabel 'data', -s "{*search_keyword*}" Menentukan string pencarian yang sudah didefinisikan sebelumnya dalam variabel '*search_keyword*', -l {*limit*} Menentukan jumlah maksimum tweet yang akan diambil, yang sudah didefinisikan sebelumnya dalam variabel '*limit*', --token "9a762107d9e14122a22e8c189499abd57fa9fd02" Menentukan token akses yang diperlukan untuk mengakses API X. Token ini harus disediakan agar utilitas dapat mengambil data dari X.

Tabel 1. Contoh Hasil *Crawling Data*

Tanggal	Username	Tweet
Sun Feb 25 22:59:25 +0000 2024	gemasvidya	Semalem beli beras di daily foodhall cuma boleh 2 karung... asli deh ini beras lagi super langka ya
Sun Feb 25 23:36:10 +0000 2024	mamihnyawin	@WidasSatyo @brbhyy Oke gas oke gas silent majority tiba2 beras langka, bbm ilang, listrik naiik wkwk Tuh tuh pilihan siapa cobaa

2.2 *Pre-Processing*

Data yang diperoleh dari X umumnya belum terstruktur dengan kaidah penulisan. Oleh karena itu, perlu dilakukan penyesuaian seperti membersihkan dan menyamaratakan seluruh kata sehingga membantu agar data lebih mudah diproses dan mendapatkan hasil akurasi yang diharapkan (Kusuma & Pamungkas, 2023). Proses *pre-processing* dilakukan secara manual menggunakan kode python. Terdapat 6 tahapan pada proses *pre-processing* antara lain *data cleaning*, *case folding*, *tokenization*, normalisasi, *stopword removal*, *stemming*.

2.2.1 *Data Cleaning*

Data cleaning dilakukan untuk membersihkan kalimat dataset dari apapun yang dapat memengaruhi hasil analisis, seperti simbol, angka, tautan, emoji, nama pengguna (@namapengguna), tagar. Pembersihan data ini dilakukan agar data dapat lebih mudah diproses. Proses *data cleaning* menggunakan kode python yang melibatkan library *regular expression*. Library regular expression ini yang nantinya akan digunakan untuk membuat kode menghapus simbol, angka, tautan, emoji, username, dan tagar.

Tabel 2. Contoh Hasil *Data Cleaning*

Tweet	***Data Cleaning***
Dulu pas Minyak langka masih agak santai lah solusinya bisa kurangi konsumsi minyak tapi skrg beras yg langka kek anjir??!!! makanan pokok woy elah buset dah	Dulu pas Minyak langka masih agak santai lah solusinya bisa kurangi konsumsi minyak tapi skrg beras yg langka kek anjir makanan pokok woy elah buset dah

2.2.2 *Case Folding*

Case Folding merupakan operasi pemrosesan data yang mengonversi atau menghapus semua karakter kapital dari dokumen dan menggantinya dengan huruf kecil (Lestandy et al., 2021). Case folding perlu dilakukan karena membantu meningkatkan konsistensi data, mengurangi dimensi data, meningkatkan akurasi model, memperbaiki kualitas fitur, memberikan keseragaman dalam pencocokan pola, dan mengurangi noise. Proses *case folding* dilakukan secara manual menggunakan python dengan membuat sebuah fungsi '*case_folding*', fungsi ini akan menerima sebuah argumen '*text*' dan memeriksa apakah argumen tersebut adalah *string*. Jika itu adalah *string*, fungsi akan menggunakan metode '.*lower()*' dari tipe data *string* untuk mengubah semua huruf dalam teks menjadi huruf kecil.

Tabel 3. Contoh Hasil *Case Folding*

Data Cleaning	***Case Folding***
Dulu pas Minyak langka masih agak santai lah solusinya bisa kurangi konsumsi minyak tapi skrg beras yg langka kek anjir makanan pokok woy elah buset dah	dulu pas minyak langka masih agak santai lah solusinya bisa kurangi konsumsi minyak tapi skrg beras yg langka kek anjir makanan pokok woy elah buset dah

2.2.3 *Tokenization*

Teks yang telah dipotong menjadi kata-kata atau diuraikan, dikenal sebagai token yaitu hasil dari operasi ini (Listyarini & Anggoro, 2021). Seperti *case folding*, pada proses *tokenization* juga menggunakan python dan membuat fungsi '*tokenize*'. Fungsi ini menerima argumen '*text*' dan kemudian menggunakan metode '*.split()*' dari tipe data string untuk memisahkan teks menjadi token-token berdasarkan spasi atau karakter pemisah lainnya secara *default*. Hasilnya adalah sebuah daftar yang terdiri dari berbagai token.

Tabel 4. Contoh Hasil *Tokenizati*on

Case Folding	***Tokenization***
dulu pas minyak langka masih agak santai lah solusinya bisa kurangi konsumsi minyak tapi skrg beras yg langka kek anjir makanan pokok woy elah buset dah	'dulu', 'pas', 'minyak', 'langka', 'masih', 'agak', 'santai', 'lah', 'solusinya', 'bisa', 'kurangi', 'konsumsi', 'minyak', 'tapi', 'skrg', 'beras', 'yg', 'langka', 'kek', 'anjir', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'dah'

2.2.4 Normalisasi

Normalisasi merupakan tahapan untuk mengubah kata-kata yang tidak standar pada dataset menjadi kata-kata yang standar sesuai dengan kaidah bahasa Indonesia. Proses normalisasi akan menggunakan kamus 'kbba.csv', sebuah kamus kata-kata yang sudah dinormalisasi sesuai dengan kaidah Bahasa Indonesia. Kamus ini diambil dari https://github.com/ramaprakoso/analisis-sentimen/blob/master/kamus/kbba.txt. Setelah kamus kata-kata yang dinormalisasi dimuat, sebuah fungsi bernama '*normalized_term*' akan dibuat. Fungsi ini menerima dokumen, dalam hal ini daftar token dari proses *tokenization* sebelumnya sebagai argumen. Fungsi ini kemudian menggunakan kamus kata-kata yang telah dimuat sebelumnya untuk mengembalikan daftar token yang dinormalisasi. Jika sebuah token ditemukan dalam kamus kata-kata yang dinormalisasi, token tersebut akan diganti dengan bentuk yang dinormalisasi.

Tabel 5. Contoh Hasil Normalisasi

Tokenization	**Normalisasi**

'dulu', 'pas', 'minyak', 'langka', 'masih', 'agak', 'santai', 'lah', 'solusinya', 'bisa', 'kurangi', 'konsumsi', 'minyak', 'tapi', 'skrg', 'beras', 'yg', 'langka', 'kek', 'anjir', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'dah'	'dulu', 'pas', 'minyak', 'langka', 'masih', 'agak', 'santai', 'lah', 'solusinya', 'bisa', 'kurangi', 'konsumsi', 'minyak', 'tapi', 'sekarang', 'beras', 'yang', 'langka', 'seperti', 'anjing', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'deh'

2.2.5 *Stopword Removal*

Stopword digunakan untuk menghilangkan kata-kata yang dianggap tidak memiliki arti penting dan dapat memperlambat atau menghambat proses analisis (Farisi & Hadi, 2023). Proses *stopword removal* dilakukan menggunakan kode python, yang dimana kode tersebut akan mengimpor modul '*stopwords*' dari *library Natural Language Toolkit* atau biasa disingkat NLTK untuk mengakses daftar *stopwords* dalam bahasa Indonesia.

