**Analisis Sentiment Pada Ulasan Vision+ Di Google Play Store Menggunakan Metode Algoritma Bidirectional Encoder Representations from Transformers (BERT)**

**BAB 1: PENDAHULUAN**

* 1. **Latar Belakang**

Dalam era digital yang semakin berkembang pesat, penggunaan teknologi untuk memenuhi kebutuhan hiburan semakin meningkat. Salah satu bentuk hiburan yang sedang populer saat ini adalah platform streaming, yang memungkinkan pengguna untuk menonton konten video secara online. Beberapa platform streaming yang terkenal di Indonesia antara lain Netflix, iFlix, Vidio, dan Vision+.

Vision+ merupakan salah satu platform streaming yang saat ini sedang berkembang pesat di Indonesia. Vision+ menawarkan berbagai jenis konten, seperti film, serial televisi, program acara, dan dokumenter. Namun, dengan semakin banyaknya pilihan platform streaming, persaingan antarplatform semakin ketat. Untuk mempertahankan eksistensinya, platform streaming perlu memperhatikan kualitas konten dan pengalaman pengguna yang disajikan.

Salah satu aspek penting dalam menilai kualitas pengalaman pengguna adalah analisis sentimen. Analisis sentimen merupakan teknik untuk mengekstrak dan memahami perasaan dan opini dari teks yang dituliskan oleh pengguna di media sosial, forum, atau platform online lainnya. Analisis sentimen dapat membantu platform streaming untuk memahami kebutuhan dan preferensi pengguna, sehingga dapat meningkatkan kualitas konten dan pengalaman pengguna.

Dalam konteks Vision+, analisis sentimen berbasis aspek dapat membantu platform untuk memahami bagaimana pengguna merespons konten yang disajikan. Analisis sentimen berbasis aspek merupakan teknik analisis sentimen yang memeriksa perasaan dan opini pengguna terhadap aspek tertentu dari konten dan juga user eksperien dari pengguna platform streaming. Misalnya, dalam konteks film, aspek yang diperiksa dapat meliputi akting, alur cerita, efek visual, kemudahan dalam navigasi aplikasi, kestabilan dari aplikasi dan sebagainya .

Oleh karena itu, penelitian ini akan fokus pada analisis sentimen berbasis aspek terhadap konten dan eksperien penggunaan dari aplikasi platform streaming Vision+. Tujuan dari penelitian ini adalah untuk memberikan pemahaman yang lebih dalam mengenai respon pengguna terhadap aspek-aspek tertentu dari konten dan user eksperien dari penggunaan platform streaming Vision+. Dengan demikian, platform streaming dapat meningkatkan kualitas konten dan pengalaman pengguna dan meningkatkan fitur yang ada di vision+ sehingga dapat mempertahankan eksistensinya di tengah persaingan yang semakin ketat di industri platform streaming.

* 1. **Rumusan Masalah**

Berdasarkan tujuan penelitian yang telah disebutkan sebelumnya, maka rumusan masalah yang akan dijawab dalam penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan analisis sentimen terhadap konten Vision+ menggunakan Bidirectional Encoder Representations from Transformers (BERT)?
2. Bagaimana respon pengguna terhadap aspek-aspek tertentu dari konten Vision+ dalam analisis sentimen berbasis aspek?
3. Apa saja faktor-faktor yang mempengaruhi respon pengguna terhadap konten Vision+ dalam analisis sentimen berbasis aspek?
4. Bagaimana penerapan analisis sentimen berbasis aspek dapat membantu meningkatkan kualitas konten dan pengalaman pengguna di Vision+?

Dengan menjawab rumusan masalah di atas, diharapkan penelitian ini dapat memberikan wawasan dan pemahaman yang lebih mendalam tentang analisis sentimen berbasis aspek dalam konteks platform streaming, khususnya Vision+.

* 1. **Tujuan Penulisan**

Tujuan dari penelitian analisis sentimen adalah sebagai berikut:

1. Untuk melakukan analisis sentimen terhadap konten Vision+ dengan menggunakan metode algoritma Bidirectional Encoder Representations from Transformers (BERT).
2. Untuk memahami respon pengguna terhadap respon positive dan negative dari konten Vision+.
3. Untuk mengukur tingkat akurasi metode algoritma Bidirectional Encoder Representations from Transformers (BERT).
4. Untuk mengevaluasi penerapan analisis sentimen dalam meningkatkan kualitas konten dan pengalaman pengguna di Vision+.

Dengan tujuan tersebut, penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan pemahaman tentang penggunaan teknik analisis sentimen berbasis aspek untuk meningkatkan kualitas konten dan pengalaman pengguna di platform streaming. Selain itu, penelitian ini juga diharapkan dapat memberikan rekomendasi dan masukan bagi platform Vision+ dalam mengoptimalkan konten dan pengalaman pengguna.

* 1. **Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat akademis, yaitu sebagai referensi atau sumber penelitian bagi peneliti lain yang tertarik dalam bidang analisis sentimen, khususnya dalam konteks platform streaming.
2. Manfaat praktis, yaitu sebagai masukan atau rekomendasi bagi platform Vision+ dalam meningkatkan kualitas konten dan pengalaman pengguna.
3. Manfaat bagi pengguna platform Vision+, yaitu dalam meningkatkan kualitas layanan dan pengalaman pengguna dalam menikmati konten di Vision+.
4. Manfaat bagi industri platform streaming, yaitu sebagai wawasan dalam menerapkan teknik analisis sentimen berbasis aspek untuk meningkatkan kualitas layanan dan pengalaman pengguna di platform streaming.
5. Manfaat dari tim internal Vision+ yaitu bisa dijadikan masukan kepada tim product dan engineering untuk melakukan improvement dari aplikasi vision+ agar lebih berkembang dan fitur aplikasi lebih relevan dengan kebutuhan user saat ini
   1. **Batasan Penelitian**

Pada subbab ini peneliti menyatakan Batasan serta ruang lingkup yang menjadi titik fokus dari penelitian tugas akhir yang dilakukan peneliti. Yaitu diantaranya;

1. Mengklasifikasikan ulasan positif. netral dan negatif.
2. Penelitian ini menggunakan data ulasan dari google play yang menggunakan bahasa Indonesia
3. Data yang diambil menggunakan ulasan dari pengguna aplikasi Vision+ yang didapat dengan menggunakan package library google play scraper pada Python
4. Penelitian ini menggunakan data yang diambil dari ulasan aplikasi Vision+ dengan Teknik crawling data mencapai 8000 ulasan
   1. **Metodologi Penelitian**

Pada subbab ini peneliti menggunakan python sebagai alat utama dalam mengelola analisis sentiment berikut ini adalah tahapanya:

1. Koleksi Data

Data yang digunakan dalam penelitian ini akan dikumpulkan dari review google play store dari platform streaming Vision+ menggunakan teknik web scraping. Python memiliki beberapa library web scraping yang dapat digunakan, seperti BeautifulSoup , Scrapy dan Google play scraper.Untuk penelitian ini penulis menggunakan Google play scraper untuk memperoleh data mentah.

1. Pra-Pemrosesan Data

Data yang telah dikumpulkan akan melalui proses pra-pemrosesan, seperti menghilangkan duplikasi data, membersihkan data dari karakter-karakter yang tidak diperlukan, serta memfilter data yang tidak relevan dengan penelitian.

1. Analisis Sentiment

Untuk melakukan analisis sentimen, penelitian ini akan menggunakan teknik Natural Language Processing (NLP) dengan Python. Terdapat beberapa library NLP yang dapat digunakan, seperti NLTK dan SpaCy. Teknik ini akan digunakan untuk memisahkan kalimat-kalimat dalam data ke dalam token-token dan kemudian mengklasifikasikan setiap token ke dalam aspek yang sesuai.

1. Pembangunan Model Analisis Sentimen

Model analisis sentimen akan dibangun dengan menggunakan algoritma Machine Learning. Python memiliki banyak library Machine Learning yang dapat digunakan, seperti Scikit-Learn dan TensorFlow. Model ini akan dilatih dengan menggunakan data yang telah dikumpulkan dan telah melalui proses pra-pemrosesan, serta dilakukan validasi dengan menggunakan teknik k-fold cross-validation.

1. Evaluasi Model dan Visualisasi Hasil

Hasil dari analisis sentimen berbasis aspek akan dievaluasi dengan menggunakan beberapa metrik, seperti akurasi, presisi, recall, dan F1-score. Selain itu, hasil dari analisis akan divisualisasikan menggunakan library plotting Python, seperti Matplotlib dan Seaborn, untuk mempermudah pemahaman hasil oleh pembaca.

* 1. **Sistematika Penulisan**

Sistematika penulisan dijelaskan secara garis besar mengenai apa yang terdapat

pada ball-bab yang ada. Adapun sistematika penulisan terdiri dari

1. BAB 1 PENDAHULUAN

Bab ini berisi uraian latar belakang masalah, rumusan masalah, Batasan masalah, tujuan penelitian, metode dan sistematika penulisan.

1. BAB 2 TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori yang berkaitan dengan penelitian, seperti Analisis Sentimen, Neural Network, Bidirectional Encoder Represenlolions

from Transformers (BERT).

1. BAB 3 METODE PENELITIAN

Bab ini berisi tentang gambaran umum penelitian, sumber data dan tahapan-tahapan yang dilakukan dalam penelitian.

1. BAB 4 IMPLEMEMASI

Bab ini berisi tentang dari hasil pengolahan data. pengujian data, Serta hasil klasifikasi yang telah dilakukan.

BAB 5 PENUTUP

Bab ini berisi kesimpulan dari seluruh hasil penelitian yang merupakan jawaban

dari tujuan dan saran untuk pengembangan yang lebih baik di masa

**BAB 2: Tinjauan Pustaka**

* 1. **Crawling Data**

Crawling Data adalah sebuah proses pengambilan data yang telah tersedia secara online dan terbuka untuk umum yang bertujuan untuk mengumpulkan atau mengunduh data dari suatu database yang bertujuan untuk mengumpulkan atau mengunduh data dari suatu database (Eka Sembodo et al., 2016).

Proses crawling data ini menghasilkan sebuah informasi atau data yang telah kita cari pada worldwide web dan menyimpannya ke dalam file lokal. Data Crawling adalah program yang mengubungkan halaman web, kemudian mengunduh kontennya. Pada penerapannya Crawling Data digunakan untuk mencari suatu hal yang dibutuhkan, salah satunya mencari suatu data dengan penjelajahan target yang lebih luas.

* 1. **Aplikasi Streaming**

Aplikasi streaming adalah program atau aplikasi yang memungkinkan pengguna untuk menonton atau mendengarkan konten multimedia seperti film, acara televisi, musik, dan podcast secara online melalui internet tanpa harus mengunduh terlebih dahulu. Aplikasi streaming ini menggunakan teknologi streaming untuk memutar konten multimedia secara langsung dan memungkinkan pengguna untuk menikmati konten tersebut dalam waktu nyaris real-time.

Menurut Novita (2019), aplikasi streaming adalah suatu jenis aplikasi yang memungkinkan pengguna untuk menonton atau mendengarkan konten multimedia secara online dengan bantuan koneksi internet. Konten multimedia tersebut dapat berupa film, acara televisi, musik, dan podcast yang dapat diputar dengan bantuan teknologi streaming. Aplikasi streaming ini sangat populer dan banyak digunakan oleh masyarakat di seluruh dunia karena memudahkan mereka untuk menikmati konten multimedia tanpa harus membeli atau mendownload konten tersebut terlebih dahulu.

Selain itu, menurut Rianto (2018), aplikasi streaming juga memungkinkan pengguna untuk mengakses konten multimedia dari berbagai perangkat seperti smartphone, tablet, laptop, atau komputer, sehingga memudahkan mereka untuk menikmati konten multimedia kapan saja dan di mana saja.

Dapat disimpulkan bahwa aplikasi streaming merupakan suatu aplikasi yang memudahkan pengguna untuk menonton atau mendengarkan konten multimedia secara online melalui internet dengan bantuan teknologi streaming. Aplikasi ini sangat populer dan banyak digunakan oleh masyarakat di seluruh dunia karena kemudahannya dalam mengakses konten multimedia dari berbagai perangkat dan waktu yang fleksibel.

* 1. **Vision+**

Vision+ adalah platform layanan streaming video on-demand (VOD) yang dikembangkan oleh MNC Group, salah satu perusahaan media terkemuka di Indonesia. Aplikasi ini menyediakan akses ke berbagai konten video seperti film, serial TV, dokumenter, dan program hiburan dari MNC Group dan mitra bisnisnya.

Dalam aplikasi Vision+, pengguna dapat menonton konten video kapan saja dan di mana saja dengan menggunakan perangkat mobile atau tablet yang terhubung ke internet. Aplikasi ini juga menyediakan fitur-fitur seperti fitur pencarian, kategori, rekomendasi, dan nonton offline (download). Aplikasi Vision+ tersedia untuk diunduh secara gratis di Google Play Store dan App Store. Namun, untuk menikmati layanan streaming video, pengguna harus berlangganan layanan berbayar yang disediakan oleh Vision+.

