

**ANALISIS KINERJA AUGMENTASI TEKS *BACK TRANSLATION* DAN
SYNONYM REPLACEMENT DALAM KLASIFIKASI STATUS
KESEHATAN MENTAL MENGGUNAKAN *DISTILLED-BERT*
(DISTILBERT)**

Skripsi

Oleh

**LUSIANA FERISCA
NPM. 2117031056**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRACT

PERFORMANCE ANALYSIS OF BACK TRANSLATION AND SYNONYM REPLACEMENT TEXT AUGMENTATION IN MENTAL HEALTH STATUS CLASSIFICATION USING DISTILLED-BERT (DISTILBERT)

By

Lusiana Ferisca

Mental health is an important aspect that supports the overall quality of life of individuals. In today's digital age, social media has become a space for people to express their psychological conditions. This phenomenon opens up opportunities to utilize text data as a source of information in the process of identifying mental health disorders. This study aims to analyze the performance of two text-based data augmentation techniques, namely back translation and synonym replacement, in improving the performance of mental health status classification using the DistilBERT model. The study was conducted on English-language text data labeled according to the type of mental disorder. The synonym replacement and back translation augmentation techniques were applied to balance the data distribution in the minority class. The dataset was divided into 80% training data and 20% test data, with 20% of the training data used as validation data. The classification process was performed using a fine-tuned DistilBERT model. The results showed that the DistilBERT model with synonym replacement augmentation achieved the highest accuracy, namely 87%, while back translation achieved an accuracy of 86%. This indicates that the synonym replacement augmentation technique is more effective in increasing data variation and classification model performance on the dataset used in this study.

Keywords: Mental Health, Text Augmentation, Text Classification, DistilBERT.

ABSTRAK

ANALISIS KINERJA AUGMENTASI TEKS *BACK TRANSLATION* DAN *SYNONYM REPLACEMENT* DALAM KLASIFIKASI STATUS KESEHATAN MENTAL MENGGUNAKAN *DISTILLED-BERT* (DISTILBERT)

Oleh

Lusiana Ferisca

Kesehatan mental merupakan aspek penting yang menunjang kualitas hidup individu secara menyeluruh. Pada era digital saat ini, media sosial telah menjadi ruang ekspresi bagi masyarakat dalam mengungkapkan kondisi psikologis yang dialami. Fenomena ini membuka peluang untuk memanfaatkan data teks sebagai sumber informasi dalam proses identifikasi gangguan kesehatan mental. Penelitian ini bertujuan untuk menganalisis kinerja dua teknik augmentasi data berbasis teks, yaitu *back translation* dan *synonym replacement*, dalam meningkatkan kinerja klasifikasi status kesehatan mental menggunakan model DistilBERT. Penelitian dilakukan terhadap data teks berbahasa Inggris yang telah diberi label berdasarkan jenis gangguan mental. Teknik augmentasi *synonym replacement* dan *back translation* diterapkan guna menyeimbangkan distribusi data pada kelas minoritas. Dataset dibagi menjadi 80% data latih dan 20% data uji, dengan 20% dari data latih digunakan sebagai data validasi. Proses klasifikasi dilakukan dengan model DistilBERT yang telah di *fine-tuning*. Hasil penelitian menunjukkan bahwa model DistilBERT dengan augmentasi *synonym replacement* mencapai akurasi tertinggi, yaitu sebesar 87%, sedangkan *back translation* mencapai akurasi sebesar 86%. Hal ini menunjukkan bahwa teknik augmentasi *synonym replacement* lebih efektif dalam meningkatkan variasi data dan kinerja model klasifikasi pada dataset yang digunakan dalam penelitian ini.

Kata kunci: Kesehatan Mental, Augmentasi Teks, Klasifikasi Teks, DistilBERT.

**ANALISIS KINERJA AUGMENTASI TEKS *BACK TRANSLATION* DAN
SYNONYM REPLACEMENT DALAM KLASIFIKASI STATUS
KESEHATAN MENTAL MENGGUNAKAN *DISTILLED-BERT*
(DISTILBERT)**

LUSIANA FERISCA

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**

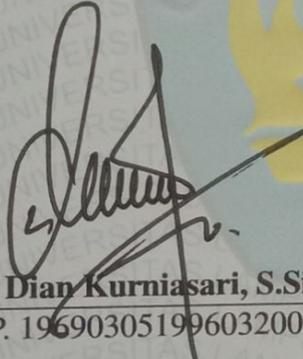
2025

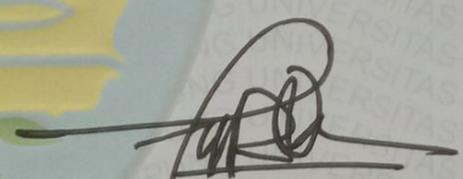
Judul Skripsi : **ANALISIS KINERJA AUGMENTASI TEKS
BACK TRANSLATION DAN SYNONIM
REPLACEMENT DALAM KLASIFIKASI
STATUS KESEHATAN MENTAL
MENGUNAKAN *DISTILLED-BERT*
(DISTILBERT)**

Nama Mahasiswa : **Lusiana Ferisca**
Nomor Pokok Mahasiswa : **2117031056**
Program Studi : **Matematika**
Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

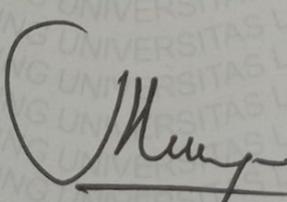
MENYETUJUI

1. Komisi Pembimbing


Dr. Dian Kurniasari, S.Si., M.Sc.
NIP. 196903051996032001


Favorisen R. Lumbanraja, M.Si., Ph.D.
NIP. 198301102008121002

2. Ketua Jurusan Matematika


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

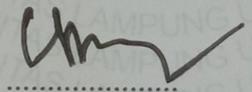
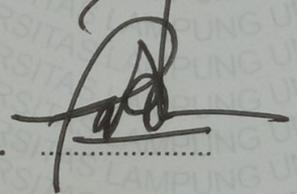
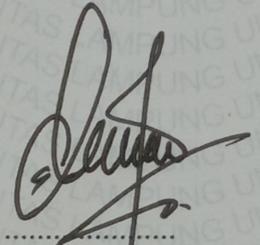
MENGESAHKAN

1. Tim Penguji

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

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

Penguji
Bukan Pembimbing : **Ir. Warsono, M.S., Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



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

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **12 Juni 2025**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : Lusiana Ferisca
Nomor Pokok Mahasiswa : 2117031056
Jurusan : Matematika
Judul Skripsi : Analisis Kinerja Augmentasi Teks *Back Translation* dan *Synonym Replacement* dalam Klasifikasi Status Kesehatan Mental Menggunakan *Distilled-BERT* (DistilBERT)

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, 12 Juni 2025

Penulis,



Lusiana Ferisca

RIWAYAT HIDUP

Penulis bernama lengkap Lusiana Ferisca, lahir di Pesawaran pada 28 Maret 2004. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Iskandar dan Ibu Maimunah.

Penulis menempuh pendidikan Sekolah Dasar (SD) di SD Negeri 4 Hanau Berak pada tahun 2008-2015, kemudian melanjutkan pendidikan ke Sekolah Menengah Pertama (SMP) Negeri 14 Pesawaran pada tahun 2015-2018, dan Sekolah Menengah Atas (SMA) Negeri 1 Padang Cermin 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 anggota Bidang Keilmuan Himpunan Mahasiswa Jurusan Matematika Periode 2022.

Sebagai bentuk pengaplikasian ilmu yang didapat selama perkuliahan, pada bulan Januari 2024 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 Pelindung Jaya, Kecamatan Gunung Pelindung, Kabupaten Lampung Timur, Provinsi Lampung.

KATA INSPIRASI

”Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap”

(Q.S. Al-Insyirah: 6-8)

”Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum, sebelum mereka mengubah keadaan diri mereka sendiri”

(Q.S. Ar-Rad: 11)

”Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu”

(Ali bin Abi Thalib)

”Di hadapan doa, kemustahilan kehilangan makna”

PERSEMBAHAN

Alhamdulillahillobbil' alamin, puji dan syukur kehadiran 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. Penulis persembahkan karya ini sebagai bentuk penghargaan dan rasa terima kasih kepada:

Ayah, Ibu, dan Adik

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

Alhamdulillah rabbil'alamin, puji syukur kehadiran Allah SWT. atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Kinerja Augmentasi Teks *Back Translation* dan *Synonym Replacement* dalam Klasifikasi Status Kesehatan Mental Menggunakan Distilled-BERT (DistilBERT)".

Terselesainya 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 Favorisen 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. Bapak Ir. Warsono, M.S., Ph.D. selaku Pembahas yang telah meluangkan waktu untuk memberikan kritik, saran, serta masukan yang membangun demi perbaikan skripsi ini.
4. Bapak Prof. Dr. La Zakaria, S.Si., M.Sc. selaku Pembimbing Akademik yang telah memberikan bimbingan dan arahnya 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. Kedua orang tua penulis, Bapak Iskandar dan Ibu Maimunah, serta adik-adik penulis, Nayla dan Faris, yang tidak pernah lelah memberikan doa, dukungan moral, serta materiil yang tidak ternilai harganya, sehingga penulis dapat menyelesaikan skripsi ini.
9. Rekan-rekan terbaik penulis selama menjalani perkuliahan Citra, Damai, Ida, Rohana, Irma, Dede, Windi, yang selalu mendengarkan keluh kesah penulis dan banyak membantu penulis selama perkuliahan maupun penyelesaian skripsi ini.
10. Teman-teman seperbimbingan, Andi, Anggy, Ariz, Dita, Maya, Mey, Nabila, Sherina, Adinda, Anastasia, Dina, Erwin, Fathan, Rhea, dan Yulina, yang telah kebersamai 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. 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, 12 Juni 2025

Lusiana Ferisca

DAFTAR ISI

	Halaman
DAFTAR ISI	xiii
DAFTAR TABEL	xvi
DAFTAR GAMBARxviii
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	6
II TINJAUAN PUSTAKA	7
2.1 Penelitian Terkait	7
2.1.1 Penelitian Pertama (Rahma & Suadaa, 2023)	9
2.1.2 Penelitian Kedua (Oktariansyah dkk., 2024)	10
2.1.3 Penelitian Ketiga (Kapusta dkk., 2024)	11
2.2 <i>Natural Language Processing</i> (NLP)	12
2.3 <i>Text Mining</i>	13
2.4 Ketidakseimbangan Data	14
2.5 Data Augmentasi	14
2.5.1 <i>Easy Data Augmentation</i> (EDA)	15
2.5.2 <i>Back Translation</i>	16
2.6 Representasi Kata (<i>Word Embedding</i>)	20
2.7 <i>Splitting Data</i>	23
2.8 Klasifikasi	24
2.9 <i>Deep Learning</i>	24
2.9.1 <i>Layers</i>	26
2.9.2 Fungsi Aktivasi	26
2.9.3 Fungsi Kerugian	30

2.9.4	Algoritma Optimasi	31
2.10	<i>Transformer</i>	32
2.10.1	<i>Attention</i>	33
2.10.2	<i>Scaled Dot-Product Attention</i>	33
2.10.3	<i>Multi-Head Attention</i>	34
2.10.4	<i>Position-Wise Feed-Forward Networks</i>	35
2.10.5	<i>Embeddings dan Softmax</i>	36
2.10.6	<i>Positional Encoding</i>	36
2.11	<i>Distilled-BERT</i>	37
2.12	Evaluasi Kinerja Model	41
2.13	Uji-t	46
III	METODE PENELITIAN	48
3.1	Tempat dan Waktu Penelitian	48
3.1.1	Tempat Penelitian	48
3.1.2	Waktu Penelitian	48
3.2	Data dan Alat	48
3.2.1	Data	48
3.2.2	Alat	50
3.3	Metode Penelitian	52
IV	HASIL DAN PEMBAHASAN	55
4.1	Input Data	55
4.2	<i>Exploratory Data Analysis (EDA)</i>	56
4.2.1	Analisis Struktur Data	56
4.2.2	Analisis Persebaran Data	57
4.2.3	Analisis Frekuensi Kata	58
4.3	<i>Pre-processing Data</i>	63
4.3.1	Seleksi Variabel	63
4.3.2	Penanganan <i>Missing Value</i> dan Data Duplikat	64
4.3.3	<i>Data Filtering dan Case Folding</i>	64
4.4	<i>Splitting Data</i>	66
4.5	Augmentasi Data	68
4.5.1	<i>Back Translation</i>	68
4.5.2	<i>Synonym Replacement</i>	72
4.6	DistilBERT <i>Embedding</i>	74
4.7	Pemodelan Klasifikasi	77

4.8	Pengujian Model Klasifikasi	78
4.8.1	Pengujian Pada Data Tanpa Augmentasi	79
4.8.2	Pengujian Data Dengan Augmentasi	93
4.9	Evaluasi Peningkatan Kinerja Model DistilBERT dengan Teknik Augmentasi Data	104
4.10	Uji Signifikansi Kinerja Model	105
4.11	<i>Benchmarking</i> dengan Penelitian Terdahulu	106
V	KESIMPULAN DAN SARAN	109
5.1	Kesimpulan	109
5.2	Saran	110
	DAFTAR PUSTAKA	111

