IMPLEMENTASI MODEL *ROBUSTLY OPTIMIZED* BERT *PRETRAINING APPROACH* (ROBERTA) UNTUK KLASIFIKASI *TWEET CYBERBULLYING* DENGAN AUGMENTASI DATA *BACK TRANSLATION*

Skripsi

Oleh

FATHAN ALHINDAMI
NPM. 2117031068

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

ABSTRACT

IMPLEMENTATION OF ROBUSTLY OPTIMIZED BERT PRETRAINING APPROACH (ROBERTA) MODEL FOR CYBERBULLYING TWEET CLASSIFICATION WITH BACK TRANSLATION DATA AUGMENTATION

By

Fathan Alhindami

The rapid development of internet technology has led to a significant increase in the number of social media users worldwide. Despite its benefits, social media can also be a space for negative behavior such as cyberbullying. This condition requires an automatic classification system that can accurately detect and identify types of cyberbullying in text. This research aims to build a RoBERTa-based text classification model and evaluate the effect of data augmentation techniques on improving model performance. The data used is a collection of English tweets that have been labeled according to the type of cyberbullying. The back translation data augmentation technique is applied to overcome imbalanced data and increase the variety of training data. The RoBERTa model that has gone through the fine-tuning process is implemented to perform classification and its performance is evaluated using accuracy, precision, recall, F1 score, and ROC-AUC metrics. The results showed that the method without data augmentation produced an accuracy of 88%, while the method with data augmentation achieved the highest accuracy of 93%, with precision, recall, and F1-score of 92% each, and an average AUC value of 95.19%. These results show that data augmentation techniques can significantly improve the model's performance in classifying cyberbullying tweets.

Keywords: Cyberbullying, RoBERTa, Data Augmentation, Text Classification.

ABSTRAK

IMPLEMENTASI MODEL *ROBUSTLY OPTIMIZED* BERT *PRETRAINING APPROACH* (ROBERTA) UNTUK KLASIFIKASI *TWEET CYBERBULLYING* DENGAN AUGMENTASI DATA *BACK TRANSLATION*

Oleh

Fathan Alhindami

Perkembangan teknologi internet yang pesat telah memberikan peningkatan yang signifikan dalam jumlah pengguna media sosial di seluruh dunia. Terlepas dari manfaatnya, media sosial juga bisa menjadi ruang munculnya perilaku negatif seperti *cyberbullying*. Kondisi ini memerlukan adanya sistem klasifikasi otomatis yang dapat mendeteksi dan mengidentifikasi jenis *cyberbullying* dalam teks secara akurat. Penelitian ini bertujuan untuk membangun model klasifikasi teks berbasis RoBERTa dan mengevaluasi pengaruh teknik augmentasi data terhadap peningkatan kinerja model. Data yang digunakan merupakan kumpulan *tweet* berbahasa Inggris yang telah diberi label sesuai jenis *cyberbullying*. Teknik augmentasi data *back translation* diterapkan untuk mengatasi data yang tidak seimbang dan menambah variasi data pelatihan. Model RoBERTa yang telah melalui proses *fine-tuning* diimplementasikan untuk melakukan klasifikasi dan kinerjanya dievaluasi menggunakan metrik *accuracy, precision, recall, F1-score*, dan ROC-AUC. Hasil penelitian menunjukkan bahwa metode tanpa augmentasi data menghasilkan *accuracy* sebesar 88%, sedangkan metode dengan augmentasi data mencapai *accuracy* tertinggi sebesar 93%, dengan *precision, recall*, dan *F1-score* masing-masing sebesar 92%, serta rata-rata nilai AUC sebesar 95,19%. Hasil ini menunjukan bahwa teknik augmentasi data secara signifikan mampu meningkatkan kinerja model dalam mengklasifikasikan *tweet cyberbullying*.

Kata kunci: *Cyberbullying*, RoBERTa, Augmentasi Data, Klasifikasi Teks.

IMPLEMENTASI MODEL *ROBUSTLY OPTIMIZED* BERT *PRETRAINING APPROACH* (ROBERTA) UNTUK KLASIFIKASI *TWEET CYBERBULLYING* DENGAN AUGMENTASI DATA *BACK TRANSLATION*

FATHAN ALHINDAMI

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

Judul Skripsi	:	**IMPLEMENTASI MODEL *ROBUSTLY OPTIMIZED* BERT *PRETRAINING APPROACH* (ROBERTA) UNTUK KLASIFIKASI *TWEET CYBERBULLYING* DENGAN AUGMENTASI DATA *BACK TRANSLATION***
Nama Mahasiswa	:	Fathan Alhindami
Nomor Pokok Mahasiswa	:	**2117031068**
Program Studi	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. Dian Kurniasari, S.Si., M.Sc.	**Favoriser R. Lumbanraja, M.Si., Ph.D.**
NIP 196903051996032001	NIP 198301102008121002

2. Ketua Jurusan Matematika

Dr.Aang Nuryaman, S.Si.,M.Si.
NIP. 197403162005011001

MENGESAHKAN

1. Tim penguji

Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.** .....

Sekretaris : **Favorisen R. Lumbanraja, M.Si., Ph.D.** .....

Penguji
Bukan Pembimbing : **Prof. Dra. Wamiliana, MA., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **13 Juni 2025**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama	:	**Fathan Alhindami**
Nomor Pokok Mahasiswa	:	**2117031068**
Jurusan	:	**Matematika**
Judul Skripsi	:	**Implementasi Model *Robustly Optimized* BERT *Pretraining Approach* (RoBERTa) untuk Klasifikasi *Tweet Cyberbullying* dengan Augmentasi *Data Back Translation***

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 13 Juni 2025

Penulis

Fathan Alhindami

RIWAYAT HIDUP

Penulis bernama lengkap Fathan Alhindami, lahir di Bandar Lampung pada 21 Desember 2002. Penulis merupakan anak tunggal dari pasangan Bapak Lukman Karim dan Ibu Yulia Fajar Sari.

Penulis memulai perjalanan pendidikannya di Sekolah Dasar (SD) di SD Negeri 1 Hanura pada tahun 2009-2015, kemudian melanjutkan pendidikan ke Sekolah Menengah Pertama (SMP) Lazuardi Haura GIS pada tahun 2015-2018, dan Sekolah Menengah Atas (SMA) Swasta YP Unila pada tahun 2018-2021.

Pada tahun 2021 penulis melanjutkan pendidikan di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis pernah menjadi pengurus Himpunan Mahasiswa Jurusan Matematika (HIMATIKA) FMIPA Unila Periode 2023 sebagai anggota Bidang Eksternal.

Sebagai bentuk pengaplikasian ilmu yang didapat selama perkuliahan, pada bulan Desember 2023 hingga Februari 2024, penulis melaksanakan Kerja Praktik (KP) di PT Pegadaian (persero) CP Teluk Betung. Kemudian, pada bulan Juni hingga Agustus 2024, penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Kedung Ringin, Kecamatan Pasir Sakti, Kabupaten Lampung Timur, Provinsi Lampung.

KATA INSPIRASI

”*Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan*”
(Q.S Al-Insyirah: 5-6)

”*Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Dia mendapat pahala dari kebajikan yang dikerjakannya dan dia mendapat siksa dari kejahatan yang diperbuatnya*”
(Q.S Al-Baqarah: 286)

”*Hatiku tenang karena mengetahui bahwa apa yang melewatkanku tidak akan pernah menjadi takdirku, dan apa yang ditakdirkan untukku tidak akan pernah melewatkanku*”
(Umar Bin Khattab)

”*Don’t be afraid of mistakes. Mistakes are part of the learning process and the path to knowledge* ”
(Marie Curie)

PERSEMBAHAN

Alhamdulillahirobbil'alamin,

Puji dan syukur kehadirat Allah Subhanahu Wata'ala atas limpahan nikmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Shalawat serta salam senantiasa tercurahkan kepada junjungan kita Nabi Muhammad Shallallahu 'Alaihi Wassalam. Dengan penuh syukur, penulis persembahkan karya ini kepada:

Ayah dan Ibu

Terima kasih atas segala pengorbanan, doa, serta dukungannya selama ini. Terima kasih telah menjadi sumber semangat, mengupayakan yang terbaik, dan selalu menguatkan penulis dalam kondisi apapun. Tanpa kasih sayang, keteguhan, dan doa tulus kalian, perjalanan ini tidak akan sampai sejauh ini.

Dosen Pembimbing dan Pembahas

Terima kasih kepada dosen pembimbing dan pembahas yang sudah sangat berjasa membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga untuk penulis.

Sahabat-sahabatku

Terima kasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasi, serta doa dan dukungan dalam hal apapun.

Almamater Tercinta
Universitas Lampung

SANWACANA

Alhamdulillahirabbil'alamin, puji syukur kehadirat Allah SWT. atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Implementasi Model *Robustly Optimized* BERT *Pretraining Approach* (RoBERTa) untuk Klasifikasi *Tweet Cyberbullying* dengan Augmentasi *Data Back Translation*".

Terselesaikannya skripsi ini, tidak lepas dari bimbingan, arahan, motivasi, serta doa dari berbagai pihak. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku Pembimbing I yang dengan penuh dedikasi membimbing penulis melalui setiap tahap penulisan skripsi ini. Bimbingan, motivasi, serta ilmu dan wawasan yang diberikan sangat membantu penulis dalam menyelesaikan skripsi ini.
2. Bapak Favorisesn Rosyking Lumbanraja, S.Kom., M.Si., Ph.D. selaku Pembimbing II yang telah memberikan ilmu, wawasan, motivasi, serta masukan yang membangun kepada penulis. Dukungan dan arahan yang diberikan sangat membantu penulis dalam menghadapi berbagai tantangan selama proses penulisan skripsi ini.
3. Ibu Prof. Dra. Wamiliana, MA., Ph.D. selaku Pembahas yang telah meluangkan waktu untuk memberikan kritik, saran, serta masukan yang membangun demi perbaikan skripsi ini.
4. Bapak Dr. Agus Sutrisno, S.Si., M.Si. selaku Pembimbing Akademik atas bimbingan dan arahannya selama penulis menjalani proses perkuliahan.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Seluruh dosen dan staf Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan wawasan, ilmu, dan pengetahuan yang sangat berharga bagi penulis selama menjalani proses perkuliahan.

8. Ayah, Ibu, dan keluarga yang selalu memberikan doa, dukungan moral, serta material yang tidak ternilai harganya, sehingga penulis dapat menyelesaikan skripsi ini.
9. Rekan-rekan terbaik penulis selama menjalani perkuliahan Lusiana, Adinda, Mei, Rahma, dan Atun yang banyak membantu penulis selama perkuliahan maupun penyelesaian skripsi ini.
10. Teman-teman seperbimbingan lainya, Anggy, Ariz, Dita, Maya, Mey, Nabila, Sherina, Anastasia, Dina, Erwin, Rhea, dan Yulina, yang telah membersamai dan banyak membantu penulis selama proses penyelesaian skripsi.
11. Teman-teman Jurusan Matematika angkatan 2021 serta Abang Yunda yang telah membantu selama proses perkuliahan.
12. Teman-teman dari Bidang Eksternal Himatika FMIPA Unila 2023 yang telah menemani.
13. Seluruh pihak terkait yang telah membantu menyelesaikan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penulisan skripsi ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak.

Bandar Lampung, 13 Juni 2025

Fathan Alhindami

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi internet dalam beberapa dekade terakhir telah memberikan perubahan yang signifikan di berbagai aspek kehidupan manusia, mulai dari komunikasi, pendidikan, kesehatan, dan lain-lain. Internet telah menjadi bagian dari kehidupan masyarakat di seluruh dunia saat ini. Berdasarkan laporan DIGITAL 2024 yang dirilis oleh *We Are Social and Kepios*, jumlah pengguna internet di seluruh dunia mencapai 5.45 miliar di tahun 2024. Sedangkan, jumlah pengguna internet di Indonesia mencapai angka 221.5 juta pada tahun 2024 dengan kenaikan 1.4% dari tahun sebelumnya. Pertumbuhan tersebut menunjukan adanya tren positif dalam jumlah pengguna internet di Indonesia (APJII, 2024). Pertumbuhan jumlah pengguna internet diikuti dengan meningkatnya penggunaan media sosial sebagai salah satu produk utama dari internet. Media sosial adalah wadah seseorang untuk bersosialisasi antar individu meskipun tidak saling mengenal satu sama lain (Fadli & Sazali, 2023). Pengguna media sosial di Indonesia terus berkembang pesat setiap tahunnya dengan 167 juta pengguna aktif pada tahun 2023 (Kemp, 2023).

Saat ini, salah satu platform media sosial yang populer digunakan untuk berinteraksi adalah Twitter atau yang sekarang dikenal sebagai X. Pada tahun 2024, Indonesia menempati peringkat ke-4 dalam jumlah pengguna aktif Twitter dengan sekitar 24,85 juta pengguna, hal ini membuat twitter menjadi salah satu *platform* media sosial yang popular di Indonesia (Statista, 2023). Namun, di balik dampak positifnya, peningkatan penggunaan internet dapat memberikan dampak negatif, salah satunya adalah *cyberbullying*. Menurut Ali & Syed (2020), *Cyberbullying* adalah bentuk penindasan atau pelecehan yang dilakukan melalui media atau sarana elektronik dan umumnya terjadi di kalangan anak muda dan remaja. Berdasarkan data Komisi Perlindungan Anak Indonesia (KPAI), tercatat ada 361 anak yang menjadi korban *cyberbullying* di media sosial pada periode 2016-2020. Sementara itu, di Jawa Timur

tercatat ada 1.283 kasus *cyberbullying* yang dilaporkan ke KPAI Jawa Timur pada tahun 2021, jumlah kasus tersebut meningkat sebanyak dua kali lipat dibandingkan tahun-tahun sebelumnya (Rovida & Sasmini, 2024).

Twitter adalah salah satu media sosial yang digunakan oleh banyak orang untuk saling bertukar informasi dan berinteraksi dengan orang lain. Namun, Twitter juga sering digunakan sebagai *platform* untuk menyebarkan pesan agresif dan intimidatif melalui *tweet* yang ditujukan untuk merendahkan individu atau kelompok. *Tweet* tersebut biasanya didasarkan pada beberapa karakteristik tertentu, yaitu demografi, jenis kelamin, ras, atau orientasi seksual (Sterner & Felmlee, 2017).

Menurut survei yang dilakukan oleh *Center for Countering Digital Hate* yang dikutip dari CNN Indonesia (2022), menunjukkan bahwa jumlah tweet berisi hinaan terhadap pengguna media sosial mencapai rata-rata 3.876 *tweet* per hari. Selain itu, survei tersebut juga mengungkapkan bahwa pengguna Twitter yang merupakan pria *gay* atau homoseksual menerima rata-rata 3.964 *tweet cyberbullying* setiap harinya. Dampak yang dapat ditimbulkan dari *cyberbullying* antara lain adalah korban merasa depresi, cemas, hingga memiliki keinginan untuk melakukan bunuh diri (Rahayu, 2012). Melihat dampak serius serta data kasus *cyberbullying* yang sering terjadi di media sosial Twitter, perlu dilakukan langkah pencegahan dan pengawasan terhadap *tweet* yang mengandung unsur *cyberbullying*. Hal tersebut perlu dilakukan untuk mengurangi dampak negatif bagi para pengguna khususnya bagi anak-anak dan remaja yang rentan menjadi korban *cyberbullying*.

Metode klasifikasi merupakan salah satu pendekatan penting dalam upaya menangani kasus *cyberbullying* (Wang & Potika, 2021). Pendekatan klasifikasi teks memiliki kemampuan secara otomatis mengidentifikasi tindakan *cyberbullying* di media sosial dengan memanfaatkan teknik komputasi dan *machine learning*. Klasifikasi adalah proses mengelompokkan data ke dalam kategori yang telah ditentukan. Data pelatihan berlabel digunakan untuk menetapkan aturan yang kemudian diterapkan pada data uji agar dapat dikelompokkan ke dalam kategori tersebut (Priyambodo & Prihati, 2020). Salah satu tantangan dalam pemrosesan data teks adalah struktur yang tidak konsisten, yang dapat menyebabkan ketidakakuratan dalam model. Oleh karena itu, diperlukan pendekatan inovatif agar data tersebut dapat diolah dan dipahami secara efektif (Alwehaibi, *et al*., 2022).

Natural Language Processing (NLP) atau pemrosesan bahasa alami adalah cabang dari ilmu komputer dan *machine learning* yang berfokus pada pengembangan sistem yang memungkinkan interaksi antara komputer dan bahasa manusia secara alami. Tujuan utama dari NLP adalah untuk memungkinkan komputer tidak hanya memahami dan menganalisis, tetapi juga memproses dan menghasilkan bahasa manusia dengan cara yang efektif dan kontekstual (Arrasyid, *et al*., 2024). Beberapa teknik utama dalam NLP, seperti tokenisasi, *stemming, lemmatization, part-of-speech tagging*, dan *named entity recognition* yang memungkinkan komputer untuk memahami struktur dan makna teks dengan lebih baik (Oktavia, *et al*., 2024). Komputer tidak mampu memahami bahasa alami seperti manusia, sehingga diperlukan metode NLP yang dapat mengklasifikasikan teks dengan akurasi tinggi.

Pada proses klasifikasi teks, sering muncul masalah ketidakseimbangan data atau *imbalanced data*, yang dapat mempengaruhi kinerja model prediksi. Ketidakseimbangan data terjadi ketika distribusi data pelatihan tidak merata, di mana kelas mayoritas jauh lebih banyak daripada kelas minoritas. Hal ini membuat model cenderung lebih fokus pada kelas mayoritas dan mengabaikan kelas yang lebih sedikit (Rahma & Suadaa, 2023). Ada berbagai metode yang dapat diterapkan untuk mengatasi masalah ini. Teknik seperti *random oversampling* dan *Synthetic Minority Over-sampling Technique* (SMOTE) saat melakukan *resampling* dapat mengabaikan distribusi asli data dan meningkatkan risiko *overfitting* (Wang, *et al*., 2012). Selain itu, SMOTE juga memiliki kelemahan lain, seperti menghasilkan sampel yang *noisy* sehingga menyebabkan *over generalization*, pengambilan sampel yang kurang informatif, serta peningkatan *overlapping* antar label (Soltanzadeh & Hashemzadeh, 2021).

Salah satu metode alternatif untuk menangani masalah ketidakseimbangan data adalah augmentasi data. Augmentasi data merupakan proses yang bertujuan untuk memperluas ukuran dataset pelatihan dengan membuat variasi teks baru dari data yang sudah ada. Teknik ini menghasilkan data teks yang berbeda dari versi aslinya, tetapi tetap mempertahankan konteks aslinya (Taylor & Nitschke, 2018). Pada penelitian yang dilakukan oleh Wei dan Zou (2019), menunjukan bahwa augmentasi data dapat mengurangi risiko terjadinya *overfitting* dan meningkatkan kinerja dalam tugas klasifikasi teks. Salah satu teknik dari augmentasi data adalah *back translation*. *Back translation* adalah teknik yang menerjemahkan teks asli ke dalam bahasa target, kemudian diterjemahkan kembali ke bahasa asal. *Back translation* menghasilkan dataset yang lebih besar dan dapat digunakan untuk meningkatkan performa model

(Bucos & Tucudean, 2023). Hal ini dibuktikan dalam penelitian Ma dan Li (2020), menyatakan bahwa teknik *back translation* telah terbukti efektif dalam menangani ketidakseimbangan distribusi sampel dan meningkatkan kinerja model klasifikasi secara keseluruhan.

Model *transformer* adalah salah satu model yang sering digunakan pada pemrosesan bahasa alami akhir-akhir ini karena kinerja yang baik mengatasi berbagai masalah dalam NLP. Model ini memiliki struktur yang lebih kompleks dan mekanisme *self-attention* yang memungkinkan untuk memahami struktur bahasa serta tulisan yang lebih kompleks. Oleh kerena itu, para peneliti mulai mengembangkan model *transformer* dalam tugas NLP. Beberapa tahun ini, *Biderectional Encoder Representations from Transformer* (BERT) telah menjadi model representasi yang digunakan secara luas, mencapai tingkat kinerja yang mutakhir pada tugas-tugas tingkat kalimat, dan mengungguli banyak arsitektur untuk tugas khusus NLP (Devlin, *et al*., 2019). Model BERT memiliki keunggulan dibandingkan algoritma NLP lainya. Model BERT akan memproses sebuah kata dengan memahami konteks dari kata-kata disekitarnya. Algoritma BERT menganalisis keseluruhan konteks, baik dari kata-kata sebelum atau sesudah dari kata tersebut sehingga menemukan pola yang relevan (Nayla, *et al*., 2023).