* 1. **Information Technology**

(Brian K. Williams & Stacey C. Sawyer., 2007) mengemukakan pendapat bahwa, teknologi informasi merupakan suatu teknologi yang merupakan hasil gabungan dari jalur komunikasi dengan kecepatan tinggi dengan teknologi komputasi. Menurut pendapat Martin(1999), teknologi informasi tidak hanya memiliki keterbatasan terhadap teknologi komputer yang meliputi perangkat lunak yang biasa disebut software dan perangkat keras atau yang bisa disebut hardware yang digunakan sebagai alat untuk menyimpan serta memproses informasi, melainkan juga mencangkum suatu teknlogi komunikasi yang digunakan untuk mengirimkan suatu informasi ataupun menerima informasi

* 1. **Text Mining**

Text mining merupakan proses menambang data yang berupa teks dimana sumber data secara umum biasanya didapat dari suatu dokumen serta tujuannya adalah mencari kata yang dapat mewakili isi dari suatu dokumen tersebut sehingga dapat dilakukan sebuah analisis yang memiliki keterhubungan antar dokumen tersebut (Zaki Izzani Akbar, 2021). Menurut (Ronen & James, 2007)

Text mining merupakan metode penambangan yang dilakukan oleh komputer dengan tujuan mendapatkan ilmu atau hal baru yang tidak diketahui sebelumnya serta menemukan Kembali informasi yang telah tersebar yang secara otomatis terekstrak dari sumber sumber data teks yang berbeda-beda Text mining merupakan salah satu Teknik yang dapat digunakan untuk melakukan klasifikasi. Teknik ini menemukan pola menarik dari suatu pengumpulan data tekstual (Santi et al., 2020). Text mining mirip dengan data mining. Perbedaannya, Teknik yang dirancang untuk bekerja pada data yang terstruktur atau semi terstruktur.

* 1. **Analisis Sentiment**

Analisis sentiment adalah suatu proses mengestraksi, mengolah, serta memahami suatu data yang berbentuk teks yang tidak terstruktur secara otomatis yang bertujuan untuk mengambil suatu informasi sentiment yang terdapat pada sebuah kalimat pendapat atau opini terhadap suatu topik baik negatif maupun positif (Imam Fahrur Rozi et al., 2012). Metode pemrosesan ini dapat diterapkan pada semua bidang opini yang ada. Diantaranya pada topik ekonomi, politik, sosial ataupun hukum.

*Sentiment analysis* atau analisis sentimen dalam bahasa indonesia adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen negative (Muchammad Shiddieqy Hadna et al., 2016). Selain itu (Medhat et al., 2014) mengemukakan bahwa Sentiment Analysis atau Opinion Mining merupakan sebuah studi yang mempelajari mengenai pendapat, sikap, dan emosi seseorang terhadap suatu entitas yang dapat mewakili suatu individu, acara atau suatu topik tertentu. Analisis sentimen mencakup deteksi, analisis, dan evaluasi keadaan pikiran manusia terhadap berbagai peristiwa, masalah, layanan atau minat lainnya (Steven, 2020) Lebih tepatnya, bidang ini bertujuan untuk menggali pendapat, sentimen dan emosi berdasarkan pengamatan orang-orang yang bisa didapatkan melalui tulisan, ekspresi wajah, ucapan, musik, gerakan, dan lain sebagainya (Steven, 2020)

Tujuan dari analisis sentimen sendiri adalah untuk menemukan pendapat, mengidentifikasi sentimen yang mereka ungkapkan, dan kemudian mengklasifikasikan polaritasnya (Medhat et al., 2014). Dengan kata lain, analisis sentimen berfungsi untuk mengklasifikasikan teks kedalam kelas positif, negatif atau netral (Steven, 2020) Beberapa pendapat mengenai analisis sentimen dapat diambil kesimpulan bahwa analisis sentimen adalah suatu proses untuk menentukan sentimen atau opini dari seseorang yang biasanya di wujudkan dalam bentuk teks dan bisa dikategorikan sebagai sentimen positif atau negatif (Muchammad Shiddieqy Hadna et al., 2016)***.***

* 1. **Neural Network**

Neural Network atau jaringan syaraf tiruan adalah salah Satu teknik machine

learning yang vuvuler dengan mensimulasikan mekanisme 'Embelajaran yang

terinspirasi dari bagaimana cara sistem syaraf manusia atau makhluk biologis lainnya.Sistem Saraf terdiri dari sel yang disebut dengan neuron. Neuron-neuron tersebut Saling terhubung dengan satu sama lain axon dan dendrites. Sinapsis adalah penghubung antara axon dan dendriles (Aggarwal, 2018).

Network sendiri disebut sebagai arsitektur, di mana berbagai layers saling

terhubung dengan Satu sama lain. Layer yang ada di antara input layer dan Output layer disebut dengan hidden layers dan output dari hidden layer disebut dengan hidden units

(Osinga, 2018). Istilah hidden diberikan karena unit-unit tersebut tidak dapat langsung terlihat dari luar sebagai input atau output. Inti dari neural network adalah hidden layer yang dibentuk dari hidden units, yang masing-masing merupakan neural units,mengambil weighted sum dari input dan kemudian menerapkan non-lineariw.

Setiap unit pada tiap layer mengambil input dan Output dari semua unit di layer

dan hubungan antara setiap pasangan unit daridua layer yang saling berdekatan

sehingga setiap layer saling terhubung. Tiap hidden units akan menjumlahkan semua input unit (Jurafsky & Martin, 2019).Jaringan syarat tiruan memiliki beberapa parameter, seperti weights W, bias b yang dipelajari Oleh gradient descent. Sedangkan hyperparameters adalah parameter yang dipilih Oleh desainer algoritma nilai optimal yang diatur pada sebuah devser,bukan dengan sebuah gradient descent di dataset training. Hyperparameters mencakup learning rate n, mini-barch size, arsitektur model (jumlah layer, jumlah hidden node tiap layer, fungsi aktivasi yang dipilih) dan lain-lain (Jurafsky & Maltin, 2019).

Gradient descent (penurunan gradien) adalah sebuah algoritma optimasi yang

digunakan ketika melatih sebuah model machine learning. Gradient descent hanya

digunakan untuk menemukan nilai parameter fungsi (koefisien) yang meminimalkan

cost. Tujuan utama dari gradient descent adalah menemukan weight yang optimal

dengan meminimalisasi loss function. Seberapa besar step yang dilakukan gradient

descent ditentukan Oleh learning rare yang menunjukkan seberapa cepat atau lambat

bergeraknya fungsi ke weight yang optimal. Terdapat tiga jenis gradient descent yang

sering digunakan, yaitu Balch Gradient Descent, Stochastic Gradient Descent, dan

Mini -Batch Gradient Descent.

1. Batch gradient descent (BGD) atau vanilla gradient descent menghitung error dari

tiap contoh yang ada pada dataset training. proses ini seperti sebuah siklus yang

disebut juga training epoch. Semua cost dari training pada daraser dihitung.

Algoritma ini memiliki kelebihan yaitu dapat membuat jaringan syaraf tiruan

lebih cepat, menghasilkan gradien kesalahan yang stabil, Akan tetapi

gradien kesalahan yang stabil dapat pula membuat model tidak dapat menghasilkan

kondisi konvergensi yang terbaik\_ Selain itu,seluruh training dataset harus berada

dalam memori.

1. Stochastic gradient descent (SGD) adalah algoritma yang meminimalkan loss

function dengan menghitung gradiennya tiap kali 'raining dilakukan (Jurafsky &

Martin, 2019). Algoritma ini disebut stokastik karena hanya memiliki Satu contoh

acak dalam satu waktu, memindahkan weight-nyasehingga dapat meningkatkan

Irrforma dari contoh tersebut. Sehingga, SGD akan melakukan proses tersebut pada

setiap contoh dataset Satu per Satu. Weight yang terus diperbarui memungkinkan

untuk mendapatkan peningkatan yang cukup detail. Akan tetapi pembahaman yang

terlalu sering dilakukan lebih mahal dan dapat menyebabkan gradient yang tidak rata karena error rare dapat naik turun.

1. Mini barch gradient descent menggabungkan konsep dari SGD dan BGD. Mini

batch adalah pruses training sekumpulan dataset (umumnya512 atau 1024) yang

lebih sedikit dari dataset asli. Algoritma ini akanmembagi training dataset menjadi

batch-barch yang kecil dan memperbarui weight dari masing-masing batch.

Salah Satu algoritma optimasi yang sering digunakan adalah Adam. Adam atau

Adaptive Momenl Eslimarion Algorithm (Kingma & Ba, 2015) menghitung estimasi

momen dan menggunakannya untuk mengoptimmsi fungsi. Adam merupakan

kombinasi dari dua algoritma yaitu AdaGrad dan RMSProp. AdaGrad akan

mempertahankan learning rare per parameter yang meningkatkan peforma pada

masalah dengan gradien yang renggan dan RMSProp juga mempertahankan learning

rare per parameter yang diadaptasi berdasarkan rata-rata besaran gradien untuk weight

(s&rapa cepat berubah). Algoritma ini akan menghitung rata-rata eks1Nmensial

weighted bergerak kemudian mengkuadratkan gradien yang dihitung. Alih•alih

mengadaptasi learning rate parameter berdasarkan rata-rata momen Ikrtama seperti di

RMS Prop, Adam juga menggunakan rata-rata momen kedua. Dengan menggunakan

model dan dataset yang besar, Adam dapat secara efektif menyelesaikan masalah•

masalah deep learning praktikal (Kingma & Ba, 2015). Hal ini membuat Adam

optimizer menjadi metode yang efisien secara komputasi, membutuhkan sedikit

memori, tidak berbeda dengan gradien dengan skala diagonal, cocok untuk masalah

yang menggunakan data atau parameter yang besar (Kingma & Ba, 2015).

* 1. **Artificial Neural Network ANN**

Artificial neural network (ANN) atau jaringan saraf tiruan adalah model matematis yang terinspirasi dari sistem saraf biologis. ANN terdiri dari jaringan neuron buatan (artificial neuron) yang terhubung secara paralel, dan mampu melakukan pemrosesan informasi melalui propagasi sinyal yang disebut "forward pass" dan "backward pass".

Setiap neuron dalam ANN menerima masukan (input) dari neuron lain atau dari luar, dan kemudian menghasilkan keluaran (output) berdasarkan suatu fungsi aktivasi yang dapat berbeda-beda, seperti fungsi sigmoid atau fungsi ReLU. Hubungan antara neuron-neuron dalam ANN dapat direpresentasikan oleh bobot (weight), yaitu nilai numerik yang menentukan seberapa besar pengaruh suatu neuron terhadap neuron lainnya.

Proses pembelajaran (learning) ANN terjadi melalui pengaturan bobot dengan tujuan untuk meminimalkan suatu fungsi error. Terdapat berbagai macam algoritma pembelajaran ANN, seperti backpropagation, gradient descent, dan stochastic gradient descent, yang digunakan untuk menentukan perubahan bobot yang optimal.

ANN digunakan dalam berbagai aplikasi seperti klasifikasi, prediksi, pengenalan pola, dan optimasi. Contoh penggunaan ANN antara lain dalam pengenalan wajah, deteksi objek, dan prediksi harga saham.

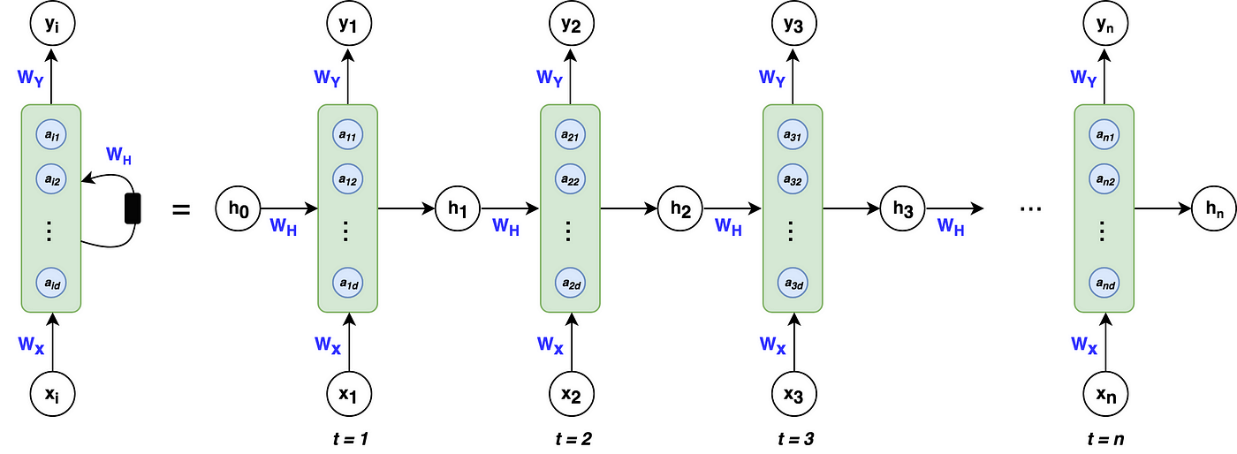
* 1. **Recurrent Neural Network**

Recurrent neural network (RNN) atau jaringan saraf rekuren adalah model ANN yang dirancang untuk mengolah data sekuensial, seperti teks atau waktu. RNN memungkinkan informasi dari masa lalu untuk mempengaruhi pengambilan keputusan di masa sekarang melalui penggunaan sel memori (memory cell) dan loop feedback (kembali).

RNN terdiri dari serangkaian sel memori (memory cell) yang terhubung ke neuron-neuron lainnya. Setiap sel memori menghasilkan output berdasarkan input sekarang dan output sebelumnya. Output sebelumnya disimpan dalam sel memori, dan digunakan untuk mempengaruhi pengambilan keputusan selanjutnya.

Salah satu jenis RNN yang populer adalah LSTM (Long Short-Term Memory) yang memungkinkan sel memori untuk menyimpan informasi dalam jangka panjang dan jangka pendek. Hal ini memungkinkan LSTM untuk lebih efektif mengolah data sekuensial, terutama pada kasus di mana ada jarak waktu yang cukup jauh antara suatu kejadian dan kejadian berikutnya.