Tabel 6. Contoh Hasil *Stopword Removal*

Normalisasi	***Stopword Removal***
'dulu', 'pas', 'minyak', 'langka', 'masih', 'agak', 'santai', 'lah', 'solusinya', 'bisa', 'kurangi', 'konsumsi', 'minyak', 'tapi', 'sekarang', 'beras', 'yang', 'langka', 'seperti', 'anjing', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'deh'	'pas', 'minyak', 'langka', 'santai', 'solusinya', 'kurangi', 'konsumsi', 'minyak', 'beras', 'langka', 'anjing', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'deh'

2.2.6 *Stemming*

Merupakan tahapan dalam proses pembentukan kata dasar yang sesuai dengan kaidah bahasa Indonesia dengan menghilangkan imbuhan berupa awalan, akhiran, atau paduan (prefiks) (Assuja, 2016). Library sastrawi python, merupakan paket yang dibuat untuk memudahkan menemukan kata dasar bahasa Indonesia, digunakan dalam prosedur *stemming* ini.

Tabel 7. Contoh Hasil *Stemming*

Stopword Removal	***Stemming***
'pas', 'minyak', 'langka', 'santai', 'solusinya', 'kurangi', 'konsumsi', 'minyak', 'beras', 'langka', 'anjing', 'makanan', 'pokok', 'woy', 'elah', 'buset', 'deh'	pas minyak langka santai solusi kurang konsumsi minyak beras langka anjing makan pokok woy elah buset deh

2.3 Labelling Data

Labeling dataset adalah langkah penting dalam penelitian analisis sentimen, terutama untuk penelitian ini yang menggunakan model BERT. Model akan menggunakan label yang diberikan pada dataset untuk mempelajari pola hubungan antar kata dan memprediksi sentimen tweet.

Proses pemberian label dataset pada penelitian ini menggunakan metode kamus sentimen. Metode ini menggunakan kamus sentimen, yang berisi daftar kata-kata dan label sentimennya. Kamus sentimen diambil dari repository github https://github.com/fajri91/InSet. Kata-kata yang termasuk dalam tweet akan dicocokkan dengan kata-kata dalam kamus sentimen, dan label sentimen untuk tweet tersebut akan dipilih berdasarkan label sentimen kata-kata yang cocok. Proses ini juga dibantu dengan menggunakan kode program Python yang melakukan analisis sentimen pada kata-kata yang diambil dari file kamus sentimen berupa positive.csv dan negative.csv.

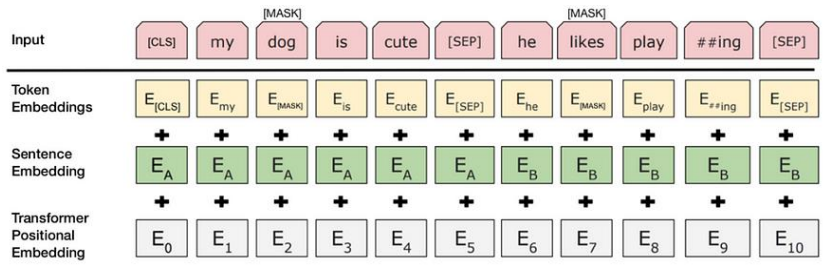
```
fungsi load_teks_dari_csv(path_berkas)
    teks = daftar kosong
    buka berkas di path_berkas dengan mode 'r' dan encoding utf-8
    untuk setiap baris dalam berkas:
        ambil teks dari kolom 'stemming_data', hapus spasi ekstra, dan ubah menjadi
huruf kecil
        jika teks tidak kosong:
            tambahkan teks ke dalam daftar teks
    kembalikan teks

fungsi load_kamus(path_berkas)
    kamus = himpunan kosong
    buka berkas di path_berkas dengan mode 'r' dan encoding utf-8
    untuk setiap baris dalam berkas:
        jika baris tidak kosong:
            tambahkan huruf kecil dari baris[0] ke dalam kamus
    kembalikan kamus

fungsi analisis_sentimen(teks, kata_positif, kata_negatif)
    token = pecah teks menjadi kata-kata dan ubah menjadi huruf kecil
    num_positif = hitung token yang ada di dalam kata_positif
    num_negatif = hitung token yang ada di dalam kata_negatif
    skor_sentimen = num_positif - num_negatif
    kembalikan skor_sentimen

teks = load_teks_dari_csv('Hasil_preprocessing (6).csv')

kata_positif = load_kamus('positive1-1.csv')
kata_negatif = load_kamus('negative1-1.csv')

sentimen = daftar kosong
skor = daftar kosong
```

```
untuk setiap teks dalam teks:
    skor_sentimen = analisis_sentimen(teks, kata_positif, kata_negatif)
    jika skor_sentimen > 0:
        sentimen = 'positif'
    else jika skor_sentimen < 0:
        sentimen = 'negatif'
    lainnya:
        sentimen = 'netral'
    tambahkan sentimen ke dalam daftar sentimen
    tambahkan skor_sentimen ke dalam daftar skor

buat DataFrame sentimen_df dengan kolom 'Teks', 'Sentimen', 'Skor' menggunakan
teks, sentimen, dan skor

tampilkan 500 baris pertama dari sentimen_df
```

Gambar 3. *Pseudocode Labelling Data*

2.4 Implementasi Model

Model BERT akan digunakan untuk analisis sentimen setelah *labelling dataset*. BERT adalah model representasi bahasa terlatih yang dibuat oleh peneliti *Google AI Language* menggunakan metodologi *Deep learning*. BERT memanfaatkan *Transformers*, sebuah metode yang menguji hubungan antara kata dan konteksnya dalam sebuah teks (Pradipta et al., 2023).



Gambar 4. Arsitektur BERT

Gambar 4 merupakan arsitektur BERT yang menunjukkan bagaimana setiap kata ditokenisasi menjadi *Word Embedding* dalam struktur vektor. Struktur vektor ini merupakan kontribusi dari lapisan *Encoder* dari *Transformers* BERT, yang memungkinkan model *transformer* untuk memahami posisi setiap kata, yang mungkin memiliki makna kontekstual yang berbeda meskipun memiliki bentuk yang sama (Pradipta et al., 2023). Dengan menggunakan *encoder*, BERT membuat representasi kontekstual dari input teks.

Tokenisasi teks adalah langkah pertama dalam proses ini, membuat token kecil seperti kata, subkata, atau huruf. Selanjutnya, setiap token diubah menjadi vektor kata menggunakan prosedur *embedding*, proses ini menggunakan *embedding* yang telah dilatih

sebelumnya oleh model. Setelah itu, token tersebut diproses oleh *encoder* BERT, yang terdiri dari beberapa lapisan *transformers*. Setiap lapisan menggunakan operasi *multi-head self-attention* dan *full-connected feedforward networks* untuk menghasilkan representasi kontekstual.

BERT merupakan algoritma yang unik karena dapat mengevaluasi token kiri dan kanan untuk memahami konteks secara bidireksional atau dua arah dan menangkap keterkaitan kontekstual yang lebih dalam. Proses ini dilakukan berulang pada setiap lapisan *encoder* yang ditumpuk. Pemahaman input teks yang semakin hierarkis dihasilkan dengan mengulangi proses ini pada setiap lapisan *encoder* yang ditumpuk (Aljabar et al., 2024).