Meskipun BERT dapat memberikan kinerja yang baik dalam melakukan tugas-tugas NLP, namun sebuah penelitian yang dilakukan oleh Liu, *et al*., (2019), mengungkapkan bahwa BERT secara signifikan kurang terlatih dan mengusulkan model lain untuk meningkatkan pelatihan model BERT agar menghasilkan kinerja yang lebih baik dibandingkan model BERT tersebut. Pada tahun 2019 tim peneliti Facebook AI berhasil mengembangkan model BERT untuk mengatasi permasalahan tersebut. *Robustly Optimized* BERT *Pre-training Approach* (RoBERTa) adalah pengembangan dari model BERT yang dimodifikasi secara sederhana yaitu dapat melatih model lebih lama dan dengan jumlah data yang lebih banyak, menghapus tugas *next sentence prediction*, melatih data dengan urutan yang lebih panjang, dan menggunakan *dynamic masking* pada data pelatihan. Menurut penelitian Liu, *et al*., (2019), RoBERTa berhasil mencapai hasil yang lebih baik dibandingkan model NLP lainya dalam beberapa *benchmark*, termasuk *General Language Understanding Evaluation* (GLUE), *Reading Comprehension From Examination* (RACE), dan *Stanford Question Answering Dataset* (SQuAD). Model RoBERTa sudah banyak digunakan dalam tugas klasifikasi teks dan memberikan hasil klasifikasi yang baik. Model RoBERTa terbukti memberikan kinerja yang terbaik

dalam mengklasifikasikan emosi dari data teks dibandingkan dengan model *transformer* yang lain (Adoma, *et al*., 2020).

Penelitian tentang klasifikasi teks dengan menggunakan model RoBERTa telah banyak dilakukan sebelumnya. Penelitian pertama dilakukan oleh Murarka, *et al*. (2021). Penelitian tersebut melakukan klasifikasi penyakit mental menggunakan data teks postingan dari media sosial Reddit dan menerapkan *easy data augmentation* untuk mengatasi ketidakseimbangan label. Model yang dibangun memperoleh *accuracy* sebesar 89%, *precission* sebesar 89%, *recall* sebesar 89%, dan *F1-score* sebesar 89%.

Berikutnya penelitian kedua tentang klasifikasi *tweet* pernah dilakukan oleh Basbeth & Fudholi (2024). Penelitian ini melakukan perbandingan kinerja klasifikasi emosi pada *tweet* berbahasa Indonesia menggunakan model BERT, RoBERTa, dan Distil-BERT. Hasil klasifikasi dengan menggunakan model RoBERTa memperoleh *accuracy* sebesar 77.78%, *precision* sebesar 78.07%, *recall* sebesar 77.78%, dan *F1-score* sebesar 77.57%.

Kemudian penelitian ketiga selanjutnya mengenai klasifikasi *tweet* pernah dilakukan oleh Setiadi, *et al*. (2024). Peneltian ini melakukan klasifikasi sentimen pada ulasan hotel berbahasa Inggris dengan menggunakan model RoBERTa. Penelitian ini memperoleh metrik evaluasi dengan *accuracy* sebesar 88.03%, *precision* sebesar 87.60%, *recall* sebesar 88.03%, dan *F1-score* sebesar 87.75%.

Selanjutnya penelitian keempat mengenai klasifikasi teks ujaran kebencian bahasa Indonesia menggunakan model RoBERTa oleh Desiani, *et al*. (2023). Penelitian ini menggunakan teknik *back translation* dan *easy data augmentation* untuk meningkatkan jumlah data pelatihan. Model yang dibangun memperoleh *accuracy* sebesar 85.21%, *precision* sebesar 83.09%, *recall* sebesar 88.78%, dan *F1-score* sebesar 85.84%.

Penelitian kelima mengenai klasifikasi *tweet* berbahasa Inggris dengan menggunakan model RoBERTa pernah dilakukan oleh Tsani dan Suhartono (2023). Penelitian ini melakukan identifikasi kepribadian seseorang ke dalam lima label yaitu *Openness, Conscientiousness, Extraversion, Agreeableness*, dan *Neuroticism*. Penelitian ini menerapkan teknik *back translation* dengan model terjemahan *Text to Text Transfer Transformer* (T5) untuk mengatasi ketidakseimbangan data. Model yang dibangun memperoleh rata-rata *F1-score* untuk dataset pertama sebesar 73% dan untuk dataset kedua sebesar 74.1%.

Berdasarkan beberapa penelitian di atas dapat disimpulkan bahwa model RoBERTa memiliki kinerja yang baik untuk melakukan tugas klasifikasi teks dan analisis sentimen. Selain itu, teknik augmentasi data dapat digunakan pada model berbasis *transformer* untuk mengatasi ketidakseimbangan label dan menambahkan variasi data pelatihan. Namun, masih sedikit penelitian mengenai implementasi model RoBERTa dan teknik augmentasi data pada data *cyberbullying*. Sehingga, hal tersebut menjadi motivasi untuk melakukan penelitian mengenai klasifikasi teks dengan menggunakan RoBERTa dengan teknik augmentasi data *back translation* pada data yang berisikan *tweet* mengenai *cyberbullying*. Oleh karena itu, penelitian ini akan membahas "Implementasi Model *Robustly Optimized* BERT *Pretraining Approach* (RoBERTa) untuk Klasifikasi *Tweet Cyberbullying* dengan Augmentasi *Data Back Translation*".

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang tersebut, adapun rumusan masalah dalam penelitian ini diantaranya:

1. Melakukan klasifikasi pada data *tweet cyberbullying* menggunakan model RoBERTa untuk memprediksi jenis *tweet* yang mengandung unsur *cyberbullying*.
2. Menganalisis pengaruh jumlah data terhadap performa model dalam melakukan klasifikasi *tweet cyberbullying*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun model klasifikasi teks berbasis RoBERTa untuk mengklasifikasikan jenis *cyberbullying* pada data *tweet cyberbullying*.
2. Menerapkan teknik augmentasi data *back translation* untuk mengukur pengaruh penambahan jumlah data terhadap performa model dalam melakukan klasfikasi *tweet cyberbullying*.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Membantu instansi atau *platform* media sosial dalam memantau dan menangani kasus *cyberbullying* secara lebih cepat dan efisien.
2. Mengurangi kebutuhan pengumpulan data baru dengan memanfaatkan teknik augmentasi *back translation* sebagai solusi alternatif untuk menambah variasi data pelatihan dan menangani ketidakseimbangan data.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang berkaitan dengan penelitian ini digunakan sebagai bahan acuan dan perbandingan untuk hasil klasifikasi. Topik penelitian yang menjadi acuan adalah klasifikasi dengan menggunakan metode RoBERTa. Secara umum, gambaran mengenai beberapa penelitian terdahulu akan dijelaskan pada Tabel 1:

Tabel 1. Penelitian terkait klasifikasi dengan metode RoBERTa

No	Penelitian	Data	Metode	Hasil			
				Acc	*Prec*	*Rec*	*F1*
1.	*Classification of Mental Illnesses on Social Media using RoBERTa* (Murarka, *et al*., 2021).	Data Teks: *Depression*: 3062 *Anxiety*: 3027 *PTSD*: 2501 *Bipolar*: 3082 *None*: 2478 Sumber Data: *Scrapping* data Reddit	Metode Klasifikasi: RoBERTa Metode *balancing data*: *Synonim replacement* dengan korpus *WordNet*	89%	89%	89%	89%
2.	Klasifikasi Emosi pada Data Teks Bahasa Indonesia dengan	Data *Tweet*: *Fear*: 691 *Anger*: 1347 *Sadness*:1071 *Love*:709 *Happy*: 1169	Metode Klasifikasi: BERT, RoBERTa, dan Distill-BERT	78%	78%	77%	77%

No	Penelitian	Data	Metode	Hasil			
				Acc	*Prec*	*Rec*	*F1*
	Algoritma BERT, RoBERTa, dan Distil-BERT (Basbeth & Fudholi, 2024).						
3.	Optimisasi Klasifikasi Sentimen Pada Review Hotel Bahasa Inggris dengan Model RoBERTa Twitter (Setiadi, *et al*., 2024).	Data Ulasan: Positif: 5000 Negatif: 4110 Netral: 1660 Sumber Data: *Scrapping data* Agoda, TripAdvisor, Booking.com, dan lain-lain	Metode Klasifikasi: RoBERTa Twitter Metode *Balancing Data: Random Oversampling*	88%	88%	88%	88%
4.	*Back Translation-EDA and Transformer for Hate Speech Classification in Indonesian* (Desiani, *et al*., 2023).	Data *Tweet*: *Hate speech*: 27805 *None-hate Speech*: 30432 Sumber Data: Situs web Github	Metode Klasifikasi: RoBERTa + Leaky RELU *Function*	85%	83%	89%	86%
5.	*Personality Identification from Social using*	Data Teks A: 50112 Data Teks B: 9898	Metode Klasifikasi: BERT dan RoBERTa	-	-	-	73%

No	Penelitian	Data	Metode	Hasil			
				Acc	*Prec*	*Rec*	*F1*
	Ensemble BERT and RoBERTa (Tsani & Suhartono, 2023).	Masing-masing data teks memiliki 5 label (*Openness, Conscientiousness, Extraversion, Agreeableness*, dan *Neuroticism*) Sumber Data *Scrapping data* Twitter Sumber Data Teks B: *Scrapping data* situs web *ChaLearn*	Metode *Balancing Data*: *Back translation T5 Model*				

2.1.1 Penelitian Pertama (Muraka, *et al*., 2021)

Muraka *et al*. (2021), melakukan penelitian klasifikasi penyakit mental berdasarkan postingan media sosial Reddit menggunakan model RoBERTa. Data teks yang digunakan pada penelitian ini diperoleh dengan cara *scraping* data postingan Reddit. Data berjumlah 17159 postingan yang berisikan judul dan teks di mana 3062 data berlabel *depression*, 3027 data berlabel *anxiety*, 2501 data berlabel *ptsd*, 3082 data berlabel *bipolar*, dan 2478 data berlabel *none*.

Data teks terlebih dahulu dilakukan *preprocessing* untuk menghilangkan *URLs, usernames*, dan lain-lain sebelum dibangun model. Setelah itu, untuk menambah variasi data dan mengatasi label yang tidak seimbang maka penelitian ini menerapkan *easy data augmentation* (EDA) menggunakan teknik *synonim replacement*. Hal ini dilakukan karena EDA akan memiliki kinerja yang lebih baik untuk dataset yang

kecil yaitu kurang dari 5000 sampel.

Korpus yang digunakan untuk *synonim replacement* adalah WordNet. Cara kerja dari *synonim replacement* pada penelitian ini adalah dengan memilih sejumlah token secara acak dari setiap postingan. Proses ini melakukan percobaan pada beberapa persentase token yaitu 10%, 50%, dan 100%, namun tidak termasuk *stop words* dan kata dasar. Kemudian setiap token diganti dengan salah satu sinonimnya. Sedangkan untuk teks dan judul akan diterapkan cara kerja yang sama, namun untuk label kelas tidak akan diubah. Hasil terbaik diperoleh ketika 10% token diganti, sehingga penelitian ini menggunakan 10% token yang diubah pada setiap teks.

Penelitian ini menggunakan model berbasis RoBERTa, namun untuk melakukan perbandingan terhadap model yang diusulkan, maka data diterapkan model LSTM dan BERT. Model LSTM yang diterapkan menggunakan parameter seperti 2 *embedding layer* ukuran 100 dan *hidden layer* ukuran 256, *dropout* sebesar 0.5, *optimizer* dengan menggunakan Adam, nilai *learning rate* sebesar 0.005, nilai *batch size* 32, dan dilatih sebanyak 25 *epochs*.

Model yang berbasis *transformer* menerapkan BERT dan RoBERTa *embedding* yaitu menambahkan token khusus di awal [CLS] dan diakhir [SEP]. Selanjutnya menambahkan token *padding* [PAD] untuk memotong menjadi satu panjang konstan. Panjang token yang digunnakan 512 untuk data postingan. Model yang digunakan adalah model *pre-trained* BERT dan RoBERTa. Model *fine-tuning* yang digunakan untuk klasifikasi adalah *BertForSequenceClassification* dan *RobertaFor SequenceClassification*. Parameter yang digunakan untuk model berbasis *transformer* adalah *optimizer* dengan menggunakan Adam, nilai *dropout* sebesar 0.3, nilai *batch size* sebesar 16, nilai *learning rate* sebesar 1×10^{-5}, dan dilatih sebanyak 10 *epochs*. Pada ketiga model yang digunakan dalam penelitian ini, model yang memiliki hasil metriks evaluasi terbaik adalah model RoBERTa. Model ini memperoleh nilai *accuracy* sebesar 89%, *precission* sebesar 89%, *recall* sebesar 89%, dan *F1-score* sebesar 89%.

2.1.2 Penelitian Kedua (Basbeth & Fudholi, 2024)

Basbeth dan Fudholi (2024), melakukan penelitian tentang klasifikasi emosi pada data teks bahasa Indonesia menggunakan algoritma BERT, RoBERTa, dan

Distil-BERT. Data yang digunakan pada penelitian ini berisikan teks dari Twitter yang diperoleh dari *scrapping* media sosial Twitter. Dataset berjumlah 4987 teks yang terbagi menjadi 5 label yaitu 691 data berlabel *Fear*, 1347 data berlabel *Anger*, 1071 data berlabel *Sadness*, 709 data berlabel *Love*, dan 1169 data berlabel *Happy*.

Sebelum masuk ke dalam model, dilakukan *preprocessing* untuk mengelolah data agar mudah digunakan dalam model. *Preprocessing* yang dilakukan adalah menghapus simbol (*username, retweet, hastags, emoticon, URL*), *case folding* yaitu membuat kata menjadi huruf kecil, *tokenization* yaitu membuat kata menjadi sebuah token, *steaming* yaitu membuat kata menjadi kata dasar, *slangword* yaitu menbuat kata singkatan menjadi kata yang baku, *remove whitespace* yaitu menghapus spasi, *remove stopword* yaitu menghapus kata penghubung, dan yang terakhir *label encoding* yaitu membuat label menjadi representasi biner untuk setiap kategori dengan menggunakan model *one hot encoder*. Kemudian dataset dibagi menjadi 80% untuk data *training*, 10% untuk data *validation*, dan 10% untuk data *testing*.

Model pelatihan yang digunakan adalah *bert-base-uncased*, *roberta-base*, dan *distillbert-base-uncased*. Model RoBERTa dalam penelitian ini menggunakan *encoder* dengan 12 lapisan *transformer*¸12 *self-attention head*, *hidden layer* sebesar 768, panjang token sebesar 512 token dan *vocab size* sebesar 50265. Pada pembangunan model dilakukan proses *fine-tuning* dengan menggunakan klasifikasi *Roberta ForSequenceClassification*. Selanjutnya model diterapkan *hyperparameter-tuning* untuk menemukan parameter yang menghasilkan kinerja terbaik untuk model tersebut. Parameter yang terbaik dari proses *hyperparameter-tuning* untuk model RoBERTa antara lain adalah nilai *batch size* sebesar 16, *learning rate* sebesar 1×10^{-5} , dan dilatih selama 10 *epochs*. Hasil klasifikasi yang diperoleh dengan menggunakan model RoBERTa adalah *accuracy* sebesar 78.07%, *precision* sebesar 78.07%, *recall* sebesar 77.78%, dan *F1-score* sebesar 77.57%.

2.1.3 Penelitian Ketiga (Setiadi, *et al*., 2024)

Setiadi, *et al*. (2024), melakukan penelitian mengenai klasifikasi sentimen pada ulasan hotel berbahasa inggris dengan menggunakan model RoBERTa. Data penelitian ini berasal dari data ulasan hotel dengan bahasa inggris. Data diperoleh dengen melakukan *scraping* pada *platform* hotel *online* seperti Agoda, TripAdvisor,

Booking.com, dan lain-lain. Dataset berjumlah 10770 ulasan dengan 5000 ulasan positif, 4110 ulasan negatif, dan 1660 ulasan netral. Selanjutnya, data dilakukan *preprocessing* sebelum dilakukan pelatihan. *Preprocessing* yang dilakukan adalah menghilangkan *noise*, memberikan label sentimen, *text cleaning* untuk menghilangkan simbol-simbol yang tidak dibutuhkan, *spliting data* (80% untuk data *train*, 10% untuk data *validation*, dan 10% untuk data *testing*), tokenisasi data dan *label encoder*.

Penelitian ini melakukan *hyperparameter-tuning* untuk menentukan kombinasi parameter optimal yang mempengaruhi kinerja model. Parameter yang di *tuning* adalah *epochs, batch size, warmup steps*, dan *weight regularization*. Parameter terbaik yang dihasilkan dari proses *hyperparameter-tuning* adalah pelatihan sebanyak 3 *epochs*, nilai *batch size* sebesar 16, jumlah *warmup steps* sebanyak 500, dan *weight regularization* sebesar 0.02. Hasil metrik evaluasi terbaik yang didapatkan adalah *accuracy* sebesar 88.03%, *precision* sebesar 87.60%, *recall* sebesar 88.03%, dan *F1-score* sebesar 87.75%.

2.1.4 Penelitian Keempat (Desiani, *et al*., 2023)

Desiani, *et al*. (2023), melakukan penelitian tentang yaitu klasifikasi teks ujaran kebencian bahasa indonesia dengan menggunakan model RoBERTa. Data penelitian ini diperoleh dari situs web Github yang berisikan 13169 data teks yang dikumpulkan melalui media sosial Twitter dalam bahasa Indonesia. Sebelum membangun model, dilakukan tahapan *preprocessing data*, antara lain adalah *case folding, remove punctuation, stopword removal*, dan *stemming.*

Penelitian ini menggunakan teknik augmentasi data untuk meningkatkan kualitas data pelatihan. Teknik-teknik yang digunakan adalah *back translation* dan *easy data augmentation* (EDA). Cara kerja teknik augmentasi pada penelitian ini adalah yang pertama data teks yang berbahasa Indonesia diterjemahkan ke dalam bahasa Inggris, kemudian empat metode EDA diterapkan dalam data bahasa inggris. Keempat metode EDA tersebut antara lain, *random insertion* yaitu mencari sinonim dari kata acak dalam sebuah kalimat dan menambahkan kata sinonim tersebut dalam sebuah kalimat, *random deletion* yaitu menghapus kata acak dalam sebuah kalimat, *random swap* yaitu memilih secara acak dua kata dalam sebuah kalimat dan menukar posisi kedua kata tersebut, dan yang terakhir adalah *synonim replacement* yaitu memilih

satu kata secara acak lalu mengganti kata tersebut dengan sinonimnya. Setelah diterapkan metode EDA maka data diterjemahkan kembali ke dalam bahasa Inggris. Setelah proses augmentasi dilakukan, jumlah data bertambah menjadi 58237 dengan 27805 data berlabel *hate speech* dan 30432 data berlabel *none-hate speech*.

Selanjutnya data dibagi menjadi 80% data *train* dan 20% data *testing*. Selanjutnya dilakukan RoBERTa *embdiding* untuk menambahkan token khusus diawal [CLS] dan diakhir [SEP]. Kemudian menambahkan token *padding* [PAD] untuk memotong teks menjadi sama panjang yang konstan. Pada penelitian ini klasifikasi dilakukan menggunakan model RoBERTa dengan parameter nilai *batch size* 16, fungsi aktivasi *Leaky Relu*, dan *epochs* sebesar 70. Hasil metriks evaluasi menghasilkan *accuracy* sebesar 85.21%, *precision* sebesar 83.09%, *recall* sebesar 88.78%, dan *F1-score* sebesar 85.84%.

2.1.5 Penelitian Kelima (Tsani & Suhartono, 2023)

Tsani dan Suhartono (2023), melakukan penelitian mengenai identifikasi kepribadian dari media sosial dengan menggunakan model BERT dan RoBERTa. Penelitian ini menggunakan 2 dataset yang diperoleh dari *scraping* di media sosial Twitter dan *website* ChaLearn. Dataset pertama adalah data yang berisikan tweet dengan jumlah 46000 teks. Sedangkan dataset kedua berisikan video pendek dengan transkrip bahasa Inggris berbasis teks dengan jumlah 10000 data. Data dilakukan *preprocessing* terlebih dahulu sebelum dimasukan ke dalam model. Tahapan *preprocessing* yang dilakukan adalah *remove URL, remove the symbol, translate* bahasa *into English , case folding, remove stop words*, dan lematisasi.

Pada penelitian ini untuk mengatasi data yang memiliki label tidak seimbang adalah dengan menerapkan augmentasi data menggunakan *back translation*. Cara kerja dari *back translation* pada penelitian ini adalah dengan menerjemahkan teks *tweet* dan transkrip Youtube ke dalam bahasa Jerman dan kemudian diterjemahkan kembali ke bahasa Inggris dengan model terjemahan *Text to Text Transfer Transformer* (T5). Model T5 digunakan untuk penerjemahan disebabkan model ini dapat menghasilkan parafrase yang baik dari bahasa aslinya.

Selanjutnya menerapkan RoBERTa *embdiding* untuk menambahkan token khusus diawal [CLS] dan diakhir [SEP]. Kemudian menambahkan token *padding* [PAD]

untuk memotong teks menjadi sama panjang yang konstan. Kemudian data dilatih dengan menggunakan model *pre-trained* RoBERTa dan dilakukan *fine-tuning* dengan menggunakan model klasifikasi *RobertaForSequenceClassification*. Parameter model RoBERTa yang digunakan untuk klasifikasi pada penelitian ini antara lain adalah *batch size* sebesar 16 untuk data Twitter, sedangkan untuk dataset Youtube menggunakan *batch size* sebesar 32, *optimizer* Adam, nilai *learning rate* sebesar 1×10^{-5}, dan data dilatih pada 10 *epochs*.