Penggunaan RNN terutama berfokus pada pengolahan data sekuensial seperti teks, waktu, dan suara. Beberapa aplikasi dari RNN antara lain terjemahan mesin, pengenalan suara, dan analisis sentimen.Berikut ini adalah gambaran dari Recurrent Neural Network (RNN)



Gambar 2.1 Recurrent Neural Network

* 1. **Machine Learning**

Machine learning (ML) adalah bidang ilmu komputer yang mempelajari algoritma untuk membuat mesin dapat belajar dari data, tanpa perlu diprogram secara eksplisit. Dalam ML, mesin belajar dengan mengidentifikasi pola pada data yang diberikan, dan kemudian menggunakannya untuk membuat prediksi atau mengambil tindakan.

Ada tiga jenis utama dari machine learning:

1. Supervised learning, di mana model belajar dari data yang diberi label dan melakukan prediksi untuk data baru.
2. Unsupervised learning, di mana model belajar dari data tanpa label dan mencoba untuk mengidentifikasi pola atau klasifikasi pada data tersebut.
3. Reinforcement learning, di mana model belajar melalui interaksi dengan lingkungannya dan mendapatkan reward atau hukuman berdasarkan tindakan yang diambil.

ML telah digunakan secara luas dalam berbagai aplikasi, termasuk pengenalan wajah, klasifikasi teks, deteksi spam, dan pengenalan suara. Peningkatan teknologi komputer dan perkembangan algoritma ML, seperti deep learning, telah memungkinkan mesin untuk menyelesaikan tugas yang lebih kompleks dan lebih presisi.

* 1. **Deep Learning**

Deep learning merupakan cabang dari machine learning yang masih menjadi

bagian dari kecerdasan buatan. Deep learning merupakan neural network yang lebih modern dan bersifat deep atau mendalam karena memiliki jauh lebih banyak layer dibandingkan dengan dengan neural network biasanya (Jurafsky & Martin, 2010) (Chollet. 2018). Deep learning bekerja untuk sehingga tidak hanya dapat memprediksi tetapi juga merepresentasikan data dengan benar, sehingga cocok untuk melakukan prediksi (Goldberg, 2017). Deep learning dapat dibagi kedalam tiga metode pendekatan yaitu supervised, semi-supervised dan unsupervised learning. Deep learning didukung Oleh banyak framework seperti TensorFlow, Torch dan Theano.

* 1. **Bidirectional Encoder Representations from Transformers (BERT)**

Bidirectional Encoder Representations from Transformers atau disingkat BERT

adalah suatu model representasi bahasa terlatih yang dikembangkan Oleh para peneliti di Google AI Language pada tabun 2018. BERT merupakan sebuah teknik open source yang dikembangkan berdasarkan teknik deep learning dan berbagai metode seperti semi-supervised learning, ELMO, OpenAI Transformer dan Transformer. Sesuai dengan namanya, BERT menggunakan Transformer. Transformer adalah mekanisme yang hubungan kontekstual antara kata-kata dalam teks (Vaswani et al. 2017). Transformer dapat memahami dan mengkonversi yang diperoleh dengan mekanisme yang bernama self-attention mechanism. Self-arrention mechanism adalah cara Transformer untuk mengubah "pemahaman" kata terkait lainnya menjadi kata-kata yang akan diproses dengan mekanismenya. Terdapat dua mekanisme pada Transformer, yaitu:

1. Encoder

Encoder memiliki fungsi untuk membaca keseluruhan teks inputan sekaligus.

encoder terdiri dari stack N = 6 layers yang identik. Setiap layer memiliki dua sub

layer yaitu self-attention layer dan feed-forward neural network. Dengan self-

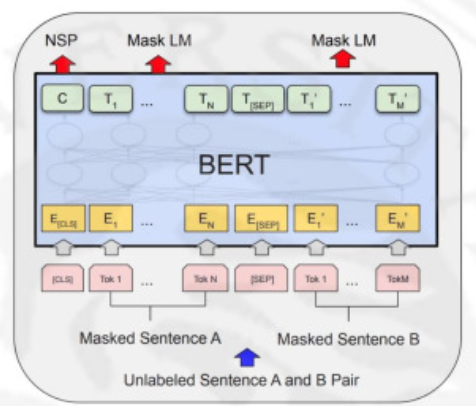
attention layer, encoder dapat membantu node untuk tidak hanya fokus kepada

kata yang sedang dilihat tetapi juga untuk mendapatkan konteks semantik dari

kata tersebut.

1. Decoder

Decoder berfungsi untuk menghasilkan urutan output yang benlpa prediksi.Decoder juga terdiri dari stack N = 6 layers yang identik. Setiap layer terdiri dari dua sub layer Seperti yang ada pada encoder dengan tambahan attention layer diantara dua layers tersebut untuk membantu node Saat ini mendapatkan key content yang membutuhkan attention (Vaswani et al. 2017) dengan melakukan multi-head attention pada output dari encoder. Sama dengan di encoder, self-attention layer di encoder membuat setiap posisi di decoder dapat menangani semua sebelumnya dan posisi saat itu.



Gambar 2.2. Arsitektur Model BERT

BERT di-training dengan dataset, dataset harus disesuaikan dengan representasi input yang dapat diterima Oleh BERT. Oleh karena itu dibutuhkan sebuah tokenizer yang lx-rtujuan untuk melakukan tokenisasi pada kalimat-kalimat dan menghasilkan input yang sesuai. Hal ini dilakukan karena BERT menggunakan vocabulary yang spesifik yang mana tergantung dengan model apa yang dipakai.Vocabulary dibuat dengan menggunakan sebuah model WordPiece. Vocabulary terdiri

dari semua karakter dan tambahan sekitar 30 kata-kata yang sering digunakan Serta

sub kata yang sering muncul pada corpus yang model latih. BERT menerima panjang yang tetap dan sama untuk setiap inputnya. BERT telah menentukan panjang urutan kalimat maksimumnya, yaitu 512 karena encoder pada Transformer hanya menghasilkan output dengan dimensi 512 saja. Jika panjang kalimat lebih dari panjang maksimum yang telah ditentukan, kalimat akan dikurangi (truncate).Sedangkan jika panjang urutan kalimat kurang dari panjang maksimum yang sudah ditentukan, kalimat akan ditambah dengan padding. Akan tetapi, panjang maksimum juga dapat ditentukan sendiri tergantung dengan kebutuhan dan tugas yang akan dilakukan. Proses mempersiapkan kalimat menjadi representasi input pada BERT dilakukan Oleh tokenizer, dengan langkah-langkah sebagai berikut:

1. Setiap kalimat ditokenisasi menjadi per kata atau sub kata menggunakan

Wordpiece. Untuk melakukan tokenisasi pada sebuah kata, tokenizer akan

memeriksa apakah tiap kata pada kalimat terdapat pada vocabulary. Jika tidakada,

tokenizer akan memecah kata menjadi sub-sub kata yang kemungkinan

kemunculannya pada vocabulary paling be.sar. Jika tokenizer juga tidak menemukan

sub kata pada vocabulary, kata tersebut dilrcah menjadi per karakternya. Akan

tetapi, jika semua kata diubah menjadi sub kata atau karakter individual, akan terjadi

overload Kata-kata yang tidak ada pada vocabulary akan diganti dengan token

[UNKI atau unknown. Namun jika semua kata diubah menjadi token tersebut,

banyak informasi yang akan hilang\_ Oleh kalena itu, kata-kata dapat dipecah

menjadi sub kata dengan simbol BERT melakukan ini karena dua hal yaitu

pertama untuk mem1Rrcepat processing dan mengurangi jumlah parameters yang

harus dilatih, dan kedua untuk mengatasi masalah out-of-vocabulary.

1. Setiap kalimat diberi token-token khusus yaitu [CLSI di awal kalimat dan [SEPI di

akhir kalimat. Token [CIS] menjadi indikator sebuah kalimat sekaligus

sentimennya Saat klasifikasi sentimen dilakukan. Token [SEP]adalah token yang

digunakan untuk memisahkan kalimat satu dengan kalimatselanjutnya. Kalimat

yang sudah diberi token khusus ini menjadi token embeddings.

1. Setelah itu kalimat-kalimat disesuaikan dengan panjang maksimum yang telah

ditentukan dengan mengurangi atau memberi padding dengan token khusus [PAD].

1. Kemudian setiap kalimat dicocokkan dengan bilangan unik atau ID sesuai dengan vocabulary dan bilangan unik tersebut disimpan sebagai token id. Bilangan unik atau ID ini didapatkan ketika model di-rraining, yang mana membuat setiap kata,sub kata, dan karakter pada vocabulary memilikibilangan uniknya masing-masing.ID tersebut diperoleh berdasarkan indeks kata pada vocabulary karena pada vocabulary disusun berdasarkan tingkat kemunculannya. Kata dan sub kata sendiri harus dikonversi menjadi ID-nya karena model pre-trained BERT hanya dapat mengerti ID dari token.
2. Sentence embedding diberi pada kalimat untuk membedakan kalimat pertama dan kalimat kedua atau padding. Tahap ini dapat dilakukan dengan memberi angka 1 pada kalimat lwrtama dan memberi angka 0 pada padding. Tokenizer dapat mengetahui mana kalimat dan mana padding dengan melihat token [SEPI] yang berfungsi memisahkan antara dua kalimat.
3. Positional embedding juga ditambahkan untuk tiap token untuk menunjukkan isi tiap kata pada kalimat\_ Tahap ini dilakukan karena BERT tidak mengetahui posisi dari tiap kata. Sehingga walaupun pada awal kalimat terdapat suatu kata dan di akhir kalimat juga terdapat kata yang sama tetapi memiliki makna yang berbeda, BERT tidak akan memperlakukan token tersebut dengan makna yang sama. BERT akan menerima unltan kata-kata atau sebuah kalimat sebagai input yangakan terus melalui tumpukan encoder. Tiap encoder mengaplikasikan self-attention dan memberikan output melalui feed-forward network yang kemudian dilanjutkan Oleh encoder selanjutnya. Pada penelitian ini, model yang dipilih adalah BERT-base sehingga proses ini berlanjut sebanyak dua belas Setelah melewati semua encoder, tiap token per posisi memberikan output vektor dengan ukuran hidden size yaitu 768 pada BERT-base. Untukproses analisis sentimen, output yang diperhatikan adalah output dari pertama yaitu token [CLSI. Vektor tersebut digunakan sebagai input untuk classifier. BERT dapat mencapai hasil yang baik hanya dengan menggunakan neural network tunggal sebagai classifieFnya (Devlin et al., 2019). Layer yang digunakan untuk klasifikasi adalah fully connected neural network dengan fungsi sofrmax. Sehingga output dari BERT yang digunakan untukklasifikasi berasal vektor token [CLSI] karena token [CIS] dianggap melakukan pengumpulan rata-rata ata.S token kata untuk mendapatkan vektor dari kalimat. Layer terakhir pada classifier layer menghasilkan logits. Logits adalah output yanglR-rupa prediksi probabilitas kasar dari kalimat yang akan diklasifikasikan. Softmax akan mengubah logits tersebut menjadi probabilitas dengan mengambil eksponen dari tiap nilai logit sehingga total probabilitasnya adalah tepat sehingga nilai probabilitas akan berada di antara 0 atau angka positif.
   1. **IndoNLU**

IndoNLU merupakan salah satu produk dari komunitas yang bernama *IndoBenchmark*. IndoNLU *Benchmark* adalah sekumpulan sumber daya untukmelatih, mengevaluasi, dan menganalisis system pemahaman bahasa alami untuk bahasa Indonesia. IndoNLU mencakup dua belas *tasks*, mulai dari klasifikasi kalimat tunggal hingga pelabelan urutan pasangan kalimat dengan tingkat kompleksitas yang berbeda. Kumpulan data untuk *tasks* terletak pada *domain* dan *styles* yang berbeda untuk memastikan keragaman *task*.

* 1. **IndoBERT-BASE**

IndoNLU mengikuti pengaturan *hiperparameter pre-training* BERT untuk melatih model IndoBERT terlebih dahulu. IndoBERT merupakan model berbasis BERT berbahasa Indonesia yang dilatih megggunakan dataset indo4B dengan 4 milyar kata bahasa Indonesia dan 20 GB ukuran data teks. Terdapat dua model IndoBERT berdasarkan ukurannya yaitu IndoBERTBASE dan IndoBERTLARGE. IndoBERT-base memiliki parameter sebesar 124,5M, 12 *layer*, 12 *head*, 768 *embedding size*, 728 *hidden size*, dan tipe *pre-train* dengan metode kontekstual.

* 1. **Confusion Matrix**

*Confusion Matrix* dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali *tuple* dari kelas yang berbeda. nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

Tabel 2.1 *Confusion Matrix*

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

Keenam istilah tersebut adalah :

* *True Positive* (TP), merupakan data positif yang diprediksi benar
* *True Negative* (TN), merupakan data negatif yang diprediksi benar
* T*rue Neutral* (TNt) adalah kalimat memiliki sentimen netral dan hasil prediksinya juga menunjukkan sentimen netral.
* F*alse Neutral* (FNt) adalah kalimat yang memiliki sentimen postif atau negatif tetapi hasil prediksinya menunjukkan sentimen entral
* *False Positive* (FP), merupakan data netral atau negatif namun diprediksi sebagai data positif
* *False Negative* (FN), merupakan data positif atau netral namun diprediksi sebagai data negatif

Beberapa hal yang dapat dilakukan oleh *Confusion Matrix* dalam hal perhitungan :

1. *Accuracy*

*Accuracy* memberikan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. nilai *accuracy* diperoleh dengan persamaan berikut :

1. *Precision*

*Precision* memberikan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Nilai *precision* diperoleh dengan persamaan berikut :

1. *Recall*

*Recall* memberikan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *Recall* diperoleh dengan persamaan berikut :

1. *F1 Score*

*F1 Score* merupakan perbandingan rata-rata presisi dan *recall* yang dibibitkan. Nilai *F1 Score* dapat diperoleh dengan persamaan berikut :

* 1. **Dataset**

Dataset adalah sekumpulan data yang bersifat sebagai himpunan data yang berasal dari informasi-informasi pada masa sebelumnya dan siap untuk di kelola menjadi sebuah informasi baru dengan menggunakan teknik pembelajaran *supervised learning*.Dimana digunakan untuk sistem prediksi sebagai acuan pendukung keputusan (Khalimi, 2020).