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu Terkait Augmentasi Teks	7
2. <i>Token Embedding</i>	22
3. Sebaran Jumlah Data Berdasarkan Status Kesehatan Mental	49
4. Sampel Data Penelitian	50
5. Data Penelitian	55
6. Distribusi Panjang Kata	57
7. Data Setelah Proses Seleksi Variabel	64
8. Perbandingan Teks Sebelum dan Sesudah Dilakukan <i>Data Filtering</i> dan <i>Case Folding</i>	65
9. Sebaran Jumlah Data Setelah <i>Pre-processing</i>	65
10. Jumlah kelas Setelah Dilakukan <i>Splitting</i> Kumpulan Data	67
11. Contoh <i>Back Translation</i> Menggunakan Bahasa Indonesia	69
12. Contoh <i>Back Translation</i> Menggunakan Bahasa Prancis	69
13. Contoh <i>Back Translation</i> Menggunakan Bahasa Spanyol	70
14. Contoh <i>Back Translation</i> Menggunakan Bahasa Jerman	70
15. Contoh <i>Back Translation</i> Menggunakan Bahasa Belanda	71
16. Contoh <i>Back Translation</i> Menggunakan Bahasa Italia	71
17. Sebaran Jumlah Data Setelah Augmentasi <i>Back Translation</i>	72
18. Contoh Teks Hasil <i>Synonym Replacement</i>	73
19. Sebaran Jumlah Data Setelah Augmentasi <i>Synonym Replacement</i>	74
20. Contoh Proses DistilBERT <i>embedding</i>	75
21. Kombinasi <i>Hyperparameter</i> yang Akan Digunakan	78
22. Kombinasi <i>Hyperparameter</i> Terbaik untuk Ketiga Data	78
23. Kriteria pengujian Untuk Model Klasifikasi	79
24. Kinerja Model DistilBERT Tanpa Augmentasi	89
25. Kinerja Model DistilBERT dengan Augmentasi <i>Back Translation</i>	96

26. Perhitungan ROC-AUC Model DistilBERT dengan Augmentasi <i>Back Translation</i>	97
27. Kinerja Model DistilBERT dengan Augmentasi <i>Synonym Replacement</i>	101
28. Perhitungan ROC-AUC Model DistilBERT dengan Augmentasi <i>Synonym Replacement</i>	102
29. Teks yang Salah Klasifikasi	104
30. Perbandingan Kinerja Model DistilBERT Pada Setiap Data	105
31. Uji t-Berpasangan antara Akurasi Data Asli dan Data Augmentasi	106
32. <i>Benchmarking</i> Hasil Penelitian	107

DAFTAR GAMBAR

Gambar	Halaman
1. Ilustrasi <i>Word Embedding</i> (Wjijayanto dkk., 2021).	21
2. <i>Position Embedding</i> (Novac, 2023).	22
3. <i>BERT Embedding Layer</i> (Novac, 2023).	23
4. Perbedaan <i>Machine Learning</i> dan <i>Deep Learning</i> (Alzubaidi dkk., 2021).	25
5. Fungsi Aktivasi ReLU (Sharma dkk., 2017).	27
6. Fungsi Aktivasi GeLU (Lee, 2023).	28
7. Fungsi Aktivasi <i>Sigmoid</i> (Sharma dkk., 2017).	29
8. Fungsi Aktivasi <i>Softmax</i> (Purwitasari & Soleh, 2022).	30
9. Arsitektur Model <i>Transformer</i> (Vaswani dkk., 2017).	33
10. <i>Scaled Dot-Product Attention</i> (Vaswani dkk., 2017).	34
11. <i>Multi-Head Attention</i> (Vaswani dkk., 2017).	35
12. Represetasi <i>Input</i> BERT (Devlin dkk., 2018).	38
13. Prosedur <i>Pre-training</i> dan <i>Fine-tuning</i> BERT (Devlin dkk., 2018).	38
14. Arsitektur Model DistilBERT (Adel dkk., 2022).	39
15. Proses <i>Feature Extraction</i> Model DistilBERT (Adel dkk., 2022).	40
16. <i>Confusion Matrix Klasifikasi Biner</i> (Markoulidakis dkk., 2021).	41
17. <i>Confusion Matrix klasifikasi Multiclass</i> (Markoulidakis dkk., 2021).	42
18. Sebaran Jumlah Data Berdasarkan Status Kesehatan Mental	50
19. <i>Flowchart</i> Prosedur Penelitian.	53
20. Distribusi Data.	56
21. Visualisasi Persebaran Data.	58
22. <i>Wordcloud</i> Kata Pada Status normal.	59
23. <i>Wordcloud</i> Kata Pada Status <i>Depression</i>	60
24. <i>Wordcloud</i> Kata Pada Status <i>Anxiety</i>	60
25. <i>Wordcloud</i> Kata Pada Status <i>Bipolar</i>	61
26. <i>Wordcloud</i> Kata Pada Status <i>Stress</i>	62

27. <i>Wordcloud</i> Kata Pada Status <i>Personality Disorder</i>	62
28. Distribusi Data Setelah <i>Pre-processing</i>	65
29. <i>Splitting</i> Data Teks Status Kesehatan Mental.	66
30. Distribusi Data Sebelum dan Sesudah Augmentasi <i>Back Translation</i>	72
31. Distribusi Data Sebelum dan Sesudah Augmentasi <i>Synonym Replacement</i>	74
32. Grafik Akurasi dan <i>loss</i> Model DistilBERT Tanpa Augmentasi.	80
33. <i>Confusion Matrix</i> Model DistilBERT Tanpa Augmentasi.	80
34. Kurva ROC-AUC Model DistilBERT Tanpa Augmentasi.	91
35. Grafik Akurasi dan <i>loss</i> Model DistilBERT dengan Augmentasi <i>Back Translation</i>	94
36. <i>Confusion Matrix</i> Model DistilBERT dengan Augmentasi <i>Back Translation</i>	95
37. Kurva ROC-AUC Model DistilBERT dengan Augmentasi <i>Back Translation</i>	98
38. Grafik Akurasi dan <i>loss</i> Model DistilBERT dengan Augmentasi <i>Synonym Replacement</i>	99
39. <i>Confusion Matrix</i> Model DistilBERT dengan Augmentasi <i>Synonym Replacement</i>	100
40. Kurva ROC-AUC Model DistilBERT dengan Augmentasi <i>Synonym Replacement</i>	103
41. Histogram <i>Benchmarking</i> Hasil Penelitian.	108

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kesehatan mental merupakan suatu keadaan di mana seseorang tidak mengalami gejala-gejala gangguan mental. Menurut *American Psychological Association* (APA) kesehatan mental dapat diartikan sebagai keberhasilan dalam menyesuaikan diri atau tidak adanya psikopatologi dan sebagai kondisi di mana individu tidak mengalami gangguan dalam aspek psikologis, emosional, perilaku, dan sosial (Nur dkk., 2023). Gangguan mental adalah kondisi kesehatan yang memengaruhi pola pikir, perasaan, perilaku, suasana hati, atau kombinasi di antaranya. Gangguan ini dapat dipengaruhi oleh berbagai faktor di antaranya adalah faktor genetik, biologis, psikologis, dan lingkungan (Vitoasmara dkk., 2024).

Saat ini, jumlah orang yang mengalami gangguan kesehatan mental terus mengalami peningkatan dan menjadi suatu perhatian yang serius. Menurut Laporan Survei Kesehatan Indonesia (SKI) Tematik oleh Kementerian Kesehatan (2023), sekitar 450 juta orang di dunia menghadapi masalah gangguan mental, neurologis, serta penyalahgunaan obat, yang menyumbang sekitar 14% beban penyakit global, berdasarkan jumlah tersebut, sekitar 154 juta individu mengalami depresi. Gangguan mental di kalangan remaja juga cukup tinggi, dengan 1 dari 7 remaja berusia 10-19 tahun mengalami masalah psikologis. Depresi menjadi penyebab utama disabilitas pada remaja dan juga dapat memicu bunuh diri, yang merupakan penyebab kematian ke-4 pada remaja di seluruh dunia. Lebih dari 90% kasus bunuh diri dikaitkan dengan gangguan mental.

Seiring dengan peningkatan gangguan kesehatan mental yang dialami oleh masyarakat, di sisi lain makin banyak juga orang yang memanfaatkan media sosial untuk mengekspresikan perasaan dan pengalaman pribadi mereka. Melalui berbagai media sosial atau komunitas kesehatan sosial daring, orang-orang sering

kali memilih untuk menyampaikan masalah atau keluhan kesehatan mental mereka secara anonim. Selain sebagai ruang berbagi, media sosial sering digunakan untuk memperoleh informasi kesehatan yang berkaitan dengan gejala yang mereka rasakan (Kim dkk., 2020). Karena banyaknya individu yang menggunakan media sosial untuk mengekspresikan perasaannya, mengakibatkan unggahan terkait kesehatan mental di media sosial sangat bervariasi. Unggahan ini dapat dijadikan sumber yang berharga untuk memahami masalah kesehatan mental yang dialami oleh masyarakat.

Berdasarkan analisis dan pengelompokan unggahan di media sosial, kecenderungan spesifik dari gangguan kesehatan mental yang dialami oleh pengguna media sosial dapat diidentifikasi sehingga memungkinkan para profesional untuk mengenali pola umum dan jenis gangguan yang sering muncul di media sosial lalu dapat memberikan penanganan yang efisien. Melalui pemahaman yang lebih baik mengenai gangguan kesehatan mental diharapkan dapat dikembangkan kebijakan yang lebih efektif untuk mengurangi permasalahan kesehatan mental yang ada (Tanugeraha dkk., 2023).

Klasifikasi teks adalah proses mengelompokkan teks ke dalam dua kategori atau lebih. Maka untuk melakukannya, model perlu melalui proses pelatihan terlebih dahulu dengan memanfaatkan dataset yang sesuai untuk mendukung pembelajaran. Namun, pada kenyataannya data yang diperoleh terutama data yang bersumber dari media sosial sering kali tidak sempurna, sehingga memerlukan pemrosesan tambahan sebelum data dapat digunakan. Salah satu masalah yang sering ditemui dalam klasifikasi teks adalah ketidakseimbangan data, di mana jumlah data pada kelas tertentu jauh lebih rendah dibandingkan jumlah data pada kelas lainnya (Rahma & Suadaa, 2023). Dataset yang tidak seimbang dapat menyebabkan kesalahan dalam menentukan hasil klasifikasi di mana data dari kelas minoritas sering diklasifikasikan sebagai kelas mayoritas. Penerapan algoritma klasifikasi secara langsung tanpa memperhatikan distribusi kelas yang tidak seimbang dapat mengakibatkan hasil prediksi yang baik pada kelas mayoritas namun kurang baik pada kelas minoritas. Jika algoritma klasifikasi tetap diterapkan tanpa melakukan *resampling* terhadap dataset yang memiliki kelas *imbalance* maka akan mengalami penurunan kinerja dalam klasifikasinya (Gumelar dkk., 2023).

Pendekatan *oversampling* dapat digunakan untuk mengatasi masalah ketidakseimbangan data. Teknik *oversampling* yang umum digunakan adalah

random oversampling dan *synthetic minority oversampling technique* (SMOTE) tetapi keduanya tidak mempertimbangkan pola atau struktur asli dari distribusi data saat menambahkan sampel baru sehingga rentan menyebabkan *overfitting*. SMOTE sering digunakan untuk mengatasi ketidakseimbangan data pada klasifikasi teks. Pernyataan ini didukung oleh penelitian yang menunjukkan bahwa, penambahan jumlah data dengan menggunakan SMOTE lebih efektif dibandingkan *random oversampling* (Sanya & Suadaa, 2022). Akan tetapi SMOTE memiliki beberapa kekurangan, seperti menghasilkan sampel yang kurang informatif, menimbulkan *noisy* atau data yang tidak relevan dan meningkatkan *overlapping* karena *oversampling* dilakukan secara berlebihan (Soltanzadeh & Hashemzadeh, 2021). Oleh karena itu diperlukan alternatif lain untuk mengatasi masalah ketidakseimbangan data ini. Salah satu alternatif yang dapat digunakan untuk meningkatkan kualitas data sambil mempertahankan informasi yang ada pada teks adalah teknik augmentasi teks. Augmentasi teks adalah teknik yang digunakan untuk menghasilkan teks sintetis berdasarkan kumpulan data teks yang ada (Rahma & Suadaa, 2023).

Teknik augmentasi teks yang sering digunakan untuk menangani ketidakseimbangan data adalah *back translation* dan *synonym replacement*. Hal ini didukung oleh penelitian yang menunjukkan bahwa teknik augmentasi *back translation* dan *synonym replacement* efektif dalam meningkatkan kinerja model klasifikasi pada data informal (Oktariansyah dkk., 2024). *Back translation* digunakan dengan memanfaatkan model penerjemahan mesin di mana teks diterjemahkan ke bahasa perantara dan kemudian dikembalikan ke bahasa asli sehingga menghasilkan teks baru yang memiliki variasi dalam struktur kalimat dan pilihan kata (Beddiar dkk., 2021), sedangkan *synonym replacement* digunakan untuk menghasilkan variasi kata baru dengan mengganti suatu kata dengan kata sinonimnya.

Selain mengatasi ketidakseimbangan data, pemilihan metode klasifikasi yang tepat juga sangat berpengaruh pada kinerja sistem klasifikasi. Saat ini model berbasis *transformer* sangat populer digunakan, karena memiliki kinerja yang sangat baik untuk mengatasi masalah dalam tugas-tugas NLP (Awalina dkk., 2022). Salah satu model berbasis *transformer* yang populer adalah *Distilled-BERT* (DistilBERT). Model DistilBERT merupakan model yang dihasilkan dari distilasi pengetahuan pada model BERT sehingga memiliki 40% lebih sedikit parameter daripada BERT, berjalan 60% lebih cepat sambil mempertahankan 97% kinerja BERT

sehingga sangat cocok digunakan untuk aplikasi dengan sumber daya komputasi yang terbatas (Sanh dkk., 2019). Model DistilBERT ini sudah banyak diterapkan dalam klasifikasi teks pada data informal, dengan arsitektur yang lebih ringan dibandingkan model BERT, model DistilBERT mampu memproses data secara efisien sambil mempertahankan tingkat akurasi yang tinggi (Fajri dkk., 2022); (Basbeth & Fudholi, 2024)

Penelitian mengenai efektivitas augmentasi teks sudah pernah dilakukan dalam banyak penelitian sebelumnya di antaranya dilakukan oleh Rahma & Suadaa (2023) penelitian ini menerapkan teknik augmentasi teks *synonym replacement* dan *back translation* pada data *clickbait* dan *hate speech* menggunakan algoritma *Support Vector Machine* (SVM) dan IndoBERT. Penelitian ini menghasilkan peningkatan akurasi 5,19% pada teknik *back translation* dan 3,23% pada teknik *synonym replacement*.