Penelitian ini menggunakan enam model percobaan yaitu model pertama menggunakan model BERT, model kedua menggunakan model RoBERTa, model ketiga menggunakan model BERT dengan augmentasi data, model keempat menggunakan model RoBERTa dengan augmentasi data, model kelima menggunakan gabungan model BERT dan RoBERTa, sedangkan model keenam yaitu model yang diusulkan dalam penelitian ini menggunakan gabungan model BERT dan RoBERTa dengan augmentasi data. Metriks evaluasi memberikan hasil yang terbaik terhadap model keenam. Hasil klasifikasi dataset Twitter dengan model keenam menghasilkan rata-rata *F1-score* sebesar 0.730, di mana hasil ini lebih tinggi 5% dibandingkan dengan model lainya tanpa proses augmentasi data. Sementara itu, untuk klasifikasi dataset Youtube dengan model keeenam dan menggunakan augmentasi data menghasilkan nilai rata-rata *F1-score* sebesar 0.741, hasil ini lebih tinggi 3% dibandingkan dengan model lainya tanpa augmentasi data.

2.2 Klasifikasi Teks

Klasifikasi adalah proses pengelompokan objek ke dalam suatu kategori atau kelas berdasarkan karakteristik yang sudah didefinisikan sebelumnya. Menurut Budiman, *et al*. (2015), Klasifikasi adalah suatu proses membangun model atau fungsi yang dapat menjelaskan dan membedakan berbagai konsep dan kelas data, sehingga dapat digunakan untuk memprediksi kelas dari objek yang belum diketahui. Klasifikasi dapat dilakukan melalui pendekatan parametrik dan non parametrik. Pendekatan parametrik adalah pendekatan yang bergantung terhadap asumsi-asumsi distribusi data sehingga jika asumsi tidak terpenuhi maka hasil yang diperoleh tidak valid. Sedangkan untuk pendekatan non parametrik adalah pendekatan yang digunakan untuk mengatasi keterbatasan dari pendekatan parametrik. Pendekatan ini tidak bergantung pada asumsi distribusi tertentu, sehingga memberikan kemudahan dalam analisis data, namun tetap mampu memberikan hasil dengan tingkat akurasi yang

tinggi (Yona, 2021).

Klasifikasi memiliki dua jenis yang umum digunakan yaitu *binary classification* dan *multiclass classification*. *Binary classification* atau klasifikasi biner adalah kegiatan yang mengklasifikasikan data ke dalam dua kelas, sedangkan *multiclass classification* atau klasifikasi multikelas merupakan kegiatan yang mengklasifikasikan data pada lebih dari dua kelas yang berbeda (Tantika & Kudus, 2022). Klasifikasi biner memisahkan data ke dalam dua kelas yang berlawanan seperti "Positif" atau "Negatif" dan "Laki-laki" atau "Perempuan". Sementara itu, klasifikasi multikelas digunakan ketika terdapat lebih dari dua kelas yang saling eksklusif, seperti klasifikasi jenis penyakit yang dialami pasien berdasarkan diagnosis berbagai kategori penyakit.

Klasifikasi teks adalah teknik yang umum digunakan dalam *machine learning* dan merupakan salah satu bagian dari bidang NLP. Klasifikasi teks adalah proses mengelompokan dokumen atau teks ke dalam kelas yang sudah ditentukan sebelumnya berdasarkan kontennya. Klasifikasi teks biasanya terdiri dari empat tahapan utama yaitu ektrasi fitur, reduksi dimensi, pemilihan algoritma klasifikasi, dan evaluasi hasil (Kowsari, *et al*, 2019).

2.3 *Natural Language Processing*

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer dan *machine learning* yang berfokus tentang pengembangan dan pengelolaan sistem yang dapat berinteraksi dengan bahasa manusia secara alami. Tujuan utama NLP adalah memungkinkan komputer untuk memahami, menganalisis, memproses dan menghasilkan bahasa manusia dengan cara yang efektif dan sesuai konteks (Arrasyid, *et al*., 2024). Seiring dengan perkembangan zaman, penggunaan NLP semakin meningkat setiap harinya dan NLP dapat menghasilkan data teks yang tidak terstruktur dalam jumlah besar setiap harinya. Beberapa teknik utama dalam NLP adalah tokenisasi *,stemming, lemmatization, part-of-speech tagging,* dan *named entity recognition*. Teknik-teknik tersebut memungkinkan komputer memahami struktur dan makna teks. NLP sudah banyak diterapkan secara luas, seperti pada mesin terjemahan otomatis, klasifikasi teks, mesin pencarian, dan lain-lain (Eisenstein, 2018). Berbagai bidang pengaplikasian yang memanfaatkan NLP antara lain adalah *information retrieval, information extraction, question-answering,*

summarization, machine translation, dan *dialogue systems* (Liddy, 2021).

2.4 *Text Mining*

Text Mining atau penambangan teks adalah proses yang dilakukan oleh komputer untuk memperoleh informasi yang baru atau belum diketahui sebelumnya dan menemukan kembali informasi yang tersembunyi secara implisit. Proses ini bertujuan mengekstrak data yang tidak terstruktur secara otomatis hingga menjadi semi terstruktur (Feldman & Sanger, 2007). *Text mining* adalah proses yang biasa digunakan untuk mengatasi masalah klasifikasi, klastering, *information extraction*, dan *information retrival*. Proses kerja dari *text mining* memiliki banyak kemiripan dengan *data mining*, yang menjadi perbedaannya adalah pola yang digunakan *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur sedangkan dalam *data mining* pola yang diambil dari data yang terstruktur (Han & Kamber, 2006).

Text mining dapat memberikan solusi untuk permasalahan yang sering ditemukan dalam pemrosesan dan analisis data yang tidak terstruktur dalam jumlah besar. Data yang diolah pada proses ini berasal dari teks yang tidak terstruktur dan berjumlah besar, sehingga dibutuhkan pendekatan yang berbeda dengan analisis data numerik biasa yaitu *text preprocessing*. *Text preprocessing* adalah proses pembersihan, penghilangan, dan pengubahan isi data seperti simbol, karakter non-alfabet maupun kata-kata yang dianggap tidak diperlulkan pada proses selanjutnya (Aulia, *et al*., 2023). Menurut Surjandari, *et al*., (2016), beberapa langkah yang dilakukan dalam *Text preprocessing* adalah sebagai berikut:

1. Tokenisasi
   Tokenisasi adalah proses pemisahan dokumen menjadi bagian-bagian kata yang disebut sebagai token.
2. *Case Folding*
   Case Folding adalah proses mengubah semua huruf besar menjadi huruf kecil pada sebuah dokumen.
3. Normalisasi
   Normalisasi adalah proses pergantian kata yang salah eja atau disingkat menjadi bentuk yang benar.
4. *Filtering*
   Filtering adalah proses menghilangkan *mention* (@), *hashtag* (#), *url*, tanda baca, *emoticon*, dan karakter non abjad.

5. *Lemmatization*

   Lemmatization adalah proses penghapusan akhiran atau imbuhan dari sebuah kata sehingga hanya menyisakan kata dalam bentuk dasar.

2.5 *Imbalanced Data*

Imbalanced data atau data yang tidak seimbang adalah salah satu masalah yang sering ditemukan dalam melakukan klasifikasi. *Imbalanced data* terjadi ketika data tidak terdistribusi secara merata dengan jumlah kelas minoritas lebih kecil dibandingkan dengan kelas mayoritas. Kelas minoritas adalah kelas yang memiliki persentase lebih kecil dari kelas lainya, sedangkan kelas mayoritas adalah kelas yang memiliki persentase lebih besar dari seluruh kelas. Model akan sulit mengklasifikasikan kelas minoritas dengan baik daripada kelas mayoritas pada data yang tidak seimbang, terkadang kelas minoritas memiliki informasi yang lebih penting (Ustyannie & Suprapto, 2020). Masalah ketidakseimbangan data dapat diatasi dengan mencari keseimbangan jumlah data pada setiap kelas, baik kelas mayoritas maupun kelas minoritas.

Mengatasi ketidakseimbangan data terbukti dapat menambah variasi data pelatihan yang tidak seimbang dan meningkatkan akurasi (Muliono, *et al*., 2022). Berbagai upaya untuk mengatasi data yang tidak seimbang, salah satunya adalah *resampling*. Menurut Kaope dan Pristyanto (2023), terdapat tiga metode *resampling* yaitu *undersampling, oversampling*, dan *hybrid sampling*. *Undersampling* adalah metode yang mengurangi jumlah kelas mayoritas sampai jumlahnya sama dengan kelas minoritas. Kemudian, *oversampling* adalah metode yang memilih secara acak data dari kelas minoritas, sehingga menghasilkan data baru. *Hybrid sampling* adalah kombinasi dari metode *oversampling* dan *undersampling*. Metode pengambilan sampel ini menambahkan objek baru ke kelas minoritas dan mengurangi objek dari kelas mayoritas untuk menyeimbangkan data. Metode *undersampling* jarang digunakan disebabkan dapat mengurangi atau mengambil informasi yang penting di dalam dataset. Namun, metode *oversampling* dapat meningkatkan risiko *overfitting* karena dapat menyebabkan duplikasi pada dataset (Fithriasari, *et al*., 2020).

2.6 *Data Augmentation*

Salah satu cara untuk mengatasi data yang tidak seimbang dan menambah variasi data pelatihan dataset berbentuk teks adalah *data augmentation. Data augmentation* adalah proses menghasilkan teks baru dari data yang sudah ada dengan memiliki arti dan konteks yang sama tetapi dengan pilihan kata atau struktur bahasa yang berbeda (Taylor & Nitschke, 2018). Pada klasifikasi teks, salah satu tantangan utamanya adalah menghasilkan teks baru tanpa mengubah label aslinya. Menurut penelitian yang dilakukan Wei dan Zou (2019), menyatakan bahwa penerapan *data augmentation* dapat meningkatkan kinerja klasifikasi dan mengurangi *overfitting* dibandingkan dengan dataset yang lebih kecil. Hal itu dapat disimpulkan bahwa dengan bantuan augmentasi data, ketika ukuran data pelatihan semakin meningkat, maka kinerja model juga akan meningkat. Teknik augmentasi data yang umum digunakan untuk menangani data yang tidak seimbang dan menambah variasi data pelatihan adalah *back translation* dan *Easy Data Augmentation* (EDA).

2.6.1 *Back Translation*

Back translation adalah metode menerjemahkan teks asli ke dalam bahasa target dan kemudian diterjemahkan kembali ke bahasa asal. Hasil Proses ini menghasilkan dataset yang lebih bervariasi dan dapat digunakan untuk meningkatakan performa dari model pelatihan (Bucos & Tucudean, 2023). Data teks yang diperoleh setelah proses *back translation* akan memiliki perbedaan dalam pilihan kata dan struktur bahasa, namun perubahan yang dilakukan dalam proses ini akan mempertahankan makna dari data teks asli, sehingga perbedaan semantik tidak mempengaruhi pemodelan. Teknik *back translation* efektif dalam mengatasi masalah *overfitting* dan menambah variasi data pelatihan. Menurut penelitian yang dilakukan oleh Edunov, *et al*. (2018), penerapan *back translation* dalam mesin penerjemah akan memberikan hasil data augmentasi yang mendekati keakuratan dengan teks asli.

Gambar 1. Contoh skema proses *back translation* (Beddiar, *et al*., 2021).

2.6.1.1 MarianMT

MarianMT adalah model terjemahan otomatis yang dikembangkan oleh Jörg Tiedemann dari *Technology Research Group* di *University of Helsinki*. Model ini dilatih di *Open Parallel Corpus* (OPUS) yaitu kumpulan data multibahasa dengan menggunakan *framework neural machine translation* yang disebut sebagai Marian MT (Chaudhary, 2020). Model MarianMT merupakan model yang dibangun khusus untuk menyelesaikan tugas yang berkaitan tentang *Neural Machine Translation* (NMT). Model MarianMT menggunakan arsitektur *transformer encoder-decoder* dengan 6 lapisan di setiap komponen dengan 8 *head attention* di setiap lapisan. (Soliman, *et al*., 2022). Model ini telah dilatih untuk berbagai pasangan bahasa dan model ini didukung untuk pelatihan dan penerjemahan yang cepat. Oleh karena itu, model ini dapat digunakan dalam tugas NLP untuk mendapatkan variasi teks terjemahan yang dapat digunakan untuk menambah variasi dataset tanpa mengubah makna aslinya (Junczys-Dowmunt, *et al*, 2018).

Penerjemahan dengan MarianMT dapat dimodelkan pada berbagai tingkat seperti tingkat dokumen, paragraf, atau kalimat. Namun, berbagai penelitian lebih berfokus pada penerjemahan di tingkat kalimat. MarianMT adalah model yang menggunakan arsitektur berbasis *transformer*, maka lapisan *encoder* dan *decoder* adalah arsitektur utama dari MarianMT. *Encoder* digunakan untuk menghitung representasi kalimat sumber, sedangkan *decoder* digunakan untuk menghasilkan kalimat target dari representasi *encoder*. Misalkan diberikan sebuah kalimat sumber $x = (x_1, \ldots, x_S)$ dengan panjang S dan sebuah kalimat target $y = (y_1, \ldots, y_T)$ dengan panjang T. MarianMT adalah model yang bersifat auto-regresif sehingga distribusi probabilitas untuk kalimat target $P(y|x)$ dapat dilihat seperti pada Persamaan (1) (Tan, *et al*., 2020).

$$P(y|x) = \prod_{t=1}^{T} P(y_t|y_0, \ldots, y_{t-1}, x) \tag{1}$$

Keterangan:
y_t = Token atau kata ke-t dalam kalimat target
y_1^{t-1} = Urutan kata yang sudah diprediksi sebelumnya dalam kalimat target
x = Kalimat sumber

Setiap token x_i direpresentasikan ke dalam ruang vektor dengan dimensi d_{model} melalui *embedding* $E(x_i) = \mathbf{e}_i \in \mathbb{R}^{d_{model}}$, di mana $E(x_i)$ merupakan baris ke-x_i matriks *embedding* berukuran $|V| \times d_{model}$ dengan $|V|$ adalah ukuran kosakata.

MarianMT adalah model berbasis *self-attention* yang hanya memperhatikan persamaan konteks dalam kalimat tanpa memperhatikan urutan. Kekurangan model ini dapat diatasi melalui *positonal encoding* dengan fungsi untuk menambahkan informasi posisi ke dalam setiap representasi kata. *Positional Encoding* merupakan vektor dengan dimensi d_{model} yang ditambahkan ke dalam *embedding* untuk memberikan urutan posisi kata dalam sebuah kalimat. *Positional encoding* memiliki dimensi yang sama dengan *embedding*, sehingga keduanya dapat dijumlahkan dan membentuk vektor input *embedding*. Cara kerja *positional encoding* menggunakan kombinasi fungsi sinus dan kosinus dengan frekuensi untuk merepresentasikan posisi seperti pada Persamaan (2) dan (3) (Vaswani, *et al*., 2017).

$$PE_{(\text{pos, 2i})} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{2}$$

$$PE_{(\text{pos, 2i+1})} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3}$$

Keterangan:
$pos =$ Posisi dalam urutan
$i =$ Indeks dimensi *embedding*

MarianMT menggunakan mekanisme *attention* yang memungkinkan model fokus pada bagian-bagian tertentu dari kalimat sumber saat menghasilkan terjemahan dalam kalimat target. Pada lapisan *encoder* menggunakan *self-attention*, sedangkan lapisan *decoder* menggunakan mekanisme *self-attention* dan *cross attention*. Mekanisme *attention* dapat menghitung relevansi setiap vektor nilai berdasarkan *query* dan *key*. Secara formal, diberikan satu set m vektor *query* $Q \in \mathbb{R}^{m\times d}$, satu set n vektor *query* $K \in \mathbb{R}^{m\times d}$, dan satu set vektor *value* yang terkait $V \in \mathbb{R}^{m\times d}$. Perhitungan dalam attention melibatkan dua tahap. Tahap pertama adalah menghitung relevansi antara suatu *key* dan *value* dengan menggunakan Persamaan (4) (Tan, *et al*., 2020).

$$\textit{Attention score}(i,j) = \frac{Q_i K_j^T}{\sqrt{d_k}} \tag{4}$$

d_k adalah dimensi dari vektor *key*. *Attention score* adalah matriks yang menyimpan skor relevansi antara setiap *key* dan *value*. Setelah menghitung semua *attention score* untuk setiap pasangan token, maka fungsi *softmax* akan diterapkan untuk menormalkan *attention score* dan menghasilkan *attention weights* yang menunjukan

seberapa penting setiap token dengan token lainya. Fungsi *softmax* yang diterapkan pada setiap *attention score* dapat dilihat pada Persamaan (5) (Tan, *et al*., 2020).

$$\textit{softmax}(z)_j = \frac{\exp(z_j)}{\sum_{i=1}^{|V|} \exp(z_i)} \tag{5}$$

Keterangan:
$z_j =$ *Attention score* untuk kata ke-j
$\sum_{i=1}^{|V|} \exp(z_i) =$ jumlah eksponensial dari seluruh *attention score*

Tahap kedua adalah menghitung vektor *output*. Hasil dari proses ini adalah representasi yang memperhitungkan konteks kata-kata lain dalam kalimat (Stahlberg, 2020). Setiap vektor *query*, vektor *output* yang sesuai dinyatakan sebagai jumlah berbobot dari vektor-vektor nilai yang di representasikan pada Persamaan (6).

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

Setelah melewati lapisan *self-attention* pada *encoder* dan memperoleh sebuah vektor konteks yang sudah diproses. Vektor tersebut akan melalui lapisan *feed-forward network* yang berfungsi untuk memperkaya dan memperdalam representasi kata yang sudah diproses. Pada proses matematisnya menggunakan dua transformasi linier yang dipisahkan dengan fungsi aktivasi ReLU yang memungkinkan model dapat menangkap hubungan antar kata secara lebih kompleks. Hasil representasi tersebut digunakan dalam lapisan berikutnya.

Hasil akhir pada lapisan *encoder* merupakan sebuah matriks vektor yang memuat informasi kontekstual di setiap token pada kalimat sumber. Matriks vektor tersebut akan digunakan sebagai *input* pada lapisan *decoder*. Lapisan *decoder* berfungsi untuk menghasilkan kalimat target berdasarkan representasi yang dihasilkan oleh *encoder*. Tahapan pertama adalah memulai dengan token khusus seperti `<sos>` yang menandakan awal kalimat yang akan diterjemahkan. Token `<sos>` adalah *input* pertama untuk *decoder* yang akan diteruskan ke lapisan pertama dari *decoder*, yaitu *Masked Self-Attention*. Lapisan *Masked Self-Attention* berfungsi untuk memungkinkan model memproses setiap token dalam *output* yang sedang dihasilkan dengan hanya mempertimbangkan kata-kata yang muncul sebelumnya, tanpa melihat kata yang akan datang. Proses ini dilakukan dengan melakukan *masking* atau penutupan kata-kata yang belum diprediksi. Proses perhitungan *attention* sama

seperti pada Persamaan (6) di lapisan *encoder*. Namun, ketika menghitung hasil *attention score* pada bagian yang di *masking* akan menghasilkan nilai $-\infty$. Hal tersebut akan membuat hasil dari nilai *softmax* bernilai 0. Sehingga, hal tersebut akan membuat token yang di *masking* tidak diperhatikan dan diabaikan saat menghitung representasi konteks.

Hasil dari lapisan *masked self-attention* akan diproses ke lapisan selanjutnya yaitu *cross-attention* yang menghubungkan informasi dari *encoder* dengan *decoder*. Pada lapisan ini, *decoder* menggunakan representasi yang dihasilkan oleh *encoder* untuk menyelaraskan konteks antara *input* dan *output*. Lapisan *cross-attention* menggunakan representasi yang dihasilkan oleh *encoder* sebagai *key* dan *value*, sedangkan *query* berasal dari *output* sebelumnya di *decoder*. Perhitungan matematis untuk *attention* pada lapisan ini sama dengan perhitungan pada lapisan *encoder*. Setelah itu, hasil dari *cross-attention* akan diteruskan ke *Feed Forward Network* (FNN). Lapisan FFN bertugas memproses representasi gabungan berdasarkan informasi dari *input encoder* dan token yang sudah diproses sebelumnya pada *decoder*. Setelah itu, akan diterapkan *layer normalization* untuk menstabilkan pelatihan.

Setelah proses *layer normalization*, *output* dari lapisan *decoder* akan diteruskan melalui *residual connection*. Proses ini memungkinkan *input* asli dari lapisan sebelumnya ditambahkan kembali ke *output* setelah dinormalisasi, sehingga model dapat mempertahankan informasi yang telah diproses sebelumnya. Selanjutnya, *output* diproyeksikan ke ruang kosakata $R^{|V|}$ menggunakan lapisan linear yang berfungsi untuk mengubah dimensi *ouput decoder* yang berukuran d_{model} menjadi kosakata $|V|$. Langkah selanjutnya adalah menerapkan fungsi *softmax* untuk mengubah vektor *output* menjadi distribusi probabilitas (Tan, *et al*., 2020). Setelah probabilitas dihitung, maka token dengan probabilitas tertinggi akan dipilih sebagai token berikutnya dalam urutan *output*. Proses ini akan berulang untuk setiap token yang dihasilkan hingga kalimat selesai diterjemahkan, dengan model memasukkan token yang dihasilkan pada langkah sebelumnya sebagai input *decoder* untuk menghasilkan token berikutnya.