* 1. **Web Scrapping**

Web scraping mempakan sebuah metode pengumpulan data melalui internet,

metode web scraping populer diigunakan dalam lymenuhan data mining. Menurut teori, web scraping adalah cara untuk mengumpulkan data menggunakan melode yang berbeda dengan penggunaan API (*Application programming Interfaces*). Cara seperti ini biasanya dimulai dengan penulisna kode program, dimana digunakan sebagai otomatisasi *query* untuk melakukan *request* data terhadap server. Data hasil tersebut dapat dilakukan ekstrasi untuk menghasilkan informasi yang akan dicari (Sahrian. 2020).

* 1. ***Case Folding***

Case folding merupakan suatu bentuk tert preprocessing yang sederhana dan

efektif meskipun sering diabaikan. Case folding memiliki tujuan untuk mengubah

semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampa 'z' yang

diterima. Karakter selain huruf dihilangkan dan dianggap delimiter. Ada beberapa cara yang dapat digunakan dalam tahap case folding, anda dapat menggunakan atau semuanya, tergantung pada tugas yang diberikan (Nugroho, 2019).

* 1. ***Dana Cleaning***

Data Cleaning merupakan tahap pembersihan dengan kata-kata yang tidak perlu sehingga dapat meningkatkan akurasi dari analisis yang akan dilakukan. Proses yang dilakukan dalam tahap cleansing adalah sebagai berikut:

1. Menghapus https
2. Menghapus mention
3. Menghapus hashtags
4. Menghapus tanda baca
5. Menghapus angka
6. Menghapus spasi berlebih
7. Menghapus kalimat duplikat.
   1. **Normalisasi Teks**

Normalisasi teks merupakan proses pengolahan teks yang bertujuan untuk

mengubah struktur atau bentuk teks yang asalnya sulit dimengerti komputer hingga

akhirnya mudah dimengerti dan diolah lebih lanjut. Normalisasi teks meliputi

pengubahan singkatan, akronim, kesalahan penulisan dan penggunaan kata tidak baku.

* 1. **Dataset Splitting**

Dataset splitting merupakan suatu metode yang digunakan untuk mengevaluasi

performa model. Metode ini membagi dataset menjadi beberapa bagian yakni bagian

yang digunakan untuk training data, validation data dan untuk testing data dengan

proporsi tenentu. Dataset training digunakan untuk melatih model, dataset validasi

digunakan untuk meminimalisir overfilling yang sering terjadi pada jaringan syaraf turuan sedangkan dataset testing sendiri digunakan sebagai test akhir untuk melihat keakuratan jaringan yang sudah dilatih dengan dataset training

* 1. **Python**

*Python* adalah bahasa pemrograman tingkat tinggi yang banyak digunakan akhir-akhir ini. dengan filosofi dimana kode mudah dibaca dan sintaksnya mengizinkan *programmer* untuk mengekspresikan konsep dalam kode baris yang lebih sedikit daripada di bahasa pemrograman C. kontruksi bahasa memungkinkan pengguna untuk menulis program secara jelas baik dalam skala kecil maupun besar.

Fitur terpenting dalam *python* adalah mendukung beberapa paradigma pemrograman, termasuk berorientasi objek, pemograman atau prosedural imperatif dan fungsional gaya. *Python* mendukung sistem tipe dinamis dan manajemen memori otomatis dan memiliki besar dan perpustakaan standar yang komprehensif. Penerjemah *python* tersedia untuk banyak sistem operasi.

* 1. **Google Colaboratory**

Google Colab atau Google Colaboratory, adalah sebuah *executable document* yang dapat digunakan untuk menyimpan, menulis, serta membagikanprogram yang telah ditulis melalui Google Drive. *Software* ini pada dasarnya serupa dengan Jupyter Notebook gratis berbentuk *cloud* yang dijalankan menggunakan *browser*, seperti Mozilla Firefox dan Google Chrome. Google Colabmemungkinkan user untuk menjalankan kode Python tanpa perlu melakukan proses instalasi dan setup lainnya karena semua keperluan *setting* dan *adjustment* akan diserahkan ke cloud (Glints.com, 2021).

Google Colaboratory dapat menulis dan mengeksekusi Python di *browser* dengan beberapa keuntungan yaitu tidak memerlukan konfigurasi, akses gratis ke GPU, berbagi dengan mudah serta dapat menggabungkan kode yang dapat dijalankan dan *rich text* dalam satu dokumen, beserta gambar, HTML, LaTeX, dan lainnya (*colab.research.google.com*). Dengan sifat yang fleksibel Google Colab dapat membuat *notebook* Colab oleh *user* itu sendiri, *notebook* tersebut akan disimpan di akun Google Drive pengguna. Pengguna dapat dengan mudah membagikan *notebook* Colab dengan pengguna lainnya, sehingga dapat saling memberikan komentar di *notebook* pengguna satu sama lainnya atau bahkan mengeditnya.

* 1. **Penelitian Terkait**

Penelitian ini dilakukan dengan mempelajari penelitian terdahulu yang memiliki keterkaitan tentang analisis sentimen. Beberapa penelitian terdahulu yang digunakan sebagai dasar pembelajaran pada Tabel 2.1

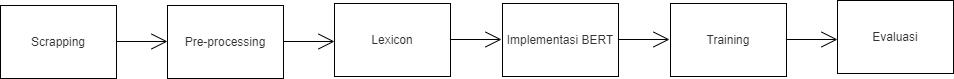
Tabel 2.1 Penelitian Terkait

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama | Tahun | Judul | Hasil |
| 1 | Cindy Alifia  Putri,  Adiwijaya,  Said Al Faraby | 2020 | Analisis Sentimen  Review Film  Berbahasa Inggris  dengan Pendekatan  Bidirectional Encoder  Representations from  Transformers | Akurasi yang didapatkan  dari analisis sentimen  menggunakan metode  BERT sebesar 73%.  dataset yang digunakan  berbentuk dokumen.  BERT-base terbukti |
| 2 | Raden Mas  Rizi Wahyu  Panca Kusuma  Atmaja, Wily  Yustanti | 2020 | Analisis Sentimen  Customer Review  Aplikasi Ruang Guru  dengan Metode  BERT(Bidirectional  Encoder  Representations from  Transformers) | Akurasi yang didapatkan  dari analisis sentimen  menggunakan metode  BERT sebesar 99%  dengan nilai F1 Score  sebesar 98.9%.  menggunakan 5437  testing data dimana 5254  komentar positif, 16  netral dan 167 komentar  negatif. |
| 3 | Muhammad  Mahrus Zain,  Rizky  Nathamael  Simbolon,  Harlem  Sulung, Zaidan  Anwar | 2021 | Analisis Sentimen  Pendapat Masyarakat  Mengenai Vaksin  Covid-19 Pada Media  Sosial Twitter dengan  Robustly Optimized  BERT Pretraining  Approach | Akurasi yang didapatkan  dari analisis sentimen  menggunakan metode  RoBERT sebesar 95%.  dimana rata-rata hasil  akurasi prediksi pada  label positif 84%, Netral  97% dan Negatif 93% |

**Bab 3 Metodologi Penelitian**

* 1. **Tahapan Penelitian**

Tahapan penelitian analisis sentimen yang dilakukan melalui komentar di Google Playstore pada aplikasi Vision+ untuk melihat bagaimana sentimen atau pendapat masyarakat mengenai aplikasi tersebut. Analisis sentimen ini menggunakan metode BERT untuk mengklasifikasikan data komentar yang mengandung kata Vision+ ke dalam tiga kategori, yaitu positif, netral, dan negatif. Diagram proses dari analisis sentimen dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

* 1. **Scrapping**

*Scrapping* atau pengambilan data dilakukan dengan metode *scraping* data di platform Google Play Store menggunakan *python script* dengan *library google* *play scraper*. *Scraping* dilakukan untuk mendapatkan komentar pengguna Disney+ Hotstar yang tertulis pada platform Google Play Store seperti pada contoh berikut:

#scrape jumlah ulasan yang diinginkan

from google\_play\_scraper import Sort, reviews

result, continuation\_token = reviews(

'com.zte.iptvclient.android.idmnc',

lang='id', #disini kita mau men scrape data ulasan aplikasi shopee yang berada di google play store

country='id', #kita setting bahasa nya menjadi bahasa indonesia

sort=Sort.MOST\_RELEVANT, # # kemudian kita gunakan most\_relevan untuk mendapatkan ulasan yang paling relevant

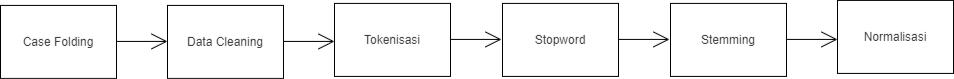
count=8000, # disini jumlah ulasan yang mau kita ambil ada seribu

filter\_score\_with=None # # kemudian di filter\_score kita gunakan None untuk mengambil semua score atau ratting bintang 1 sampai 5

)

* 1. **Preprocessing**

*Pre-processing* dilakukan untuk membersihkan data komentar, tahap ini membuat data yang tidak baik atau belum siap diolah menjadi data yang siap diubah. tahap ini terdiri dari beberapa tahapan yaitu *case folding*, *data cleaning*, tokenisasi, *stopword*, *stemming* dan normalisasi kata tidak baku. Diagram proses dari *pre-processing* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Proses *Preprocessing*

* + 1. Case Folding

*Case folding* merupakan tahap merubah semua huruf besar yang ada pada dokumen (uppercase) menjadi huruf kecil (lowercase). case folding dilakukan menggunakan fungsi *lower()* yang ada di *library python*. Perintah untuk melakukan case folding adalah sebagai berikut:

df['content'] = df['content'].str.lower()

Tabel 3.1 Contoh Hasil *case folding*

|  |  |
| --- | --- |
| Komentar | Hasil Case folding |
| Versi yg skrg byk bugnya. Ndk bs streaming video dengan lancar, walaupun sdh beli paket premium. Parah bgt dah... | versi yg skrg byk bugnya. ndk bs streaming video dengan lancar, walaupun sdh beli paket premium. parah bgt dah... |
| Parah nih aplikasi udh buffering Mulu, terus mau nonton live tv emang harus beli paket dulu yah? lawak sih klo emng gitu pelit bener gak kayak aplikasi video | parah nih aplikasi udh buffering mulu, terus mau nonton live tv emang harus beli paket dulu yah? lawak sih klo emng gitu pelit bener gak kayak aplikasi video |
| Kenapa sekarang app tidak dapat di gunakan sama sekali. Sinyal cuma muter-muter buffering terus | kenapa sekarang app tidak dapat di gunakan sama sekali. sinyal cuma muter-muter buffering terus |
| Lemot banget, saat sy tulis ulasan ini pun film nya tidak bisa diputar, padahal sudah berlangganan premium , lalu suka tiba2 putus ditengah film, mengecewakan untuk ukuran aplikasi berbayar | lemot banget, saat sy tulis ulasan ini pun film nya tidak bisa diputar, padahal sudah berlangganan premium , lalu suka tiba2 putus ditengah film, mengecewakan untuk ukuran aplikasi berbayar |

* + 1. Data Cleaning

*Data Cleaning* merupakan tahap menghilangkan angka, simbol, url, at (@)

* hashtag (#), spasi berlebih, tanda baca, emoji, dan pengulangan karakter yang ada pada kalimat. Perintah untuk melakukan cleaning adalah sebagai berikut:

def clean\_review(text):

text = text.lower()

text = re.sub(r'\n', ' ', text)

text = emoji.demojize(text)

text = re.sub(':[A-Za-z\_-]+:', ' ', text)

text = re.sub(r"([xX;:]'?[dDpPvVoO3)(])", ' ', text)

text = re.sub(r"(https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|www\.[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9]+\.[^\s]{2,}|www\.[a-zA-Z0-9]+\.[^\s]{2,})", "", text)

text = re.sub(r"@[^\s]+[\s]?", ' ', text)

text = re.sub(r'#(\S+)', r'\1', text)

text = re.sub('[^a-zA-Z,.?!]+',' ',text)

text = repeatcharClean(text)

text = re.sub('[ ]+',' ',text)

return text

Tabel 3.2 Contoh Hasil *Data Cleaning*

|  |  |
| --- | --- |
| Komentar | Hasil Data Cleaning |
| Versi yg skrg byk bugnya. Ndk bs streaming video dengan lancar, walaupun sdh beli paket premium. Parah bgt dah... | versi yg skrg byk bugnya ndk bs streaming video dengan lancer walaupun sdh beli paket premium parah bgt dah |
| Parah nih aplikasi udh buffering Mulu, terus mau nonton live tv emang harus beli paket dulu yah? lawak sih klo emng gitu pelit bener gak kayak aplikasi video | parah nih aplikasi udh buffering mulu terus mau nonton live tv emang harus beli paket dulu yah lawak sih klo emng gitu pelit bener gak kayak aplikasi video |
| Kenapa sekarang app tidak dapat di gunakan sama sekali. Sinyal cuma muter-muter buffering terus | kenapa sekarang app tidak dapat di gunakan sama sekali sinyal cuma muter buffering terus |
| Lemot banget, saat sy tulis ulasan ini pun film nya tidak bisa diputar, padahal sudah berlangganan premium , lalu suka tiba2 putus ditengah film, mengecewakan untuk ukuran aplikasi berbayar | lemot banget saat sy tulis ulasan ini pun film nya tidak bisa diputar padahal sudah berlangganan premium lalu suka tiba2 putus ditengah film mengecewakan untuk ukuran aplikasi berbayar |