Berikutnya penelitian kedua yang dilakukan oleh Oktariansyah dkk. (2024) membahas mengenai implementasi *synonym replacement* dan *back translation* untuk klasifikasi sentimen *tweet* terkait calon presiden menggunakan model klasifikasi IndoBERT. Hasil penelitian menunjukkan bahwa menggunakan *synonym replacement* akurasi meningkat sekitar 7%, sedangkan, dengan menggunakan *back translation* akurasi meningkat sekitar 3%.

Terakhir penelitian ke tiga yang dilakukan oleh Kapusta dkk. (2024) membahas mengenai dampak teknik augmentasi *synonym replacement*, *back translation*, dan *function words deletion* terhadap kinerja klasifikasi teks berita palsu menggunakan empat algoritma klasifikasi yaitu *Random Forest*, *Logistic Regression*, *BernouliNB*, dan *Support Vector Classifier* (SVC) Hasil penelitian menunjukkan bahwa teknik augmentasi *back translation*, dan *function words deletion* menghasilkan peningkatan akurasi tertinggi yaitu sebesar 3,71% menggunakan algoritma *BernouliNB*.

Berdasarkan beberapa penelitian di atas sudah banyak literatur yang membahas mengenai efektivitas teknik augmentasi data pada berbagai jenis teks. Namun, masih sedikit penelitian yang secara khusus mengevaluasi pengaruh teknik augmentasi pada teks gangguan kesehatan mental. Hal ini menjadi motivasi untuk melakukan penelitian mengenai pengaruh teknik augmentasi *synonym replacement*

menggunakan sumber sinonim dari *WordNet*, dan *back translation* menggunakan model MarianMT yaitu model terjemahan mesin berbasis *transformer* menggunakan algoritma klasifikasi DistilBERT. Penerapan teknik augmentasi ini, diharapkan dapat meningkatkan kinerja model dalam mengklasifikasikan berbagai jenis gangguan kesehatan mental, sehingga dapat mendukung pengembangan sistem deteksi dini dan intervensi berbasis kecerdasan buatan. Oleh karena itu penelitian ini akan membahas “Analisis Kinerja Augmentasi Teks *Back Translation* dan *Synonym Replacement* dalam Klasifikasi Status Kesehatan Mental Menggunakan *Distilled-BERT* (DistilBERT)”.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang di atas, adapun rumusan masalah dalam penelitian ini adalah:

1. Menilai pengaruh penambahan data hasil augmentasi teks terhadap peningkatan akurasi dan kinerja model dalam mengklasifikasikan jenis gangguan kesehatan mental.
2. Melakukan klasifikasi teks pada data status kesehatan mental untuk memprediksi jenis gangguan kesehatan mental yang dialami individu.
3. Menilai apakah terdapat perbedaan yang signifikan secara statistik terhadap kinerja model klasifikasi sebelum dan sesudah dilakukan augmentasi teks.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menerapkan augmentasi teks *back translation* dan *synonym replacement* untuk mengevaluasi pengaruh penambahan jumlah data terhadap kinerja model klasifikasi.
2. Membangun model berbasis DistilBERT untuk mengklasifikasikan jenis gangguan kesehatan mental menggunakan teks yang telah diaugmentasi.
3. Melakukan analisis statistik menggunakan uji-t untuk mengevaluasi signifikansi perbedaan kinerja model klasifikasi sebelum dan sesudah penerapan augmentasi teks.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Membantu mengidentifikasi teknik penambahan data yang efektif dapat meningkatkan kinerja model, serta memudahkan penambahan jumlah data tanpa perlu mengumpulkan data tambahan secara manual.
2. Mendukung pengembangan sistem deteksi dini jenis gangguan kesehatan mental melalui analisis teks.
3. Memberikan dasar analisis statistik untuk mengevaluasi perbedaan kinerja model sebelum dan sesudah dilakukan penambahan data.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terdahulu menjadi dasar penting dalam mengembangkan dan memperkuat landasan teori pada penelitian ini. Beberapa studi sebelumnya telah memanfaatkan teknik augmentasi data, seperti *back translation* dan *synonym replacement*, untuk mengatasi ketidakseimbangan data dan upaya untuk meningkatkan kinerja model klasifikasi. Beberapa penelitian sebelumnya menunjukkan bahwa penggunaan kedua teknik ini dapat membantu meningkatkan kinerja model dalam tugas klasifikasi teks, seperti deteksi berita palsu, dan analisis sentimen. Ringkasan beberapa penelitian terdahulu yang relevan dengan penelitian ini disajikan pada Tabel 1.

Tabel 1. Penelitian Terdahulu Terkait Augmentasi Teks

No	Penelitian	Data	Metode Augmentasi Teks	Model Klasifikasi	Akurasi Klasifikasi	Peningkatan Akurasi
1.	Penerapan <i>Text Augmentation</i> Untuk Mengatasi Data Yang Tidak Seimbang Pada Klasifikasi Teks Berbahsa Indonesia. Studi Kasus: Deteksi Judul <i>Clickbait</i> Dan Komentar	Data Teks Formal <i>Clickbait</i> : 3.316 <i>Non-clickbait</i> : 5.927 Data Teks Informal <i>Hate Speech</i> : 263 <i>Non-Hate Speech</i> : 1.307	<i>Synonym Replacement</i> <i>Back Translation</i>	IndoBERT <i>Support Vector Machine (SVM)</i>	Model SVM Data <i>Clickbait</i> : WA: 61.97% SR: 61.88% BT: 62.03% Data <i>Hate Speech</i> WA: 60,05% SR: 63,28% BT: 63,77%	<i>Back Translation</i> : SVM (<i>Clickbait</i>): +0.06% SVM (<i>Hate Speech</i>): +3.72% IndoBERT (<i>Clickbait</i>): +0,53% IndoBERT (<i>Hate Speech</i>): +5,19%

No	Penelitian	Data	Metode Augmentasi Teks	Model Klasifikasi	Akurasi Klasifikasi	Peningkatan Akurasi
	<i>Hate Speech</i> Pada Berita Online. (Rahma & Suadaa, 2023)				Model Indo-BERT Data <i>clickbait</i> WA: 83,51% SR: 83,23% BT: 84,04% Data <i>Hate Speech</i> WA: 61,78% SR: 64,50% BT: 66,97%	<i>Synonym Replacement</i> : SVM (<i>Hate Speech</i>): +3,23% IndoBERT (<i>Hate Speech</i>): +2,71%
2.	Klasifikasi Sentimen Untuk Mengetahui Kecenderungan Politik Pengguna X Pada calon Presiden Indonesia 2024 Menggunakan Metode IndoBERT (Oktariansyah dkk., 2024)	Data <i>Tweet</i> Pengguna X Positif Anies: 1901 Positif Prabowo: 888 Positif Ganjar: 1525 Negatif Anies: 264 Negatif Prabowo: 971 Negatif Ganjar: 93 Netral: 708	<i>Synonym Replacement</i> <i>Back Translation</i>	Indo-BERT	Tanpa Augmentasi: 75% Augmentasi Sinonim: 82% Augmentasi <i>Back Translation</i> : 78%	<i>Synonym Replacement</i> : +7% <i>Back Translation</i> : +3%
3.	<i>Text Data Augmentation Techniques for Word Embeddings in Fake News Classification</i> (Kapusta dkk., 2024)	Dataset WELFake : dengan label berita palsu (35.028) dan berita asli (37.106)	<i>Synonym Replacement</i> <i>Back Translation</i> <i>Reduction of Function Words</i>	<i>Random Forest</i> <i>Logistic Regression</i> <i>BernouliNB</i> <i>Support Vector Classifier</i> (SVC)	<i>Random Forest</i> Original: 85,88% SR: 85,28% FWD: 85,19% BT: 85,26% <i>Logistic Regression</i> Original: 81,67% SR: 82,15%	<i>Random Forest</i> : SR: - 0,70% FWD: -0,80% BT: -0,73% <i>Logistic Regression</i> : SR: 0,59% FWD: 1,72% BT: 1,55% <i>BernouliNB</i> : SR: -0,90%

No	Penelitian	Data	Metode Augmentasi Teks	Model Klasifikasi	Akurasi Klasifikasi	Peningkatan Akurasi
					FWD: 83,07% BT:82,94% <i>BernoulliNB</i> Original: 80,56% SR: 79,84% FWD: 81,37% BT:81,70% SVC Original: 82,90% SR: 85,75% FWD: 85,93% BT:85,96%	FWD: 1,01% BT: 1,41% SVC: SR: 3,43% FWD: 3,71% BT: 3,71%

Ringkasan dari penelitian-penelitian terkait pada Tabel 1 sebagai berikut:

2.1.1 Penelitian Pertama (Rahma & Suadaa, 2023)

Rahma & Suadaa (2023) melakukan penelitian dengan menggunakan dua jenis dataset, yaitu teks formal dan informal. Data yang digunakan dalam teks formal adalah data judul berita online berbahasa Indonesia dari CLICK-ID dengan dua label yaitu *clickbait* (3316 data) dan *non-clickbait* (5297 data). Sementara itu untuk teks informal data yang digunakan adalah komentar berita online berbahasa Indonesia dengan dua label yaitu *hate* (263 data) dan *no hate* (1307 data).

Karena adanya ketidakseimbangan jumlah data pada setiap kelas, maka dilakukan augmentasi teks menggunakan teknik *synonym replacement* dan *back translation*. *Synonym replacement* dilakukan dengan menggunakan sumber sinonim dari Tesaurus Tematis Bahasa Indonesia. Sebelum melakukan penggantian sinonim, setiap kata dalam kalimat harus diberi *tag* POS (*Part of Speech*) untuk menentukan jenis kata dalam konteks kalimat. POS *tagging* dilakukan dengan model *Conditional Random Fields* (CRF). Setelah kata diberi *tag*, penggantian sinonim

dapat dilakukan dengan syarat bahwa kata harus memiliki padanan sinonim di Tesaurus. Jika ada lebih dari satu sinonim yang ditemukan, sinonim dipilih secara acak. Sedangkan *back translation* dilakukan dengan menggunakan API *google translate*. Proses augmentasi dilakukan sebanyak lima kali menggunakan bahasa Inggris, Cina, Melayu, Jawa, dan Tagalog.

Setelah augmentasi teks selesai, proses klasifikasi dilakukan menggunakan model SVM dan *pre-trained transformer* IndoBERT-*large*. Klasifikasi teks yang dilakukan menerapkan 5-fold *cross validation* dan *grid search* untuk pencarian parameter terbaik. Hasil klasifikasi menunjukkan bahwa model IndoBERT secara konsisten lebih unggul dibandingkan model SVM pada kedua dataset. Pada dataset *clickbait*, model IndoBERT mencapai akurasi tertinggi sebesar 84,04% dengan teknik *back translation*, sementara model SVM hanya mencapai 62,03%. Pada dataset *hate speech*, IndoBERT juga lebih unggul, dengan peningkatan akurasi dari 61,78% menjadi 66,97% setelah diterapkan *back translation*, dibandingkan dengan model SVM yang hanya meningkat dari 60,05% menjadi 63,77%. Jika dilihat dari sisi teknik augmentasi, teknik *back translation* terbukti lebih efektif dibandingkan *synonym replacement*, terutama pada dataset *hate speech*, dengan peningkatan akurasi yang lebih besar pada kedua model. Teknik *back translation* meningkatkan akurasi IndoBERT hingga 5,19% dan SVM sebesar 3,72%. Dengan demikian, kombinasi IndoBERT dan *back translation* memberikan hasil terbaik dalam klasifikasi teks yang tidak seimbang.

2.1.2 Penelitian Kedua (Oktariansyah dkk., 2024)

Oktariansyah dkk. (2024) melakukan penelitian menggunakan dataset diambil dari media sosial X melalui proses *web crawling* menggunakan *Tweet Harvest* melalui API X selama 7 hari menjelang pemilihan Presiden Indonesia 2024. Data tersebut berisikan 7 label yaitu positif anies (1901 tweet), positif prabowo (888 tweet), positif ganjar (1525 tweet), negatif anies (264 tweet), negatif prabowo (971 tweet), negatif ganjar (93 tweet), netral (708 tweet). Karena adanya ketidakseimbangan jumlah data pada setiap kelas, maka dilakukan augmentasi teks menggunakan teknik *synonym replacement* dan *back translation*. *Synonym replacement* dilakukan menggunakan sumber sinonim dari *WordNet* untuk mengelompokkan kata-kata ke dalam sinonim kognitif, dan memungkinkan penggantian yang sesuai dengan konteks kalimat. Sedangkan *back translation* dilakukan dengan menggunakan API

Google Translate.

Setelah augmentasi teks selesai, proses klasifikasi dilakukan menggunakan model *pre-trained transformer* IndoBERT. Hasil klasifikasi menunjukkan bahwa model IndoBERT menunjukkan peningkatan akurasi setelah menerapkan kedua teknik augmentasi, namun hasilnya bervariasi. Tanpa augmentasi, akurasi dasar model adalah 75%. Penggunaan *synonym replacement* meningkatkan akurasi secara signifikan sebesar 7% menjadi 82%. Sementara itu, *back translation* memberikan peningkatan akurasi sebesar 3%, menghasilkan akurasi akhir 78%. Berdasarkan hasil ini dapat disimpulkan bahwa *synonym replacement* adalah teknik augmentasi yang lebih unggul dibandingkan *back translation* untuk meningkatkan kinerja model IndoBERT dalam klasifikasi tweet terkait calon presiden.