2.6.2 *Easy Data Augmentation*

Easy Data Augmentation (EDA) adalah pendekatan lain yang umum digunakan untuk teknik augmentasi data. Metode ini dapat digunakan untuk mengatasi

ketidakseimbangan data dan meningkatkan kinerja pada tugas klasifikasi teks. Menurut Wei & Zou (2019), Terdapat empat teknik EDA yang dapat dilakukan yaitu:

1. *Synonim Replacement*
   Synonim replacement atau pergantian sinonim adalah teknik yang memilih kata secara *random* dalam sebuah kalimat dan merubah kata tersebut dengan sinonimnya.
2. *Random insertion*
   Random insertion atau penyisipan acak adalah teknik yang menyisipkan sinonim dari kata-kata yang ada di dalam kalimat yang bukan termasuk *stopwords* secara *random*.
3. *Random swap*
   Random swap atau penukaran acak adalah teknik yang memilih secara *random* dua kata yang ada didalam sebuah kalimat dan menukar posisi kedua kalimat tersebut.
4. *Random deletion*
   Random deletion atau penghapusan acak adalah teknik yang menghapus kata tertentu yang ada di dalam kalimat dengan tidak mengubah konteks yang ada didalam kalimat tersebut.

2.7 *Word Embedding*

Menurut Nurdin, *et al*. (2020), pengembangan teknik representasi kata ke dalam bentuk vektor terus dikembangkan dalam bidang NLP. Teknik ini memiliki dampak yang penting terhadap kinerja model yang dibangun. Konsep tradisional seperti *one-hot encoding* merepresentasikan kata ke dalam vektor biner dengan dimensi tinggi. *Word embedding* atau representasi kata adalah fungsi yang dapat memetakan setiap kata ke dalam vektor dengan dimensi besar (Nurdin, *et al*., 2020). *Word embedding* telah umum digunakan dalam aplikasi NLP, hal ini disebabkan *word embedding* memiliki kemampuan representasi vektor yang menangkap makna dan hubungan antar kata. Teknik ini telah umum digunakan sebagai *input* untuk model *machine learning* karena memungkinkan model *machine learning* untuk memahami konteks data mentah dengan lebih baik (Wang, *et al*., 2018). Beberapa contoh teknik *word embedding* adalah Word2Vec, FastText, dan Glove. Meskipun teknik tersebut memiliki dimensi yang lebih kecil dan lebih efektif dibandingkan dengan *one-hot encoding*, namun pendekatan ini memiliki kelemahan seperti tidak mampu

menangkap konteks dinamis dalam sebuah kalimat, karena representasi kata dalam *embedding* ini bersifat statis.

Teknik *word embedding* melalui pendekatan *pre-trained word embedding* atau *embedding* yang telah dilatih sebelumnya merupakan metode untuk mengatasi beberapa kelemahan dari *word embedding* klasik. *Embedding* tersebut sudah dilatih dengan menggunakan dataset besar pada domain tertentu yang dapat digunakan untuk menyelesaikan masalah di bidang lain. Salah satu perkembangan dalam *pre-trained word embedding* adalah model berbasis *transformers*, seperti BERT. Pendekatan ini akan menghasilkan representasi kata yang lebih dinamis dengan kata-kata di sekitarnya dan menangkap makna kata secara lebih kontekstual (McCormick & Ryan, 2024).

Model BERT menggunakan tiga jenis lapisan *embedding* berbeda untuk merepresentasikan data ke dalam vektor numerik, yaitu *token embedding*, *positional embedding*, dan *token type embedding*. Proses pertama dalam BERT *embedding* adalah tokenisasi dengan menggunakan BERT *tokenizer*. Proses ini akan mengubah *input* ke dalam daftar ID token integer yang digunakan dalam model BERT, di mana setiap ID langsung memetakan ke kata atau bagian kata dalam *string* asli. Sebagai contoh, terdapat teks ”hello, world!”. Sebelum masuk ke dalam *embedding*, maka kalimat tersebut akan dipecah menjadi token-token yang lebih kecil dan diberikan token khusus [CLS] diawal teks dan token khusus [SEP] untuk mengakhiri suatu teks (Novack, 2024). Setiap token tersebut akan direprsentasikan kedalam sebuah angka yang dapat dilihat pada Tabel 2.

Tabel 2. Contoh hasil *Token Embeddinng*

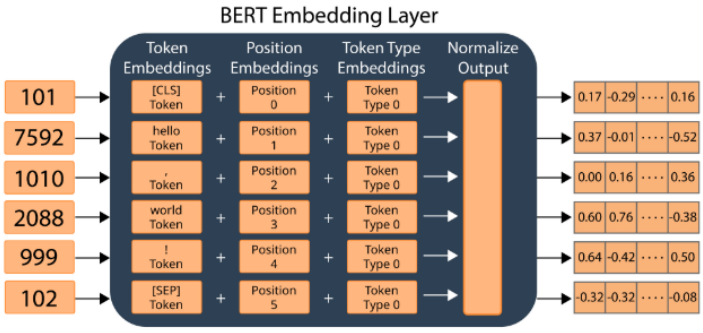
CLS	hello	,	world	!	SEP
101	7592	1010	2088	999	102

Setiap token memiliki ID yang direpresentasikan sebagai angka berdasarkan kosakata yang telah dilatih oleh model BERT bukan dari angka acak. Model BERT memiliki *vocabulary* dengan ukuran 30.522 kata dan sub-kata. *Vocabulary* ini berisikan token dan ID numeriknya yang digunakan untuk merepresentasikan kata atau sub-kata dalam suatu teks. Selanjutnya, ID tersebut digunakan sebagai indeks untuk mengambil representasi vektor numerik dari matriks *embedding*. Model BERT berisi *embedding* yang telah dilatih dengan tujuan untuk mengubah token menjadi representasi vektor yang sesuai (Novack, 2024).

Selain *token embedding*, model BERT juga menggunakan *position embedding* untuk memahami urutan kata dalam sebuah kalimat. Meskipun BERT memiliki 30.522 *token embeddings*, tetapi jumlah *position embeddings* hanya terdapat 512. Hal ini disebabkan oleh panjang *input* maksimum yang dapat diproses BERT hanya 512 token (Novack, 2024). Ilustrasi *postion embedding* dapat dilihat pada Gambar 2.

Gambar 2. Ilustrasi *Position Embedding* (Novack, 2024).

Jenis *embedding* terakhir yang digunakan model BERT adalah *toke type embeddings* atau yang dikenal sebagai *segment embeddings*. Model BERT dilatih untuk menentukan apakah sebuah kalimat B mengikuti kalimat A. Model BERT menggunakan dua jenis *token type embeddings*, yaitu tipe 0 untuk merepresentasikan kalimat pertama dan tipe 1 untuk merepresentasikan kalimat kedua. Selanjutnya, BERT *embeddings* menghitung *embedding* akhir untuk setiap token dengan menggabungkan ketiga nilai dari *token embeddings*, *position encoding*, dan *token type embeddings*. Kemudian menerapkan normalisasi pada jumlah tersebut (Novack, 2024). Ilustrasi BERT *embedding layer* dalam mengitung *embedding* dapat dilihat pada Gambar 3.



Gambar 3. Ilustrasi BERT *Embeddings Layer* (Novack, 2024).

Pengembangan konsep *embedding* BERT terus dilakukan dan menghasilkan varian yang lebih canggih, salah satunya adalah RoBERTa. Tim peneliti dari Facebook AI pada tahun 2019 telah melakukan pengembangan modifikasi model BERT yaitu RoBERTa (*Robustly Optimized* BERT *Pretraining Approach*). Model RoBERTa adalah salah satu pendekatan model berbasis *transformer* dengan menggunakan

mekanisme *self-attention* untuk memperoleh representasi teks yang memahami hubungan kontekstual antar kata dan nuansa bahasa lebih efektif. Model RoBERTa menggunakan teknik tokenisasi subword berbasis *Byte-Pair Encoding* (BPE), yang artinya kata-kata yang jarang atau belum pernah muncul akan direpresentasikan ke dalam bentuk *subword* yang terdiri dari suku kata yang sering muncul. RoBERTA hanya menggunakan dua jenis *embeddings* yaitu *token embeddings* dan *position encoding* tanpa *token type embeddings*. Model RoBERTa memiliki kemampuan generalisasi yang lebih baik dan mampu menangani bahasa atau kata yang baru lebih efisien (Liu, *et al*., 2019).

2.8 *Splitting Data*

Splitting data atau pembagian data adalah teknik yang digunakan untuk membagi dataset menjadi dua data yang terpisah yaitu data *train* atau data pelatihan dan data *testing* atau data pengujian (Joseph, 2022). *Splitting data* merupakan proses yang diperlukan dalam model *machine learning* untuk mengurangi bias dan mencegah *overfitting* pada data *training*. Menurut Raykar & Saha (2015), pendekatan klasik untuk membangun model prediktif yang baik adalah dengan membagi data menjadi tiga bagian, yaitu data *train* untuk menyesuaikan dan melatih model , data *validation* untuk memperkirakan kesalahan prediksi untuk pemilihan model, dan data *testing* untuk memperkirakan kinerja model akhir yang dipilih. Tingkat *error* adalah perbadingan sampel yang diprediksi benar terhadap jumlah total sampel dalam data *testing* (Muraina, 2022).

2.9 *Deep Learning*

Deep learning adalah bagian dari *machine learning* yang didasarkan pada cara kerja otak manusia memproses informasi. *Deep learning* merupakan salah satu cabang ilmu *machine learning* yang berbasis Jaringan Saraf Tiruan (JST) (Ilahiyah & Nilogiri, 2018). *Deep learning* memiliki arsitektur yang lebih kompleks dan menggunakan lebih banyak lapisan dibandingkan *neural network*, sehingga hal ini membuat *deep learning* lebih efektif dalam menangani masalah yang lebih kompleks dengan jumlah data yang besar seperti mengenali teks, gambar, dan suara (Purnama, 2021).

Deep leaning memiliki pendekatan yang lebih adaptif yaitu dapat menyesuaikan diri dengan data baru yang belum pernah ditemui sebelumnya. Model *deep learning* memiliki beberapa komponen utama yang membuat model mempelajari pola dari data, membuat prediksi, dan meningkatkan kinerja secara bertahap. Komponen tersebut terdiri dari *layer*, *loss function*, *optimization algorithms*, *forward propagation* dan *backpropagation* (Mienye & Swart, 2021).

Salah satu contoh algoritma *deep learning* adalah RoBERTa. model RoBERTa adalah pengembangan dari model BERT yang memiliki arsitektur *transformer*. Lapisan *transformer* dapat digunakan untuk mengekstrak fitur-fitur dari data dan memahami pola dari data secara mendalam dan kontekstual (Liu, *et al*., 2019). Model RoBERTa memiliki lapisan dan *neuron* dengan jumlah jutaan hingga miliaran parameter yang saling terhubung. Hal ini memungkinkan RoBERTa meningkatkan performa berbagai tugas NLP dan menjadi salah satu bukti pencapaian terbaik dalam bidang *deep learning*.

2.9.1 *Layer*

Layer atau lapisan adalah komponen dalam model *deep learning* yang bertugas memproses dan mentransformasikan data saat bergerak melalui jaringan. Model *deep learning* memiliki tiga jenis lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* berperan menerima data *input*, lalu diproses lebih lanjut oleh *hidden layer*. *Hidden layer* pada model *deep learning* memiliki jumlah yang banyak untuk membuat komposisi algoritma yang tepat supaya meminimalisir nilai *error* pada output (Yanto, *et al*., 2021). Pada tahap ini, model akan mempelajari pola yang relevan dari data. Selanjutnya, *Output layer* akan menghasilkan hasil akhir, baik dalam bentuk prediksi ataupun klasifikasi berdasarkan informasi yang telah diproses (Mienye & Swart, 2021).

2.9.2 *Loss Function*

Loss function atau fungsi kerugian adalah fungsi yang bertugas untuk mengukur sejauh mana prediksi model berbeda dari nilai sebenarnya. Fungsi ini juga membantu menentukan tingkat *error* yang terjadi sehingga dapat dijadikan acuan dalam memperbarui parameter model agar *error* diminimumkan. Pemilihan *loss function* sangat bergantung pada jenis tugas dan data yang digunakan. Salah satu *loss function*

yang umum digunakan dalam tugas klasifikasi adalah *cross-entropy loss*. Fungsi ini efektif dalam mengukur perbedaan antara distribusi probabilitas kelas yang sebenarnya (p) dan distribusi kelas yang diprediksi oleh model (q). Secara matematis fungsi *cross-entropy loss* didefinisikan pada Persamaan (7) (Mienye & Swart, 2021).

$$\textit{Cross Entropy} = -\sum_{i} p(y_i) \log q(y_i) \tag{7}$$

Keterangan:
$p(y_i)$ = Vektor *one-hot encoding* kelas sebenarnya
$q(y_i)$ = Distribusi probabilitas hasil prediksi model untuk seluruh kelas

Dalam klasifikasi biner, *cross-entropy loss* disederhanakan menjadi *binary cross-entropy loss* yang digunakan ketika *output* berupa probabilitas yang mewakili dua kelas (0 dan 1). Untuk klasifikasi multi-kelas, fungsi ini diperluas menjadi *categorical cross-entropy loss*. *Cross-entropy loss* banyak digunakan dalam tugas klasifikasi karena memberikan penalti yang besar terhadap prediksi yang memiliki tingkat keyakinan tinggi tetapi salah, sehingga membuat model untuk menghasilkan probabilitas yang lebih sesuai dengan distribusi kelas sebenarnya (Mienye & Swart, 2021).

2.9.3 *Optimization Algorithms*

Optimization algorithms atau algoritma pengoptimalan berperan penting dalam *deep learning* karena algoritma ini menyesuaikan parameter model selama proses pelatihan dan digunakan untuk meminimumkan *loss function* (Syifa & Dewi, 2022). Pengoptimalan yang baik akan menentukan keberhasilan pelatihan model, karena mempengaruhi seberapa cepat model mencapai konvergensi dan kualitas hasil akhirnya (Mienye & Swart, 2021). Berbagai algoritma optimasi telah dikembangkan untuk meningkatkan efisiensi dan kinerja pelatihan, salah satu algoritma optimasi yang paling sering digunakan dalam *deep learning* adalah Adam *Optimizer* (Mienye & Swart, 2021).

Adam berasal dari kepanjangan *Adaptive Moment Estimation* karena memiliki kemampuan memperbarui bobot dan *learning rate* secara adaptif (Mienye & Swart, 2021). Adam memiliki beberapa keunggulan, antara lain metode ini membutuhkan sedikit *tuning* dalam *learning rate*, metode ini efisien secara komputasi dan hanya

membutuhkan sedikit memori, dan mudah diterapkan dan tetap efektif meskipun terjadi perubahan skala pada gradien. Penggunaan Adam *Optimizer* pada saat pelatihan dapat menyesuaikan nilai *learning rate* dan meningkatkan kinerja model (Soydaner, 2020).

2.9.4 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang merubah atau mentransformasikan suatu sinyal *input* menjadi sinyal *output* tertentu (Sitepu & Sigiro, 2021). Sinyal *output* tersebut digunakan sebagai *input* selanjutnya pada lapisan berikutnya. Fungsi aktivasi secara khusus digunakan dalam jaringan saraf tiruan. Jika fungsi aktivasi tidak diterapkan dalam jaringan saraf tiruan maka sinyal *output* hanya menjadi linear sederhana yang hanya berupa polinomial berderajat satu (Sharma, *et al.,* 2017). Pemilihan jenis fungsi aktivasi dan pengaturan jumlah lapisan yang digunakan akan mempengaruhi keakuratan dan kinerja prediksi dari jaringan saraf tiruan. Berikut ini adalah beberapa jenis fungsi aktivasi yang biasa digunakan:

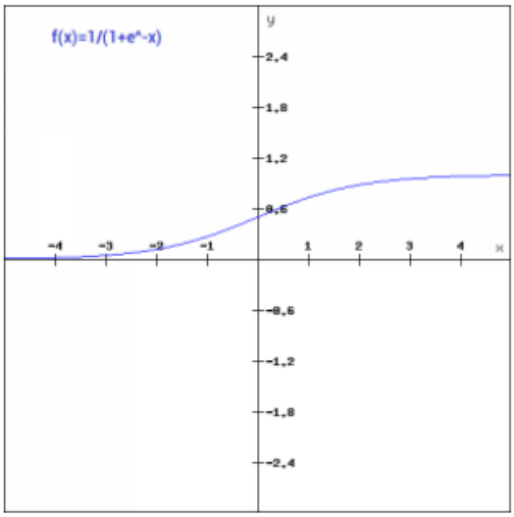
1. Fungsi Aktivasi *Sigmoid*

   Fungsi aktivasi *sigmoid* adalah salah satu fungsi aktivasi yang paling sering digunakan karena memiliki sifat non-linear. Fungsi aktivasi ini akan menerima angka tunggal dan merubah nilai dalam rentang 0 hingga 1. Secara matematis fungsi aktivasi *sigmoid* didefinisikan pada Persamaan (8) (Sharma, *et al.,* 2017).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

   Keterangan:
   x = Nilai data *input*
   e = Nilai eksponensial (2.7183)

Berdasarkan Persamaan (8), grafik fungsi aktivasi *sigmoid* dapat disajikan pada Gambar 4.



Gambar 4. Fungsi Aktivasi *Sigmoid* (Sharma, *et al.,* 2017).

Fungsi aktivasi *sigmoid* tidak simetris terhadap nilai nol yang mengakibatkan tanda dari semua nilai *output* akan sama (Sharma, *et al.,* 2017). Masalah tersebut dapat diperbaiki dengan melakukan penskalaan fungsi *sigmoid* dengan faktor skala tertentu.

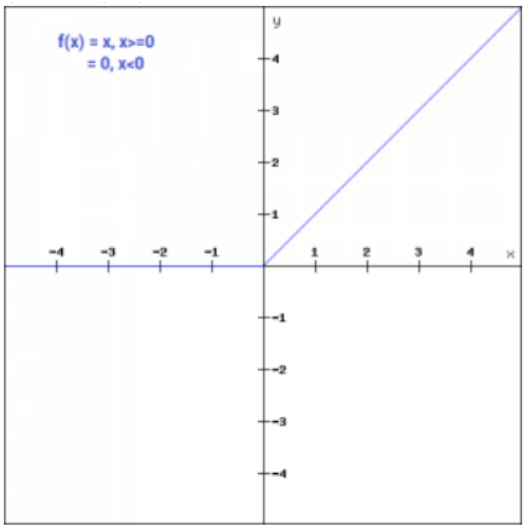
2. Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

   Fungsi aktivasi ReLU merupakan fungsi aktivasi *non-linear* yang banyak digunakan dalam jaringan saraf. Kelebihan menggunakan fungsi aktivasi ReLU adalah semua *neuron* tidak diaktifkan pada waktu yang sama. Hal ini berati bahwa sebuah *neuron* akan dinonaktifkan ketika *output* dari transformasi linear adalah nol. Secara matematis fungsi aktivasi ReLU didefinisikan pada Persamaan (9) (Sharma, *et al.,* 2017).

$$f(x) = \max(0, x) \tag{9}$$

Keterangan:
x = Nilai data *input*
$f(x)$ = Hasil *output* fungsi ReLU berupa nilai dalam bentuk 0 dan 1

Berdasarkan Persamaan (9), nilai $f(x) = x$ ketika $x > 0$ dan $f(x) = 0$ ketika $x \leq 0$. Grafik fungsi aktivasi ReLU dapat disajikan pada Gambar 5.



Gambar 5. Fungsi Aktivasi ReLU (Sharma, *et al.,* 2017).

Fungsi aktivasi ReLU lebih efisien dibandingkan fungsi aktivasi lainya karena semua *neuron* tidak diaktifkan pada waktu yang sama, melainkan hanya sejumlah *neuron* tertentu pada satu waktu. Dalam kasus tertentu, nilai gradien dapat bernilai nol yang mengakibatkan bobot dan bias tidak diperbarui selama langkah *backpropagation* dalam pelatihan jaringan saraf (Sharma, *et al.,* 2017).

3. Fungsi Aktivasi *Softmax*

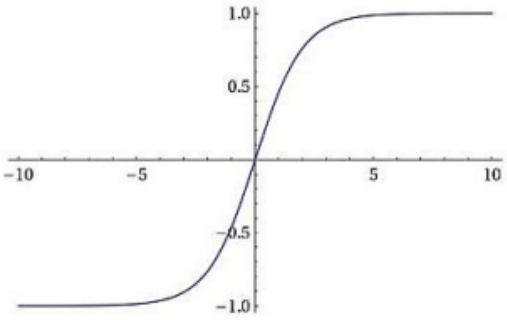
Fungsi aktivasi *softmax* adalah kombinasi dari beberapa fungsi *sigmoid*. Fungsi *sigmoid* menghasilkan nilai dalam rentang 0 hingga 1 yang dapat diinterpretasikan sebagai probabilitas pada kelas data tertentu. Berbeda dengan fungsi aktivasi *sigmoid* yang umumnya digunakan untuk klasifikasi biner, sedangkan fungsi *softmax* dapat digunakan untuk masalah klasifikasi multikelas. Ketika membangun model untuk klasifikasi multikelas, maka lapisan *output* dari jaringan akan memiliki jumlah *neuron* yang sama dengan jumlah kelas dalam target. Fungsi ini memberikan probabilitas untuk setiap kelas individu dalam satu titik data. Berikut ini adalah representasi matematis fungsi aktivasi *softmax* didefinisikan pada Persamaan (10) (Sharma, *et al.,* 2017)).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{untuk } j = 1, \ldots, K. \tag{10}$$

Keterangan:
$\sigma(z)_j$ = Probabilitas prediksi lapisan *output*
e^{z_j} = Sinyal yang dihasilkan lapisan *output*

e = Nilai eksponensial (2.7183)
K = Jumlah kelas lapisan *output*
$\sum_{k=1}^{K} e^{z_k}$ = Jumlah eksponensial seluruh sinyal *output*

Berdasarkan Persamaan (10), grafik fungsi aktivasi *softmax* dapat disajikan pada Gambar 6.