* + 1. **Tokenisasi**

Tokenisasi merupakan tahapan untuk menemukan karakter yang akan dihapus. tokenisasi digunakan untuk memecah kalimat menjadi daftar kata. tahapan ini dijalankan menggunakan “word\_tokenize” dari library NLTK. Perintah untuk melakukan tokenisasi adalah sebagai berikut:

def normalize\_review(text):

# tokenize

list\_text = word\_tokenize(text)

# ubah bahasa alay

list\_text = [normalize\_word\_dict[term] if term in normalize\_word\_dict else term for term in list\_text]

Tabel 3.3 Contoh Hasil *Tokenisasi*

|  |  |
| --- | --- |
| Komentar | Hasil Case folding |
| Versi yg skrg byk bugnya. Ndk bs streaming video dengan lancar, walaupun sdh beli paket premium. Parah bgt dah... | “versi” “yg” “skrg” “byk” “bugnya” “ndk” “bs” “streaming” “video” “dengan” “lancer” “walaupun” “sdh” “beli” “paket” “premium” “parah” “bgt” “dah” |
| Parah nih aplikasi udh buffering Mulu, terus mau nonton live tv emang harus beli paket dulu yah? lawak sih klo emng gitu pelit bener gak kayak aplikasi video | “parah” “nih” “aplikasi” “udh” “buffering” “mulu” “terus” “mau” “nonton” “live” “tv” “emang” “harus” “beli” “paket” “dulu” “yah” “lawak” “sih” “klo” “emng” “gitu” “pelit” “bener” “gak” “kayak” “aplikasi” “video” |
| Kenapa sekarang app tidak dapat di gunakan sama sekali. Sinyal cuma muter-muter buffering terus | “kenapa” “sekarang” “app”  “tidak” “dapat” “di” “ gunakan” “sama” “sekali” “sinyal” “cuma" “muter” “buffering” “terus” |
| Lemot banget, saat sy tulis ulasan ini pun film nya tidak bisa diputar, padahal sudah berlangganan premium , lalu suka tiba2 putus ditengah film, mengecewakan untuk ukuran aplikasi berbayar | “lemot” “banget” “saat” “sy” “tulis” “ulasan” “ini” “pun” “film” “nya” “tidak” “bisa” “diputar” “padahal” “sudah” “berlangganan” “premium” “lalu” “suka” “tiba2” “putus” “ditengah” “film” “mengecewakan” “untuk” “ukuran” “aplikasi” “berbayar” |

* + 1. Stemming

Pada tahap ini dilakukan untuk mengubah kata berimbuhan yang terdapat pada kata seperti pada awalan, sisipan atau akhiran kata menjadi kata dasar. Untuk melakukan *stemming* menggunakan *library* Sastrawi seperti pada perintah berikut:

factory = StemmerFactory()

stemmer = factory.create\_stemmer()

list\_text = [stemmer.stem(word) for word in list\_text]

hapus kata yang termasuk stopword

list\_text = [word for word in list\_text if word not in list\_stopwords]

Perintah di atas akan memanggil “*StemmerFactory*” dari Sastrawi untuk menjalankan proses *stemming*. Setelah dipanggil, proses akan memeriksa tiap kata. Jika kata tersebut memiliki imbuhan maka imbuhan tersebut akan dihapus.

Perintah di atas akan memanggil “*StemmerFactory*” dari Sastrawi untuk menjalankan proses *stemming*. Setelah dipanggil, proses akan memeriksa tiap kata. Jika kata tersebut memiliki imbuhan maka imbuhan tersebut akan dihapus.

* + 1. **Stopwords Removal**

Pada tahap ini digunakan untuk menghilangkan kata-kata yang umumnya sangat sering muncul pada suatu dokumen dan tidak memiliki arti khusus. Proses ini menggunakan *library stopwords* Indonesia yang disediakan oleh NLTK dan menambahkan kamus yang dibuat oleh Tala. Untuk melakukan *stopword* menggunakan perintah berikut:

# additional\_stopword:list kata tdk penting

additional\_stop = [] #['ya', 'deh', 'dia']

default\_stop = pd.read\_csv(stopword\_path, names=['stopwords'])

default\_stop = default\_stop.stopwords.to\_list()

# combine all stopword

list\_stopwords = []

list\_stopwords = default\_stop + additional\_stop

Tabel 3.4 Daftar *Stopwords*

Sumber. Tala, F.Z. (2003)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ada | Anda | Atau | Belum | Dimana |
|  |  |  |  |  |
| Adalah | Antara | Bagaimanapun | Bisa | DLL |
|  |  |  |  |  |
| Agak | Anu | Bagi | Boleh | Dsb |
|  |  |  |  |  |
| Akan | Apakah | Bahwa | Dahulu | Dst |
|  |  |  |  |  |
| Amat | Apalagi | Begitu | Dalam | Dua |
|  |  |  |  |  |

* + 1. Normalisasi Kata Tidak Baku

Pada tahap ini dataset yang mengandung kata tidak baku diubah menjadi kata yang baku dan yang sesuai dengan ejaan. Normalisasi dilakukan karena masih terdapat banyak kata *slang* yang akan mempengaruhi program. Untuk melakukan normalisasi menggunakan perintah berikut:

kamus\_alay = pd.read\_csv(kamus\_alay\_path)

normalize\_word\_dict = {}

for index, row in kamus\_alay.iterrows():

if row[0] not in normalize\_word\_dict:

normalize\_word\_dict[row[0]] = row[1]

Setelah proses tokenisasi selesai, contoh kalimat yang telah melalui proses tokenisasi akan terlihat pada tabel 3.5

Tabel 3.4 Contoh Hasil Normalisasi

|  |  |
| --- | --- |
| Komentar | Hasil Case folding |
| “versi” “yg” “skrg” “byk” “bugnya” “ndk” “bs” “streaming” “video” “dengan” “lancer” “walaupun” “sdh” “beli” “paket” “premium” “parah” “bgt” “dah” | “versi” “yang” “sekarang” “banyak” “bugnya” “tidak” “bisa” “streaming” “video” “dengan” “lancar” “walaupun” “sudah” “beli” “paket” “premium” “parah” “banget” “sudah” |
| “parah” “nih” “aplikasi” “udh” “buffering” “mulu” “terus” “mau” “nonton” “live” “tv” “emang” “harus” “beli” “paket” “dulu” “yah” “lawak” “sih” “klo” “emng” “gitu” “pelit” “bener” “gak” “kayak” “aplikasi” “video” | “parah” “nih” “aplikasi” “udh” “buffering” “mulu” “terus” “mau” “nonton” “live” “tv” “emang” “harus” “beli” “paket” “dulu” “iya” “lawak” “sih” “kalau” “memang” “gitu” “pelit” “bener” “gak” “kayak” “aplikasi” “video” |
| “kenapa” “sekarang” “app”  “tidak” “dapat” “di” “ gunakan” “sama” “sekali” “sinyal” “cuma" “muter” “buffering” “terus” | “kenapa” “sekarang” “app”  “tidak” “dapat” “digunakan” “sama” “sekali” “sinyal” “cuma" “memutar” “buffering” “terus” |
| “lemot” “banget” “saat” “sy” “tulis” “ulasan” “ini” “pun” “film” “nya” “tidak” “bisa” “diputar” “padahal” “sudah” “berlangganan” “premium” “lalu” “suka” “tiba2” “putus” “ditengah” “film” “mengecewakan” “untuk” “ukuran” “aplikasi” “berbayar” | “lemot” “banget” “saat” “saya” “tulis” “ulasan” “ini” “pun” “film” “nya” “tidak” “bisa” “diputar” “padahal” “sudah” “berlangganan” “premium” “lalu” “suka” “tiba-tiba” “putus” “ditengah” “film” “mengecewakan” “untuk” “ukuran” “aplikasi” “berbayar” |

* 1. **Lexicon Based**

Dengan menggunakan *Lexicon Based* ini data yang sudah dibersihkan pada tahap sebelumnya akan dilabelisasi menjadi 3 yaitu positif, netral, dan negatif dengan cara mencocokan kata pada data yang sudah dibersihkan dengan kamus kata bersentimen. Apabila suatu komentar memiliki score bernilai <= 2 maka akan diklasifikasikan sebagai teks bersentimen negatif, jika suatu komentar memiliki score bernilai = 3 maka akan diklasifikasikan sebagai teks bersentimen netral, dan jika suatu teks memiliki komentar dengan score bernilai >= 4 maka akan diklasifikasikan sebagai teks bersentimen positif. Pada proses Lexicon Based data yang digunakan adalah data yang sudah melalui tahap *stemming*. Hasil klasifikasi dari 8000 dataset yang digunakan menghasilkan 3000 sentimen negatif, 4000 sentimen positif dan 1000 sentimen netral. Untuk melakukan klasifikasi menggunakan Lexicon based menggunakan perintah berikut :

def to\_sentiment(rating):

rating = int(rating)

if rating <= 2:

return 0

elif rating == 3:

return 1

else:

return 2

# Apply to the dataset

df['category'] = df["score"].apply(to\_sentiment)

class\_names = ['negative', 'neutral', 'positive']

ax = sns.countplot(x = df["category"])

ax.set\_xticks(range(len(class\_names)))

ax.set\_xticklabels(class\_names)

plt.xlabel('review sentiment')

plt.show()

def to\_sentiment(rating):

rating = int(rating)

if rating <= 2:

return "negative"

elif rating == 3:

return "neutral"

else:

return "positive"

# Apply to the dataset

df['category'] = df["score"].apply(to\_sentiment)

* 1. Dataset Splitting

Data selanjutnya dibagi menjadi data *training*, *validation* dan *testing*. Sebanyak 8000 data yang ada, data dibagi dengan komposisi 80% atau sebanyak 5599 data *training*, 13,4 % atau sebanyak 1608 data validasi, dan 6.6% atau sebanyak 794 data *testing*. Data *training* digunakan untuk melatih algoritma dalam melakukan klasifikasi sentimen. Data validasi digunakan untuk meminimalisir *overfitting* yang sering terjadi pada jaringan syaraf tiruan, *Overfitting* adalah suatukedaan dimana data yang digunakan pada data training memiliki prediksi yang terlalu baik, namun prediksinya buruk pada data testing. Sedangkan, data *testing* akan digunakan sebagai penguji seberapa baik metode BERT dapat mengklasifikasi sebuah data

train\_set, val\_set = train\_test\_split(df\_v2, test\_size=0.3, stratify=df\_v2.category, random\_state=1)

val\_set, test\_set = train\_test\_split(val\_set, test\_size=0.33, stratify=val\_set.category, random\_state=1)

* 1. Implementasi BERT

Pada penelitian ini menggunakan Model *pre-trained* IndoBERT yang merupakan arsitektur yang khusus dilatih menggunakan data *corpus* bahasa Indonesia. Data set yang digunakan mencapai 4 milyar kata, baik bahasa informal maupun formal dengan 12 korpus bahasa Indonesia yang berbeda. Dataset ini kemudian dilatih dengan arsitektur BERT standard yang memiliki 12 *transformers layers.*

Model ini dipilih karena model tersebut mendukung 104 bahasa termasuk Bahasa Indonesia (Yanuar & Shiramatsu, 2020). Pada penelitian ini menggunakan *library transformers* yang telah disediakan oleh *HuggingFace* yang berisi sejumlahbesar model terlatih dimana dapat digunakan untuk melakukan berbagai macam

tugas seperti klasifikasi, ekstraksi informasi, tanya jawab, *summarization*, translasi, *text generation* sebagai bantuan untuk proses yang panjang. Dua perpustakaanpembelajaran mendalam yang paling populer adalah *PyTorch* dan *TensorFlow*. *Transformers* menggunakan kedua *library* ini untuk membuat model pembelajaranmesin yang andal. Kemudian melakukan penyesuaian dataset sesuai format agar dapat dilakukan pelatihan atau finetuning. Proses penyusunan kalimat yang akan dihasilkan input pada BERT akan dilakukan oleh tokenizer, dengan tahapan-tahapan berikut :

1. Semua Kalimat ditokenisasi dan dipecah menjadi sebuah kata menggunakan WordPiece. *Tokenizer* akan memeriksa apakah setiap kata dalam kalimat tersebut berada di dalam kosa kata atau di luar kosa kata (OOV). *Tokenizer* akan memecah kata menjadi sub-kata dengan kemungkinan kemunculan paling signifikan dalam kosakata. Jika *tokenizer* tidak menemukan sub-kata dalam kosakata, kata tersebut akan dipecah menjadi per-karakternya. Kata-kata ini dipecah menjadi sub-kata atau karakter dengan simbol ##. Namun jika kata yang panjang diubah menjadi sub-kata atau karakter individu, akan terjadi kelebihan beban atau *overload*, dan kata tersebut akan diganti dengan token [UNK] atau *unknown*. BERT melakukan ini karena dua alasan, pertama untuk mempercepat pemrosesan dan mengurangi jumlahnya parameter yang harus dilatih. Kedua adalah untuk mengatasi masalah *out-of-vocabulary*.
2. Tahap selanjutnya adalah *token embedding*, yaitu penambahan *token* [CLS] di awal kalimat sebagai indikator awal kalimat saat dilakukan klasifikasi sentimen. *Token* [SEP] di akhir kalimat berfungsi untuk memisahkan satu kalimat dari kalimat berikutnya.
3. Tahap selanjutnya adalah menyesuaikan panjang kalimat dengan panjang maksimal yang telah ditentukan. Pada tahap ini dilakukan pengurangan token jika kalimat melebihi panjang maksimum atau menambahkan token [PAD] jika kalimat kurang dari panjang maksimum. *Encoder* pada *transformer* hanya menghasilkan output dengan dimensi 512 saja. Jikapanjang hukuman lebih dari panjang maksimal yang telah ditentukan,

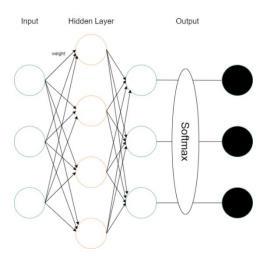
Efeknya akan dikurangi (truncate). Jika panjang rangkaian kalimat kurang dari panjang maksimal yang telah ditentukan, maka kalimat akan ditambah dengan *padding*. Panjang maksimum juga dapat ditentukan secara independen tergantung pada kebutuhan dan tugas yang harus dilakukan.