2.1.3 Penelitian Ketiga (Kapusta dkk., 2024)

Kapusta dkk. (2024) melakukan penelitian menggunakan data *WELFake*, yang terdiri dari 72.134 artikel yang berisi dua label yaitu berita palsu (35.028 artikel) dan berita asli (37.106 artikel). Augmentasi data dilakukan dengan tiga teknik yaitu *synonym replacement*, *back translation*, dan *reduction of function words*. *Synonym replacement* yang dilakukan menggunakan sumber sinonim dari *WordNet*, jika tidak ada sinonim ditemukan, maka akan dilakukan *lemmatization* pada kata tersebut menggunakan fungsi morfologi *WordNet* untuk menemukan bentuk lema kemudian cari kembali sinonimnya, setiap sinonim akan dilihat kesamaan semantiknya menggunakan *path similarity*. Sinonim yang diambil hanya sinonim yang memiliki nilai kesamaan semantik lebih dari 0,5. *Reduction of Function Words (FWD)* dilakukan dengan membagi kata dalam kalimat menjadi dua kategori, yaitu kata fungsi (kata yang tidak membawa makna utama) dan kata konten (kata yang memiliki makna utama). Teknik ini berfokus pada penghapusan acak kata-kata fungsi dari kalimat, sementara kata konten mungkin tetap dipertahankan. Sedangkan *back translation* dilakukan menggunakan *Easy Neural Machine Translation (EasyNMT)*.

Setelah augmentasi teks selesai, proses klasifikasi dilakukan menggunakan model klasifikasi *random forest*, *logistic regression*, *bernouliNB*, *Support Vector Classifier (SVC)*. Berdasarkan hasil klasifikasi, kinerja model berbeda-beda tergantung pada

teknik augmentasi yang digunakan. Model *random forest* menunjukkan akurasi tertinggi pada data asli dengan 85,88%, tetapi mengalami penurunan setelah diterapkan semua teknik augmentasi. Akurasi model setelah diterapkan teknik *Synonym Replacement* (SR) menurun 0,70% menjadi 85,28%, FWD menurun 0,80% menjadi 85,19%, dan *Back Translation* (BT) menurun 0,73% menjadi 85,26%. Ini menunjukkan bahwa augmentasi tidak meningkatkan kinerjanya.

Sebaliknya, model *logistic regression* mencatatkan peningkatan akurasi dari 81,67% menjadi 83,07% dengan teknik FWD, menunjukkan bahwa teknik augmentasi FWD dapat meningkatkan kinerja model ini. Model SVC mencatatkan peningkatan signifikan, dengan akurasi asli 82,90%, dan meningkat hingga 85,96% dengan *back translation* dan FWD yang memberikan peningkatan yang sama. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa model SVC dan teknik augmentasi FWD adalah yang paling efektif, dengan SVC yang menghasilkan akurasi tertinggi setelah augmentasi. Sebaliknya, teknik *Synonym replacement* kurang efektif pada sebagian besar model, terutama pada *random forest* dan *bernoulliNB*, yang mengalami penurunan akurasi.

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang ilmu komputer dan kecerdasan buatan yang berfokus pada pemahaman, pemrosesan, dan penghasilan bahasa manusia melalui komputer. Penerapan NLP memungkinkan komputer dapat berinteraksi dengan manusia menggunakan bahasa alami, baik dalam bentuk ucapan maupun teks. Tujuan utama dari NLP adalah untuk memahami, menganalisis, dan menghasilkan teks dalam bahasa manusia, serta mengembangkan sistem yang mampu berkomunikasi dengan manusia. NLP menghadapi beberapa tantangan dalam pemrosesan bahasa manusia, mengingat bahasa manusia memiliki struktur yang kompleks, termasuk tata bahasa, ambiguitas, variasi penggunaan tergantung konteks, serta ekspresi figuratif, dan masih banyak lagi (Rivaldi & Wismarini, 2024).

Dua teknik utama yang digunakan untuk memahami NLP adalah analisis sintaksis dan analisis semantik. Kedua teknik ini digunakan untuk memverifikasi struktur bahasa. Analisis sintaksis merujuk pada tata bahasa, sedangkan analisis semantik

mengacu pada penafsiran suatu kalimat. Analisis sintaksis merupakan teknik pengaturan pada suatu kalimat sehingga kalimat tersebut memiliki tata bahasa yang benar. Analisis sintaksis melibatkan penentuan stuktur di dalam kalimat seperti subjek, predikat, kata benda, kata kerja, kata ganti, dan lain sebagainya. Analisis semantik adalah teknik yang digunakan untuk mengidentifikasi dan memahami topik yang dimaksud dalam sebuah kalimat (Prasetyo dkk., 2021).

2.3 Text Mining

Text mining adalah salah satu teknologi yang dapat digunakan untuk mengolah data teks sehingga menghasilkan informasi tertentu. *Text mining* memungkinkan data yang tidak terstruktur dan data semi terstruktur dapat dikenali dengan baik. *Text mining* sangat dibutuhkan untuk membantu dalam pengelolaan dan ekstraksi informasi penting yang dapat diperoleh dalam sebuah dokumen (Cahyani & Arif, 2022). *Text preprocessing* adalah tahapan pertama yang dilakukan dalam *text mining*, *text preprocessing* dilakukan untuk mempersiapkan data teks sebelum digunakan pada proses lainnya. Pada tahap ini data teks akan diubah menjadi bentuk yang lebih baik sehingga menghasilkan informasi teks yang lebih berkualitas dan siap digunakan pada proses selanjutnya (Khairunnisa dkk., 2021).

Berikut adalah beberapa tahapan yang dilakukan dalam *text preprocessing* (Riyadhi, 2019):

1. *Case Folding*

Case folding merupakan proses mengubah semua huruf yang ada dalam teks menjadi huruf kecil (*lower case*).

2. *Tokenizing*

Tokenizing merupakan proses pemotongan kata dari kata-kata yang membentuknya menjadi suatu urutan list. Pada tahap ini juga beberapa karakter yang dianggap sebagai tanda baca dihilangkan.

3. *Filtering*

Filtering merupakan proses mengidentifikasi kata-kata penting yang telah dipisahkan dari sebuah kalimat, dan dilakukan pembersihan kata-kata yang dianggap tidak relevan.

4. *Lemmatization*

Lemmatization merupakan teknik yang digunakan untuk mengembalikan kata ke dalam bentuk dasarnya (Syawanodya, 2018).

2.4 Ketidakseimbangan Data

Ketidakseimbangan data atau sering disebut *imbalanced data*, adalah kondisi di mana data memiliki perbandingan jumlah yang tidak seimbang antara satu kelas dengan kelas yang lain, sehingga terdapat kelas yang memiliki lebih banyak sampel data dibandingkan dengan kelas lainnya. Prediksi terhadap data yang tidak seimbang sulit dilakukan, karena hasil klasifikasi cenderung mendeteksi kelas mayoritas dibandingkan kelas minoritas. Oleh karena itu, hasil dari klasifikasi akan menjadi bias (Hayawan & Ardhana, 2023). Misalnya, dalam sebuah dataset untuk mendeteksi gangguan kesehatan mental, jika mayoritas data berasal dari kelas normal dan hanya sedikit data yang mewakili gangguan, model dapat saja menghasilkan prediksi yang mayoritas mengklasifikasikan data sebagai kelas normal. Hal ini menyebabkan akurasi keseluruhan model terlihat tinggi, namun sebenarnya model gagal mendeteksi kasus gangguan yang sangat penting untuk dikenali. Permasalahan ketidakseimbangan data dapat diatasi dengan menggunakan pendekatan teknik *sampling*. Teknik *sampling* yang paling umum digunakan dalam mengatasi permasalahan ketidakseimbangan data adalah *oversampling*, *undersampling*, dan kombinasi keduanya (Hairani dkk., 2016). Teknik *oversampling* dilakukan dengan meningkatkan jumlah data kelas minoritas. Sedangkan teknik *undersampling* dilakukan dengan mengurangi jumlah data kelas mayoritas. Dibandingkan kedua teknik *sampling* yang lain, teknik *oversampling* lebih banyak digunakan dalam beberapa penelitian karena dapat mempertahankan informasi asli dari kumpulan data dan mudah digunakan (Ananda dkk., 2024).

2.5 Data Augmentasi

Data augmentasi merupakan teknik dalam pengolahan teks yang digunakan untuk memperluas dataset pelatihan dengan membuat variasi pada teks-teks yang ada cakupannya adalah untuk meningkatkan kuantitas dan keragaman data set pelatihan. Augmentasi teks menghasilkan data teks baru yang mirip dengan informasi teks asli (Bucos & Tucudean, 2023). Data augmentasi ini adalah salah satu teknik *oversampling* yang banyak digunakan untuk meningkatkan kinerja sistem klasifikasi (Azizah dkk., 2023). Data augmentasi untuk data teks terdapat tiga jenis yaitu data augmentasi berbasis *sampling*, *noise*, dan *paraphrase*. Augmentasi berbasis *sampling* merupakan proses pembuatan data teks baru yang menggunakan nilai distribusi data dan titik sampel pada data asli.

Kemudian augmentasi berbasis *noise* adalah menciptakan data baru dengan cara menambahkan kata atau simbol, menukar kata, menyisipkan dan mengganti kata pada data teks asli. Sedangkan augmentasi berbasis *paraphrase* adalah pembuatan data serupa dengan cara mengubah kalimat namun tetap memperhatikan isi konteks pada suatu kalimat. Teknik data augmentasi yang umum digunakan untuk mengatasi ketidakseimbangan data adalah *Easy Data Augmentation* (EDA) dan *back translation*.

2.5.1 Easy Data Augmentation (EDA)

Easy Data Augmentation (EDA) adalah salah satu teknik augmentasi data teks yang dapat digunakan untuk melakukan penambahan data dan meningkatkan kinerja pada tugas klasifikasi. Secara umum proses augmentasi EDA memiliki 4 teknik yaitu:

1. *Synonym Replacement*

Synonym replacement adalah metode yang digunakan untuk membuat data baru dengan memilih kata dalam kalimat secara acak dan kemudian menggantinya dengan sinonim dari kata yang dipilih (Fadilah & Priyanta, 2022).

2. *Random Insertion*

Random insertion atau penyisipan acak adalah proses yang digunakan untuk menyisipkan suatu kata yang ada dalam data secara acak pada kalimat (Azizah dkk., 2023).

3. *Random Swap*

Random swap atau pertukaran acak adalah proses yang digunakan untuk menukar urutan suatu kata secara acak pada suatu kalimat (Azizah dkk., 2023).

4. *Random Deletion*

Random deletion atau penghapusan acak adalah proses penambahan data dengan menghapus satu kata dalam sebuah kalimat untuk menghasilkan kalimat baru (Fadilah & Priyanta, 2022).

2.5.1.1. Algoritma Lesk

Tantangan utama dalam proses augmentasi data adalah memastikan kata-kata yang diubah tetap sesuai dengan konteks kalimat. Salah satu pendekatan yang dapat digunakan untuk mengidentifikasi makna kata dari suatu kalimat adalah algoritma lesk. *Algoritma lesk* adalah algoritma yang bekerja untuk mengetahui makna dari

kata yang memiliki makna ganda dalam suatu kalimat dengan memperhatikan kata-kata di sekitarnya.

Algoritma lesk bekerja dengan cara membandingkan definisi dari kata yang ambigu (polisemi) dengan definisi dari kata-kata di sekitarnya (Fazar & Widiastuti, 2017). Penentuan makna suatu kata polisemi dilakukan dengan mengumpulkan definisi atau glos dari kata tersebut yang terdapat dalam kamus. Glos-glos tersebut kemudian dibandingkan dengan konteks penggunaan kata dalam kalimat. Makna yang paling sesuai ditentukan berdasarkan tumpang tindih tertinggi antara glos dan konteks kalimat. Skor kesesuaian dihitung untuk setiap pasangan makna kata menggunakan rumus pada Persamaan (1) (Gujjar dkk., 2023):

$$\text{overlapscoreLesk}(S_1, S_2) = |\text{Gloss}(S_1) \cap \text{Gloss}(S_2)| \quad (1)$$

Arti masing-masing kata ditetapkan berdasarkan nilai maksimum yang diperoleh dari rumus di atas, di mana gloss (S_i) mewakili kumpulan kata dalam interpretasi tekstual dari arti yang relevan (Gujjar dkk., 2023).

2.5.2 Back Translation

Back translation adalah salah satu metode yang efektif untuk melakukan penambahan data teks. Metode ini sudah banyak digunakan dan terbukti dapat meningkatkan kinerja dalam berbagai tugas NLP salah satunya adalah klasifikasi (Bucos & Tucudean, 2023). *Back translation* dilakukan dengan cara menerjemahkan teks dari bahasa sumber ke bahasa target menggunakan sistem terjemahan mesin, kemudian teks tersebut diterjemahkan kembali dari bahasa target ke bahasa sumber (Oktariansyah dkk., 2024). *Back translation* menciptakan teks baru yang memiliki makna serupa dengan teks asli tetapi menggunakan frasa yang berbeda sehingga dapat menghasilkan teks baru dengan struktur kata berbeda. Proses menghasilkan data baru melalui *back translation* membuat teknik ini efektif dalam memperbesar ukuran dataset pelatihan. Teknik augmentasi ini dapat memperkuat kemampuan model dalam menggeneralisasi dan memahami pola yang tersembunyi dalam data, serta dapat meningkatkan kinerja model klasifikasi (Bucos & Tucudean, 2023).

2.5.2.1. MarianMT

Marian adalah mesin inti untuk layanan penerjemahan mesin saraf *microsoft translator*. Marian adalah kerangka kerja penerjemahan mesin neural yang mandiri, *open-source*, gratis, dan efisien. Kerangka kerja ini sepenuhnya dikembangkan dalam bahasa C++. Pelatihan marian dilakukan pada teks mentah, dengan pemrosesan data menggunakan *SentencePiece*. Marian digunakan oleh laboratorium *natural language preprocessing* Universitas Helsinki untuk melatih ratusan model penerjemahan dengan menggunakan data paralel yang diperoleh dari opus, dan model-model tersebut kemudian menjadi sumber terbuka dan dinamakan MarianMT (Soliman dkk., 2022). Model MarianMT merupakan model yang dirancang khusus untuk melakukan tugas yang berkaitan dengan *Neural Machine Translation* (NMT). Model MarianMT terdiri dari 6 lapisan *encoder-decoder transformer* dengan 8 *head attention* di setiap lapisan. Pencipta MarianMT, Helsinki-NLP, memiliki lebih dari 1000 model bahasa yang telah dilatih untuk model terjemahan MarianMT (Soliman dkk., 2022).