BERT menghasilkan representasi dikombinasikan dari token [CLS], yang berguna untuk klasifikasi dan sentimen analisis (Alaparthi & Mishra, 2020.). Secara keseluruhan, BERT adalah alat yang bagus untuk pemrosesan bahasa alami dengan kemampuan transfer learning yang baik karena pemahaman kontekstual dan dua arah yang kuat (Kokab et al., 2022).

Implementasi model BERT ini akan menggunakan optimizer Adam untuk proses pelatihan. Adam, singkatan dari *Adaptive Moment Estimation*, merupakan optimizer yang sering digunakan dalam pelatihan model *deep learning* karena kemampuannya menggabungkan keuntungan dari dua metode optimasi lainnya, yaitu RMSProp dan *Stochastic Gradient Decent with Momentum*. Rumus utama dari optimizer Adam adalah:

$$m_t = \beta_1 - 1 + (1 - \beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} - 1 + (1 - \beta_2) g_t^2 \quad (2)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (4)$$

$$\theta_t = \theta_{t-1} - \frac{\eta \widehat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (5)$$

Secara sederhana, Adam menggunakan dua nilai rata-rata yang bergerak atau *moving averages* untuk mengestimasi gradien atau kemiringan dari fungsi yang sedang dioptimalkan, m_t untuk gradien pertama dan v_t untuk gradien kedua atau kuadrat gradien. Gradien gtg_tgt adalah gradien dari fungsi loss terhadap parameter θ_t pada iterasi ke-t. Parameter θ_t adalah parameter model pada iterasi ke-t. Faktor peluruhan β_1 dan β_2 mengatur seberapa cepat nilai rata rata ini berubah. Bias-koreksi $\widehat{m}_t$ dan $\hat{v}_t$ memastikan bahwa estimasi ini tidak biased, terutama pada awal pelatihan. Parameter η

adalah *learning rate* yang menentukan seberapa besar langkah perubahan pada setiap iterasi pelatihan. *Epsilon* (ϵ) adalah nilai kecil untuk mencegah pembagian dengan nol. Dengan menggabungkan informasi dari gradien dan rata-rata gradien, Adam dapat menyesuaikan langkah pembelajaran secara adaptif, yang membantu dalam mencapai konvergensi yang lebih cepat dan stabil selama pelatihan model (Kingma & Ba, 2014).

Pada implementasi model juga melibatkan pemisahan data atau data splitting menjadi *data training* dan *data testing*. Berdasarkan penelitian oleh (Ardika et al., 2020) proposi pembagian *data training* dan *data testing* yaitu 90:10, proporsi pembagian ini menghasilkan hasil yang sangat baik, dengan memberikan 90% data untuk dilatih akan membantu dalam pembelajaran pola yang lebih baik.

2.5 Evaluasi Model

Setelah model BERT diimplementasikan, langkah selanjutnya adalah mengevaluasi kinerjanya. Metrik-metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score* digunakan untuk menilai performa dan kinerja model dalam klasifikasi sentimen seperti pada penelitian oleh (Septian & Kharisma, 2023).

Accuracy adalah ukuran kinerja yang paling intuitif dan ini hanyalah rasio observasi yang diprediksi dengan benar terhadap total observasi. Berikut formula dari *Accuracy*:

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi} \quad (6)$$

Precision adalah rasio observasi positif yang diprediksi dengan benar terhadap total observasi positif yang diprediksi. Berikut formula dari *precision*:

$$Precision = \frac{(TP)}{(TP + FP)} \quad (7)$$

Recall adalah rasio observasi positif yang diprediksi dengan benar terhadap semua observasi di kelas aktual. Berikut formula dari *Recall*:

$$Recall = \frac{(TP)}{(TP + FN)} \quad (8)$$

F1 score adalah rata-rata tertimbang dari precision dan recall. Berikut formula dari *F1 score*:

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \quad (9)$$

Evaluasi model juga akan memanfaatkan *confusion matrix*. *Confusion matrix* adalah alat yang membantu dalam visualisasi kinerja algoritma klasifikasi, terutama dalam mengidentifikasi jumlah prediksi benar dan salah yang dibuat oleh model dibandingkan dengan nilai aktual dalam data uji. *Confusion matrix* menampilkan perbandingan antara hasil prediksi dan nilai sebenarnya dalam bentukam matriks yang menunjukkan *True*

Positive (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Selain itu, model akan dibuatkan antarmuka web untuk memudahkan pengguna dalam mengakses dan menggunakan model BERT yang telah dibangun. Untuk tujuan ini, Flask akan digunakan sebagai framework pengembangan web. Flask adalah framework micro web berbasis Python yang ringan dan fleksibel, sehingga sangat cocok untuk proyek-proyek yang membutuhkan kustomisasi dan kecepatan pengembangan.

2.6 Visualisasi Data

Visualisasi data mempermudah pembaca dalam memahami informasi yang disajikan. Dengan memanfaatkan *Pie Chart*, visualisasi ini menghasilkan *wordcloud* dari data ulasan. Hal ini menjadikan informasi penting dalam data ulasan lebih mudah diinterpretasikan, sehingga mendukung pengambilan keputusan dan analisis yang lebih efektif.

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan dataset yang dikumpulkan dari media sosial X dengan cara *crawling data* menggunakan *tools tweet-harvest* dengan kata kunci "beras langka". Proses *crawling data* dilakukan pada tanggal 16 Februari 2024 sampai 24 Februari 2024. Data yang diperoleh sebanyak 3030 tweet, dengan kolom tanggal, *username*, dan tweet. Hasil *Crawling Data* dapat dilihat pada tabel 8.

Tabel 8. Hasil *Crawling Data*

Tanggal	Username	Tweet
Fri Feb 16 23:46:39 +0000 2024	kztanpopo	@convomf Sembako udah naik mana beras langka lagi hiks. ini belum ramadhan pdhl biasanya ramadhan tambah naik
Fri Feb 16 23:45:43 +0000 2024	rahmwt	Walau sebagian komoditas tersebut masih mahal harganya. Coba aja ya bisa berdayakan petani untuk tanam komoditas lain selain padi dan bisa menggaet minat masyarakat terhadap komoditas tersebut sepertinya kita tidak perlu repot impor beras kalau beras langka
...	...	...

Sat Feb 24 00:01:50 +0000 2024	chefanja_Cassie	@jokowi Gabah ditempat saya tdk langka cuma harga gabah mahal jd klo di selep utk menghasilkan 1 kg beras perlu biaya 15rb disini yg mahal pupuknya krn yg subsidi sedikit banyak beli yg non subsidi. Petani minta subsidi pupuknya yang diperbanyak. Solusi buat petani mana?

Pada tahap *pre-processing*, data dibersihkan dari elemen-elemen yang dapat memengaruhi hasil analisis melalui proses *data cleaning*, untuk memastikan data dapat lebih mudah diproses. Proses *data cleaning* ini menggunakan kode python yang melibatkan *library Reguler expression*. Hasil *data cleaning* dapat dilihat pada tabel 9.