1. Tahap selanjutnya adalah *token* disesuaikan dengan nomor unik atau id token sesuai dengan *vocabulary*. model *pre-trained* BERT hanya dapat menerima ID token. Dalam model IndoBERT, token [PAD] memiliki token ID 0, token [UNK] memiliki token ID 1, token [CLS] memiliki token ID 2,

dan token [SEP] memiliki token ID 3. Begitu juga token-token lain dalam *vocabulary* masing-masing sudah memiliki token mereka sendiri.Pemberian *attention mask* berfungsi untuk membedakan antara token yang harus diperhatikan dan tidak perlu diperhatikan. *Attention mask* akan menunjukkan posisi indeks yang diberi *padding* sehingga model tidak perlu memproses token tersebut. Tahap ini dapat dilakukan dengan memberikan angka 1 pada token yang harus diperhatikan dan memberikan angka 0 pada *padding*.

|  |  |  |
| --- | --- | --- |
|  | Tabel 3.6 Contoh Hasil Tahapan Tokenizer | |
|  |  |  |
| Tahap |  | Hasil |
|  |  |  |
| Tahap 1 |  | coba aplikasi lancar |
|  |  |  |
| Tahap 2 |  | [CLS], coba, aplikasi, lancar, [SEP] |
|  |  |  |
| Tahap 3 |  | [CLS], coba, aplikasi, lancar, [SEP], [PAD], |
|  |  | [PAD],[PAD], [PAD] |
|  |  |  |
| Tahap 4 |  | 2, 1841, 1339, 4844, 3, 0, 0, 0,0 |
|  |  |  |
|  |  | 1,1,1,1,1, 0,0,0,0 |
|  |  |  |
|  |  | 1,2,3,4,5,6,7,8,9 |
|  |  |  |

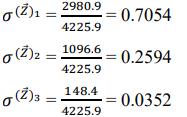
BERT akan menerima urutan kata-kata atau sebuah kalimat sebagai *input* yang akan terus melalui tumpukan *encoder*. Tiap *encoder* mengaplikasikan self-*attention* dan memberikan *output* melalui *feed-forward network* yang kemudiandilanjutkan oleh *encoder* selanjutnya. Pada penelitian ini, model yang dipilih adalah BERTBASE sehingga proses ini berlanjut sebanyak dua belas kali. Setelah melewati semua *encoder*, tiap token per posisi memberikan *output* berupa vektor dengan *hidden size* yaitu 768 pada BERTBASE. Untuk proses analisis sentimen, *output* yangdiperhatikan adalah *output* dari posisi pertama yaitu token [CLS]. Vektor tersebut digunakan sebagai input untuk *classifier*. BERT dapat mencapai hasil yang baik hanya dengan menggunakan *neural network* tunggal sebagai classifier-nya (Devlin et al., 2019). *Layer* yang digunakan untuk klasifikasi adalah *fully connected neural network* dengan fungsi *softmax*. Sehingga *output* dari BERT yang digunakan untukklasifikasi berasal vektor token [CLS] karena token [CLS] dianggap melakukan pengumpulan rata-rata atas token kata untuk mendapatkan vektor dari kalimat. *Layer* terakhir pada *classifier layer* menghasilkan *logits*. *Logits* adalah *output* yangberupa prediksi probabilitas kasar dari kalimat yang akan diklasifikasikan. *Softmax* akan mengubah *logits* tersebut menjadi probabilitas dengan mengambil eksponen dari tiap nilai *logit* sehingga total probabilitasnya adalah tepat 1. Sehingga nilai probabilitas akan berada di antara 0 atau angka positif



Gambar 3.3 Ilustrasi *Layer* untuk Analisis Sentimen

Misalkan terdapat vektor logits = yang harus dikonversi ke distribusi probabilitas dari sebuah komentar agar dapat diklasifikasi ke dalam kategori positif, negatif, dan netral. Maka, langkah untuk mendapatkan probabilitas yaitu:

1. Hitung seluruh eksponensial dari setiap elemen pada vektor.
2. Normalisasi nilai dengan menjumlahkan semua eksponensial.
3. Bagi eksponensial dari setiap elemen dengan normalisasi untuk Memperoleh output softmax dari tiap elemen yang ada.



Untuk memastikan apakah hasil prediksi tiap dari tiap probabilitas adalah 1, maka semua probabilitas dijumlahkan

Hasil prediksi probabilitas tersebut akan menunjukkan probabilitas dari komentar ke kategori sentimen. Berarti komentar tersebut memiliki probabilitas sebesar 0.7054 kepada kategori sentimen positif, sedangkan terhadap sentimen negatif sebesar 0.2594, dan sentimen netral sebesar 0.0352. Hal ini juga dilakukan kepada semua komentar yang ada pada dataset. Pada penelitian ini terdapat 3 kategori sentimen yaitu positif, netral, dan negatif. Selain itu, dibutuhkan pula dropout untuk membuat jaringan tidak mengalami *overfitting*. *Overfitting* adalah suatu masalah yang timbul ketika data di-training, loss berkurang dan akurasi meningkat. Tetapi akurasi yang diperoleh saat testing tidak meningkat atau terus menurun.

Setelah BERT *di-pre-train* dengan dataset yang ada, maka dilakukan *training*. Sebelum melakukan *training*, dataset terlebih dahulu dibagi menjadi tigajenis yaitu dataset untuk *training*, validasi, dan *testing*. Karena dataset yang digunakan cukup banyak, memasukkan dataset ke dalam memori sekaligus dapat membuat memori berhenti bekerja dan memperlambat program. Oleh karena itu

dibutuhkan sebuah data loader untuk membantu menghemat memori dan meningkatkan kecepatan selama training. *DataLoader* merupakan fungsi pada *PyTorch* yang berperan sebagai iterator. *DataLoader* akan menggabungkan datasetdan sample untuk iterasi. Karena dataset dibagi menjadi tiga, maka terdapat juga 3 data *loader* yaitu data *loader* untuk *training*, data *loader* untuk validasi, dan data *loader* untuk *testing*. Seperti pada paper aslinya (Devlin et al., 2019), *optimizer* yang digunakan adalah AdamW yang bertujuan untuk mengoreksi *weight* dari kalimat.

* 1. **Memuat Model Pre-trained**

Penelitian ini menggunakan *pre-trained* model indobenchmark/indobert-base-p1 dari indoNLU.*Pre-trained* model dari indoNLU diakses melalui Hugging-Face. Model *pre-trained* untuk penelitian ini menggunakan kelas model *BertForSequenceClasssification* dari *library transformers* milik *HuggingFace*

# Load Tokenizer and Config

tokenizer = BertTokenizer.from\_pretrained('indobenchmark/indobert-base-p1')

config = BertConfig.from\_pretrained('indobenchmark/indobert-base-p1')

config.num\_labels = DocumentSentimentDataset.NUM\_LABELS

# Instantiate model

model = BertForSequenceClassification.from\_pretrained('indobenchmark/indobert-base-p1', config=config)

* 1. Training Data (Fine Tuning)

*Training* didapatkan dari proses pembagian data sebelumnya yaitu *train data*, *validation data* dan *testing data*. *Data training* di *load* sesuai *hyperparameter tuning* yang telah ditetapkan sebelumnya yaitu *batch size* sebesar 32, *batch size* adalah jumlah *sample* yang dimasukkan ke dalam *network* sebelum *weight* disesuaikan. Jumlah *worker* sebanyak 16, maksimal dari panjang *sequence* adalah 512 dimana 512 adalah dimensi panjang token yang bisa diterima oleh BERT. Selama proses pelatihan, akan dilakukan pemantauan terhadap *loss* dan akurasi dari *data training* dan data validasi pada setiap *epoch*. Variasi *epoch* yang digunakan yaitu 5 dikarenakan Semakin besar batch size maka semakin lama waktu yang dibutuhkan untuk menyelesaikan satu batch (Osinga,2018), *epoch* adalah jumlah berapa kali jaringan melihat seluruh *dataset*. Proses pelatihan akan dijalankan menggunakan Google Collab. Data latih menggunakan menggunakan *learning rate* 3e-6, *learning rate* menentukan seberapa banyak *weight* pada *neural network* yang akan diubah. Pada tahapan sebelumnya, didapatkan hasil pencarian parameter yang optimal. Hasil pencarian ini berguna untuk menyesuaikan penyetelan parameter. Perintah untuk melakukan *training* menggunakan perintah berikut :

# Train

n\_epochs = 5

history = defaultdict(list)

for epoch in range(n\_epochs):

    model.train()

    torch.set\_grad\_enabled(True)

    total\_train\_loss = 0

    list\_hyp\_train, list\_label = [], []

    train\_pbar = tqdm(train\_loader, leave=True, total=len(train\_loader))

    for i, batch\_data in enumerate(train\_pbar):

        # Forward model

        loss, batch\_hyp, batch\_label = forward\_sequence\_classification(model, batch\_data[:-1], i2w=i2w, device='cuda')

        # Update model

        optimizer.zero\_grad()

        loss.backward()

        optimizer.step()

        tr\_loss = loss.item()

        total\_train\_loss = total\_train\_loss + tr\_loss

        # Hitung skor train metrics

        list\_hyp\_train += batch\_hyp

        list\_label += batch\_label

        train\_pbar.set\_description("(Epoch {}) TRAIN LOSS:{:.4f} LR:{:.8f}".format((epoch+1),

            total\_train\_loss/(i+1), get\_lr(optimizer)))

    metrics = document\_sentiment\_metrics\_fn(list\_hyp\_train, list\_label)

    print("(Epoch {}) TRAIN LOSS:{:.4f} {} LR:{:.8f}".format((epoch+1),

        total\_train\_loss/(i+1), metrics\_to\_string(metrics), get\_lr(optimizer)))

    # save train acc for learning curve

    history['train\_acc'].append(metrics['ACC'])

    # Evaluate di validation set

    model.eval()

    torch.set\_grad\_enabled(False)

    total\_loss, total\_correct, total\_labels = 0, 0, 0

    list\_hyp, list\_label = [], []

    pbar = tqdm(valid\_loader, leave=True, total=len(valid\_loader))

    for i, batch\_data in enumerate(pbar):

        batch\_seq = batch\_data[-1]

        loss, batch\_hyp, batch\_label = forward\_sequence\_classification(model, batch\_data[:-1], i2w=i2w, device='cuda')

        # Hitung total loss

        valid\_loss = loss.item()

        total\_loss = total\_loss + valid\_loss

        # Hitung skor evaluation metrics

        list\_hyp += batch\_hyp

        list\_label += batch\_label

        metrics = document\_sentiment\_metrics\_fn(list\_hyp, list\_label)

        pbar.set\_description("VALID LOSS:{:.4f} {}".format(total\_loss/(i+1), metrics\_to\_string(metrics)))

    metrics = document\_sentiment\_metrics\_fn(list\_hyp, list\_label)

    print("(Epoch {}) VALID LOSS:{:.4f} {}".format((epoch+1),

        total\_loss/(i+1), metrics\_to\_string(metrics)))

    # save validation acc for learning curve

    history['val\_acc'].append(metrics['ACC'])

* 1. Evaluasi

Tahap evaluasi bertujuan untuk mendapatkan prediksi dari performa model menggunakan *confusion matrix* yang didapatkan dari tahap training sebelumnya*.* seperti yang ditunjukkan pada Tabel 3.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Predicted Class |  |  |
|  |  |  |  |  |  |
|  |  | Positive | Neutral | Negative |  |
|  |  |  |  |  |  |
| True | Positive |  |  |  |  |
| Class |  | True Positive | False Neutral | False Negative |  |
|  |  |  |  |  |
|  |  | (TP) | (FNt) | (FN) |  |
|  |  |  |  |  |  |
|  | Neutral | False Positive | True Netral | False Negative |  |
|  |  | (FP) | (TNt) | (FN) |  |
|  |  |  |  |  |  |

*Confusion matrix* terdiri dari 6 kategori yaitu :

1. *True Positive* (TP) adalah kalimat positif yang diprediksi positif.
2. *False Positive* (FP) adalah kalimat netral/negatif tetapi diprediksi positif.
3. *True Neutral* (TNt) adalah kalimat netral yang diprediksi netral.
4. *False Neutral* (FNt) adalah kalimat positif atau negatif tetapi diprediksi netral
5. *True Negative* (TN) adalah kalimat negatif yang diprediksi negatif.
6. *False Negative* (FN) adalah kalimat positif/netral tetapi

diprediksi negatif.