Model MarianMT dapat melakukan terjemahan pada berbagai tingkatan seperti tingkat dokumen, paragraf, dan kalimat. Karena MarianMT merupakan model terjemahan mesin berbasis *transformer* maka lapisan *encoder-decoder* merupakan arsitektur dominan pada MarianMT. Lapisan *encoder* digunakan untuk menghitung representasi kalimat sumber, dan lapisan *decoder* digunakan untuk menghasilkan kalimat target dari representasi tersebut. Kalimat sumber dinotasikan dengan $x = (x_1, \dots, x_S)$ dan kalimat target dinotasikan dengan $y = (y_1, \dots, y_T)$. Model MarianMT merupakan model auto-regresif sehingga distribusi peluang atas kalimat target $P(y|x)$ dapat didefinisikan menggunakan aturan rantai seperti pada Persamaan (2) (Tan dkk., 2020):

$$P(y|x) = \prod_{t=1}^T P(y_t|y_1, \dots, y_{t-1}, x) \quad (2)$$

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.

Pada proses awal terjemahan setiap token x_i akan direpresentasikan ke dalam ruang vektor berdimensi d_{model} melalui *embedding* $E(x_i) = \mathbf{e}_i \in \mathbb{R}^{d_{model}}$, dimana

$E(x_i)$ merupakan baris ke- x_i matriks *embedding* berukuran $|V| \times d_{model}$ dengan $|V|$ adalah ukuran kosakata. Karena MarianMT ini merupakan model berbasis *self-attention* dimana model hanya memperhatikan kesamaan konteks dalam kalimat tanpa memperhatikan urutan kata maka *positional encoding* ditambahkan untuk mengatasi masalah ini. *Positional encoding* ini adalah vektor berdimensi d_{model} yang merepresentasikan angka urutan kata dalam kalimat. Vektor ini ditambahkan ke *embedding* kata untuk memberi tahu model tentang posisi kata tersebut dalam urutan kalimat. *Positional encoding* memiliki dimensi yang sama dengan *embedding* sehingga keduanya dapat dijumlahkan. *Positional encoding* menggunakan fungsi *sinus* dan *cosinus* dan dapat diuraikan seperti pada Persamaan (3) dan (4) (Vaswani dkk., 2017):

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (3)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (4)$$

Keterangan:

pos = Posisi dalam urutan.

i = Index dimensi *embedding*.

Vektor-vektor *embedding* yang telah diproses dengan *positional encoding* dimasukkan ke dalam lapisan *encoder*. Lapisan *encoder* pada MarianMT menggunakan mekanisme *self-attention*, mekanisme *self-attention* menghitung relevansi setiap vektor nilai berdasarkan *query* dan *key* (Stahlberg, 2020). Secara formal, diberikan sekumpulan m vektor *query* $Q \in \mathbb{R}^{m \times d}$, sekumpulan n vektor *key* $K \in \mathbb{R}^{n \times d}$, dan vektor nilai (*value*) terkait $V \in \mathbb{R}^{n \times d}$. Komputasi mekanisme *self-attention* dalam lapisan *encoder* melibatkan dua langkah. Langkah pertama adalah menghitung relevansi antara *query* dan *key*, menggunakan Persamaan (5) (Tan dkk., 2020):

$$Attention\ score(i, j) = \frac{Q_i K_j^T}{\sqrt{d_k}} \quad (5)$$

Dengan d_k merupakan dimensi dari vektor *key*. Setelah menghitung relevansi untuk semua pasangan token, fungsi *softmax* digunakan untuk menormalkan *attention score* dan menghasilkan *attention weights* yang menunjukkan seberapa penting setiap token terhadap token lainnya. Fungsi *softmax* dapat diterapkan pada *attention*

score dengan menggunakan rumus pada Persamaan (6) (Tan dkk., 2020):

$$\text{softmax}(z_j) = \frac{\exp(z_j)}{\sum_{i=1}^{|V|} \exp(z_i)} \quad (6)$$

Keterangan:

$z_j = \text{Attention score}$ untuk kata ke- j .

$\sum_{i=1}^{|V|} \exp(z_i) = \text{Jumlah eksponensial dari seluruh attention score.}$

Langkah selanjutnya adalah menghitung vektor *output*. Untuk setiap vektor *query*, vektor *output* sebagai jumlah berbobot dari vektor-vektor *value*. Vektor *output* dapat dihitung dengan menggunakan rumus pada Persamaan (7) (Vaswani dkk., 2017):

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (7)$$

Setelah melewati lapisan *self-attention* pada *encoder*, diperoleh *output* berupa vektor konteks atau representasi kata yang sudah diproses. *Output* tersebut akan melalui lapisan *feed-forward network* yang bertugas untuk memperdalam representasi kata atau token yang sudah diproses oleh *self-attention*. Proses matematisnya melibatkan dua transformasi linier yang dipisahkan dengan fungsi aktivasi ReLU, sehingga memungkinkan model untuk mempelajari hubungan antar kata dengan lebih kompleks, dan kemudian meneruskan representasi tersebut untuk digunakan dalam lapisan berikutnya. Hasil akhir dari *encoder* adalah representasi vektor dari seluruh kalimat *input* yang menyimpan konteks seluruh kalimat.

Hasil akhir dari *encoder* akan dijadikan *input* pada lapisan *decoder*. *Decoder* bertugas untuk menghasilkan kalimat target dari representasi yang dihasilkan oleh *encoder*. Langkah pertama pada lapisan *decoder* dimulai dengan token `<start>` yang merupakan simbol khusus untuk menandakan awal kalimat yang akan diterjemahkan. Token ini merupakan *input* pertama untuk *decoder* dan akan diteruskan ke lapisan *masked self-attention*. Pada lapisan ini model memproses setiap token dalam *output* yang sedang dihasilkan dengan hanya memperhatikan kata-kata yang muncul sebelumnya, tanpa melihat kata yang akan datang. Hal ini dilakukan dengan menggunakan *masking* atau penutupan informasi kata-kata yang belum terproses. Perhitungan *attention* pada *masked self-attention* dimulai dengan cara yang sama seperti pada Persamaan (7), tetapi ketika menghitung matriks *score attention* QK^T , *masking* diterapkan. *Masking* ini digunakan untuk memberikan

nilai $-\infty$ pada *attention score* yang berasal dari token setelah token yang sedang diproses, sehingga hasil dari fungsi *softmax* bernilai 0. Karena nilainya 0 maka hubungan kata *query* dengan kata setelahnya menjadi tidak ada.

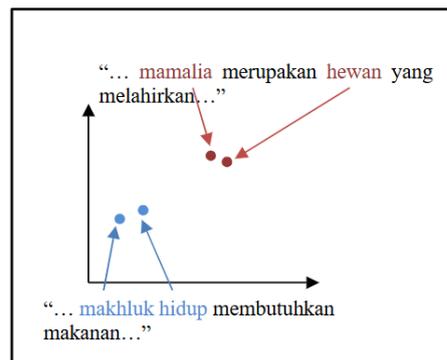
Hasil dari *masked self-attention* diproses dengan menggunakan *cross-attention* atau *encoder decoder attention* untuk menghubungkan informasi dari *encoder* dengan *decoder*. Pada lapisan ini, *decoder* memperhatikan representasi dari *encoder* untuk mencocokkan konteks *input* dan *output*. Lapisan *cross-attention* menggunakan representasi dari *encoder* sebagai *key* dan *value*, sementara *query* tetap berasal dari *output* sebelumnya di *decoder*. Perhitungan matematis pada lapisan ini sama dengan perhitungan *self-attention* pada lapisan *encoder*. Hasil dari *cross-attention* diteruskan ke *feed forward network* (FFN). *Feed forward network* memiliki fungsi yang sama seperti pada lapisan *encoder*. Setelah melalui FFN, hasilnya akan diproses pada *layer normalization* untuk meningkatkan stabilitas pelatihan.

Setelah proses *decoding* selesai dan setiap token dalam kalimat target mendapatkan representasi kontekstual melalui seluruh lapisan *decoder*. Kemudian representasi yang dihasilkan oleh *decoder* dipetakan ke dalam vektor z diruang kosakata $R^{|V|}$ dengan pemetaan linier. Kemudian, fungsi *softmax* digunakan untuk memastikan vektor *output* tersebut adalah peluang yang valid, secara matematis fungsi *softmax* dapat ditulis seperti pada Persamaan (6) sebelumnya, dengan z_i menyatakan komponen ke- i dalam z_j (Tan dkk., 2020). Proses pada lapisan ini merupakan tahap terakhir dalam proses pengolahan *output* di *decoder*. Model menghasilkan peluang untuk setiap token dalam kosakata, lalu kemudian token dengan peluang tertinggi akan dipilih, dan dimasukkan kembali ke dalam *decoder* untuk menghasilkan token berikutnya. Proses ini akan terus berulang hingga kalimat selesai diterjemahkan.

2.6 Representasi Kata (*Word Embedding*)

Word embedding sangat umum digunakan untuk menyelesaikan masalah NLP, karena memiliki dampak yang sangat penting terhadap kinerja dan akurasi model pembelajaran yang dibangun (Muhammad & Pulungan, 2022). *Word embedding* adalah vektor bernilai real yang merepresentasikan kata sehingga dapat mewakili konteks kata tersebut di dalam kalimat. Pada vektor ini kata-kata akan digambarkan dalam bentuk numerik dan setiap katanya dapat dipetakan ke dalam ruang vektor.

Kata-kata yang memiliki hubungan semantik atau sintaksis akan dipetakan saling berdekatan satu sama lain daripada kata yang tidak memiliki hubungan sama sekali seperti pada Gambar 1, keterkaitan antarkata tersebut bergantung kepada data teks atau korpus di mana kata tersebut berasal. *Word embedding* dapat dilatih secara langsung dari sebuah korpus atau kumpulan data berisikan teks dan tidak membutuhkan cara manual dalam pelabelan atau ekstraksi fitur (Wjijayanto dkk., 2021). Ilustrasi *word embedding* direpresentasikan seperti pada Gambar 1.



Gambar 1. Ilustrasi *Word Embedding* (Wjijayanto dkk., 2021).

Saat ini *word embedding* terus mengalami perkembangan dengan munculnya algoritma-algoritma baru yang memiliki model arsitektur yang berbeda dari sebelumnya (Muhammad & Pulungan, 2022). Awalnya *One Hot Encoding* digunakan untuk merepresentasikan kata-kata sebagai vektor *biner*, lalu dikembangkan pendekatan yang lebih canggih menggunakan *word embedding* berbasis jaringan saraf. *Word embedding* merepresentasikan kata-kata dalam vektor berdimensi tetap yang dihasilkan dari model seperti Word2Vec, FastText, atau Glove. Meskipun cukup efektif, pendekatan ini juga memiliki sedikit keterbatasan, yaitu karena setiap kata memiliki representasi yang tetap maka konteks kata dalam kalimat tidak dipertimbangkan. Keterbatasan ini mendorong pengembangan model *Bidirectional Encoder Representations from Transformers* (BERT) yang mampu menghasilkan representasi kata bersifat kontekstual, dengan *embedding* kata berubah secara dinamis berdasarkan kata-kata di sekitarnya. Dengan demikian, BERT mampu menangkap makna kata yang lebih akurat dalam berbagai konteks, sehingga dapat meningkatkan kinerja model dalam berbagai tugas pemrosesan bahasa alami (McCormick & Ryan, 2024).

BERT menggunakan tiga jenis lapisan *embedding* yang berbeda yaitu *token embedding*, *position embedding*, dan *token type embedding*. Tahap pertama

yang dilakukan dalam BERT *embedding* adalah melakukan tokenisasi menggunakan BERT *tokenizer*. *Tokenizer* digunakan untuk mengubah *input* dari suatu *string* menjadi daftar ID token integer, di mana setiap ID secara langsung merepresentasikan kata atau bagian dari kata dalam *string* asli (Novac, 2023).

Sebagai contoh, terdapat teks yang berisi kalimat "hello, world!". Sebelum proses *embedding* dilakukan, teks "hello, world!" akan dipecah terlebih dahulu menjadi token-token yang lebih kecil menjadi ["hello", ",", "world", "!"] lalu diberikan token [CLS] di awal *input* teks dan token [SEP] di akhir untuk menandakan akhir dari suatu teks. Selanjutnya, setiap token ini akan direpresentasikan oleh sebuah angka yang disajikan pada Tabel 2:

Tabel 2. *Token Embedding*

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

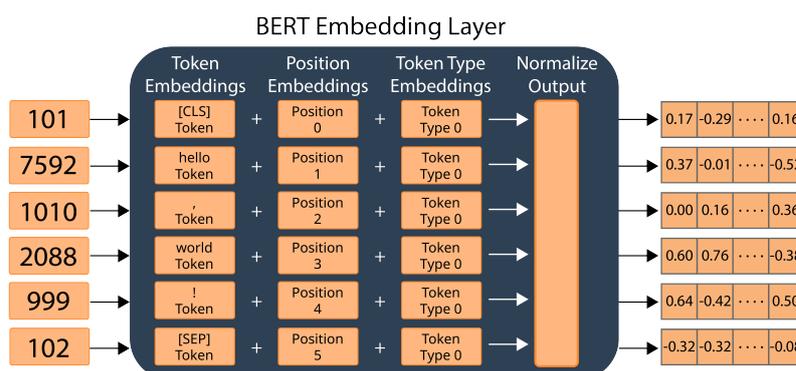
Setiap angka yang merepresentasikan token dalam teks di atas bukanlah angka yang dihasilkan secara acak melainkan model BERT memiliki *vocabulary* yang telah ditetapkan saat proses *pretraining* model. *Vocabulary* ini berisi daftar token beserta ID numeriknya yang digunakan untuk merepresentasikan kata atau bagian kata dalam teks. Saat ini, model BERT memiliki 30.522 token dalam *vocabulary*nya. Selain *token embedding* BERT juga menggunakan *poitional embedding* untuk merepresentasikan posisi token dalam urutan *input* seperti pada Gambar 2.



Gambar 2. *Position Embedding* (Novac, 2023).