Tabel 9. Hasil *Data Cleaning*

Tweet	***Data Cleaning***
@RaisaTjokrodnta Pak @jokowi gimana atuh emak2 pada ngeluh harga beras naik dan langka - gimana mau makan gratis pak @prabowo @gibran_tweet makin susah x ya, bisa2 diganti pake singkong atau jagung	Pak gimana tuh emak pada ngeluh harga beras naik dan langka gimana mau makan gratis pak makin susah x ya bisa diganti pake singkong atau jagung

Proses *pre-processing* juga melibatkan *case folding* atau konversi semua karakter kapital menjadi huruf kecil, sehingga mempermudah proses selanjutnya. Hasil *case folding* dapat dilihat pada tabel 10.

Tabel 10. Hasil *Case Folding*

Data Cleaning	***Case Folding***
Pak gimana tuh emak pada ngeluh harga beras naik dan langka gimana mau makan gratis pak makin susah x ya bisa diganti pake singkong atau jagung	pak gimana tuh emak pada ngeluh harga beras naik dan langka gimana mau makan gratis pak makin susah x ya bisa diganti pake singkong atau jagung

Setelah dilakukan *case folding* selanjutnya yaitu pemotongan data menjadi kata-kata melalui proses tokenisasi. Tokenisasi ini dilakukan menggunakan kode python dengan membuat fungsi ‘tokenize’. Hasil tokenisasi dapat dilihat pada tabel 11.

Tabel 11. Hasil *Tokenization*

Case Folding	***Tokenization***
pak gimana tuh emak pada ngeluh harga beras naik dan langka gimana mau makan gratis pak makin susah x ya bisa diganti pake singkong atau jagung	'pak', 'gimana', 'tuh', 'emak', 'pada', 'ngeluh', 'harga', 'beras', 'naik', 'dan', 'langka', 'gimana', 'mau', 'makan', 'gratis', 'pak', 'makin', 'susah', 'x', 'ya', 'bisa', 'diganti', 'pake', 'singkong', 'atau', 'jagung'

Proses selanjutnya yaitu normalisasi untuk mengubah kata-kata tidak standar menjadi sesuai dengan standar Bahasa Indonesia. Proses normalisasi memanfaatkan sebuah kamus ‘kbba.csv’, sebuah kamus kata-kata yang sudah dinormalisasi sesuai dengan kaidah Bahasa Indonesia. Hasil Normalisasi dapat dilihat pada tabel 12.

Tabel 12. Hasil Normalisasi

Tokenization	**Normalisasi**
'pak', 'gimana', 'tuh', 'emak', 'pada', 'ngeluh', 'harga', 'beras', 'naik', 'dan', 'langka', 'gimana', 'mau', 'makan', 'gratis', 'pak', 'makin', 'susah', 'x', 'ya', 'bisa', 'diganti', 'pake', 'singkong', 'atau', 'jagung'	'pak', 'bagaimana', 'itu', 'ibu', 'pada', 'ngeluh', 'harga', 'beras', 'naik', 'dan', 'langka', 'bagaimana', 'mau', 'makan', 'gratis', 'pak', 'makin', 'susah', 'sekali', 'iya', 'bisa', 'diganti', 'pakai', 'singkong', 'atau', 'jagung'

Selanjutnya, pada proses *stopword removal*, kata-kata yang dianggap tidak memiliki arti penting dan dapat menghambat proses analisis akan dihilangkan. Proses *stopword removal* ini memanfaatkan modul *‘stopword’* dari *library Natural Language Toolkit* atau biasa disingkat NLTK. Hasil *stopword removal* pada tabel 13.

Tabel 13. Hasil *Stopword Removal*

Normalisasi	***Stopword Removal***
'pak', 'bagaimana', 'itu', 'ibu', 'pada', 'ngeluh', 'harga', 'beras', 'naik', 'dan', 'langka', 'bagaimana', 'mau', 'makan', 'gratis', 'pak', 'makin', 'susah', 'sekali', 'iya', 'bisa', 'diganti', 'pakai', 'singkong', 'atau', 'jagung'	'ibu', 'ngeluh', 'harga', 'beras', 'langka', 'makan', 'gratis', 'diganti', 'pakai', 'singkong', 'jagung'

Proses terakhir dari *pre-processing* yaitu *stemming*, dimana kata kata yang memiliki imbuhan, awalan, akhiran atau paduan akan diubah menjadi kata dasar. Proses *stemming* ini memanfaatkan *library* Sastrawi untuk memudahkan menemukan kata dasar Bahasa

Indonesia. Pada hasil stemming token-token kata sebelumnya akan diubah lagi menjadi string atau kalimat biasa. Hasil *Stemming* pada tabel 14.

Tabel 14. Hasil *Stemming*

Stopword Removal	***Stemming***
'ibu', 'ngeluh', 'harga', 'beras', 'langka', 'makan', 'gratis', 'diganti', 'pakai', 'singkong', 'jagung'	ibu ngeluh harga beras langka makan gratis ganti pakai singkong jagung

Data hasil *pre-processing* selanjutnya akan diberi label sentimen melalui proses *labelling*. Data tersebut dikategorikan ke dalam tiga label sentimen, negatif, netral, positif. Proses *labelling* dilakukan menggunakan metode kamus sentimen, Metode ini menggunakan kamus sentimen yang berisi daftar kata-kata dan label sentimennya. Kata-kata yang termasuk dalam data akan dicocokkan dengan kata-kata dalam kamus sentimen, dan label sentimen untuk tweet tersebut akan dipilih berdasarkan label sentimen kata-kata yang cocok. Hasil *labelling data* dapat dilihat pada tabel 15.

Tabel 15. Hasil *labelling data*

No	**Tweet**	**Sentimen**
1	ibu ngeluh harga beras langka makan gratis ganti pakai singkong jagung	Negative
2	sikap beras sinyalir langka gubernur arinal tinjau pasar	Neutral
3	beras langka kaya nasi bubur tindak darurat impor stabil harga beras	Positive
...	**...**	**...**
3030	gabah tempat langka harga gabah mahal selep hasil kg beras biaya rb mahal pupuk subsidi beli non subsidi tani subsidi pupuk banyak solusi tani	Negative

Setelah dataset diberi label, selanjutnya dataset akan dijadikan input untuk proses analisis. Dalam penelitian ini, analisis sentimen akan dilakukan menggunakan algoritma

BERT. Dataset akan dilakukan proses data splitting atau membagi data menjadi dua bagian, yaitu *data training* dan *data testing*. *Data training* digunakan untuk melatih model, sementara *data testing* digunakan untuk menguji seberapa baik kinerja model pada data yang belum pernah dilihat sebelumnya (Anggoro & Kurnia, 2020). Pembagian data dilakukan dengan proporsi 90:10 yang menghasilkan 2727 data training dan 303 data testing. Hasil Data splitting dapat dilihat pada tabel 16.