Nilai *akurasi, presisi, recall,* dan *F1-score* juga dapat diperoleh setelah mendapatkan nilai *confusion matrix*. Akurasi adalah perbandingan prediksi Benar (Positif, Netral, dan Negatif) dengan keseluruhan data. *Precision, recall,* dan *F1-score* dibagi dengan jumlah kelas. Sedangkan *F1-score* merupakan perbandinganberbobot dari rata-rata *presisi* dan *recall*. Pada penelitian ini menggunakan fungsi *classification\_report()* dari *library Sklearn* untuk menghitung proses evaluasi.Rumus perhitungan untuk mendapatkan *akurasi, presisi, recall,* dan *F1-score* ditunjukkan pada persamaan berikut :

1. **Hasil Dan Pembahasan**
   1. **Evaluasi Hasil**

Setelah dilakukan tahap implementasi BERT maka tahap selanjutnya yaitu tahap evaluasi. Sebanyak 792 data *testing* yang dilakukan dari proses pengujian menggunakan model BERT menghasilkan prediksi komentar *true positive* sebanyak 418 data, *true neutral* sebanyak 71 data, dan *true negative* sebanyak 303 data. Hasil prediksi beberapa data *testing* dapat dilihat pada tabel-tabel berikut.

Tabel 4.1 *True Positive*

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
| good aplikasi . jangan ragu untuk mendownload aplikasi ini . semua saluran ada disini . tingkatkan terus kualitasnya . mantap | Positive | Positive |
| lumayan buat cadangan menonton tv disaat mati listrik , tapi masih bisa menonton di vision , senang banget | Positive | Positive |
| puas dengan vision tidak mengecewakan , fitur lengkap dan mudah penggunaannya | Positive | Positive |
| request supaya app bisa berjalan di background dong min seperti youtube . menghemat baterai kan keren | Positive | Positive |
| dari banyak aplikasi untuk menonton siaran tv , vision terbaik menurut sy.bs menonton siaran ulang nya dan siaran langsung tentu nya dan gratis | Positive | Positive |

Tabel diatas merupakan contoh dari hasil *true positive* yang dihasilkan dari *testing.* *True positive* merupakan kalimat positif yang diprediksi positif oleh model.

Tabel 4.2 *True Neutral*

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
| kenapa bintang , karena kalau menonton champion atau pertandingan bola yang sekiranya big match ypasti susah masuk ke chenel tv tersebut | Neutral | Neutral |
| maaf , . saya kasih bintang dulu , soalnya mau cek dulu antv lancar apa enggak , soalnya kalo saya download apps tv lain selalu antv enggak bisa lancar alias hilang . kalo lancar saya kasih bintang . | Neutral | Neutral |
| loding masih terlalu berat , mohon tampilan di tv android dibuat memudahkan untuk ganti chanel , agar tidak harus keluar masuk saat akan ganti chanel tv . semoga kedepan loading lebih ringan dan fitur memudahkan pengguna | Neutral | Neutral |
| bisa ditayangkan di tv android saya mohon tingkatkan lagi kedepannya dari kualitas dan lainnya | Neutral | Neutral |
| membutuhka koneksi yang benar kuat untuk bisa menonton film | Neutral | Neutral |

Tabel diatas merupakan contoh dari hasil *true neutral* yang dihasilkan dari *testing.* *True neutral* merupakan kalimat netral yang diprediksi netral oleh model.

Tabel 4.3 *True Negative*

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
| katanya sudah di perbaiki tapi masih saja kayak tadi kalau begini gua enggak suka lagi dukung timnas .. mengapai di dukung lihat mereka main saja enggak bisa | Negative | Negative |
| sudah di perbarui masih saja enggak bisa di buka | Negative | Negative |
| aplikasi busuk enggak bisa dibuka ! uninstall saja | Negative | Negative |
| sangat buruk susah banget buat berlangganan lagi tolong di perbaiki kembali sistematis pelayanannya | Negative | Negative |
| sudah kecewa banget , ku kira bisa menonton siaran ulang di tv nasional yang terlewat malah error , tolong diperbaiki lah , kalau aplikasi enggak berguna buat apa orang tertarik beli premium nya ? | Negative | Negative |

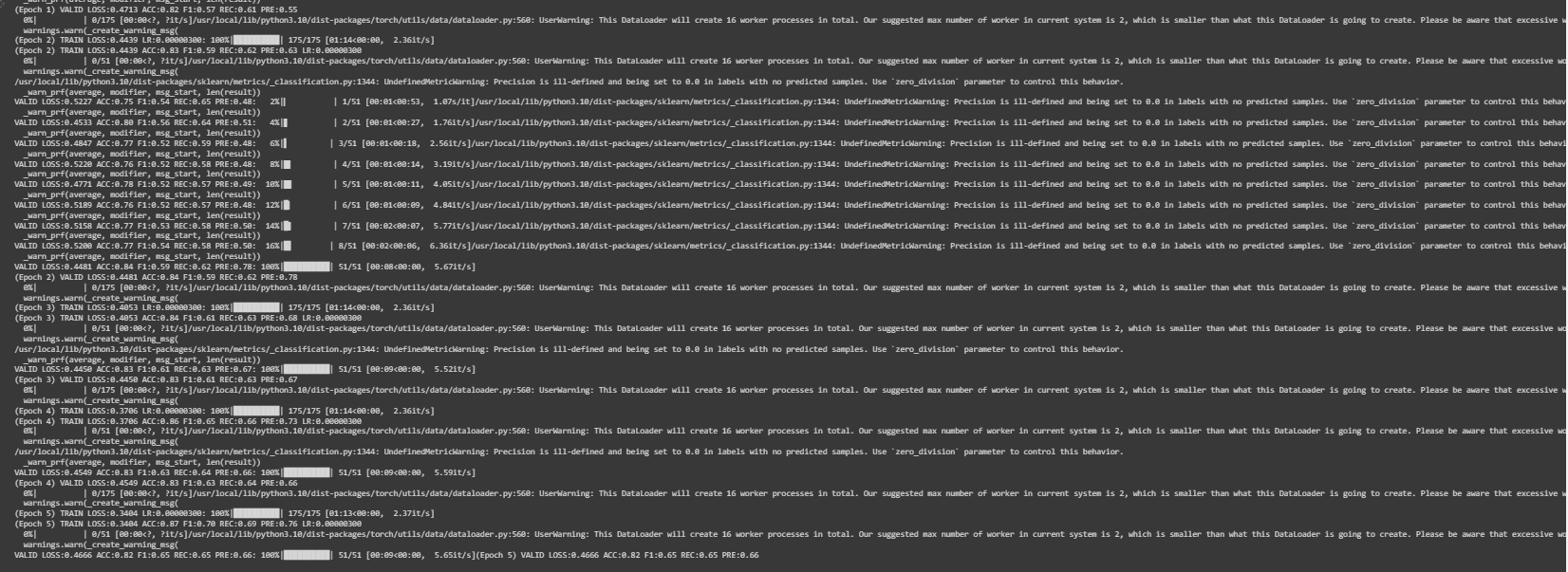
Tabel diatas merupakan contoh dari hasil *true negative* yang dihasilkan dari *testing.*

*True negative* merupakan kalimat negatif yang diprediksi negatif oleh model.

* 1. Uji Akurasi

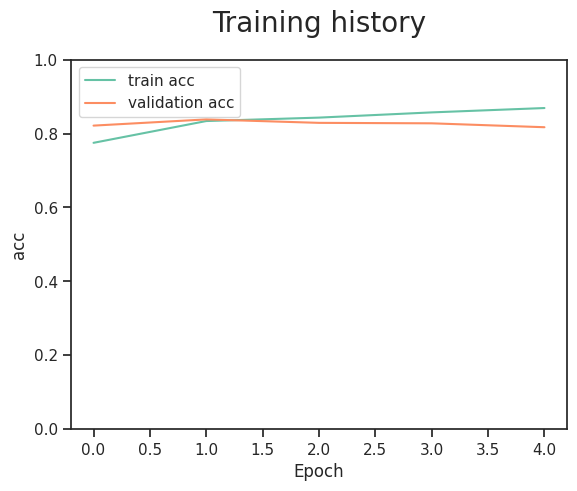
Tahap ini akan membahas tentang hasil uji coba model yang telah dilatih. Pada tahap ini akan ditampilkan hasil pelatihan model dan hasil uji coba model dengan data uji yang akan ditunjukkan dengan *confusion matrix* untuk mengukur seberapa banyak model berhasil memprediksi dengan benar semua sentimen pada *data testing*.

Kemudian akan ditampilkan juga *classification report* yang akan menunjukkan skor *accuracy, precision, recall,* dan *f1-score*. Proses *training* dan evaluasi dapat dilihat pada gambar 4.1



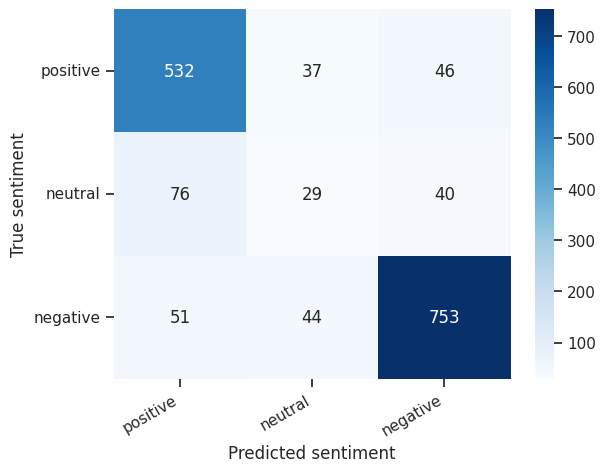
Gambar 4.1 Proses Uji Akurasi

Setelah melalui proses uji coba dengan perulangan per-*epoch*, hasil uji coba akan disimpan. Kemudian akan dibuat *learning curve* sebagai visualisasi perbandingan antara akurasi yang didapat dari *training* dan validasi.



Gambar 4.2 Learning Curve

Pada gambar 4.2 menghasilkan learing curve yang sesuai atau *Good Fit*. *Good fit* diidentifikasi oleh training dapat beradaptasi ketika diberi dataset baru.Hasil data *training* dan validasi yang telah diuji menunjukkan bahwa data *training* mempunyai hasil akurasi yang lebih tinggi dari pada hasil akurasi data validasi. Kemudian akan dilakukan uji coba kurasi terhadap 396 dataset yang ada menggunakan *confusion matrix* yang hasilnya dapat dilihat pada gambar 4.3



Gambar 4.3 *Confusion Matrix*

Confusion matrix yangdidapatkan pada penelitian ini menghasilkan 201 t*rue positive*, *True positive* merupakan kalimat positif yang diprediksi positif olehmodel. 40 *false positive*, *false positive* merupakan kalimat positif yang diprediksi netral atau negatif oleh model. 15 *true neutral, true neutral* merupakan kalimat netral yang diprediksi netral oleh model. 4 *false neutral*, *false neutral* merupakan kalimat netral yang diprediksi positif atau negatif oleh model. 102 *true negative*, *true negative* merupakan kalimat negatif yang diprediksi negatif oleh model. 34 *false negative, false negative* merupakan kalimat negatif yang diprediksi posiitifatau netral oleh model. Proses uji coba dengan *confusion matrix* dilakukan menggunakan perintah berikut :

def show\_confusion\_matrix(confusion\_matrix):

  hmap = sns.heatmap(confusion\_matrix, annot=True, fmt="d", cmap="Blues")

  hmap.yaxis.set\_ticklabels(hmap.yaxis.get\_ticklabels(), rotation=0, ha='right')

  hmap.xaxis.set\_ticklabels(hmap.xaxis.get\_ticklabels(), rotation=30, ha='right')

  plt.ylabel('True sentiment')

  plt.xlabel('Predicted sentiment');

cm = confusion\_matrix(test\_real, test\_pred)

df\_cm = pd.DataFrame(cm, index=['positive', 'neutral', 'negative'], columns=['positive', 'neutral', 'negative'])

show\_confusion\_matrix(df\_cm)

Hasil dari *confusion matrix* yang ada pada penelitian ini menunjukkan bahwa model memiliki kekurangan dalam hal memprediksi sentimen netral, namun untuk sentimen positif dan negatif sudah cukup baik.

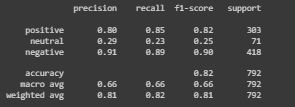
Setelah dilakukan *confusion matrix* maka langkah selanjutnya yaitu menghitung *accuracy, precision, recall,* dan *F1-Score* menggunakan fungsi *classification\_report()* dan sklearn. untuk menampilkan hasil perhitunganmenggunakan perintah seperti berikut :

cm = confusion\_matrix(test\_real, test\_pred)

df\_cm = pd.DataFrame(cm, index=['positive', 'neutral', 'negative'], columns=['positive', 'neutral', 'negative'])

show\_confusion\_matrix(df\_cm)

Hasil perhitungan *accuracy, precision, recall,* dan *F1-Score* yang ada pada penelitian ini ada pada gambar 4.4



Gambar 4.4 Hasil Evaluasi

Dari hasil uji coba akurasi yang telah dilakukan, didapatkan hasil akurasi dari metode BERT dalam memprediksi data testing yaitu 80%. Precision merupakan nilai untuk memberikan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif, tingkat precision untuk kelas positif sebesar 80%, tingkat precision untuk kelas netral sebesar 29%, sedangkan tingkat precision untuk kelas negatif sebesar 91%. Recall memberikan rasio prediksi benar positif dibandingkan dengan keluruhan data yang benar positif, nilai recall yang didapat dari kelas positif sebesar 85%, nilai recall yang didapat dari kelas netral sebesar 23% sedangkan nilai recall yang didapat dari kelas negatif sebesar 89%. F1-score merupakan perbandingan rata-rata presisi dan recall yang dibibitkan, nilai f1-score yang didapat dari kelas positif sebesar 82%, nilai f1-score yang didapat dari kelas netral sebesar 25% sedangkan nilai f1-score yang didapat dari kelas negatif sebesar 81%.