Gambar 6. Fungsi Aktivasi *Softmax* (Purwitasari & Soleh, 2022).

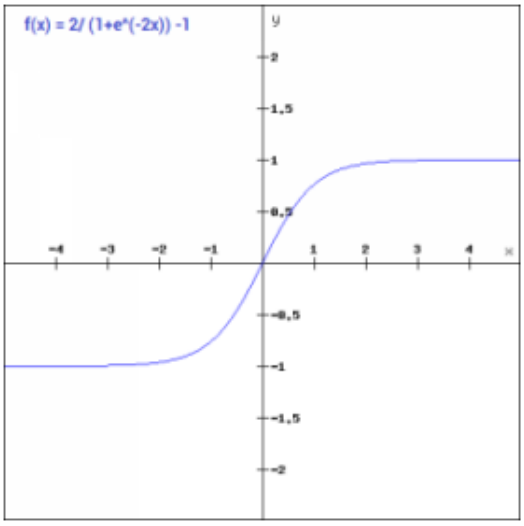
4. Fungsi Aktivasi *Tangent Hyperbolic* (tanh)

Fungsi aktivasi $\tanh$ identik dengan fungsi *sigmoid* tetapi memiliki simetri di sekitar nilai nol. Fungsi ini menghasilkan nilai *output* dengan tanda berbeda dari lapisan sebelumnya yang kemudian digunakan sebagai *input* untuk lapisan selanjutnya. Fungsi ini menghasilkan nilai pada rentang -1 hingga 1 (Firmansyah & Hayadi, 2022). Secara matematis, fungsi $\tanh$ ditunjukan oleh Persamaan (11).

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{11}$$

Keterangan:
e^{x} = Nilai eksponensial positif dari nilai data *input*
e^{-x} = Nilai eksponensial negatif dari nilai data *input*
$tanh(x)$ = hasil *output* fungsi $tanh$ berupa nilai dalam rentang -1 sampai 1

Berdasarkan Persamaan (8), grafik fungsi aktivasi $\tanh$ dapat disajikan pada Gambar 7.



Gambar 7. Fungsi Aktivasi $\tanh$ (Sharma, *et al.,* 2017).

Fungsi $\tanh$ bersifat kontinu dan dapat diturunkan. Dibandingkan dengan fungsi *sigmoid*, gradien dari fungsi $\tanh$ lebih curam. Fungsi $\tanh$ lebih disukai daripada fungsi *sigmoid* disebabkan fungsi ini memiliki gradien yang tidak terbatas pada arah tertentu dan berpusat disekitar nilai nol (Sharma, *et al.,* 2017).

5. Fungsi Aktivasi *Gaussian Error Linear Unit* (GELU)

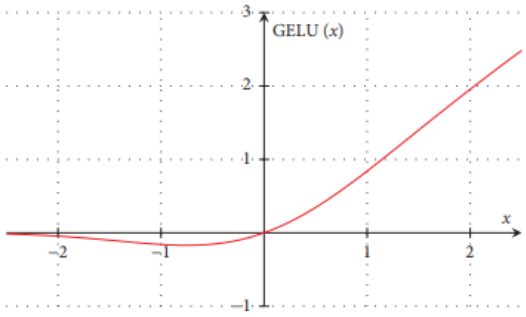
   Fungsi aktivasi GELU dikembangan sebagai fungsi alternatif dari fungsi ReLU tanpa menghilangkan keunggulan yang dimiliki ReLU. Fungsi ReLU didefinisikan sebagai $\text{ReLU}(x) = \max(0, x)$ menambahkan *nonlinearitas* pada jaringan, namun fungsi tersebut tidak dapat diturunkan pada $x = 0$. sehingga menimbulkan masalah dalam optimasi berbasis gradien, seperti dinamika pelatihan yang tidak stabil (Lee, 2023).

   Untuk mengatasi beberapa kekurangan dari ReLU, fungsi aktivasi GELU dikembangkan sebagai versi yang lebih halus daripada fungsi ReLU. fungsi GELU dapat diturunkan untuk setiap titik, selanjutnya mempertahankan karakteristik *nonlinear* yang penting dalam *deep learning*. Fungsi GELU terinspirasi dari fungsi disribusi kumulatif *gaussian* (CDF) yang dikenal dengan sifat kehalusanya dan mudah untuk diturunkan. Secara matematis, fungsi GELU didefinisikan pada Persamaan (12).

$$\text{GELU}(x) = 1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right) \tag{12}$$

Keterangan:
$tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
e^x = Nilai eksponensial positif
e^{-x} = Nilai eksponensial negatif
x = Nilai data *input*
$\frac{2}{\pi}$ = konstanta dari normalisasi distribusi normal standar

Berdasarkan Persamaan (12), grafik fungsi aktivasi GELU dapat disajikan pada Gambar 8.



Gambar 8. Fungsi Aktivasi GELU (Lee, 2023).

Berdasarkan Gambar 8, fungsi aktivasi GELU memiliki bentuk halus tanpa sudut tajam, sehingga memungkinkan perubahan *output* yang mulus seiring peningkatan nilai *input*. Saat *input* bernilai negatif, maka *output* GELU mendekati nol, namun tidak sepenuhnya mencapai nol. Fungsi aktivasi GELU memiliki berbagai karakteristik matematis seperti kemampuan diferensiasi, stasioneritas, *smoothing*, dan *nonlinearitas*. Berdasarkan karakteristik tersebut, fungsi aktivasi GELU mendukung efektivitas dalam arsitektur *deep learning*. GELU juga dapat menjadi komponen penting dalam model-model canggih seperti BERT dan GPT (Lee, 2023).

2.10 *Transformer*

Model yang berbasis *Recurrent Neural Network* (RNN) seperti model *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Neural Networks* (GRU) telah menjadi model terbaik dalam pemodelan data-data sekuens, seperti pemodelan bahasa dan penerjemahan otomatis. banyak upaya yang dilakukan untuk mengembangkan model berbasis RNN. Model tersebut didasarkan pada jaringan kompleks yang berulang atau saraf konvolusi yang melibatkan *encoder* dan *decoder*

dengan memanfaatkan mekanisme *attention*. Model berbasis RNN memproses data secara berurutan berdasarkan posisi simbol dalam *input* dan *output*. Setiap pemrosesan dihitung berdasarkan urutan simbol tersebut, sehingga menghasilkan sifat komputasi yang berurutan. Sifat tersebut menghambat proses dalam pelatihan, terutama ketika memproses data yang sangat panjang (Vaswani, *et al*., 2017).

Self-attention adalah sebuah mekanisme yang digunakan untuk menghubungkan berbagai posisi dalam satu urutan untuk menghasilkan representasi dari keseluruhan urutan tersebut. Mekanisme *self-attention* telah banyak digunakan bersama dengan RNN. *Transformer* adalah model pertama yang sepenuhnya bergantung pada mekanisme *self-attention* untuk menghasilkan representasi *input* dan *output* tanpa menggunakan RNN atau lapisan konvolusi yang memproses data secara berurutan, sehingga mempercepat pemrosesan data dan meningkatkan efisiensi (Rush, 2018). Penerapan *self-attention* dalam model *transformer* telah berhasil digunakan dalam menyelesaikan berbagai tugas-tugas NLP, seperti pemahaman bacaan, *abstractive summarization, textual entailment*, dan pembelajaran representasi kalimat (Vaswani, *et al*., 2017).

Transformer memiliki arsitektur yang dibangun berdasarkan lapisan-lapisan *encoder-doceder*. *Encoder* memiliki fungsi untuk mengubah urutan simbol input $(x_1, \ldots, x_n)$ menjadi urutan representasi kontinu $z = (z_1, \ldots, z_n)$. Setelah itu, *decoder* menggunakan representasi z untuk menghasilkan urutan output $(y_1, \ldots, y_m)$ dengan memproses simbol-simbol secara bertahap. Pada setiap langkah, model ini bekerja secara autoregresif yang artinya simbol yang sudah didapatkan sebelumnya digunakan sebagai *input* tambahan untuk menghasilkan simbol selanjutnya (Rush, 2018). Pada arsitektur *transformer, encoder* terdiri dari 6 *layer* dengan setiap lapisan memiliki dua *sub-layer. Sub-layer* pertama adalah mekanisme *multi-head self-attention*, sedangkan *sub-layer* kedua adalah jaringan *feed-forward* sederhana dan terhubung. Kemudian *decoder* juga terdiri dari 6 *layer* dengan dua *sub-layer* yang sama dengan lapisan *encoder*. Selain itu, *decoder* memiliki *sub-layer* ketiga yaitu *multi-head attention* pada *output* dari tumpukan *encoder*. *Masked Multi-Head attention* adalah *Sub layer self-attention* yang terdapat pada *decoder* yang dimodifikasi untuk mencegah token memperhatikan token setelahnya, sehingga hasil prediksi menjadi lebih akurat dan sesuai dengan konteks kata. Meskipun *transformer* dibangun dari arsitektur layer *encoder* dan *decoder*, arsitektur model RoBERTa terinspirasi dari model BERT, sehingga model RoBERTa hanya menggunakan lapisan *encoder* dalam strukturnya (Liu, *et al*., 2019). Arsitektur model *transformer*

diilustrasikan pada Gambar 9.

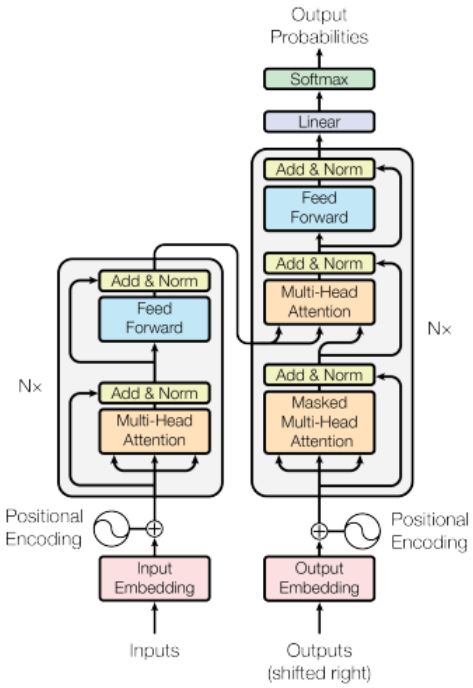


Figure 1: The Transformer - model architecture.

Gambar 9. Arsitektur Model *Transformer* (Vaswani, *et al*., 2017).

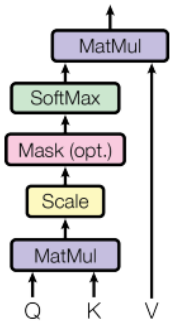
2.10.1 *Attention*

Fungsi *attention* adalah fungsi yang berfungsi untuk memetakan *query* dengan pasangan *key-value* yang menghasilkan sebuah *output* dengan *query, key, value*, dan *output* semuanya dalam bentuk vektor. Proses ini melibatkan perhitungan *output* sebagai penjumlahan tertimbang dari setiap *value* dengan bobot yang diberikan pada masing-masing *value* dihitung berdasarkan tingkat kecocokan antara *query* dan *key* yang sesuai (Vaswani, *et al*., 2017).

2.10.2 *Scaled Dot-Product Attention*

Scladed dot-product attention terdiri dari himpunan vektor *query* dan *key* dengan dimensi d_k serta himpunan vektor *value* dengan dimensi d_v. Langkah awalnya adalah perhitungan *dot product* antara *query* dengan setiap *key* untuk mengukur kesesuainya. Selanjutnya, Hasil perhitungan kemudian dibagi dengan $\sqrt{d_k}$. Setelah itu, fungsi *softmax* diterapkan pada nilai tersebut untuk mendapatkan bobot yang digunakan

pada vektor *value* (Vaswani, *et al*., 2017). Ilustrasi Operasi *scaled dot-product* dapat dilihat pada Gambar 10 dan Perhitungan *scaled dot-product* diformulasikan pada Persamaan (13).



Gambar 10. *Scaled Dot-Product Attention* (Vaswani, *et al*., 2017).

$$\textit{Attention}(Q, K, V) = \textit{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{13}$$

Keterangan:
Q = *Query* representasi *input* untuk mencari hubungan antar token
K = *Key* representasi *input* untuk mencocokan *query*
V = *Value* representasi *input* yang akan dihitung atensinya
$\sqrt{d_k}$ = Dimensi dari *key*

Dua fungsi *attention* yang paling sering digunakan dalam model adalah *additive attention* dan *dot-product (multiplicative) attention*. Kedua jenis *attention* ini memiliki kompleksitas teoritis yang sama, namun *dot-product attention* jauh lebih cepat dan efisien dalam praktik. Hal ini disebabkan *dot-product attention* menggunakan perkalian matriks yang dioptimalkan (Vaswani, *et al*., 2017).

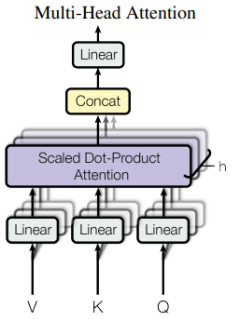
2.10.3 *Multi-Head Attention*

Multi-Head attention adalah struktur dalam model *transformer* yang memungkinkan model melakukan mekanisme *attention* secara bersamaan sebanyak h kali terhadap beberapa proyeksi *query, key*, dan *value*. Setiap *query, key*, dan *value* diubah ke dimensi yang lebih kecil, yaitu d_k untuk *query* dengan *key* dan d_v untuk *value*. *Multi head attention* memungkinkan model untuk menangkap informasi dari berbagai representasi dan posisi yang berbeda secara bersamaan. *Multi Head Attention* dapat dirumuskan secara matematis oleh Persamaan (14) dan (15) (Vaswani, *et al*., 2017).

$$MultiHead(Q, K, V) = Concat\left(head_1, \ldots, head_h\right) W^O \tag{14}$$

$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{15}$$

di mana proyeksi tersebut adalah matriks parameter $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$, dan $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. Di mana W_i^Q, W_i^K, W_i^V adalah matriks proyeksi untuk setiap *head* dan W^o adalah matriks proyeksi akhir (Chen, *et al*., 2021).



Gambar 11. *Multi-head Attention* (Vaswani, *et al*., 2017).

2.10.4 *Position-Wise Feed-Forward Networks*

Setiap lapisan dalam *encoder* dan *decoder* memiliki jaringan *feed-forward* yang sepenuhnya terhubung (*fully connected*). Jaringan ini diterapkan ke setiap posisi secara terpisah dan identik. Struktur jaringan *feed-forward* terdiri dari dua transformasi linier yang dipisahkan oleh fungsi aktivasi ReLU (Vaswani, *et al*., 2017). Secara matematis *feed-forward* dapat diuraikan pada Persamaan (16).

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{16}$$

Keterangan:
W = Matriks bobot
b = Basis vektor

2.10.5 *Embeddings* dan *Softmax*

Transformer menguunakan *embedding* yang dipelajari untuk mengubah token *input* dan *output* menjadi vektor berdimensi d_{model}. Transformasi linear dan fungsi *softmax*

diterapkan untuk mengubah *output* dari *decoder* menghasilkan probabilitas token berikutnya yang diprediksi. Model memanfaatkan matriks bobot yang sama untuk dua lapisan *embedding* dan transformasi linear sebelum *softmax* dalam proses ini. Pada lapisan *embedding*, bobot tersebut dikalikan dengan $\sqrt{d_{model}}$ untuk menjaga kestabilan represebtasi selama proses pelatihan (Vaswani, *et al*., 2017).

2.10.6 *Positional Encoding*

Di karenakan model berbasis *transformer* tidak menggunakan mekanisme *recurrent* dan konvolusi, maka model perlu menambahkan informasi mengenai posisi setiap token dalam urutan agar model dapat memahami konteksnya. Model transformer menggunakan *positional encoding* pada *input embedding* di bagian bawah tumpukan *encoder* dan *decoder* untuk menambahkan informasi mengenai urutan kata. *Positional encoding* memiliki dimensi yang sama dengan *embedding*, sehingga keduanya dapat dijumlahkan secara langsung (Vaswani, *et al*., 2017). *Positional encoding* menggunakan fungsi sinus dan kosinus dengan frekuensi berbeda yang dapat direpresentasikan pada Persamaan (17) dan (18).

$$PE_{(\text{pos, 2i})} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{17}$$

$$PE_{(\text{pos, 2i+1})} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{18}$$

Pada Persamaan (17) dan (18) terdapat pos adalah representasi posisi dalam urutan dan i adalah dimensi. Setiap dimensi dari *positional encoding* dipetakan ke fungsi sinusoidal. Panjang gelombang fungsi ini berkembang secara geometris yang berkisar antara 2π hingga 10000 * 2π. Fungsi tersebut dipilih karena fungsi ini membantu model untuk belajar dengan baik mengenai posisi relatif (Vaswani, *et al*., 2017).

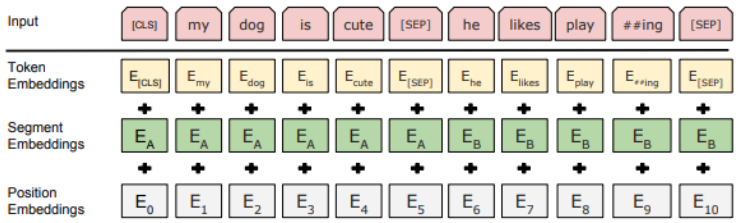
2.11 *Robustly Optimized BERT Approach* (RoBERTa)

Robustly Optimized BERT *Approach* adalah salah satu model berbasis *transformer* yang dikembangkan oleh tim peneliti dari Facebook AI pada tahun 2019 sebagai pengembangan dari model BERT untuk meningkatkan performa kinerja model

dalam menyelesaikan tugas NLP yang lebih kompleks. Model RoBERTa memiliki arsitektur sama dengan model BERT dengan menggunakan lapisan *encoder* tanpa *decoder* dalam *transformer*. Namun, tidak seperti BERT, Model RoBERTa hanya dilatih menggunakan *Masked Language Modeling* (MLM) sebagai tujuan *pretraining* dan menghilangkan tujuan *Next Sentences Prediction* (NSP) (Liu, *et al*., 2019).

Bidirectional Encoder Representations from Transformer adalah model yang dikembangkan oleh tim peneliti Google AI pada tahun 2019. Model BERT memperkenalkan konsep pendekatan *bidirectional* melalui arsitektur *transformer* dalam pelatihan, yang artinya model ini dirancang untuk melatih representasi dua arah yang mendalam dari teks yang tidak berlabel, dengan memperhatikan konteks di sebelah kiri dan kanan secara bersamaan di setiap lapisan. Oleh karena itu, BERT dapat menghasilkan representasi bahasa yang sangat canggih (Devlin, *et al*., 2019).

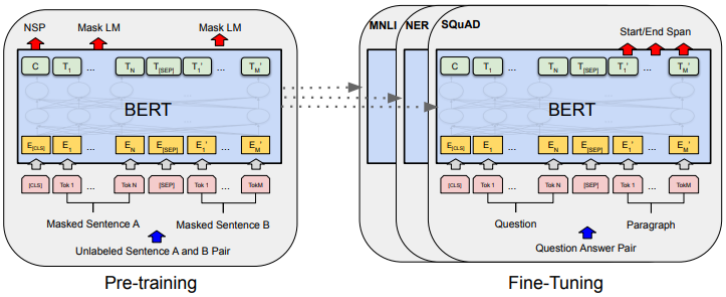
Model BERT menggunakan 2 jenis token yaitu token [CLS] dan [SEP]. Token pertama dalam setiap urutan selalu berupa token token [CLS] sebagai token khusus untuk klasifikasi. Kalimat-kalimat tersebut akan dipisahkan oleh Token [SEP] yang berada di akhir kalimat. Model BERT membuat representasi *input* untuk dapat memahami konteks dari kalimat dalam satu urutan token. Representasi ini dimulai dengan token *embedding* yaitu setiap token akan diubah menjadi representasi vektor dengan kosakata *WordPiece* yang berisikan 30.000 token. Selanjutnya, memberikan tanda posisi pada setiap token dalam kalimat dengan menggunakan *positional embedding*. Pada setiap token, representasi *input* dibangun dengan menjumlahkan *token embedding*, *segment embedding*, dan *positional embedding* (Devlin, *et al*., 2019). Representasi *input* BERT dapan dilihat pada Gambar 12.



Gambar 12. Representasi *Input* BERT (Devlin, *et al*., 2019).

Model BERT memiliki dua tahap proses pelatihan, yaitu *pre-training* dan *fine-tuning*. Tahap *pre-training* adalah proses di mana model BERT dilatih untuk memahami bahasa serta konteksnya. Selama proses *pre-training*, model BERT dilatih menggunakan data yang tidak berlabel pada beberapa tugas *pretraining* yang berbeda.

Selanjutnya untuk proses *fine-tuning,* model BERT diinisialisasi terlebih dahulu dengan parameter hasil *pre-training*, dan kemudian semua parameternya disesuaikan (*fine-tuned*) menggunakan data berlabel sesuai dengan tugas-tugas yang spesifik. *Fine-tuning* BERT sangat mudah dilakukan karena mekanisme *self-attention* dalam arsitektur transformer. Mekanisme ini memungkinkan BERT menangani berbagai tugas yang melibatkan kalimat tunggal ataupun kalimat berpasangan, hanya dengan menyesuaikan *input* dan *output* yang sesuai (Devlin, *et al*., 2019). Ilustrasi prosudur *pre-train* dan *fine-tuning* dapat dilihat pada Gambar 13.