Meskipun BERT memiliki 30.522 *token embedding* yang berbeda, *position embedding* yang tersedia hanya 512. Hal ini dikarenakan panjang batas maksimum urutan *input* yang dapat diproses oleh BERT hanya 512 token. Kemudian jenis terakhir dari *embedding* yang digunakan oleh BERT adalah *token type embedding*, *embedding* ini digunakan untuk membedakan antara dua kalimat dalam satu *input* teks. *Token type embedding* ini hanya terdiri dari dua jenis *embedding* yang berbeda, *token type embedding* 0 digunakan untuk merepresentasikan token dalam kalimat pertama, dan *token type embedding* 1 digunakan untuk merepresentasikan token

dalam kalimat kedua. Selanjutnya *embedding layer* menghitung *embedding* akhir untuk setiap token dengan cara menjumlahkan ketiga jenis *embedding* sebelumnya, lalu menerapkan normalisasi pada jumlah tersebut. Pada model BERT *base* dimensi akhir dari setiap token berukuran 768 sesuai dengan ukuran vektor tersembunyi (*hidden size*). Ilustrasi bagaimana proses BERT *embedding layer* menghitung *embedding* untuk teks “hello, world!” dapat di lihat pada Gambar 3.



Gambar 3. BERT Embedding Layer (Novac, 2023).

Konsep *embedding* yang diperkenalkan dalam BERT tidak hanya terbatas pada model ini saja, tetapi juga dapat menjadi inspirasi bagi pengembangan model lainnya. Salah satu contohnya adalah model DistilBERT. Pada model DistilBERT, proses *embedding* masih mempertahankan prinsip utama dari BERT, namun terdapat sedikit penyederhanaan yaitu DistilBERT hanya menggunakan *token embedding* dan *position embedding*, tanpa *token type embedding*. Meskipun demikian, hasil *embedding* yang dihasilkan tetap sesuai dengan berbagai tugas NLP.

2.7 Splitting Data

Splitting data adalah teknik yang digunakan untuk membagi dataset menjadi beberapa bagian data yang digunakan untuk menghilangkan dan mengurangi bias pada data *training*. *Splitting* data adalah salah satu dari beberapa aspek yang memengaruhi kinerja model klasifikasi pada algoritma pembelajaran mesin. Pendekatan klasik yang digunakan untuk membangun model prediktif yang baik adalah dengan membagi data menjadi tiga bagian utama yaitu *data training*, *data validation*, dan *data testing* (Muraina, 2022). *Data training* adalah bagian dari dataset yang digunakan untuk melatih model dalam melakukan prediksi atau

menjalankan fungsi dari sebuah algoritma sesuai dengan tujuan masing-masing. *Data testing* merupakan bagian dataset yang digunakan untuk melihat keakuratan atau kinerja model terhadap suatu data (Azmi dkk., 2023). Tingkat kesalahan merupakan rasio antara sampel yang diprediksi dengan benar berdasarkan jumlah total sampel dalam kumpulan data pengujian (Muraina, 2022).

2.8 Klasifikasi

Klasifikasi adalah salah satu teknik yang paling umum digunakan dalam pembelajaran mesin. Klasifikasi dapat diterapkan secara mandiri, seperti pada klasifikasi teks. Klasifikasi teks adalah proses mengelompokkan data ke dalam kategori atau kelas yang sudah ditentukan sebelumnya. Biasanya klasifikasi teks menggunakan data pelatihan berlabel untuk menghasilkan aturan yang nantinya akan digunakan untuk mengklasifikasikan data uji ke dalam kelompok yang telah ditentukan (Priyambodo & Prihati, 2020). Klasifikasi teks termasuk ke dalam metode *supervised learning* atau klasifikasi teks terawasi. Klasifikasi teks terawasi merupakan metode klasifikasi untuk mengelompokkan teks ke dalam kelompok yang sudah ada berdasarkan aturan yang telah ditentukan sebelumnya melalui proses pembelajaran (Pratiwi & Yaqin, 2022).

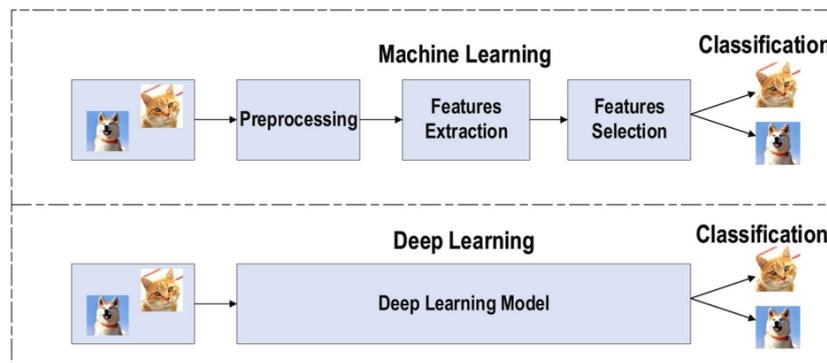
Klasifikasi teks dapat dibedakan berdasarkan jumlah label yang dimiliki, yaitu klasifikasi biner dan klasifikasi *multiclass*. Klasifikasi biner adalah kegiatan mengkategorikan data ke dalam dua label yang berbeda misalnya “0” atau “1”, “True” atau “False”, “Positif” atau “Negatif” dan lain sebagainya. Sedangkan klasifikasi *multiclass* adalah kegiatan mengkategorikan setiap data pada label yang berbeda, dimana banyaknya label lebih dari dua. Klasifikasi *multiclass* ini dapat diterapkan pada berbagai bidang, seperti untuk menganalisis suasana hati seseorang dan mengelompokkan lebih dari sekedar kategori positif atau negatif. (Tantika & Kudus, 2022).

2.9 Deep Learning

Deep learning merupakan bagian dari *machine learning* dan kecerdasan buatan yang terinspirasi oleh pola pemrosesan informasi pada otak manusia. Proses kerja

deep learning tidak memerlukan aturan yang dirancang oleh manusia melainkan menggunakan kumpulan data yang besar untuk memetakan *input* ke label tertentu. *Deep learning* dirancang menggunakan berbagai lapisan algoritma yang dikenal dengan jaringan syaraf tiruan, di mana setiap lapisan memberikan interpretasi yang berbeda terhadap data yang diinputkan (Alzubaidi dkk., 2021).

Pada *machine learning* konvensional, tugas klasifikasi melibatkan langkah-langkah berurutan, yaitu *pre-processing*, ekstraksi fitur, pemilihan fitur yang tepat, pembelajaran, dan klasifikasi. Selain itu, pemilihan fitur memiliki dampak besar terhadap kinerja *machine learning*. Pemilihan fitur yang bias dapat menyebabkan kesalahan dalam membedakan antar kelas. Sebaliknya, *deep learning* memiliki kemampuan untuk secara otomatis mempelajari himpunan fitur untuk berbagai tugas sehingga memungkinkan proses pembelajaran dan klasifikasi dilakukan dalam satu langkah. Perbedaan *machine learning* konvensional dan *deep learning* dapat dilihat pada Gambar 4.



Gambar 4. Perbedaan *Machine Learning* dan *Deep Learning* (Alzubaidi dkk., 2021).

Model *deep learning* tersusun dari berbagai komponen penting yang memungkinkan model belajar dari data, membuat prediksi dan optimasi kinerja model. Setiap komponen memiliki peran spesifik dalam menentukan efektivitas model dalam melakukan generalisasi data. Adapun komponen utama dalam model *deep learning* meliputi (Mienye & Swart, 2024):

2.9.1 Layers

Layers adalah elemen dasar pada *deep learning* yang menentukan bagaimana data diproses dan ditransformasikan saat melewati jaringan. Model *deep learning* umumnya terdiri dari lapisan *input*, satu atau lebih lapisan tersembunyi, dan lapisan *output*. Lapisan *input* digunakan untuk menerima data awal yang kemudian diteruskan ke lapisan tersembunyi di mana berbagai perhitungan dan transformasi dilakukan. Lapisan tersembunyi adalah tempat model belajar mengekstrak fitur dan pola yang relevan dari data. Kemudian, lapisan *output* menghasilkan prediksi akhir atau klasifikasi berdasarkan informasi yang telah diproses.

2.9.2 Fungsi Aktivasi

Fungsi aktivasi memiliki peran penting dalam *deep learning*, karena mereka memperkenalkan *non-linearitas* ke dalam jaringan, sehingga memungkinkan model untuk mempelajari pola-pola yang kompleks dan membuat prediksi yang canggih. Beberapa fungsi aktivasi yang biasa digunakan adalah sebagai berikut:

1. Fungsi Aktivasi *Rectified Liner Unit* (ReLU)

ReLU merupakan fungsi aktivasi non linear yang banyak digunakan dalam jaringan saraf. Keunggulan menggunakan fungsi ReLU adalah semua *neuron* tidak diaktifkan dalam waktu yang bersamaan. Hal ini berarti bahwa *neuron* akan dinonaktifkan hanya ketika *output* dari transformasi linear adalah nol. Secara matematis fungsi aktivasi ReLU dapat didefinisikan pada Persamaan (8) (Sharma dkk., 2017).

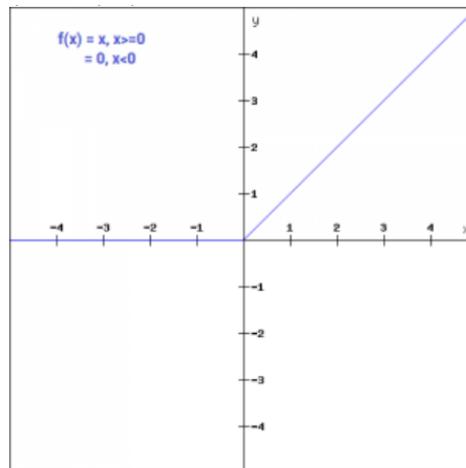
$$f(x) = \max(0, x) \tag{8}$$
$$f(x) = \begin{cases} 0, & \text{untuk } x \leq 0, \\ x, & \text{untuk } x > 0. \end{cases}$$

Keterangan:

x = Nilai data *input*

$f(x)$ = Hasil *output* fungsi ReLU

Berdasarkan Persamaan (8), grafik fungsi aktivasi ReLU dapat disajikan pada Gambar 5.



Gambar 5. Fungsi Aktivasi ReLU (Sharma dkk., 2017).

Fungsi aktivasi ReLU lebih efisien dibandingkan dengan fungsi-fungsi aktivasi yang lain karena, semua *neuron* tidak diaktifkan pada saat yang bersamaan, melainkan hanya sejumlah *neuron* yang diaktifkan pada satu waktu. Pada fungsi aktivasi ReLU jika terdapat elemen dengan nilai negatif maka nilai tersebut akan diubah menjadi 0. Oleh karena itu bobot dan bias tidak diperbarui selama *backpropagation* dalam pelatihan jaringan saraf.

2. Fungsi Aktivasi *Gaussian Error Linear Unit* (GeLU)

Fungsi aktivasi GeLU adalah alternatif halus dari fungsi aktivasi ReLU, tanpa menghilangkan keunggulannya. Fungsi aktivasi ReLU dinyatakan sebagai $ReLU(x) = \max(0, x)$, memberikan nonlinieritas pada jaringan, namun fungsi tersebut tidak dapat dibedakan (*non-differentiable*) pada $x = 0$, sehingga dapat menimbulkan komplikasi dalam optimasi berbasis gradien, seperti neuron mati atau dinamika pelatihan yang tidak stabil (Lee, 2023).

Sebagai solusi terhadap permasalahan tersebut, fungsi aktivasi GeLU dikembangkan sebagai pendekatan yang lebih halus dibandingkan ReLU. Fungsi ini dirancang untuk mempertahankan sifat non-linier yang penting dalam *deep learning* sekaligus menjamin *diferensiabilitas* di seluruh titik. Fungsi aktivasi GeLU sendiri didasarkan pada fungsi distribusi kumulatif gaussian (CDF) untuk mengatur aktivasi neuron secara halus. Berdasarkan kemampuannya dalam menghadirkan non-linieritas, *diferensiabilitas*, dan kehalusan secara bersamaan, fungsi ini telah digunakan secara luas dalam berbagai arsitektur

model mutakhir, seperti BERT, ViT, dan GPT. Secara matematis, fungsi aktivasi GeLU dirumuskan pada Persamaan (9) (Lee, 2023).

$$GELU(x) = 0.5x \left(1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right) \quad (9)$$

Keterangan:

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

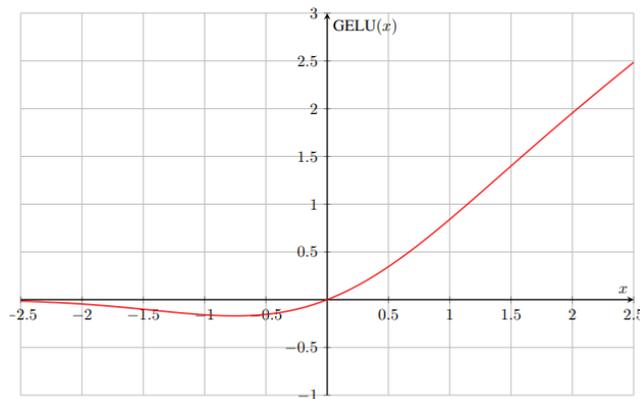
e^x = Nilai eksponensial Positif

e^{-x} = Nilai eksponensial Negatif

x = Nilai data *input*

$\sqrt{\frac{2}{\pi}}$ = Konstanta dari normalisasi distribusi normal standar

Berdasarkan Persamaan (9), grafik fungsi aktivasi GeLU dapat disajikan pada Gambar 6.



Gambar 6. Fungsi Aktivasi GeLU (Lee, 2023).

Berdasarkan Gambar 6, Fungsi aktivasi GeLU memiliki grafik yang lebih halus. Ketika nilai *input* sangat negatif, fungsi GeLU menghasilkan *output* yang mendekati nol. Sementara itu, untuk nilai *input* yang berada di sekitar nol, grafik *output* meningkat secara bertahap dengan transisi yang lebih halus. Hal ini menjadikan fungsi aktivasi GeLU lebih stabil dan lebih baik dalam pelatihan jaringan saraf mendalam.