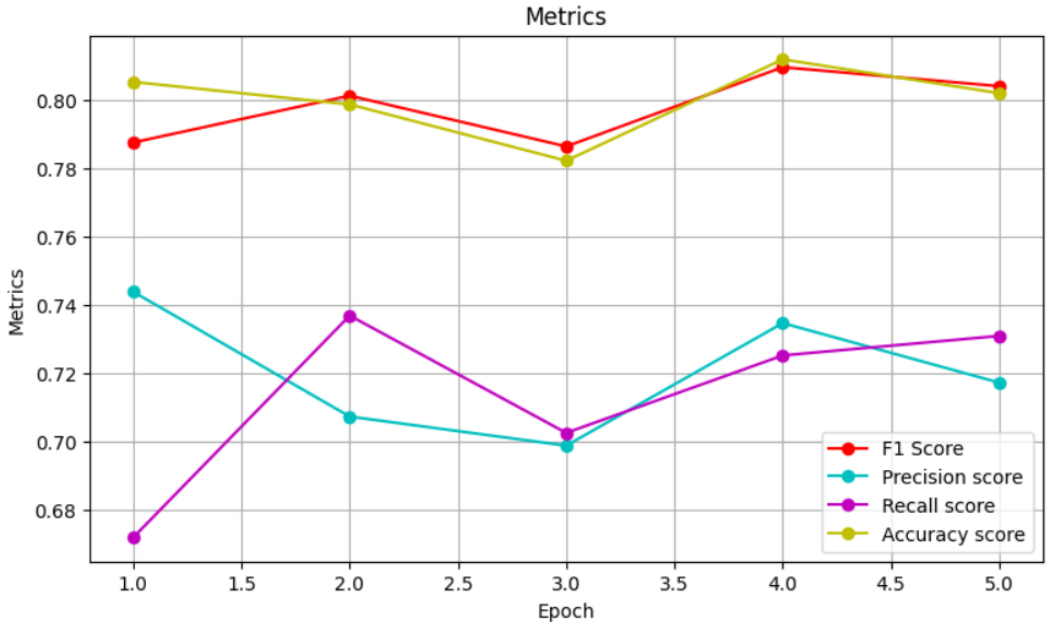
Tabel 16. Hasil Data Splitting

Category	**Label**	**Data type**	**Jumlah**
Negative	0	Train	1818
		Test	202
Neutral	1	Train	577
		Test	64
Positive	2	Train	332
		Test	37

Selanjutnya, menyiapkan BERT pretrained model. Model pretrained yang digunakan dalam penelitian ini adalah indoBERTweet-base-uncased dari IndoLEM, yang telah dilatih menggunakan korpus teks Bahasa Indonesia, hal ini berarti model sudah memiliki pemahaman mendalam tentang Bahasa Indonesia dan dapat menghasilkan representasi teks yang bermakna. Indobertweet adalah versi adaptasi dari BERT yang dioptimalkan untuk teks berbahasa Indonesia, khususnya teks yang berasal dari platform media sosial seperti Twitter dalam hal ini media sosial X. model ini dirancang untuk menangani karakteristik unik dari Bahasa informal dan singkatan yang sering ditemukan dalam tweet berbahasa Indonesia. penggunaan indobertweet memungkinkan peningkatan akurasi dalam tugas-tugas pemrosesan bahasa alami yang melibatkan teks media sosial, berkat pelatihan yang ekstensif pada data yang relevan (Koto et al, 2021.).

Dalam penelitian ini, parameter model diatur dengan *batch size* sebanyak 32 dan jumlah epoch sebanyak 5. Pemilihan *batch size* 32 bertujuan untuk mencapai keseimbangan antara efisiensi komputasi dan stabilitas pelatihan. Jumlah epoch sebanyak

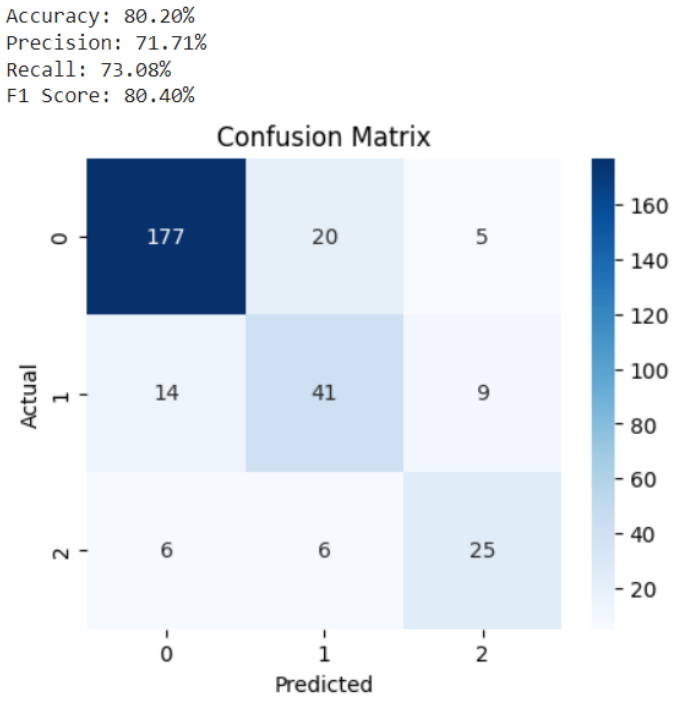
5 dipilih untuk memastikan model memiliki cukup waktu untuk belajar dari data tanpa *overfitting*. Optimizer yang digunakan adalah Adam, dengan nilai *learning rate* sebesar 4e-5 dan epsilon sebesar 1e-10. *Learning rate* 4e-5 dipilih karena merupakan nilai yang sering direkomendasikan untuk fine-tuning model BERT, dan setelah percobaan dengan beberapa nilai learning rate, nilai 4e-5 mendapatkan hasil yang terbaik. Epsilon sebesar 1e-10 digunakan untuk mencegah pembagian dengan nol dalam perhitungan pembaruan parameter.



Gambar 7. Grafik Metrik Evaluasi Model

Gambar 7 menunjukkan grafik metrik evaluasi model selama lima epoch pelatihan dengan data training. Pada grafik ini, terlihat bahwa nilai *F1 Score, Precision, Recall* dan *Accuracy* bervariasi di setiap epoch. *F1 score* dan *Accuracy* terlihat stabil dan tinggi, berkisar di sekitar 80%, menunjukkan konsistensi model dalam menjaga keseimbangan antara presisi dan *recall* serta ketepatan prediksi. *Precision* dan *recall* mengalami fluktuasi yang lebih signifikan, terutama pada epoch awal, namun cenderung stabil dan meningkat pada epoch selanjutnya, masing-masing berakhir di sekitar 74%. Hal ini menunjukkan bahwa model semakin baik dalam mengidentifikasi *true positives* dan mengurangi *false positives* serta *false negatives* seiring berjalannya waktu pelatihan. *Precision* Mengukur seberapa banyak prediksi positif model yang benar-benar positif. Nilai *precision* yang tinggi berarti model menghasilkan sedikit *false positives* dan itu adalah hal yang bagus. Sementara *recall* mengukur seberapa banyak sampel positif yang sebenarnya dapat diidentifikasi dengan benar oleh model. Nilai recall yang tinggi berarti model mampu menangkap sebagian besar dari sampel positif yang sebenarnya, mengurangi *false negatives*, semakin tinggi nilainya semakin bagus.

Untuk mengevaluasi performa dan kinerja algoritma BERT yang telah diimplementasikan, dapat digunakan *confusion matrix*. Dengan memanfaatkan *library* scikit-learn pada python, *confusion matrix* dapat dibuat dengan fungsi *confusion_matrix*. Selain itu, metrik-metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, *F1 Score* juga dapat dihitung untuk memberikan gambaran yang komprehensif tentang kemampuan model dalam mengklasifikasikan data dengan benar.