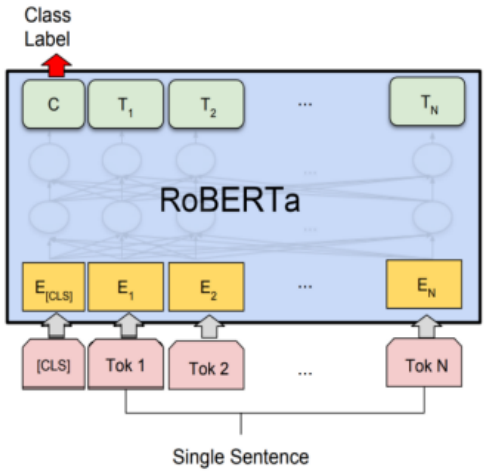


Gambar 13. Prosedur *Pre-train* dan *Fine-tuning* BERT (Devlin, *et al*., 2019).

Berdasarkan Gambar 13, dapat dilihat bahwa prosedur *pre-train* dan *fine-tuning* model BERT menggunakan arsitektur yang sama kecuali pada lapisan *output*. Model yang telah dilatih pada tahap *pre-train* digunakan untuk menginisialisasi model pada berbagai tugas khusus. Selama tahapan *fine-tuning*, semua parameter pada model disesuaikan (*fine-tuning*) untuk setiap tugas (Devlin, *et al*., 2019).

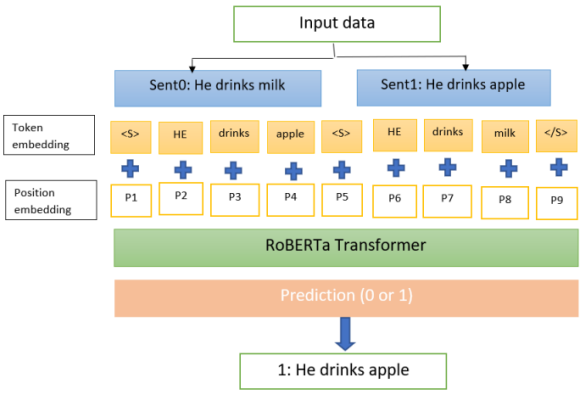
Meskipun model BERT dapat memberikan kinerja yang baik dalam melakukan tugas-tugas NLP, namun model BERT belum dioptimalkan secara maksimal dalam pelatihannya. Oleh karen itu, penelitian yang dilakukan oleh tim Facebook AI *Research* pada tahun 2019 mengusulkan modifikasi model BERT yang lebih dioptimalkan yaitu model RoBERTa. Model RoBERTa dibangun untuk dapat menyamai atau bahkan melampaui kinerja semua metode yang dikembangkan setelah BERT. RoBERTa menggunakan arsitektur yang sama dengan BERT dengan beberapa modifikasi sederhana yaitu dengan 1) melatih model lebih lama, dengan ukuran *batch* yang lebih besar dan menggunakan lebih banyak data, 2) menghapus tujuan *next sentence prediction*, 3) melatih model dengan urutan *input* yang lebih panjang, dan 4) menerapkan *dynamic masking* pada data pelatihan (Liu, *et al*., 2019).

Arsitektur Model RoBERTa dibangun dengan arsitektur yang sama dengan model BERT. Dalam model RoBERTa kalimat diubah menjadi token, setelah itu kalimat token tersebut digunakan sebagai input dalam model. RoBERTa menggunakan dua jenis *input* yaitu *input_ids* dan *attention_mask*. *Input_ids* adalah representasi numerik dari setiap token. Pada awal setiap urutan token, ditambahkan token $< s >$ untuk mengidentifikasi klasifikasi, sementara token $< /s >$ diambahkan di akhir urutan token. Sedangkan, *Attention_mask* adalah representasi biner yang menunjukan apakah suatu token merupakan *padding*. Jika panjang urutan token lebih pendek dari urutan terpanjang, *padding* akan ditambahkan untuk mencapai panjang maksimal yang digunakan oleh RoBERTa, yaitu 512 token. Selanjutnya, setelah semua *input* dimasukan, model RoBERTa memprosesnya dengan 12 lapisan *encoder*, mengubah *input* menjadi vektor *embedding* yang terdiri dari *embedding token* dan *positional encoding*. Hasil akhir dari lapisan tersebut adalah *last hidden state* yang bertugas menyimpan semua vektor kata *embedding*. Vektor-vektor ini kemudian dilatih untuk memahami bahasa, sehingga model dapat disesuaikan untuk berbagai tugas NLP (Khusuma, *et al*, 2023). Arsitektur model RoBERTa dapat dilihat pada Gambar 14.



Gambar 14. Arsitektur Model RoBERTa (Khusuma, *et al*, 2023).

Mekanisme representasi *input* model RoBERTa sangat identik dengan model BERT, namun dengan beberapa perbedaan yang membuat model RoBERTa berbeda dengan model BERT. Ilustrasi mekanisme representasi *input* model RoBERTa dapat dilihat pada Gambar 15.



Gambar 15. Representasi *Input* Model RoBERTa (Al-Jarrah, *et al*., 2020).

Proses kerja model RoBERTa dimulai dari kalimat *input* yang diproses melalui beberapa tahapan komputasi hingga menghasilkan *output* untuk prediksi. Hal pertama yang dilakukan adalah tokenisasi, yaitu mengubah kalimat input menjadi bagian-bagian kecil yang disebut token. Model RoBERTa menggunakan tokenisasi berbasis *Byte-Pair Encoding* (BPE), yang mengubah kata menjadi unit subkata, yang memungkinkan model untuk mengatasi kata-kata yang jarang atau tidak diketahui dengan memecahnya menjadi potongan kata-kata yang diketahui. Sebagai contoh, kata "*unbelievably*" dipecah menjadi $['un', 'believ', 'ably']$. Hal tersebut memberikan pemahaman kata yang lebih baik dan fleksibel dalam berbagai konteks. Selanjutnya token-token tersebut akan diubah kedalam bentuk numerik yang disebut *input ids*. Misalnya, hasil tokenisasi tersebut dikonversi menjadi urutan angka seperti $[0, 31414, 6, 232, 2]$. Model RoBERTa menggunakan dua token khusus, yaitu token $< s >$ sebagai token awal dan token $< /s >$ untuk memisahkan dua kalimat atau penanda akhir teks. Misalnya, hasil tokeniasi tersebut dikonversi menjadi urutan angka seperti $[0, 31414, 6, 232, 2]$, di mana setiap angka merepresentasikan token dari teks, angka 0 menunjukan token $< s >$ dan angka 2 menunjukan token $< /s >$ (Liu, *et al*., 2019).

Selanjutnya melakukan penambahan *positional encoding* untuk memberikan informasi posisi setiap kata terhadap model. *Positional embedding* ditambahkan ke setiap token *embedding* sebelum masuk ke dalam lapisan *encoder*. Model RoBERTa memiliki 12 lapisan *encoder*. Setiap lapisan *encoder* melakukan mekanisme *self attention* secara pararel yang memungkinkan model memahami hubungan antar kata dalam kalimat. Proses ini berlangsung secara berulang dan menghasilkan *last hidden state*, yaitu representasi akhir dari seluruh token (Liu, *et al*., 2019).

Selanjutnya, pada tugas klasifikasi, hanya token pertama $< \ s \ >$ yang digunakan sebagai representasi seluruh kalimat. Kemudian, vektor dari token ini diproses menggunakan fungsi aktivasi *softmax* untuk menghasilkan probabilitas dari setiap kelas. Sebagai contoh, misalnya *output* akhir berupa vektor $[0.02, 0.01, 0.05, 0.91, 0.01]$, di mana nilai 0.91 menunjukan model memprediksi kalimat tersebut ke dalam kelas *not cyberbullying* (Al-Jarrah, *et al*., 2020).

Model RoBERTa memiliki struktur utama yang hanya menggunakan lapisan *encoder* pada arsitektur *transformer*. Model RoBERTa memiliki dua varian ukuran, yaitu RoBERTa BASE dan RoBERTa LARGE. Perbedaan keduanya terletak pada jumlah lapisan *encoder*, jumlah *multi-head attention*, *hidden size*, dan jumlah parameter. Model RoBERTa BASE memiliki 12 lapisan *encoder*, 12 *multi-head attention*, 768 *hidden size*, dan 110 juta parameter. Sementara itu, model RoBERTa LARGE memiliki 24 lapisan *encoder*, 16 *multi-head attention*, 1024 *hidden size*, dan 340 juta parameter (Liu, *et al*., 2019). Perbadingan varian model RoBERTa BASE dan RoBERTa LARGE direpresentasikan pada Tabel 3 berikut.

Tabel 3. Perbandingan Varian Model RoBERTa

Hyperparam	**RoBERTa$_{LARGE}$**	**RoBERTa$_{BASE}$**
Number of Layers	24	12
Hidden size	1024	768
FFN inner hidden size	4096	3072
Attention heads	16	12
Attention head size	64	64
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
Warmup Steps	30k	24k
Peak Learning Rate	4e-4	6e-4
Batch Size	8k	8k
Weight Decay	0.01	0.01
Max Steps	500k	500k
Learning Rate Decay	*Linear*	*Linear*
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.98	0.98
Gradient Clipping	0.0	0.0

Model RoBERTa adalah perbaikan dari model BERT dengan beberapa modifikasi yang signifikan dalam prosedur *pre-training*. Modifikasi ini diharapkan dapat meningkatkan performa dari model. Dalam penelitian yang dilakukan Liu, *et al*. pada tahun 2019, tahapan *pretraining* model RoBERTa dibangun menggunakan arsitektur model yang sama dengan model BERT BASE. Berikut ini adalah beberapa modifikasi pada tahapan *pretraining* model RoBERTa (Liu, *et al*., 2019):

1. *Dynamic Masking*

   Model RoBERTa menggunakan teknik *dynamic masking* yang merupakan lawan dari teknik *static masking* yang digunakan BERT. Modifikasi ini dilakukan untuk menghindari penggunaan pola yang sama pada setiap *epoch* dengan menduplikasi data pelatihan sebanyak 10 kali, sehingga setiap urutan teks disamarkan dengan 10 cara berbeda selama 40 *epoch* pelatihan. *Dynamic masing* adalah teknik *masking* yang memungkinkan posisi *masking* terus berubah selama saat proses pelatihan. Modifikasi ini akan meningkatkan keacakan data, sehingga model dapat mempelajari konteks yang lebih luas dan meningkatkan kemampuan model dalam memahami pola bahasa (Gao, *et al*., 2022).

   Static masking adalah teknik di mana pola *masking* pada urutan *input* diterapkan hanya sekali selama *preprocessing data* dan tidak berubah pada setiap *epoch* pelatihan. Sementara itu, *dynamic masking* adalah teknik yang menghasilkan pola masking baru pada setiap *epoch*, sehingga urutan *input* disamarkan secara acak setiap pelatihan. *Dynamic masking* menjadi penting saat melakukan *pretraining* dengan jumlah langkah yang lebih banyak dan dataset yang lebih besar. Penerapan *dynamic masking* menghasilkan kinerja yang setara atau sedikit lebih baik dibandingkan dengan *static masking* (Liu, *et al*., 2019). Perbandingan kinerja *dynamic masking* dan *static masking* dapat dilihat pada Tabel 4.

   Tabel 4. Kinerja *Static Masking* VS *Dynamic Masking*

Masking	**SQuAD 2.0**	**MNLI-m**	**SST-2**
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

2. *Model Input Format and Next Sentence Prediction*

   Selain tujuan dari *masked language modeling*, model juga dilatih untuk memprediksi apakah dua segmen tersebut berasal dari dokumen yang sama atau tidak melalui *Next Sentence Prediction* (NSP). NSP diyakini menjadi faktor penting dalam efektivitas pelatihan model BERT. Model RoBERTa menerapkan *full sentence* sebagai input yang terdiri dari kalimat-kalimat yang diambil secara berurutan dari satu atau beberapa dokumen dengan total panjang maksimum 512 token dan menambahkan token pemisah tambahan antar dokumen (Liu, *et al*., 2019).

   Tabel 5 menunjukan perbandingan hasil dari empat pengaturan format *input* yang berbeda (Liu, *et al*., 2019). Pertama dengan membandingkan format *input segment-pair* dengan *sentence-pair*, di mana kedua format tersebut mempertahankan NSP *loss*, tetapi *sentence-pair* menggunakan kalimat tunggal. Format tersebut menunjukan bahwa penggunaan kalimat tunggal merusak kinerja pada tugas *downstream* disebabkan ketidakmampuan model dalam mempelajari ketergantungan jarak jauh. Selanjutnya melakukan perbadingan pelatihan tanpa NSP *loss* dan pelatihan dengan blok teks dari satu dokumen. Format kedua mengungguli hasil BERT BASE dan menghapus NSP *loss* menghasilkan kinerja yang setara atau sedikit lebih baik dalam tugas *downstream*. Pada Tabel 5 menunjukan bahwa membatasi sekuens dari satu dokumen (*doc-sentences*) menunjukan sedikit peningkatan dibandingkan dengan menyusun sekuens dari beberapa dokumen (*full-sentences*). Namun, model RoBERTa tetap menggunakan *full-sentences* disebabkan *doc-sentences* menghasilkan ukuran *batch* yang bervariasi (Liu, *et al*., 2019).

   Tabel 5. Kinerja Model *Input Format* & NSP

Model	**SQuAD 1.1/2.0**	**MNLI-m**	**SST-2**	**RACE**
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
BERT_{BASE}	88.5/76.3	84.3	92.3	64.3
XLNet_{BASE} (K = 7)	–/81.3	85.8	92.7	66.1
XLNet_{BASE} (K = 6)	–/81.0	85.6	93.4	66.7

3. *Training with Large Batches*

   Model BERT awalnya hanya dilatih dengan 1 juta langkah dengan *batch size* 256 urutan. Hal ini setara dengan biaya komputasi yang diperlukan untuk melatih dengan 125000 langkah dengan *batch size* 2000 urutan atau 31000 langkah dengan *batch size* 8000 urutan. Sedangkan, Model RoBERTa dilatih dengan 125000 langkah dan *batch size* yang lebih besar dari 2000 urutan dengan kata lain, RoBERTa melatih 500000 langkah dengan *batch size* 8000 dan menghasilkan kinerja yang lebih baik daripada model BERT (Liu, *et al*., 2019).

   Dalam penelitian yang dilakukan Liu, *et al*. (2019), menyatakan bahwa pelatihan dengan *batch size* yang lebih besar dapat mengurangi *perplexity* dalam tujuan pemodelan bahasa dan meningkatkan akurasi pada tugas akhir. *Perplexity* adalah sebuah metrik yang digunakan dalam pemodelan bahasa untuk menilai seberapa baik model memprediksi urutan kata dan mengukur akurasi suatu informasi dari sebuah teks terkait topik yang dihasilkan (Patmawati & Yusuf, 2021). Perbandingan *perplexity* dan kinerja tugas akhir ketika *batch size* ditingkatkan dan mengontrol jumlah lintasan yang digunakan dalam pelatihan dapat dilihat pada Tabel 6.

   Tabel 6. Perbandingan *Perplexity* dan Kinerja Tugas Akhir

bsz	***steps***	**lr**	**ppl**	**MNLI-m**	**SST-2**
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	**3.68**	**85.2**	**92.9**
8K	31K	1e-3	3.77	84.6	92.8

4. *Larger Byte-Pair Encoding* (BPE)

   Byte-Pair Encoding adalah pendekatan yang menggabungkan representasi tingkat karakter dan kata, sehingga dapat mengelola kosakata besar yang sering dijumpai dalam korpus bahasa alami. *Byte-Pair Encoding* tidak mengkodekan kata secara utuh, melainkan memecahnya menjadi huruf-huruf yang direpresentasikan sebagai *subword* seperti contohnya adalah '[*play*]' dan '[#*ing*]'. Ukuran kosakata BPE umumnya berkisar antara 10000 hingga 100000 unit *subword*. Namun, karakter *unicode* memberikan porsi yang cukup besar dari kosakata ini ketika memodelkan korpus yang besar dan beragam (Liu, *et al*., 2019).

   Radford, *et al*. pada tahun 2019 memperkenalkan versi BPE dengan menggunakan *byte*, bukan karakter *unicode* sebagai unit *subword* dasar, versi ini memungkinkan

untuk mempelajari kosakata *subword* yang lebih ringkas sekitar 50000 unit dan tetap dapat menyandikan teks tanpa menghasilkan token tidak dikenal. Model BERT menggunakan kosakata BPE berbasis karakter dengan ukuran 30000 unit *subword* yang diperoleh setelah *preprocessing* atau tokenisasi tambahan pada *input*. Sedangkan, RoBERTa menggunakan kosakata BPE berbasis *byte* yang terdiri dari 50000 unit *subword* tanpa *preprocessing* atau tokenisasi tambahan pada *input* (Liu, *et al*., 2019).

5. *Training with a Larger Datasets*

   Model BERT dilatih selama 100000 langkah menggunakan kumpulan dataset *BookCorpus* dan *English Wikipedia* dengan ukuran 16 GB (Devlin, *et al*., 2019). Sedangkan, model RoBERTa menambahkan 3 kumpulan dataset tambahan dalam pelatihanya, yaitu *CC-News* yang berisikan 63 juta artikel berbahasa inggris, *OpenWebText* yang berisikan WebText yang diekstrak dari aplikasi Reddit, dan *Stories* yang berisikan data *CommonCrawl*. model RoBERTa dilatih menggunakan 100000 langkah pelatihan dan lebih dari 160 GB dataset, menghasilkan peningkatan performa yang lebih baik dalam tugas khusus (Liu, *et al*., 2019). Perbandingan kinerja pengembangan model RoBERTa dengan variasi jumlah data pelatihan dan jumlah langkah pelatihan dapat dilihat dalam Tabel 7.

   Tabel 7. Evaluasi Model RoBERTa pada Berbagai Dataset

Model	**data**	**bsz**	**steps**	**SQuAD 1.1/2.0**	**MNLI-m**	**SST-2**
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ *additional data* (3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ *pretrain longer*	160GB	8K	300K	94.4/88.7	90.0	96.1
+ *pretrain even longer*	160GB	8K	500K	94.6/89.4	90.2	96.4
BERT						*LARGE*
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNeT						*LARGE*
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

2.12 Evaluasi Kinerja Model

Kinerja model klasifikasi dapat dievaluasi dengan menggunakan berbagai jenis metode. Salah satu metode yang umum digunakan adalah *confusion matrix*. Dalam

masalah klasifikasi biner, *confusion matrix* memiliki ukuran $2 * 2$ dengan salah satu label dianggap sebagai "Positif" dan yang lainya sebagai "Negatif". *Confusion matrix* terdiri dari baris yang mewakili kelas prediksi dan kolom yang mewakili kelas yang sebenarnya. Setiap elemen dalam matriks ini dikategorikan berdasarkan label prediksi (positif atau negatif) dan hasil perbandingan antara prediksi dengan label yang sebenarnya (benar atau salah). Elemen-elemen pada *confusion matrix*, yaitu *True Positive* (TP) yang terjadi jika kelas yang diprediksi dan kelas sebenarnya adalah positif, *False Positive* (FP) yang terjadi jika kelas yang diprediksi adalah positif, namun kelas sebenarnya adalah negatif, *False Negative* (FN) yang terjadi jika kelas yang diprediksi adalah negatif, namun kelas sebenarnya adalah positif, dan *True Negative* (TN) yang terjadi ketika kelas yang diprediksi dan kelas sebenarnya adalah negatif (Markoulidakis, *et al*., 2021).

Pada kasus klasifikasi multikelas, metrik yang digunakan untuk klasifikasi biner tidak sepenuhnya berlaku. *Confusion matrix* untuk klasifikasi multikelas memiliki dimensi $N * N$, di mana N adalah jumlah kelas berbeda seperti $(C_0, C_1, \ldots, C_N)$ Dengan demikian definisi TP, TN, FP, dan FN tidak dapat diterapkan langsung pada kasus klasifikasi multikelas. Sebagai gantinya, analisis dapat dilakukan dengan berfokus pada kelas tertentu. Pendekatan ini mendefinisikan satu set metrik untuk setiap kelas. Kemudian, metrik-metrik ini digabungkan untuk mengukur performa klasifikasi multikelas secara keseluruhan dengan *confusion matrix*. Metrik-metrik yang dapat digunakan pada *confusion matrix* untuk klasifikasi multikelas adalah *accuracy, precision, recall* dan *F1-score*. *Confusion matrix multiclass* dapat dilihat pada Tabel 8 (Markoulidakis, *et al*., 2021).