3. Fungsi Aktivasi *Sigmoid*

Fungsi aktivasi *sigmoid* adalah fungsi aktivasi yang paling sering digunakan karena sifatnya yang non linear. Fungsi *sigmoid* mengubah suatu nilai *input* x ke dalam rentang 0 hingga 1. *Output* dari fungsi aktivasi *sigmoid* tidak simetris terhadap nol, karena nilainya selalu positif. Oleh karena itu, jika nilai *input*

sangat kecil atau mendekati $-\infty$, maka nilai *output* mendekati 0, sedangkan jika nilai *input* sangat besar atau mendekati ∞ , maka nilai *output* mendekati 1. Sehingga fungsi aktivasi *sigmoid* sangat berguna dalam klasifikasi biner, karena nilai *output* dapat diprediksi sebagai peluang. Fungsi aktivasi *sigmoid* dapat didefinisikan pada Persamaan (10) (Sharma dkk., 2017).

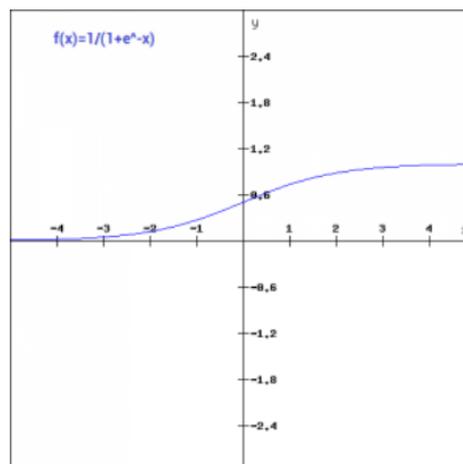
$$f(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

Keterangan:

x = Nilai data *input*

e = Bilangan Euler (2.71828)

Berdasarkan Persamaan (10), grafik fungsi aktivasi *sigmoid* dapat disajikan pada Gambar 7.



Gambar 7. Fungsi Aktivasi *Sigmoid* (Sharma dkk., 2017).

4. Fungsi Aktivasi *Softmax*

Fungsi aktivasi *softmax* adalah kombinasi dari beberapa fungsi *sigmoid*. Fungsi *sigmoid* menghasilkan nilai dalam rentang 0 hingga 1 yang dapat dijadikan sebagai peluang untuk titik data pada kelas tertentu. Fungsi *softmax* tidak seperti fungsi aktivasi *sigmoid* yang hanya digunakan untuk klasifikasi biner, fungsi *softmax* dapat digunakan untuk masalah klasifikasi *multiclass*. Fungsi ini menghasilkan peluang untuk setiap titik data terhadap semua kelas individual dengan *output* kelas yang memiliki nilai peluang paling tinggi.

Secara matematis fungsi aktivasi *softmax* dapat didefinisikan pada Persamaan (11) (Sharma dkk., 2017).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \quad \text{untuk } j = 1, \dots, K. \quad (11)$$

Keterangan:

$\sigma(z)_j$ = peluang prediksi lapisan *output*

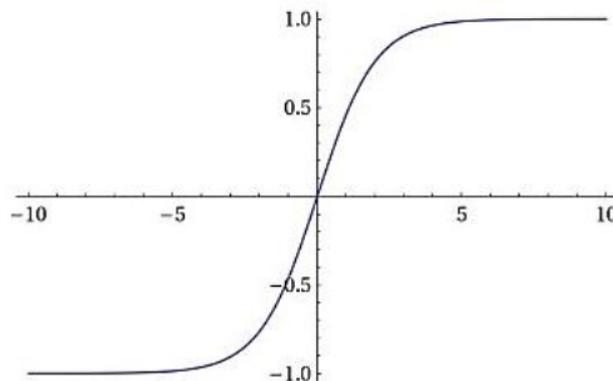
e^{z_j} = Sinyal yang dihasilkan lapisan *output*

e = Bilangan Euler (2.71828)

K = Jumlah kelas lapisan *output*

$\sum_{k=1}^K e^{z_k}$ = Jumlah eksponensial seluruh sinyal *output*

Berdasarkan Persamaan (11), grafik fungsi aktivasi *softmax* dapat disajikan pada Gambar 8.



Gambar 8. Fungsi Aktivasi *Softmax* (Purwitasari & Soleh, 2022).

2.9.3 Fungsi Kerugian

Pada *deep learning* fungsi kerugian digunakan untuk menguku perbedaan prediksi model dengan nilai sebenarnya. Fungsi kerugian memandu proses optimasi dengan menunjukkan bagaimana parameter model harus disesuaikan untuk meminimalkan *error* dan meningkatkan akurasi prediksi. Fungsi kerugian yang sering digunakan dan efektif untuk tugas klasifikasi adalah *cross-entropy loss*. *Cross-entropy* mengukur perbedaan antara distribusi label sebenarnya p dan distribusi label yang diprediksi q . *Cross-entropy loss* secara efektif menangkap ketidaksesuaian antara peluang kelas aktual dan yang diprediksi. Secara matematis, *Cross entropy loss*

dapat dituliskan pada Persamaan (12).

$$\text{Cross Entropy} = - \sum_i p(y_i) \log q(y_i) \quad (12)$$

Pada konteks klasifikasi *biner*, *cross-entropy loss* disederhanakan menjadi *binary cross-entropy loss*, yang digunakan ketika *output* adalah peluang yang mewakili dua kelas (0 atau 1). Sementara itu, pada permasalahan klasifikasi *multiclass*, *categorical cross-entropy loss* memperluas konsep ini ke beberapa kelas, di mana $p(y_i)$ adalah vektor *one-hot* yang mewakili kelas sebenarnya. dan $q(y_i)$ adalah distribusi peluang prediksi di semua kelas.

2.9.4 Algoritma Optimasi

Pada model *deep learning* pemilihan algoritma optimasi dapat digunakan untuk meningkatkan kinerja model klasifikasi. Pada saat proses pelatihan, model *deep learning* perlu melakukan perubahan atau memodifikasi atribut seperti bobot dan *learning rate* untuk membantu meminimumkan *loss function* dan meningkatkan akurasi. *Optimizer* mempelajari bagaimana mengatur parameter model agar model dapat memberikan prediksi yang lebih akurat dan sesuai dengan data yang digunakan selama proses pelatihan. Memilih bobot yang tepat untuk model adalah tugas yang cukup rumit karena model *deep learning* umumnya terdiri dari jutaan parameter (Wardani dkk., 2023). Salah satu algoritma optimasi yang sering digunakan dalam *deep learning* adalah *Adam optimizer*.

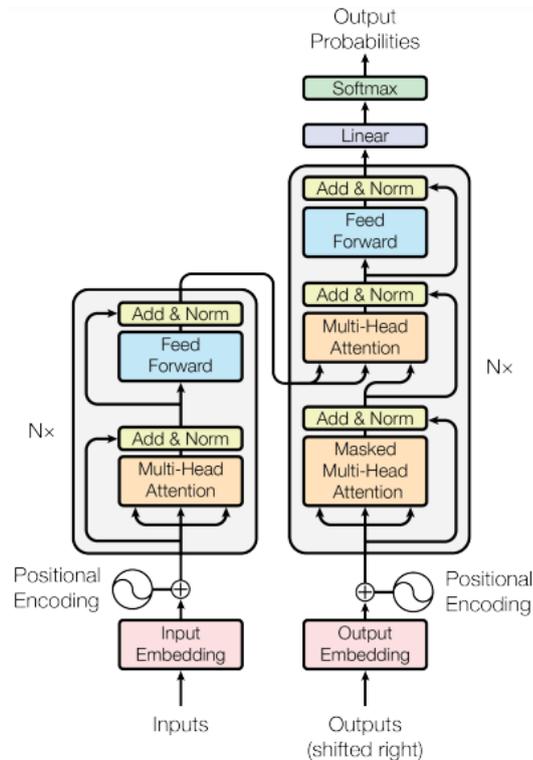
Adaptive Moment Estimation (Adam) *optimizer* adalah algoritma optimasi yang lebih canggih. Adam menggabungkan keunggulan dari algoritma optimasi *Adaptive Gradient Algorithm* (AdaGrad), yang menyesuaikan laju pembelajaran untuk setiap parameter berdasarkan gradien sebelumnya, dan *Root Mean Square Propagation* (RMSProp), yang menyesuaikan laju pembelajaran berdasarkan *moving average* dari gradien terbaru. Adam mempertahankan laju pembelajaran yang terpisah untuk setiap parameter dan mempunyai kemampuan untuk memperbarui bobot dan laju pembelajaran secara otomatis sehingga dapat mengoptimalkan kinerja model. Selain varian standar, Adam telah mengalami berbagai pengembangan untuk meningkatkan efisiensi optimasi. Salah satu variasi Adam *optimizer* yang paling populer adalah AdamW.

2.10 Transformer

Transformer merupakan model *deep learning* terkemuka yang telah banyak digunakan di berbagai bidang, seperti *Natural Language Preprocessing* (NLP), *Computer Vision* (CV), dan pemrosesan suara. *Transformer* dikembangkan untuk mengatasi permasalahan yang dihadapi model-model transduksi urutan (*sequence transduction*) tradisional seperti *Recurrent Neural Network* (RNN) khususnya *Long Short Term Memory* (LSTM) dan *Gated Recurrent Neural Networks* (GRU). Model-model transduksi tradisional ini sangat bergantung pada arsitektur *encoder-decoder* dengan mekanisme komputasi berurutan. Model ini menghitung urutan berdasarkan posisi simbol dalam urutan *input* dan *output*. Karena perhitungan harus dilakukan secara berurutan maka model mengalami keterbatasan paralelisasi. Pada urutan yang panjang penggunaan memori menjadi terbatas dan proses *batching* menjadi kurang efisien. Oleh karena itu dikembangkanlah *transformer* sebuah arsitektur model yang menggunakan mekanisme *self-attention* atau *intra attention* yang digunakan untuk menghubungkan berbagai posisi dalam satu urutan untuk menghasilkan representasi dari urutan tersebut. *Transformer* adalah model transduksi urutan pertama yang sepenuhnya menggunakan *self-attention* tanpa menggunakan RNN dan CNN yang berurutan sehingga memungkinkan *transformer* melakukan paralelisasi dengan lebih baik (Vaswani dkk., 2017).

Transformer memiliki struktur *encoder-decoder*. *Encoder* berfungsi untuk memetakan urutan *input* yang terdiri dari representasi simbol (x_1, \dots, x_n) ke dalam urutan representasi kontinu $z = (z_1, \dots, z_n)$. Setelah memperoleh z kemudian *decoder* menghasilkan urutan *output* (y_1, \dots, y_n) dari simbol satu per satu. Pada setiap langkah, model ini bersifat *auto-regresif*, yang berarti simbol yang dihasilkan sebelumnya digunakan sebagai *input* tambahan untuk menghasilkan simbol berikutnya. Arsitektur *transformer* ini memiliki 6 lapisan identik dan setiap lapisan memiliki dua sub lapisan yaitu *multi-head self-attention* dan jaringan *feed-forward*. Kemudian *decoder* juga memiliki 6 lapisan identik dengan tiga sub lapisan di mana dua sub lapisan sama dengan *encoder*, ditambah satu sub lapisan yang melakukan *multi-head attention* pada *output* dari tumpukan *decoder*. Sub lapisan *self-attention* pada *decoder* juga diubah untuk mencegah token memperhatikan token setelahnya, sehingga hasil prediksi menjadi lebih akurat dan sesuai dengan konteks kata. Meskipun arsitektur *transformer* terdiri dari lapisan *encoder-decoder*, tetapi model berbasis *transformer* seperti DistilBERT

hanya menggunakan lapisan *encoder* pada strukturnya. Arsitektur *transformer* dapat diilustrasikan pada Gambar 9 (Vaswani dkk., 2017).



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

2.10.1 Attention

Fungsi *attention* adalah proses yang memetakan sebuah *query* dengan sekumpulan pasangan *key-value* sehingga menghasilkan sebuah *output*, di mana *query*, *key*, nilai, dan *output* dari semuanya berupa vektor. *Output* dihitung sebagai total dari semua nilai yang diberi bobot. Bobot untuk setiap nilai ditentukan berdasarkan seberapa cocok *query* dengan *key* yang terkait (Vaswani dkk., 2017).

2.10.2 Scaled Dot-Product Attention

Scaled dot-product attention terdiri dari *query* dan *key* dengan dimensi d_k , serta *value* dengan dimensi d_v . Pada metode ini, *dot product* dihitung antara setiap *query* dan semua *key*, kemudian hasilnya dibagi dengan $\sqrt{d_k}$, selanjutnya diterapkan

fungsi *softmax* untuk menghasilkan bobot pada *value*. Perhitungan fungsi *attention* dilakukan secara bersamaan pada satu set *query* yang disusun dalam bentuk matriks Q , sementara *key* dan *value* disusun dalam bentuk matriks K dan V . Terdapat dua pendekatan umum dalam fungsi *attention* yaitu *additive attention* dan *dot-product (multiplicative) attention*, kedua pendekatan ini memiliki kompleksitas teoretis serupa, namun *scaled dot-product attention* lebih cepat dan efisien karena optimalisasi perkalian matriks (Vaswani dkk., 2017). Perhitungan *scaled dot-product attention* dapat diformulasikan pada Persamaan (13) dan ilustrasi operasi *scaled dot-product attention* dapat dilihat pada Gambar 10.