Gambar 6. Confusion Matrix

Pada gambar 8 Dengan menggunakan *Confusion matrix* untuk klasifikasi sentimen menggunakan algoritma BERT dapat dilihat bahwa ada 243 data prediksi dengan benar dan 60 data prediksi dengan salah. Untuk memudahkan membaca pembagian nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) dari setiap kelas, berikut adalah tabel ringkasan matrix confusion.

Tabel 17. Ringkasan Confusion Matrix

Kelas	**True Positive (TP)**	**True Negative (TN)**	**False Positive (FP)**	**False Negative (FN)**
0 (Negatif)	177	81	20	25
1 (Neutral)	41	213	26	23
2 (Positif)	25	252	14	12

Model BERT menghasilkan nilai *Accuracy* 80.20%, *Precision* 71.71%, *Recall*

73.09%, *F1 Score* 80.40%. Hasil ini juga bisa dihitung dan dibuktikan dengan menggunakan rumus Accuracy, Precision, Recall, dan F1 Score.

Accuracy adalah proporsi prediksi yang benar terhadap total prediksi. berikut cara menghitung accuracy menggunakan rumus seperti pada persamaan (6).

$$Accuracy = \frac{Jumlah\ prediksi\ benar}{Total\ prediksi} = \frac{243}{303} = 0.8020\ atau\ 80.20\%$$

Precision adalah proporsi prediksi benar untuk suatu kelas terhadap total prediksi untuk kelas tersebut. Precision dihitung untuk masing masing kelas. Berikut cara menghitung precision menggunakan rumus seperti pada persamaan (7).

$$Precision = \frac{TP}{TP+FP}$$

$$Precision_0 = \frac{TP_0}{TP_0+FP_0} = \frac{177}{177+20} = \frac{177}{197} = 0.8985$$

$$Precision_1 = \frac{TP_1}{TP_1+FP_1} = \frac{41}{41+26} = \frac{41}{67} = 0.6119$$

$$Precision_2 = \frac{TP_2}{TP_2+FP_2} = \frac{25}{25+14} = \frac{25}{39} = 0.6410$$

$$Precision\ Average = \frac{0.8985+0.6119+0.6410}{3} = 0.7171\ atau\ 71.71\%$$

Recall adalah proporsi prediksi benar untuk suatu kelas terhadap total kasus sebenarnya dari kelas tersebut. *Recall* dihitung untuk masing masing kelas. Berikut cara menghitung *recall* menggunakan rumus seperti pada persamaan (8).

$$Recall = \frac{TP}{TP+FN}$$

$$Recall_0 = \frac{TP_0}{TP_0+FN_0} = \frac{177}{177+25} = \frac{177}{202} = 0.8762$$

$$Recall_1 = \frac{TP_1}{TP_1+FN_1} = \frac{41}{41+23} = \frac{41}{64} = 0.6406$$

$$Recall_2 = \frac{TP_2}{TP_2+FN_2} = \frac{25}{25+12} = \frac{25}{37} = 0.6757$$

$$Recall\ Average = \frac{0.8762+0.6406+0.6757}{3} = 0.7308\ atau\ 73.08\%$$

F1 Score adalah rata rata tertimbang dari precision dan recall. F1 score dihitung untuk masing masing kelas. Pada *F1 score* menggunakan weighted average, yang dimana weighted *F1 Score* dihitung dengan memberikan bobot sesuai dengan jumlah nilai sebenarnya atau nilai aktual dari setiap kelas. Jadi, bobot untuk setiap kelas adalah proporsi dari total prediksi. Berikut cara menghitung *F1 Score* menggunakan rumus seperti pada persamaan (9).

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

$$F1\ Score_0 = 2 \times \frac{0.8985 \times 0.8762}{0.8985+0.8762} = 0.8872$$

$$F1\ Score_1 = 2 \times \frac{0.6119 \times 0.6404}{0.6119+0.6404} = 0.6260$$

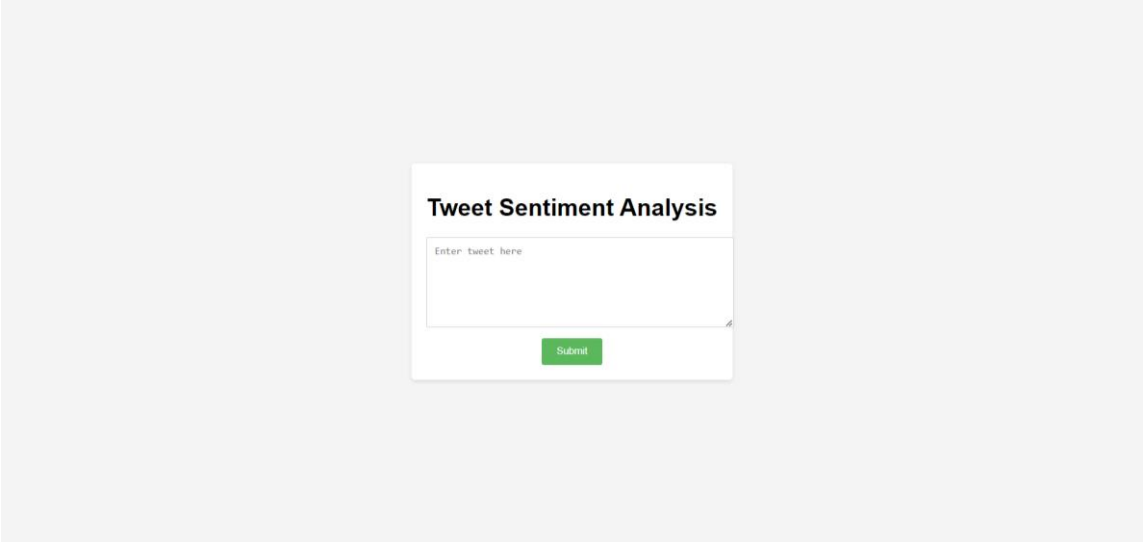
$$F1\ Score_2 = 2 \times \frac{0.6410 \times 0.6757}{0.6410 + 0.6757} = 0.6578$$

$$F1\ Score_{weighted} = \left(\frac{202}{303} \times 0.8872\right) + \left(\frac{64}{303} \times 0.6260\right) + \left(\frac{37}{303} \times 0.6578\right)$$