Tabel 8. Evaluasi Model RoBERTa pada Berbagai Dataset

		Predicted Class			
		C_1	**C_2**	**...**	**C_N**
Actual Class	**C_1**	$C_{1,1}$	FP	...	$C_{1,N}$
	C_2	FN	TP	...	FN
	...	...	...	...	...
	C_N	$C_{N,1}$	FP	...	$C_{N,N}$

Berikut ini adalah beberapa metrik yang dapat digunakan pada *confusion matrix* untuk klasifikasi multikelas:

1. *Accuracy*

Accuracy atau akurasi adalah metrik yang digunakan untuk menghitung seberapa akurat model yang dibangun dalam melakukan klasifikasi. *Accuracy* secara matematis dapat dijelaskan dalam Persamaan (19) (Markoulidakis, *et al*., 2021).

$$Accuracy = \frac{\sum_{i=0} C_{ii}}{\sum_{i=0} \sum_{j=0} C_{ij}} \tag{19}$$

2. *Precision*

   Precision atau presisi adalah metrik yang menunjukan keakuratan model dalam memprediksi hasil positif dari yang diprediksi sebagai positif. *Precision* digunakan sebagai indikator kinerja ketika tujuannya adalah untuk mengurangi jumlah FP (Yun, 2021). *Precision* secara matematis dapat dijelaskan dalam Persamaan (20) dan (21).

$$\textit{Precision} \text{ kelas } C_i = PPV(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \tag{20}$$

$$\textit{Precision} \text{ (Rata-rata Makro)} = \textit{Precision}_{\,avg} = \frac{1}{N} \sum_{i=1}^{N} PPV(C_i) \tag{21}$$

3. *Recall* / Sensitivitas

   Recall atau sensitivitas adalah rasio antara jumlah prediksi positif yang benar dengan total data data positif yang sebenarnya. *Recall* digunakan untuk mengukur proporsi data positif yang berhasil diklasifikasikan dengan benar (Hossin & Sulaiman, 2015). *Recall* secara matematis dapat dijelaskan dalam Persamaan (22) dan (23).

$$\textit{Recall} \text{ kelas } C_i = TPR(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \tag{22}$$

$$\textit{Recall} \text{ (Rata-rata Makro)} = \textit{Recall}_{\,avg} = \frac{1}{N} \sum_{i=1}^{N} TPR(C_i) \tag{23}$$

4. Spesitifitas

   Spesitifitas adalah rasio antara jumlah prediksi negatif yang benar dengan total data data negatif yang sebenarnya. Spesitifitas digunakan untuk mengukur proporsi data negatif yang berhasil diklasifikasikan dengan benar (Hossin & Sulaiman, 2015). Spesitifitas secara matematis dapat dijelaskan dalam Persamaan (24) dan (25).

$$\text{Spesitifitas kelas } C_i = TNR(C_i) = \frac{TN(C_i)}{TN(C_i) + FP(C_i)} \tag{24}$$

$$\text{Spesitifitas (Rata-rata Makro)} = \text{Spesitifitas}_{avg} = \frac{1}{N}\sum_{i=1}^{N} TPR(C_i) \quad (25)$$

5. *F1-Score*

   F1-score adalah sebuah perhitungan metrik yang menggambarkan perbandingan antara *precision* dan *recall*. Metrik ini merupakan rata-rata harmonis tertimbang dari nilai *recall* dan *precision* (Hasibuan & Heriyanto, 2022). *F1-score* membantu mengevaluasi seberapa baik model memprediksi data positif dengan benar sekaligus mengidentifikasi sebagian besar data positif yang sebenarnya. Semakin tinggi nilai *F1-score*, maka semakin baik kinerja model klasifikasi tersebut. *F1-score* secara matematis dapat dijelaskan dalam Persamaan (26) dan (27).

$$F_1\text{-}Score \text{ kelas } C_i = \frac{2 \times TPR(C_i) \times PPV(C_i)}{TPR(C_i) + PPV(C_i)} \quad (26)$$

$$F_1\text{-}Score = F_1\text{-}Score_{avg} = \frac{2 \times TPR(\text{makro}) \times PPV(\text{makro})}{TPR(\text{makro}) + PPV(\text{makro})} \quad (27)$$

6. *Receiver Operating Characteristic* (ROC)

   Salah satu metode lain yang dapat digunakan untuk mengevaluasi kinerja model klasifikasi biner ataupun multikelas adalah kurva *Receiver Operating Characteristic* (ROC) (Markoulidakis, *et al*., 2021). Metode ini menyajikan hubungan antara *True Positive Rate* (TPR) dengan *False Positive Rate* (FPR) dengan ukuran 2 dimensi di mana garis horizontal adalag nilai *false positive* dan garis vertikal adalah *true positive* (Hidayatullah, *et al*., 2014). *False Positive Rate* (FPR) adalah risiko salah mengidentifikasi kejadian negatif sebagai positif dan *False Negatif Rate* (FNR) adalah risiko salah mengidentifikasi kejadian positif sebagai negatif. Persamaan FPR dan FNR secara matematis direpresentasikan pada Persamaan (28) dan (29) (Figueiredo, *et al*., 2018).

$$\text{FPR} = 1 - \text{Spesifisitas} = 1 - \text{TNR} \quad (28)$$

$$\text{FNR} = 1 - \text{Sensitivitas} = 1 - \text{TPR} \quad (29)$$

   Kurva ROC memberikan gambaran mengenai efektivitas model dalam mebedakan antara kelas positif dan negatif. Metode ini dapat memberikan perbadingan antara model yang diuji dengan pemilihan kelas secara acak yang direpresentasikan oleh garis lurus yang menghubungkan titik (0, 0) dan (1, 1). Model yang memiliki performa lebih baik akan menghasilkan kurva ROC yang berada di atas garis ini dan model yang lebih buruk akan berada dibawahnya (Markoulidakis, *et al*., 2021).

Salah satu metrik yang penting dalam kurva ROC adalah *Area Under the Curve* (AUC) yang mengukur luas area di bawah kurva ROC (Markoulidakis, *et al*., 2021). Nilai AUC terletak di rentang 0 hingga 1 oleh karena itu, model dengan AUC mendekati 1 menunjukan model tersebut memiliki kemampuan yang sangat baik dalam membedakan antara kelas positif dan negatif. Sebaliknya, jika AUC mendekati 0 menunjukan bahwa model memiliki kinerja yang sangat buruk untuk membedakan kelas positif dan negatif. Sementara jika AUC bernilai 0.5 maka model tidak memiliki kemampuan membedakan antara kedua kelas (Narkhede, 2018).

Pada klasifikasi biner, analisis ROC-AUC dapat diterapkan dengan mudah. Namun, dalam klasifikasi multikelas, kompleksitas analisis ROC-AUC akan meningkat seiring bertambahnya jumlah kelas. Pada masalah dengan N kelas, salah satu pendekatan yang dapat digunakan adalah membuat kurva ROC untuk setiap kelas terpisah dengan setiap kelas dianggap sebagai kelas positif, sementara kelas lain digabung sebagai kelas negatif, sehingga menghasilkan sejumlah N grafik ROC yang berbeda. Kemudian rata-rata nilai AUC tiap kelas akan diambil sebagai nilai AUC makro (Markoulidakis, *et al*., 2021). Secara matematis nilai AUC dapat direpresentasikan pada persamaan (30) dan (31) (Powers, 2020).

$$\begin{aligned} \text{AUC kelas } C_i &= \frac{\text{TPR}(C_i) - \text{FPR}(C_i) + 1}{2} \\ &= \frac{\text{TPR}(C_i) + \text{TNR}(C_i)}{2} \end{aligned} \tag{30}$$

$$\text{AUC (Rata-rata Makro)} = \text{AUC}_{avg} = \frac{1}{N}\sum_{i=1}^{N} AUC(C_i) \tag{31}$$

2.13 Uji-t Berpasangan

Uji-t berpasangan adalah salah satu bagian dari analisis statistik parametrik. Uji ini bertujuan untuk mengetahui apakah terdapat perbedaan rata-rata dua sampel yang saling berpasangan atau berhubungan. Uji-t berpasangan digunakan ketika data berasal dari pasangan yang saling terkait, seperti sebelum atau sesudah perlakuan pada subjek yang sama (Manfei, *et al*., 2017). Tahapan pengujian hipotesis dengan uji-t berpasangan adalah sebagai berikut:

1. Hipotesis:
   H_0 : Tidak terdapat perbedaan yang siginifikan $(\mu_1 = \mu_2)$.
   H_1 : Terdapat perbedaan yang siginfikan $(\mu_1 \neq \mu_2)$.

2. Taraf Signifikansi
   $\alpha = 5\% = 0.05$
3. Statistik Uji:
   Secara matematis rumus untuk uji-t berpasangan dapat dilihat pada Persamaan (32)

$$t = \frac{\bar{d}}{\frac{Sd}{\sqrt{n}}} \tag{32}$$

   dengan:

$$Sd = \sqrt{var} \tag{33}$$

$$var(s^2) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{34}$$

   di mana:
   t = nilai t hitung
   $\bar{d}$ = rata-rata selisih pengukuran 1 dan 2
   Sd = standar deviasi selisih pengukuran 1 dan 2
   n = jumlah sampel
4. Kriteria Uji:
   Tolak H_0 jika $p - value < \alpha = 0.05$ atau $T_{\text{hitung}} > T_{\text{tabel}}$.
   Tidak tolak H_0 jika $p - value < \alpha = 0.05$ atau $T_{\text{hitung}} < T_{\text{tabel}}$.
5. Kesimpulan
   Kesimpulan dibuat berdasarkan keputusan uji yang telah diambil. Jika H_0 ditolak, maka dapat disimpulkan bahwa terdapat perbedaan rata-rata yang signifikan antara dua kelompok. Sedangkan, jika H_0 tidak ditolak, maka tidak terdapat perbedaan yang signifikan antara kedua kelompok tersebut.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan Tempat Penelitian ini yaitu sebagai berikut:

3.1.1 Tempat Penelitian

Penelitian ini dilaksanakan secara studi literatur di jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No.1, Gedong Meneng, Bandar Lampung.

3.1.2 Waktu Penelitian

Penelitian ini dilaksanakan pada semester ganjil tahun akademik 2024/2025, tepatnya pada bulan September 2024. Penelitian ini terdiri dari tiga tahap utama. Tahap pertama adalah studi literatur, di mana topik penelitian yang relevan digunakan sebagai refrensi dalam penyusunan proposal peneltian. Selanjutnya adalah melakukan pengumpulan data yang akan digunakan dalam penelitian dan penyusunan draf proposal penelitian. Tahap kedua adalah pengerjaan program yang dimulai dari *input data, preprocessing data, data augmentation, word embedding* dengan RoBERTa, pemodelan klasifikasi menggunakan RoBERTa, *fine-tuning* RoBERTa, dan evaluasi metrik dari model yang telah dibangun. Tahap ketiga adalah penyusunan hasil penelitian dan kesimpulan penelitian.

3.2 Data dan Alat

3.2.1 Data

Data yang digunakan dalam penelitian ini merupakan data teks yang bersumber dari situs Kaggle `https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification`. Data tersebut merupakan data teks berbahasa Inggris yang dan terakhir diperbarui pada Januari 2020 oleh Andrew Maranhao. Data ini merupakan kumpulan *tweet* dari media sosial Twitter yang sudah berlabel dengan total 6 label berdasarkan jenis *cyberbullying yaitu age, ethnicity, gender, religion*, *other_cyberbullying*, dan *not_cyberbullying*. Pada penelitian ini saya tidak menggunakan data dengan label *other_cyberbullying* disebabkan beberapa alasan yaitu, label *other_cyberbullying* memiliki kategori yang luas dan ambigu, dapat menyebabkan *overlapping* dengan label lain, dan tidak membawa informasi yang spesifik untuk klasifikasi jenis *cyberbullying*. Oleh karena itu, dengan menghapus label *other_cyberbullying*, diharapkan performa model akan meningkat dan model dapat lebih fokus terhadap klasifikasi jenis *cyberbullying* yang sudah terdefinisi.

Dataset ini memiliki 2 atribut/variabel yaitu *tweet_text* dan *cyberbullying_type*. *Tweet_text* berisikan *tweet* yang di posting dan *cyberbullying_type* adalah jenis *cyberbullying* dari *tweet* tersebut. Pada penelitian ini menggunakan data kualitatif dengan total jumlah pengamatan sebanyak 37288 pengamatan. Data menggunakan semua atribut dalam melakukan klasifikasi. Data ini termasuk data *supervised learning* di karenakan data sudah memiliki label. Sebaran jumlah tweet berdasarkan jenis *cyberbullying* dapat disajikan pada Tabel 9.

Tabel 9. Sebaran *Tweet* Berdasarkan Jenis *Cyberbullying*

Jenis *Cyberbullying*	**Jumlah**	**Persentase (%)**
Religion	7965	21.4
Age	7910	21.2
Ethnicity	7499	20
Gender	7592	20.4
Not Cyberbullying	6322	17

Kemudian sampel dari kumpulan data yang digunakan dalam penelitian ini disajikan dalam Tabel 10.

Tabel 10. Sampel Data *Tweet Cyberbullying*

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying
...	...	...
38838	Black ppl aren't expected to do anything, depe...	ethnicity
38839	Turner did not withhold his disappointment. Tu...	ethnicity
38840	I swear to God. This dumb nigger bitch. I have...	ethnicity
38841	Yea fuck you RT @therealexel: IF YOURE A NIGGE...	ethnicity
38842	Bro. U gotta chill RT @CHILLShrammy: Dog FUCK ...	ethnicity

3.2.2 Alat

Adapun peralatan yang digunakan dalam penelitian ini adalah:

a. Perangkat Keras (*Hardware*)
   Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan merek Lenovo Ideapad Slim 3 - 81W0 14ADA05 dengan 64-bit *operating system*. Spesifikasi *hardware* perangkat tersebut adalah sebagai berikut:
   1. *Preocessor* AMD Ryzen 3 3250U *with Radeon Graphics* (4 CPUs), $\tilde{2}$.6GHZ
   2. Installed RAM 4 GB

b. Perangkat Lunak (*Software*)
   Perangkat lunak yang digunakan pada penelitian ini untuk mendukung dan menunjang penelitian ini, yaitu:
   1. Sistem operasi Windows 11 *Home Single Language* 64-bit
   2. Python 3.10.14
      Peneliti menggunakan *website* Kaggle sebagai *notebook* Python. Adapun *package* yang digunakan dalam penelitian sebagai berikut:

a. Pandas versi 2.2.3
   Pandas adalah *library* Python yang digunakan untuk mengelola dan menganalisis data terstruktur atau tabel secara cepat, mudah, dan efisien (McKinney, 2022).

b. Numpy versi 1.26.4
   Numpy *Numerical Python* (Numpy) adalah *library* Python yang digunakan untuk komputasi numerik yang memungkinkan pengelolaan *array* dan matriks berukuran besar, serta menyediakan berbagai fungsi matematika untuk operasi pada *array* tersebut (McKinney, 2022).

c. Sklearn versi 1.2.2
   Scikit-learn (Sklearn) adalah *library* serba guna yang digunakan untuk mempermudah penerapan *machine learning* dalam Python sejak tahun 2010 (McKinney, 2022).

d. Matplotlib 3.7.5
   Matplotlib adalah *library* Python yang digunakan untuk membuat grafik dan plot visualisasi data dalam 2D dan 3D (McKinney, 2022).

e. Seaborn versi 0.12.2
   Seaborn adalah *library* Python yang digunakan untuk membuat grafik statistik yang lebih canggih, menarik dan informatif seperti *box plot, heat map,pair plot, line plot*, dan lain-lain (Ranjan, *et al*., 2023).

f. NLTK 3.2.4
   Natural Language Toolkit (NLTK) adalah *library* yang digunakan untuk pemrosesan bahasa alami dalam Python dengan cepat dan mudah (Wang & Hu, 2021).

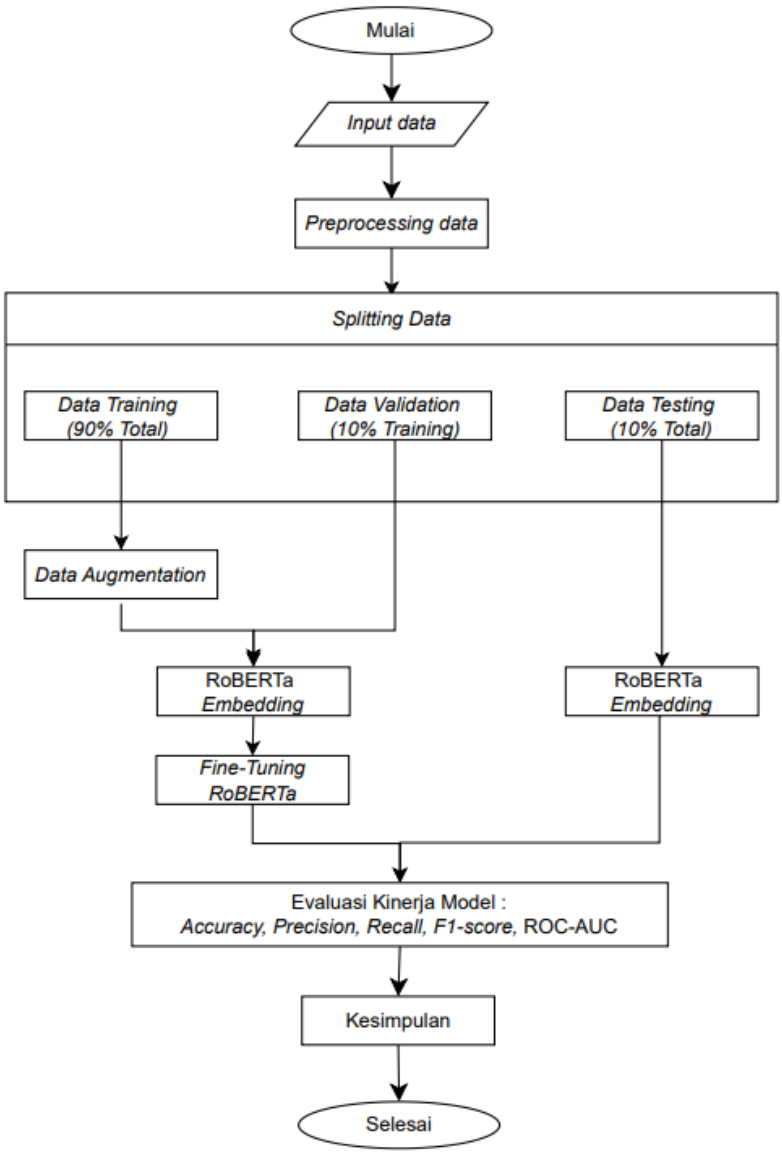
g. Transformer versi 4.45.1
   Transformer adalah *library* Python yang dibuat untuk mendukung arsitektur model berbasis *Transformer* dan distribusi model *pre-train* (Wolf, *et al*., 2020).

h. Pytorch versi 2.3.0+cpu
   Pytorch adalah *open source deep learning framework* yang populer digunakan untuk pengembangan dan pelatihan *neural network* (Osborne, *et al*., 2024).

i. Wordcloud versi 1.9.3
   Wordcloud adalah *library* Python yang digunakan untuk memvisualisasikan kata-kata yang sering muncul dalam teks, sehingga memudahkan untuk melihat dan menganalisis teks (Murthy & Madhav, 2020).

3.3 Metode Penelitian

Penelitian ini melakukan augmentasi data menggunakan teknik *back translation* dan menggunakan model RoBERTa untuk melakukan klasifikasi *tweet cyberbullying*. Penelitian ini dilakukan menggunakan perangkat lunak *Python* dengan bantuan teks editor Kaggle dan Google colab. Evaluasi model RoBERTa dalam klasifikasi teks akan didasarkan pada penggunaan *confusion matrix* dan selanjutnya membangun metrik evaluasi seperti *accuracy, precision, recall*, dan *F1-score*. Langkah-langkah yang dilakukan pada penelitian dapat diilustrasikan pada Gambar 16.



Gambar 16. *Flowchart* Prosedur Penelitian.