* 1. Visualisasi Frekuensi Data

Pada thap ini akan menampilkan frekuensi kata yang paling sering muncul pada dataset yang ada dengan batasan maksimal 20 kata. Untuk menampilkan hasil frekuensi data menggunakan perintah seperti berikut:

plt.style.use('default')

sns.set(style='ticks', palette='Set2')

mpl.rcParams['axes.titlesize'] = 20

mpl.rcParams['axes.titlepad'] = 20

# Compare plots of train and test corpus

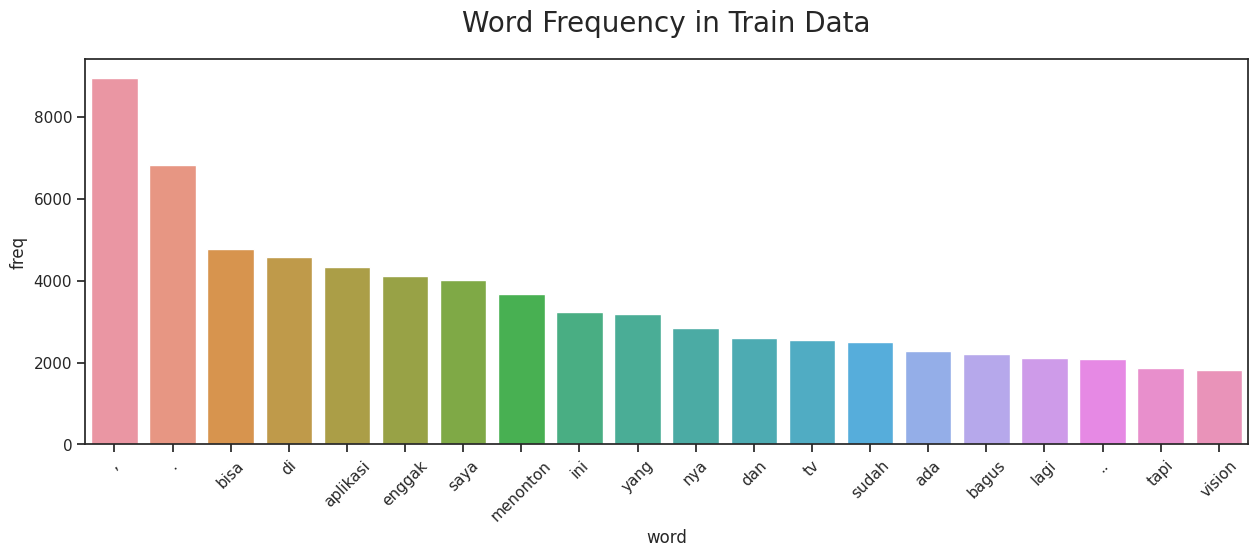
f, ax1 = plt.subplots(1,figsize=(15,5))

sns.barplot(x='word', y='freq', data=corpus\_freq, ax=ax1)

ax1.set\_title('Word Frequency in Train Data')

ax1.tick\_params(axis='x', rotation=45)

Hasil frekuensi kata yang ada pada penelitian ini ada pada gambar 4.5



Gambar 4.5 Visualisasi Frekuensi Data

* 1. Visualisasi Word Cloud

Data yang ditampilkan pada wordcloud adalah data dalam bentuk 20 kata yang paling sering muncul ada pada *dataset*. Proses visualisasi data *wordcloud* dilakukan menggunakan perintah seperti berikut ini:

wordcloud = WordCloud(max\_font\_size=50, max\_words=100, background\_color="white").generate(corpus)

plt.figure()

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis("off")

plt.show()

Hasil *wordcloud* yang ada pada penelitian ini ada pada gambar 4.6



Gambar 4.6 *Wordcloud*

* 1. Ketidak akuratan Hasil Testing

Pada tahap evaluasi yang didapatkan dari testing menghasilkan confusion matrix 14 nilai *false positive* dimana sentimen bernilai netral tetapi diprediksi positif, 26 nilai *false positive* dimana sentimen bernilai negatif tetapi diprediksi positif. Kemudian terdapat 1 *false neutral* dimana sentimen bernilai positif tetapi diprediksi netral, 3 *false neutral* dimana sentimen bernilai negatif tetapi diprediksi netral. Terakhir, 23 *false negative* dimana sentimen bernilai negatiff tetapi diprediksi positif dan 11 *false negative* dimana sentimen bernilai netral tetapi diprediksi negatif.

Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data t training sehingga prediksi model bisa berbeda dengan labelisasi.

Tabel 4.4 *False Positive*

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
|  |  |  |
| film keren keren | Netral | Positif |
|  |  |  |
| tolong langgan biasa bayar pakai gopay punya | Netral | Positif |
| gopay tolong sudah kasih bintang full |  |  |
|  |  |  |
| kasih star kouta putar film aneh kalau kouta | Netral | Positif |
| flashnya jalan kouta maxsteam bagaimana |  |  |
|  |  |  |

Tabel diatas merupakan contoh dari nilai *false positive* dimana sentimen bernilai netral tetapi diprediksi positif oleh model. Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

*Tabel 4.5 False Positive*

|  |  |  |  |
| --- | --- | --- | --- |
| Komentar |  | Sentimen | Prediksi |
|  |  |  |  |
| aplikasi segi kualitas gambar | enggak lama | Negatif | Positif |
| premium jangkau sayang update film film baru | |  |  |
| film action film indonesia baru bom | |  |  |
|  |  |  |  |
| sudah bayar |  | Negatif | Positif |
|  |  |  |  |
| aplikasi bagus menonton film | saja masang wifi | Negatif | Positif |
| tau dapat disney hotstar gratis pasang wifinya | |  |  |
| keren |  |  |  |
|  |  |  |  |

Tabel diatas merupakan contoh dari nlai *false positive* dimana sentimen bernilai negatif tetapi diprediksi positf oleh model. Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

*Tabel 4.6* False Neutral

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
|  |  |  |
| putar set top box smart tv subtitle indonesia putar | Positif | Netral |
| smartphone sedia bahasa indonesia |  |  |
|  |  |  |

Tabel diatas merupakan contoh dari nlai *false neutral* dimana sentimen bernilai positif tetapi diprediksi netral oleh model. Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

Tabel 4.7 False Neutral

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
|  |  |  |
| demon slayer season update | Negatif | Netral |
|  |  |  |
| film jangan | Negatif | Netral |
|  |  |  |
| buka aplikasi tampil muat putar | Negatif | Netral |
|  |  |  |

Tabel diatas merupakan contoh dari nlai *false neutral* dimana sentimen bernilai negatif tetapi diprediksi netral oleh model. Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

.*Tabel 4.8* False Negative

|  |  |  |
| --- | --- | --- |
| Komentar | Sentimen | Prediksi |
|  |  |  |
| putar set top box smart tv subtitle indonesia putar | Positif | Netral |
| smartphone sedia bahasa indonesia |  |  |
|  |  |  |
| loading home screen mood menonton | Positif | Netral |
|  |  |  |
| kasih bintang download putar | Positif | Netral |
|  |  |  |

Tabel diatas merupakan contoh dari nlai *false negative* dimana sentimen bernilai positif tetapi diprediksi netral oleh model. Hal ini dapat terjadi dikarenakan model

menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

Tabel 4.9 *False Negative*

|  |  |  |  |
| --- | --- | --- | --- |
| Komentar |  | Sentimen | Prediksi |
|  | |  |  |
| sudah beli paket kuota maxtream tetap kuota | | Netral | Negatif |
| reguler kepake sampai ambil pulsa habis sudah | |  |  |
| buka tetap pakai kuota reguler |  |  |  |
|  |  |  |  |
| film kadang putar terus loading | suara putus putus | Netral | Negatif |
| eror |  |  |  |
|  |  |  |  |
| bagus kadang sinyal suka lama | buka aplikasi | Netral | Negatif |
| lancar |  |  |  |
|  |  |  |  |

Tabel diatas merupakan contoh dari nlai *false negative* dimana sentimen bernilai netral tetapi diprediksi negatif oleh model. Hal ini dapat terjadi dikarenakan model menilai komentar berdasarkan pola dari data training sehingga prediksi model bisa berbeda dengan labelisasi.

1. **PENUTUP**
   1. **Kesimpulan**

Penelitian analisis sentimen Aplikasi Vision+ pada Android menggunakan metode BERT atau *Bidirectional Encoder Representations from Transformers* dengan bahasa pemrograman *Python* untuk mengklasifikasikankomentar ke dalam sentimen positif, netral, dan negatif telah berhasil dilakukan. Berdasarkan hasil pengujian yang telah dilakukan, didapatkan hasil akurasi sebesar 82% dari jumlah data komentar sebanyak 8000 data dengan data latih sebanyak 5599 data, data validasi sebanyak 1608 data, dan data uji sebanyak 792 data. Berdasarkan hasil analisis sentimen yang dilakukan menghasilkan akurasi persentase sentimen pengguna Disney+ Hotstar pada positif sebesar 83%, netral sebesar 79% dan negatif sebesar 75%.

* 1. **Saran**

Adapun saran yang diberikan pada penelitian ini untuk pengembangan lebih lanjut, pengembangan yang dilakukan dapat melakukan pengambilan dataset yang lebih banyak, proses pelabelan dengan user yang lebih memahami kosakata Bahasa Indonesia, dan menggunakan model *Bidirectional Encoder Representations* (BERT) yang lain seperti IndoBERT-large agar dapat melihat perbedaan hasil akurasi yang didapatkan dan dapat menggunakan *learning rate* yang berbeda.

**Daftar Pustaka**

B.M. Randles, L. P. (2017). Using the Jupyter Notebook as a Tool for Open Source.

*An Empirical Study*, 1-2.

C.A. Putri, A. S. (2021). Analisis Sentimen Review Film Berbahasa Inggris dengan Pendekatan Bidirectional Encoder Representaions from Transformers.

*JEISBI*.

Fimoza, D. (2021). *Analisis Sentimen Film Indonesia Dengan Pendekatan BERT.*

Medan: USU.

Gho, K. W. (2021). *Implementasi dan Pemodelan BERT untuk Analisis Sentimen Analisis Aplikasi Gojek pada Platform Playstore.* Tangerang: UMN.

Imanuel K, A. R. (2019). Analisis Sentimen Topik Viral Desa Penari Pada Media Sosial Twitter Dengan Metode Lexicon Based. *Jurnal ilmiah matrik*.

J Han, M. K. (2006). *Data Mining: Concepts And Techniques, Second Edition.*

Morgan Kaufmann.

Jacob devlin, M.-W. C. (2019). BERT: Pre-training of Deep Bidirectional Transformers For Language Understanding . *Google AI Language*.

Kamer, J. (2017). *Python - The Fastest Growing Programming Language.*

Lian Ardiani, H. S. (2020). Implementasi Sentimen Analisis Tanggapan Masyarakat Terhadap Pemangunan di Kota Pontianak. *JUSTIN*.

1. Zain, R. S. (2018). Analisis Sentimen Pendapat Masyarakat Mengenai Vaksin Covid-19 Pada Media Sosial Twitter dengan RO-BERT Pretraining Approach. *Komputer Terapan*, 2.

R. Atmajaya, w. Y. (2021). Anlisis Sentimen Customer Review Aplikasi Ruang Guru Dengan Menggunakan Metode BERT(Biderectional Encoder Representations From Transformers). *JEISBI*.

Reyhana, Z. (2018). Anlisis Sentimen Pendapat Masyarakat Terhadap Pembangunan Infrastruktur Kota Surabaya Melalui Twitter Dengan Menggunakan Support Vector Machine Dan Neural Network .

S. Gosh, S. Gosh, dan D. Das. (2016). Part-of- speech tagging of code-miced Proceeding of the second workshop on Computational Approaches to Code Swing. 90-97

Vaswani, A. (2017). Attention is all you need. *Conference on Neural Information Processing systems*.

Alammar, J. (2018). Retrieved from https://jalammar.github.io/illustrated-transformer/ diakses pada Tanggal 05 Juni 2022

DQlabs.id. (2021). *Apa itu Machine Learning dan Mengapa Machine Learning Penting*. Diambil dari www.dqlab.id/apa-itu-machine-learning-dan-mengapa-penting diakses pada tanggal 25 April 2022

Putra, Ezha Ananda (2021). *Sentiment Analysis Using Bidirectional Encoder*

*Representations from Transformers (BERT).* Diambil dari https://medium.com/@eza.a.putra/implementasi-bert-untuk-analisis sentimen-terhadap-ulasan-aplikasi-flip-berbahasa-indonesia-557d691e0440 diakses pada tanggal 27 April 2022

Glints.com. (2021). *Mengenal Google Colab: Mulai dari Definisi,* *Cara*

*Menggunakan, hingga Manfaatnya*. Diambil dari https://glints.com/id/lowongan/google-colab-adalah/. diakses pada tanggal 01 Juni 2022

Hotstar, D. (n.d.). *About US*. Diambil dari www.hotstar.com/id/about-us diakses pada tanggal 05 Juni 2022