$$F1\ Score_{weighted} = 0.5916 + 0.1322 + 0.0803 = 0.8041\ atau\ 80.41\%$$

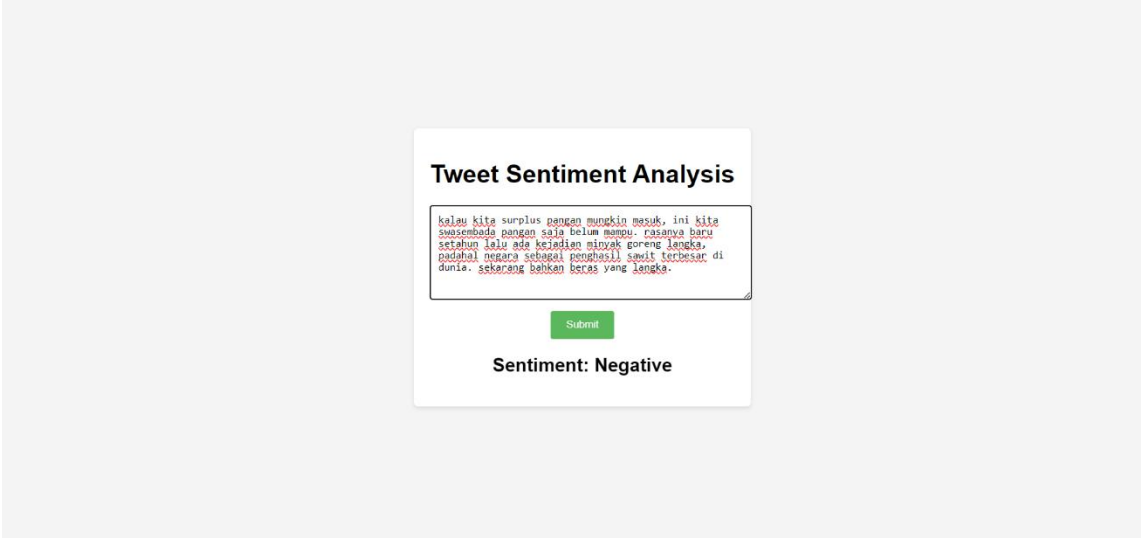
Nilai-nilai metrik evaluasi ini dipengaruhi oleh beberapa faktor, termasuk kualitas dan kuantitas data pelatihan, kompleksitas model, serta parameter yang digunakan selama pelatihan. Jika data pelatihan memiliki distribusi yang baik dan representatif dari data yang sebenarnya, model lebih mungkin untuk menghasilkan hasil yang baik. Model yang terlalu sederhana mungkin tidak cukup untuk menangkap pola dalam data, sedangkan model yang terlalu kompleks bisa *overfit* pada data pelatihan. Penyesuaian parameter seperti *learning rate*, *batch size*, dan epoch memengaruhi performa model. Nilai-nilai metrik evaluasi bisa ditingkatkan dengan memperhatikan faktor-faktor tersebut, terutama dengan menyesuaikan parameter atau juga menyesuaikan model pretrained yang paling relevan dari BERT itu sendiri. Hasil model BERT pada penelitian ini sudah cukup membuktikan bahwa BERT efektif untuk melakukan analisis sentimen. Efektivitas ini juga diperkuat dengan membandingkan hasil penelitian oleh (Santoso, 2022) dengan judul Analisis sentimen Twitter Bahasa Indonesia Menggunakan Pendekatan Machine Learning. Dalam penelitian tersebut, analisis sentimen dilakukan dengan menggunakan beberapa model, namun hasil yang terbaik yaitu menggunakan model *Logistic Regression*, dengan hasil *Accuracy* 61%, *F1 Score* 56%, *Precision* 58%, dan *Recall* 55%. Dengan menggunakan dataset yang sama pada penelitian tersebut, model BERT menunjukkan performa yang lebih baik dengan *Accuracy* 71.95%, F1 *Score* 72.07%, *Precision* 70.60%, *Recall* 71.96%. Perbandingan ini menegaskan bahwa penggunaan model BERT dapat meningkatkan performa analisis sentimen dibandingkan dengan pendekatan machine learning tradisional yang digunakan dalam penelitian sebelumnya.

Hasil model yang sudah didapatkan sebelumnya menggunakan BERT akan dibangun ke dalam sistem menggunakan framework Flask, agar model ini memiliki user interface dan dapat melakukan input. Tampilan user interface pada gambar 9.



Gambar 9. Tampilan User Interface

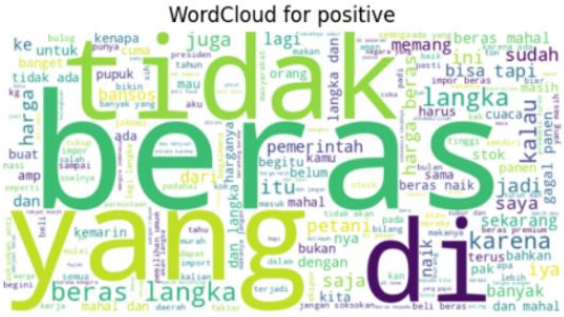
Pada web tersebut kita bisa memasukkan input sebuah tweet, jika dimasukkan tweet negatif, neutral ataupun positif maka sistem akan memberikan hasil prediksi. seperti contoh kita memasukkan input tweet negatif pada gambar 10.



Gambar 10. Tampilan Prediksi Model

Selanjutnya, penelitian ini akan menampilkan visualisasi data. Visualisasi ini memungkinkan data yang sebelumnya sulit dibaca agar dapat terbaca dengan jelas, sehingga informasi yang disajikan bisa dipahami oleh pembaca dengan mudah.

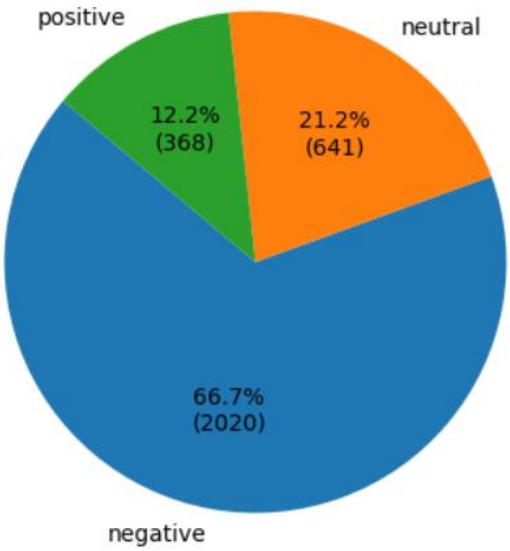
a b



c

Gambar 11. *Wordcloud* (a) sentimen negatif, (b) sentimen netral,
(c) sentimen positif

Gambar 11a, 11b, 11c merupakan representasi visual dalam bentuk *wordcloud* yang menggambarkan sebaran sentimen dari analisis twitter yang berkaitan dengan isu kelangkaan beras. Gambar 11a memperlihatkan kata kata yang paling banyak muncul pada sentimen negatif, kata kata ini menggambarkan kekhawatiran masyarakat tentang kelangkaan beras, kenaikan harga beras dan dampaknya terhadap masyarakat. Gambar 11b menunjukkan kata kata yang paling banyak muncul pada sentimen netral, kata kata ini menggambarkan informasi yang lebih netral tentang beras, seperti tentang petani, harga beras, dan cuaca. Gambar 11c menampilkan kata kata yang paling banyak muncul pada sentimen positif, kata kata ini menggambarkan harapan dan solusi terkait beras, seperti tentang upaya pemerintah untuk mengatasi kelangkaan beras dan solusi untuk menghadapai harga beras yang mahal.



Gambar 12. Diagram Pie Total sentimen

Pada gambar 10 ditampilkan sebuah diagram pie yang mengilustrasikan hasil analisis sentimen dari sebuah kumpulan data. Diagram tersebut menunjukkan bahwa sebagian besar data memiliki sentimen negatif, yaitu sekitar 66.7% yang berjumlah 2020 data. Sentimen negatif ini mungkin menunjukkan adanya ketidakpuasan, kekecewaan, atau kritik terhadap suatu hal. Selain itu, ada juga data yang bersifat netral, yaitu sekitar 21.2% yang berjumlah 641 data. Sentimen netral ini mungkin menunjukkan adanya sikap objektif, netral atau tidak berpihak terhadap suatu hal. Terakhir, ada juga data yang bersifat positif, yaitu sekitar 12.2% yang berjumlah 368 data. Sentimen positif ini mungkin menunjukkan adanya kepuasan, penghargaan, atau pujian terhadap suatu hal. Dengan demikian, diagram pie ini memudahkan pemahaman terhadap distribusi sentimen yang ada.