Penjelasan dari langkah-langkah yang dilakukan pada penelitian ini sebagai berikut:

1. Melakukan *input* data pada *software* Python dengan Google Colab sebagai teks editor untuk menulis kode Python.
2. Melakukan *preprocessing* pada data yang telah diinput. *preprocessing* data yang dilakukan meliputi *remove duplicated data*, *data cleaning*, *case folding*, dan normalisasi teks.
3. Melakukan *spliting data* menjadi 90% data model dan 10% data *testing*. Kemudian 90% data model dibagi menjadi 90% data *training* dan 10% data *validation*.
4. Melakukan augmentasi data menggunakan teknik *back translation* untuk mengatasi ketidakseibangan data dan menambah variasi data pelatihan. *Back translation* dilakukan dengan model terjemahan MarianMT pada *data training*. Data secara *random* dipilih sebanyak yang dibutuhkan, lalu diterjemahkan ke dalam bahasa target yaitu bahasa Indonesia dan diterjemahkan kembali ke bahasa sumber yaitu bahasa Inggris.
5. Menyatukan data hasil *preprocessing* diawal dan data hasil augmentasi.
6. Mengimplementasikan RoBERTa *embedding* menggunakan RoBERTa-BASE pada *data model* dan *data testing*. Setiap *tweet* dalam data memiliki panjang token yang bervariasi untuk data penelitian ini ditentukan panjang token maksimal sebanyak 38 token dari batas maksimal RoBERTa adalah 512 token. Apabila panjang kalimat melebihi batas maksimum yang telah ditentukan, maka teks akan dikurangi (*truncate*) sampai kebatas maksimum. Sedangkan jika panjang kalimat kurang dari panjang maksimum yang telah ditentukan, maka akan ditambahkan token [PAD].
7. Melakukan *fine-tuning* RoBERTa untuk tugas klasifikasi *cyberbullying* dengan menggunakan model *pre-trained RobertaForSequenceClassification*. Setelah itu, model dibangun dan dilatih menggunakan *data train* dan *data validation*. Selanjutnya menentukan parameter yang digunakan seperti *batch size, dropout, learning rate, patience, weight decay*, dan *epoch*.
8. Menguji model RoBERTa dengan *data testing*.
9. Melakukan evaluasi kinerja model untuk melihat performa model klasifikasi dengan menggunakan nilai *accuracy, precision, recall*, *F1-score*, grafik ROC, dan nilai AUC.
10. Melakukan *benchmarking* dengan penelitian terdahulu dan membuat kesimpulan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini, telah dilakukan implementasi dan evaluasi model RoBERTa dengan teknik augmentasi data *back translation* dalam klasifikasi *tweet cyberbullying* yang memberikan hasil yang cukup memuaskan. Beberapa kesimpulan yang didapatkan dalam penelitian ini adalah sebagai berikut:

1. Pembangunan model klasifikasi *tweet cyberbullying* dilakukan dengan beberapa langkah seperti *preprocessing* data, *splitting* data, augmentasi data, RoBERTa *embedding*, dan *fine tuning model*. Selanjutnya model yang telah dibangun akan dievaluasi kinerjanya dalam mengklasifikasikan *tweet cyberbullying* dengan tujuan untuk memberikan model yang memiliki kinerja yang optimal. Beberapa metrik evaluasi yang digunakan adalah *accuracy*, *precision*, *recall*, *F1 Score*, dan ROC-AUC.
2. Model RoBERTa menunjukan kinerja yang baik dalam melakukan klasifikasi *tweet cyberbullying*. Pada metode tanpa augmentasi data, model RoBERTa menghasilkan *accuracy* sebesar 88%. Sedangkan, model RoBERTa dengan metode augmentasi data menghasilkan *accuracy* sebesar 93% dengan peningkatan akurasi sebesar 5%. Selain peningkatan *accuracy*, metode dengan augmentasi data memperoleh *precision* sebesar 92%, *recall* sebesar 92%, *F1 Score* sebesar 92%, dan rata-rata nilai AUC sebesar 0.9519. Hasil tersebut didapatkan dengan menggunakan parameter yang meliputi *batch size* 32, *dropout* 0.2, dan *learning rate* 5×10^{-7}. Hal ini menunjukan bahwa teknik augmentasi data seperti *back translation* memberikan dampak positif terhadap kinerja model RoBERTa dalam melakukan klasifikasi *tweet cyberbullying*.

Secara keseluruhan, hasil dari penelitian ini dapat memberikan kontribusi penting dalam pemahaman dan penerapan model RoBERTa dan teknik *back translation* untuk tugas klasifikasi *tweet cyberbullying*. Selain itu, hasil dari penelitian ini juga dapat dijadikan refrensi penelitian selanjutnya dalam pengembangan model deteksi *cyberbullying* secara otomatis yang lebih akurat.

5.2 Saran

Berdasarkan hasil peneltian yang telah dilakukan, berikut ini adalah beberapa saran yang dapat dijadikan bahan untuk pertimbangan bagi penelitian selanjutnya:

1. Menggunakan model *pretrained* alternatif lainya yang lebih spesifik untuk teks media sosial atau klasifikasi *cyberbullying* untuk memperoleh kinerja model yang lebih baik.
2. Mengoptimaliasi *hyperparameter tuning* yang lebih luas untuk mengeksplorasi berbagai nilai *batch size*, *learning rate*, dan *dropout*, sehingga menemukan kombinasi yang optimal dan meningkatkan kinerja model.
3. Menggunakan teknik augmentasi data yang lebih bervariasi yang bertujuan untuk lebih meningkatkan keberagaman pada data *training*. Beberapa teknik augmentasi yang dapat digunakan seperti *synonim replacement*, *random insertion*, *text generative*, dan lain-lain.
4. Mempertimbangkan untuk menggunakan model *transformer* lainya seperti BERT, *Distill*-BERT, GPT, dan lain-lain untuk penelitian selanjutnya. Hal tersebut dilakukan untuk melakukan studi perbandingan kinerja antara berbagai model yang dapat memberikan wawasan yang lebih luas mengenai efektivitas berbagai pendekatan dalam konteks klasifikasi *cyberbullying*.

DAFTAR PUSTAKA

Adoma, A. F., Henry, N. M., & Chen, W. 2020. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition, 117-121. 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).

Aggrawal, N. 2018. Detection of offensive tweets: A comparative study. *Computer Reviews Journal*. **1**(1): 75-89.

Ali, A. & Syed, A. M. 2020. Cyberbullying Detection Using Machine Learning. *Pakistan Journal of Engineering and Technology*. **3**(2): 45-50.

Alwehaibi, A., Bikdash, M., Albogmi, M., & Roy, K. 2022. A Study of the Performance of Embedding Methods for Arabic Short-text Sentiment Analysis Using Deep Learning Approaches. *Journal of King Saud University-Computer and Information Sciences*. **34**(8): 6140-6149.

Al-Jarrah, H., Al-Hamouri, R., & Mohammad, A. S. 2020. HR@JUST team at SemEval-2020 Task 4: The impact of RoBERTa transformer for evaluation common sense understanding, 521-526. Proceedings of the Fourteenth Workshop on Semantic Evaluation.

APJII. 2024. APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang. https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang. Diakses pada 8 Oktober 2024.

Arrasyid, R. M., Putera, D. E., & Yusuf, A. Y. P. 2024. Analisis Sentimen Review Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing. *Jurnal Tekno Kompak*. **18**(2): 319-330.

Aulia, G. P., Widiharih, T., & Utami, I. T. 2023. Penerapan Text Mining Dan Fuzzy C-Means Clustering Untuk Identifikasi Keluhan Utama Pelanggan PDAM Tirta Moedal Kota Semarang. *Jurnal Gaussian*. **12**(1): 126-135.

Basbeth, F. & Fudholi, D. H. 2024. Klasifikasi Emosi Pada Data Text Bahasa Indonesia Menggunakan Algoritma BERT, RoBERTa, dan Distil-BERT. *Jurnal Media Informatika Budidarma*. **8**(2): 1160-1170.

Beddiar, D. R., Jahan, M. S., & Oussalah, M. 2021. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*. **24**. 100153.

Bucos, M. & Țucudean, G. 2023. Text Data Augmentation Techniques for Fake News Detection in the Romanian Language. *Applied Sciences*. **13**(13). 7389.

Budiman, I., Muliadi, & Ramadina, R. 2015. Penerapan Fungsi Data Mining Klasifikasi untuk Prediksi Masa Studi Mahasiswa Tepat Waktu pada Sistem Informasi Akademik Perguruan Tinggi. *JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer*. **7**(1): 39-50.

Chaudhary, A. 2020. Text Data Augmentation with MarianMT. https://amitness.com/posts/back-translation. Diakses pada 26 Oktober 2024.

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., & Lu, H. 2021, 8126-8135. Transformer tracking. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

CNN Indonesia. 2022. Ujaran Kebencian Hingga Akun Rasis Naik Tajam di Twitter Era Elon Musk. https://www.cnnindonesia.com/teknologi/20221203032125-185-882226/ujaran-kebencian-hingga-akun-rasis-naik-tajam-di-twitter-era-elon-musk. Diakses pada 10 Oktober 2024.

Desiani, A., Adrezo, M., Kresnawati, E. S., Akbar, M., & Hasibuan, M. S. 2023. Back Translation-EDA and Transformer for Hate Speech Classification in Indonesian, 611-616. International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, 4171–4186. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Edunov, S., Ott, M., Aulim, M., & Grangier, D. 2018. Understanding back-translation at scale. arXiv preprint arXiv:1808.09381.

Eisenstein, J. 2018. *Natural language processing*. Jacob Eisenstein. 507.

Fadli, A. & Sazali, H. 2023. Peran Media Sosial Instagram @GREENPEACEID Sebagai Media Kampanye dalam Menjaga Lingkungan. *Jurnal Ilmu Komunikasi UHO: Jurnal Penelitian Kajian Ilmu Komunikasi Dan Informasi*. **8**(2): 209-222.

Feldman, R. & Sanger, J. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

Figueiredo, D. M., Cordella, C. B., Bouveresse, D. J. R., Archer, X., Bégué, J. M., & Rutledge, D. N. 2018. A variable selection method for multiclass classification problems using two-class ROC analysis. *Chemometrics and Intelligent Laboratory Systems*. (177): 35-46.

Firmansyah, I. & Hayadi, B. H. 2022. Komparasi Fungsi Aktivasi Relu Dan Tanh Pada Multilayer Perceptron. *JIKO (Jurnal Informatika dan Komputer)*. **6**(2): 200-206.

Fithriasari, K., Hariastuti, I., & Wening, K. S. 2020. Handling imbalance data in classification model with nominal predictors. *(IJCSAM) International Journal of Computing Science and Applied Mathematics*. **6**(1): 33-37.

Gao, L., Zhang, L., Zhang, L., & Huang, J. 2022. RSVN: A RoBERTa Sentence Vector Normalization Scheme for Short Texts to Extract Semantic Information. *Applied Sciences*. **12**(21). 11278.

Han, J. & Kamber, M. 2006. *Data Mining Concepts and Techniques*. 2nd Edition. Morgan Kaufmann. San Fransisco.

Hanna, M. & Bojar, O. 2021. A fine-grained analysis of BERTScore, 507-517. Proceedings of the 6thConference on Machine Translation.

Hardiyanti, K. & Indawati, Y. 2023. Perlindungan Bagi Anak Korban Cyberbullying: Studi Di Komisi Perlindungan Anak Indonesia Daerah (KPAID) Jawa Timur. *Sibatik Journal: Jurnal ilmiah bidang sosial, ekonomi, budaya, teknologi, dan pendidikan.* **2**(4): 1179-1198.

Hasibuan, E. & Heriyanto, E. A. 2022. Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier. *Jurnal Teknik dan Science*. **1**(3): 13-24.

Hidayatullah, A. F., Prasetyo, A. D., Sari, D. P., & Pratiwi, I. 2014. Analisis Kualitas Data dan Klasifikasi Data Pasien Kanker. Seminar Nasional Informatika Medis SNIMed. 38-47.

Hossin, M. & Sulaiman, M. N. 2015. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*. **5**(2). 1.

Ilahiyah, S. & Nilogiri, A. 2018. Implementasi deep learning pada identifikasi jenis tumbuhan berdasarkan citra daun menggunakan convolutional neural network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*. **3**(2): 49-56.

Joseph, V. R. 2022. Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*. **15**(4): 531-538.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A, & Birch, A. 2018. Marian: Fast neural machine translation in C++. arXiv preprint arXiv:1804.00344.

Kaope, C. & Pristyanto, Y. 2023. The Effect of Class Imbalance Handling on Datasets Toward Classification Algorithm Performance. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*. **22**(2): 227-238.

Kemp, S. 2023 Digital 2023: Indonesia. https://datareportal.com/reports/digital-2023-indonesia. Diakses pada 8 Oktober 2024.

Kemp, S. 2024. Digital Around The World. https://datareportal.com/global-digital-overview. Diakses pada 8 Oktober 2024.

Khusuma, R., Maharani, W., & Gani, P. H. 2023. Personality Detection On Twitter User With RoBERTa. *Jurnal Media Informatika Budidarma*. **7**(1): 542-553.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). *Text classification algorithms: A survey. Information*. **10**(4): 150.

Lee, M. 2023. Mathematical analysis and performance evaluation of the gelu activation function in deep learning. *Journal of Mathematics*. (1). 4229924.

Liddy, E. D. 2001. *Natural language processing*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

Ma, J. & Li, L. 2020. Data augmentation for chinese text classification using back-translation. *Journal of Physics: Conference Series*. **1651**(1).

Manfei, X. U., Fralick, D., Zheng, J. Z., Wang, B., Xin, M. T., & Changyong, F. E. N. G. 2017. The differences and similarities between two-sample t-test and paired t-test. *Shanghai archives of psychiatry*. **29**(3): 184.

Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. 2021. Multi-class confusion matrix reduction method and its application on net promoter score classification problem, 412-419. Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference.

McCormick, C. & Ryan, N. 2019. BERT Word Embeddings Tutorial. http://www.mccormickml.com. Diakses pada 20 Februari 2025.

McKinney, W. 2022. *Python for data analysis: Data wrangling with pandas, numpy, and jupyter*. O'Reilly Media Inc.

Mienye, I. D., & Swart, T. G. 2024. A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information*. **15**(12): 755.

Muliono, Y., Gaol, F. L., Soewito, B., & Warnars, H. L. H. S. 2022. Hoax Classification in Imbalanced Datasets Based on Indonesian News Title using RoBERTa, 264-268. 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS).

Muraina, I. 2022. Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts, 496-504. 7th international Mardin Artuklu scientific research conference.

Murarka, A., Radhakrishnan, B., & Ravichandran, S. 2021. Classification of Mental Illnesses on Social Media using RoBERTa, 59-68. Proceedings of the 12th international workshop on health text mining and information analysis.

Murthy, K. N. & Madhav, S. V. (2020). Word Cloud in Python. *Complexity International*. **24**(01).

Narkhede, S. 2018. Understanding auc-roc curve. *Towards data science*. **26**(1): 220-227.

Nayla, A., Setianingsih, C., & Dirgantoro, B. 2023. Deteksi Hate Speech Pada Twitter Menggunakan Algoritma BERT. eProceedings of Engineering. **10**(1).

Novack, G. 2023. BERT Embeddings. Tinkerd. https://tinkerd.net/blog/machine-learning/bert-embeddings/. Diakses pada 22 Februari 2025.

Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. 2020. Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*. 1**4**(2): 74-79.

Oktavia, A. N., Iqbal, M., Saputra, R. W., Zulfikar, M. I., & Saifudin, A. 2024. Implementasi Metode Natural Language Processing Dalam Studi Analisis Semantik Dan Emosi Buzzer Pada Tweet Di Aplikasi X. *Buletin Ilmiah Ilmu Komputer dan Multimedia (BIIKMA)*. **2**(1): 154-159.

Osborne, C., Daneshyan, F., He, R., Ye, H., Zhang, Y., & Zhou, M. 2024. Characterising Open Source Co-opetition in Company-hosted Open Source Software Projects: The Cases of PyTorch, TensorFlow, and Transformers. arXiv preprint arXiv:2410.18241.

Patmawati, P. & Yusuf, M. 2021. Analisis Topik Modelling Terhadap Penggunaan Sosial Media Twitter oleh Pejabat Negara. *Building of Informatics, Technology and Science (BITS)*. **3**(3): 122-129.

Powers, D. M. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Priyambodo, A. & Prihati, P. 2020. Evaluasi Ekstraksi Fitur Klasifikasi Teks Untuk Peningkatan Akurasi Klasifikasi Menggunakan Naive Bayes. *Elkom: Jurnal Elektronika dan Komputer*. **13**(1): 159-175.

Purnama, A. 2021. Implementasi Metode Deep Learning Dengan Menggunakan Algoritma Convolution Neural Network (CNN) Pada Citra Tulisan Tangan Aksara Sunda. *INSERT: Information System and Emerging Technology Journal*. **2**(1): 46-58.

Purwitasari, N. A. & Soleh, M. 2022. Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing. *JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI (IPTEK)*. **6**(1): 14-21.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog*. **1**(8): 9.

Rahma, I. A. & Suadaa, L. H. 2023. Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **10**(6): 1329-1340.

Rahayu, F. S. 2012. Cyberbullying sebagai dampak negatif penggunaan teknologi informasi. *Journal of Information Systems*. **8**(1): 22-31.

Ranjan, M. K., Barot, K., Khairnar, V., Rawal, V., Pimpalgaonkar, A., Saxena, S., & Sattar, A. M. 2023. Python: Empowering Data Science Applications and Research. *Journal of Operating Systems Development & Trends*. **10**(1): 27-33.

Raykar, V. C. & Saha, A. 2015. Data split strategiesfor evolving predictive models, 3-19. Proceedings Part I 15 Machine Learning and Knowledge Discovery in Databases: European Conference. Porto, Portugal. Springer International Publishing.

Rovida, K. & Sasmini, S. 2024. Evaluasi Sistem Hukum Indonesia dalam Menangani Cyberbullying Berbasis Teknologi Informasi dan Komunikasi. *DIVERSI: Jurnal Hukum*. **10**(1): 169-205.

Rush, A. M. 2018. The annotated transformer, 52-60. Proceedings of workshop for NLP open source software (NLP-OSS).

Sartana, S. & Afriyeni, N. 2017. Perundungan Maya (CyberBullying) pada Remaja Awal. *Jurnal Psikologi Insight*. **1**(1): 25-39.

Setiadi, B., Purwanto, E., & Permatasari, H. 2024. Optimisasi Klasifikasi Sentimen Pada Review Hotel Bahasa Inggris Dengan Model Roberta Twitter. *SINTECH (Science and Information Technology) Journal*. **7**(2): 70-79.

Sharma, S., Sharma, S., & Athaiya, A. 2017. Activation functions in neural networks. *Towards Data Sci*. **6**(12): 310-316.

Sitepu, A. C. & Sigiro, M. 2021. Analisis fungsi aktivasi relu dan sigmoid menggunakan optimizer SGD dengan representasi MSE pada model backpropagation. *JUTISAL Jurnal Teknik Informatika Universal*. **1**(1): 12-25.

Soliman, A. S., Hadhoud, M. M., & Shaheen, S. I. 2022. MarianCG: a code generation transformer model inspired by machine translation. *Journal of Engineering and Applied Science*. **69**(1): 1-23.

Soltanzadeh, P. & Hashemzadeh, M. 2021. RCSMOTE: Range-Controlled Synthetic Minority Over-sampling Technique for Handling the Class Imbalance Problem. *Information Sciences*. **542**: 92-111.

Soydaner, D. 2020. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*. **34**(13). 2052013.

Stahlberg, F. 2020. Neural machine translation: A review. *Journal of Artificial Intelligence Research*. **69**: 343-418.

Statista. 2023. Countries with the Most Twitter Users 2023. https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/. Diakses pada 9 Oktober 2024.

Statista. 2024. Age Distribution of Global Twitter Users 2024. https://www.statista.com/statistics/283119/age-distribution-of-global-twitter-users/. Diakses pada 10 Oktober 2024.

Sterner, G. & Felmlee, D. 2017. The Social Networks of Cyberbullying on Twitter. *International Journal of Technoethics (IJT)*. **8**(2): 1-15.

Stieglitz, S. & Dang-Xuan, L. 2013. Emotions and information diffusion in social media sentiment of microblogs and sharing behavior. *Journal of management information systems*. **29**(4): 217-248.

Surjandari, I., Megawati, C., Dhini, A., & Hardaya, I. B. N. S. 2016. Application of text mining for classification of textual reports: a study of Indonesia's national complaint handling system. *6th International Conference on Industrial Engineering and Operations Management (IEOM 2016)*.

Syifa, S. A. & Dewi, I. A. 2022. Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru-Paru. *MIND (Multimedia Artificial Intelligent Networking Database) Journal*. **7**(2): 139-150.

Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., & Liu, Y. 2020. Neural machine translation: A review of methods, resources, and tools. *AI Open*. **1**: 5-21.

Tantika, R. S. & Kudus, A. 2022. Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid. *Bandung Conference Series: Statistics*. **2**(2): 159-166.

Taylor, L. & Nitschke, G. 2018. Improving Deep Learning with Generic Data Augmentation, 1542-1547. IEEE symposium series on computational intelligence (SSCI).

Tsani, E. F. & Suhartono, D. 2023. Personality Identification from Social Media Using Ensemble BERT and RoBERTa. *Informatica*. **47**(4).

Ustyannie, W., & Suprapto, S. 2020. Oversampling Method To Handling Imbalanced Datasets Problem in Binary Logistic Regression Algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. **14**(1): 1-10.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L.,& Polosukhin, I. 2017. Attention is all you need. Advances in Neural Information Processing Systems.

Vindua, R. & Zailani, A. U. 2023. Analisis Sentimen Pemilu Indonesia Tahun 2024 Dari Media Sosial Twitter Menggunakan Python. *JURIKOM (Jurnal Riset Komputer)*. **10**(2): 479-487.

Wang, A. & Potika, K. 2021. Cyberbullying classification based on social network analysis, 87-95. IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService).

Wang, M. & Hu, F. 2021. The application of nltk library for python natural language processing in corpus research. *Theory and Practice in Language Studies*. **11**(9): 1041-1049.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., & Liu, H. 2018. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*. **87**:12-20.

Wang, Y., Liu, F., Verspoor, K., & Baldwin, T. 2020. Evaluating the Utility of Model Configurations and Data Augmentation on Clinical Semantic Textual Similarity, 105-111. Proceedings of the 19th SIGBioMed workshop on biomedical language processing.

Wang, S., Li, Z., Chao, W., & Cao, Q. 2012. Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning, 1-8. The 2012 international joint conference on neural networks (IJCNN).

Wei, J. & Zou, K. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. M. 2020. Transformers: State-of-the-art natural language processing, 38-45. Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations.

Yanto, B., Fimawahib, L., Supriyanto, A., Hayadi, B. H., & Pratama, R. R. 2021. Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network. *Jurnal Inovtek Polbeng Seri Informatika*. **6**(2): 259-268.

Yona, M. 2011. Pembentukan Pohon Klasifikasi Biner Dengan Algoritma QUEST (Doctoral dissertation, Universitas Andalas).

Yun, H. 2021. Prediction model of algal blooms using logistic regression and confusion matrix. *International Journal of Electrical and Computer Engineering*. **11**(3): 2407-2413.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.