4. PENUTUP

4.1 Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap sentimen tweet pengguna X mengenai isu kelangkaan beras di media sosial X menggunakan algoritma BERT, dapat disimpulkan bahwa BERT efektif untuk mengidentifikasi sentimen terhadap isu kelangkaan beras pada platform tersebut. Analisis sentimen ini menghasilkan nilai *Accuracy* sebesar 80.20%, *Precision* 71.71%, *Recall* 73.09%, *F1 Score* 80.40% dengan proporsi dataset yang terdiri dari 90% *data training* dan 10% *data testing*. Efektivitas ini juga diperkuat dengan perbandingan dengan penelitian terdahulu yang menggunakan logistic regression, dengan menggunakan dataset yang sama, BERT bisa memberikan

hasil yang lebih baik. Selain itu, visualisasi data dalam bentuk *wordcloud* dan diagram pie memudahkan pemahaman distribusi sentimen terkait isu kelangkaan beras, dengan Sebagian besar data menunjukkan sentimen negatif 66.7%, disusul sentimen netral 21.2%, dan positif 12.2%.

4.2 Saran

Penulis menyadari bahwa penelitian ini masih memiliki ruang untuk peningkatan dan perbaikan di masa depan. Untuk meningkatkan kualitas hasil, diharapkan penelitian berikutnya dapat memperluas dataset yang digunakan, sehingga analisis sentimen yang dihasilkan lebih kaya informasi. Diharapkan juga untuk lebih memperinci proses pre-processing data, hal ini akan memastikan data berjalan lebih efektif dan efisien. Selain itu, penelitian ini juga dapat diperkaya dengan penggunaan algoritma yang berbeda untuk tujuan perbandingan atau juga penerapan algoritma BERT pada platform yang berbeda.

DAFTAR PUSTAKA

Abdummunib, A. (2023). *Kelangkaan Beras September 1972-Januari 1973 di Daerah Istimewa Yogyakarta*. Skripsi, Universitas Gadjah Mada.

Afshoh, F., & Pamungkas, E. W. (2017). *Analisa Sentimen Menggunakan Naïve Bayes Untuk Melihat Persepsi Masyarakat Terhadap Kenaikan Harga Jual Rokok Pada Media Sosial Twitter*. Skripsi, Universitas Muhammadiyah Surakarta.

Alaparthi, S., & Mishra, M. (2021). Bidirectional Encoder Representations from Transformers (BERT): A Sentiment Analysis Odyssey. *Journal of Marketing Analytics*, 9(2), 118-126.

Anggoro, D. A., & Kurnia, N. D. (2020). Comparison of Accuracy level of Support Vector Machine (SVM) and K-nearest neighbors (KNN) algorithms in predicting heart disease. International Journal, 8(5), 1689–1694.

Ardika, R. B. P., Irawan, B., & Setianingsih, C. (2020). Analisis sentimen data pada BPJS Kesehatan menggunakan backpropagation neural network. *eProceedings of Engineering*, 7(2). Telkom University.

Assuja, M. A., & Saniati, S. (2016). Analisis sentimen tweet menggunakan backpropagation neural network. *Jurnal Teknoinfo*, 10(2), 48-53.

Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, 37(1), 51-89

Hadi, S. F. S., Jondri, J., & Lhaksmana, K. M. (2023). *Analisis Sentimen menggunakan Recurrent Neural Network Terkait Isu Anies Baswedan Sebagai Calon Presiden 2024*. eProceedings of Engineering, 10(2), 1-10. Telkom University.

Khalidin, B., & Wahyuni, R. (2020). Intervensi Bulog terhadap kelangkaan beras menurut perspektif Tas'ir Al-Jabari (Studi kasus pada Perum Bulog Divisi Regional Aceh). *Al-Mudharabah: Jurnal Ekonomi dan Keuangan Syariah*, 1(1),

115-128.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*

Kokab, S. T., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, *14*, 100157.

Koto, F., Lau, J. H., & Baldwin, T. (2021). IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 10660-10668). Association for Computational Linguistics.

Kusuma, F. A., & Pamungkas, E. W. (2023). *Pendeteksian Hate Speech Pada Sosial Media Indonesia Dengan Algoritma Support Vector Machine (SVM) Dan Decision Tree*. Skripsi, Universitas Muhammadiyah Surakarta.

Lestandy, M., Abdurrahim, A., & Syafa'ah, L. (2021). Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes. Jurnal Rekayasa Sistem dan Teknologi Informasi (RESTI), 5(2), 81-92.

Listyarini, S. N., & Anggoro, D. A. (2021). Analisis Sentimen Pilkada di Tengah Pandemi Covid-19 Menggunakan Convolution Neural Network (CNN). *Jurnal Pendidikan dan Teknologi Indonesia (JPTI)*, 1(7), 261-268.

Mas, R., Panca, R. W., Atmaja, K., & Yustanti, W. (2021). Analisis Sentimen Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from Transformers). Journal of Emerging Information Systems and Business Intelligence (JEISBI), 2(3), 227-238.

Mudding, A. A. (2024). Mengungkap Opini Publik: Pendekatan BERT-based-caused untuk Analisis Sentimen pada Komentar Film. *Journal of System and Computer Engineering (JSCE)*, 5(1), 36-43.

Pradipta, D., Kusrini, K., & Fatta, H. A. (2023). Sentiment analysis comments Covid-19 variant Omicron on social media Instagram with Bidirectional Encoder from Transformers (BERT). *JTECS: Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*, 3(1), 51-58.

Putu, N., Saraswati, V. D., Yudistira, N., & Adikara, P. (2023). Analisis Sentimen terhadap Perundungan Siber pada Twitter menggunakan Algoritma Bidirectional Encoder Representations from Transformer (BERT). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(2), 909-916.

Santoso, A. K. (2022). Analisis Sentimen Twitter Bahasa Indonesia Menggunakan Pendekatan Machine Learning. *Jurnal Informatika Kaputama (JIK)*, 6(2), 129–136.

Septian, I. F., & Kharisma, I. L. (2023). Implementasi Metode Bidirectional Encoder Representations from Transformers (BERT) untuk Analisis Sentimen Komentar Pengguna Aplikasi Dana di Instagram. Dalam *Seminar Nasional Rekayasa, Sains dan Teknologi*. Vol. 3, 45-55. Universitas Gunadarma.

Sholikhah, M., & Anjani, M. D. (2023). Kebijakan Pemerintah dalam Mengatasi Kenaikan Harga Beras di Indonesia. *Journal of Economics and Social Sciences (JESS)*, 2(2), 122-130.

Sofiana, S. (2024). *Konsep BERT pada Natural Language Processing*. Eureka Media Aksara.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1253.

Zukhrufillah, I. (2018). Gejala Media Sosial Twitter sebagai Media Sosial Alternatif. *Al-I'lam: Jurnal Komunikasi dan Penyiaran Islam*, 1(2), 102-109.