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

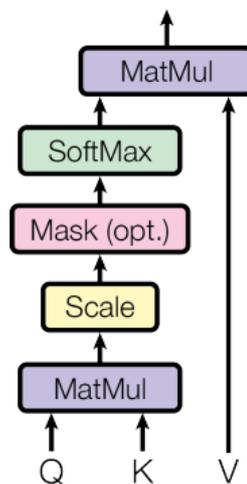
Keterangan:

Q = *Query* representasi *input* untuk mencari hubungan antar token

K = *Key* representasi *input* untuk mencocokkan *query*

V = *Value* representasi *input* yang akan dihitung atensinya

$\sqrt{d_k}$ = Dimensi dari *key*



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

2.10.3 Multi-Head Attention

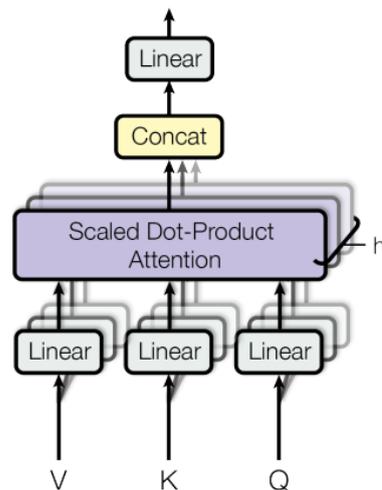
Multi-head attention memproyeksikan *query*, *key*, dan *value* secara *linear* sebanyak h kali dengan proyeksi *linear* yang berbeda yang telah dipelajari ke dimensi d_k , d_k

dan d_v . Pada setiap *query*, *key*, dan *value* yang telah di proyeksikan ini, diterapkan fungsi *attention* secara paralel sehingga menghasilkan *output* berdimensi d_v . Nilai-nilai ini lalu digabungkan dan diproyeksikan kembali sehingga menghasilkan nilai akhir seperti pada Gambar 11. *Multi-head attention* ini memungkinkan model untuk memperhatikan informasi secara bersamaan dari berbagai subruang representasi di posisi yang berbeda dengan satu *head attention* (Vaswani dkk., 2017). *Multi-head attention* dapat di interpretasikan melalui Persamaan (14) dan (15).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (14)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (15)$$

dimana W_i^Q, W_i^K, W_i^V adalah matriks proyeksi untuk setiap *head* dan W^O adalah matriks proyeksi akhir.



Gambar 11. *Multi-Head Attention* (Vaswani dkk., 2017).

2.10.4 Position-Wise Feed-Forward Networks

Pada arsitektur *transformer* masing-masing lapisan *encoder* dan *decoder* berisi jaringan *feed-forward* yang *fully connected*. Jaringan ini diterapkan ke setiap posisi secara terpisah dan identik. Struktur jaringan *feed-forward* terdiri dari dua transformasi linear dengan fungsi aktivasi ReLU di antaranya. Meskipun transformasi linear sama di setiap posisi, tetapi parameter yang digunakan berbeda dari lapisan ke lapisan (Vaswani dkk., 2017). *Feed-forward* dapat diuraikan seperti

pada Persamaan (16).

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

Keterangan:

W = Matriks Bobot

b = Basis Vektor

2.10.5 *Embeddings dan Softmax*

Transformer menggunakan *embedding* yang dipelajari untuk mengkonversi token *input* dan token *output* menjadi vektor dengan dimensi d_{model} . Transformasi linear dan fungsi *softmax* diterapkan untuk mengubah *output* dari *decoder* menjadi peluang token berikutnya yang diprediksi. Model ini menggunakan matriks bobot yang sama antara dua lapisan *embedding* dan transformasi linear *pra-softmax*. Pada lapisan *embedding*, bobot-bobot dikalikan dengan $\sqrt{d_{model}}$ agar memungkinkan hasil representasi yang lebih stabil dalam proses pembelajaran (Vaswani dkk., 2017).

2.10.6 *Positional Encoding*

Transformer tidak menggunakan *reccurent* dan konvolusi, agar model dapat memeperhatikan urutan maka harus ditambahkan informasi mengenai posisi relatif atau absolut token dalam urutan menggunakan *positional encoding*. *Positional encoding* ditambahkan ke *embedding input* pada dasar setiap tumpukan *encoder* dan *decoder*. *Positional encoding* memiliki dimensi yang sama dengan *embedding* sehingga keduanya dapat dijumlahkan. *Positional encoding* menggunakan fungsi *sinus* dan *cosinus* dari frekuensi yang berbeda dan dapat diuraikan seperti pada Persamaan (17) dan (18) (Vaswani dkk., 2017).

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

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

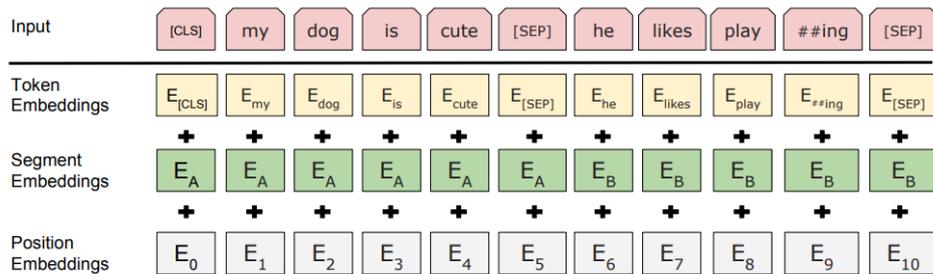
Pada Persamaan (17) dan (18), pos adalah posisi dalam urutan dan i adalah dimensi. Pada setiap dimensi pengkodean posisi berhubungan dengan *sinusoid*. Panjang gelombang *sinusoid* membentuk suatu progresi geometri dari 2π hingga $10000 - 2\pi$. Fungsi *sinusoid* dipilih karena memungkinkan model melakukan ekstrapolasi ke urutan panjang yang lebih panjang daripada saat proses pelatihan (Vaswani dkk., 2017).

2.11 *Distilled-BERT*

Model *Distilled-BERT* merupakan model berbasis *transformer* yang dikembangkan oleh Sanh dkk pada tahun 2019. Model ini dihasilkan dari distilasi pengetahuan pada model BERT. Model BERT adalah model bahasa yang dirancang untuk melakukan pra-pelatihan representasi dua arah secara mendalam pada teks yang tak berlabel dengan mempertimbangkan konteks dari kiri maupun kanan pada semua lapisan. Model BERT yang telah dilatih sebelumnya dapat disesuaikan ulang untuk melakukan berbagai tugas seperti klasifikasi dengan hanya menambahkan satu lapisan *output* tanpa melakukan modifikasi arsitektur yang signifikan (Devlin dkk., 2018). Arsitektur BERT hampir sama dengan *transformer*, namun BERT hanya menggunakan lapisan *encoder* saja, sehingga model BERT bisa diartikan sebagai tumpukan dari lapisan *encoder*. Model BERT dibedakan menjadi dua ukuran yaitu *BERT-Base* dan *BERT-Large*. Keduanya memiliki perbedaan pada jumlah lapisan *encoder*, ukuran tersembunyi, *multi head self-attention* dan jumlah parameter. Model *BERT-Base* terdiri dari 12 lapisan *encoder*, 768 ukuran tersembunyi, 12 *multi head self-attention*, dan 110 parameter. Sedangkan model *BERT-large* terdiri dari 24 lapisan *encoder*, 1024 ukuran tersembunyi, 16 *multi head self-attention*, dan 340 parameter (Devlin dkk., 2018).

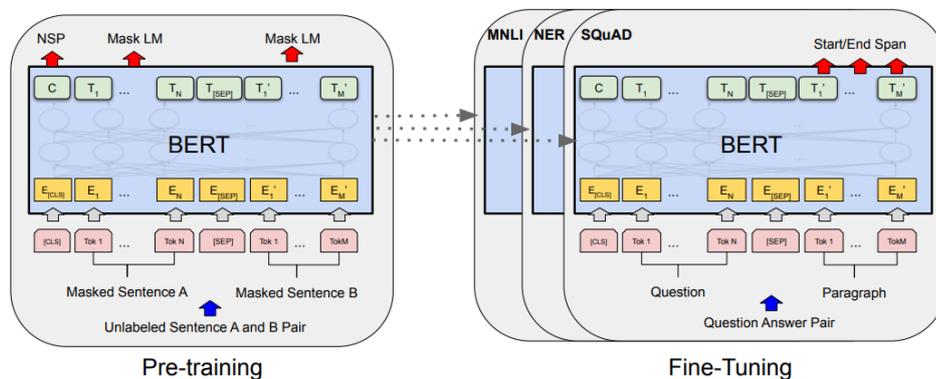
Model BERT menggunakan *embedding WordPiece* dengan 30.000 kosakata token. Hal ini memungkinkan BERT membagi kata menjadi bagian-bagian kecil untuk menangani kata yang jarang dikenali. Pada model BERT Setiap urutan kata selalu diawali dengan token khusus klasifikasi [CLS] yang memiliki fungsi sebagai representasi keseluruhan teks untuk tugas klasifikasi. Jika terdapat dua kalimat yang digabungkan menjadi satu urutan, BERT menggunakan dua cara untuk membedakannya yaitu menempatkan token khusus [SEP] di akhir kalimat, atau menambahkan *embedding* yang telah dipelajari ke setiap token untuk membedakan kalimat. Pada setiap token, representasi *input* dibangun dengan menjumlahkan

token *embedding*, *segment embedding*, dan *position embedding* (Devlin dkk., 2018). Represetasi *input* BERT dapat digambarkan seperti pada Gambar 12.



Gambar 12. Represetasi *Input* BERT (Devlin dkk., 2018).

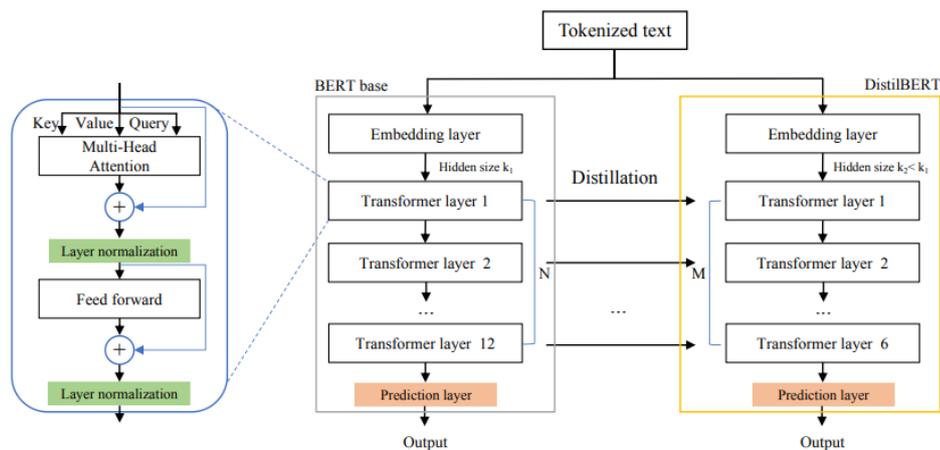
Proses pembelajaran BERT terdiri dari dua tahap, yaitu *pre-training* dan *fine-tuning*. Tahap *pre-training* dilakukan dengan melatih model BERT pada kumpulan besar data teks tidak berlabel untuk mempelajari pola dan hubungan dalam bahasa sehingga memungkinkan BERT memahami hubungan kontekstual antarkata dalam kalimat. Selanjutnya untuk tahap *fine-tuning* dilakukan dengan melatih kembali model BERT hasil *pre-training* dengan data berlabel yang spesifik untuk tugas tertentu. Proses *fine-tuning* pada model BERT cukup sederhana karena mekanisme *self attention* dalam arsitektur *transformer* memungkinkan model BERT menangani berbagai jenis tugas baik itu klasifikasi dan lain sebagainya. Pada setiap tugas, *input* dan *output* spesifik dari tugas tersebut cukup dimasukkan ke dalam model BERT kemudian dilakukan *fine-tuning* seluruh parameter secara *end-to-end* (Devlin dkk., 2018). Prosedur *pre-training* dan *fine-tuning* model BERT dapat digambarkan seperti pada Gambar 13.



Gambar 13. Prosedur *Pre-training* dan *Fine-tuning* BERT (Devlin dkk., 2018).

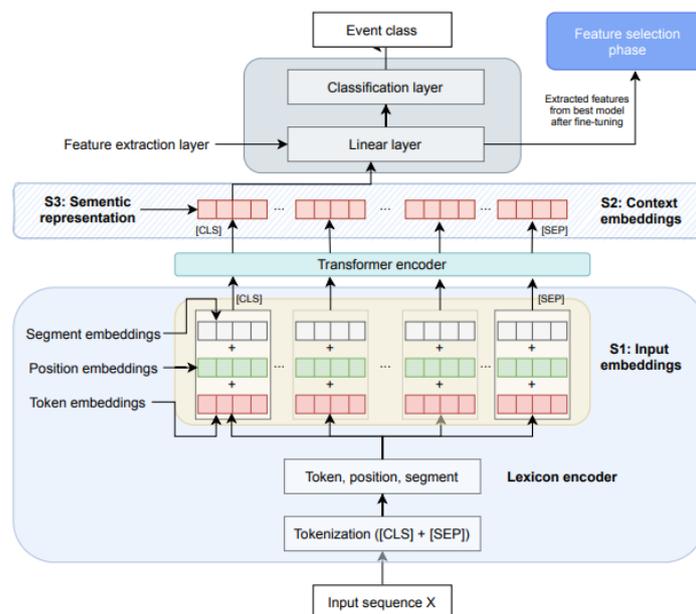
Meskipun model BERT sangat baik dalam mengerjakan berbagai tugas *Natural Language Preprocessing* (NLP), menerapkannya pada perangkat dengan keterbatasan daya komputasi seperti perangkat *on-the-edge* masih menjadi tantangan. Oleh karena itu, para peneliti dari *Hugging Face* pada tahun 2019 mengusulkan metode dengan menggunakan model bahasa yang jauh lebih kecil dan telah dilatih sebelumnya dengan penyulingan pengetahuan, sehingga lebih ringan, lebih cepat dan membutuhkan anggaran pelatihan komputasi yang lebih kecil. Model ini disebut dengan *Distilled-BERT*. Model DistilBERT dihasilkan dari distilasi pengetahuan pada model BERT dan bertujuan untuk membuat model bahasa yang lebih kecil dan efisien tetapi tetap mampu mencapai kinerja yang sebanding dengan model besar pada berbagai tugas NLP (Sanh dkk., 2019).

Model DistilBERT memiliki arsitektur umum yang sama dengan model BERT, tetapi terdapat beberapa modifikasi untuk mengurangi ukuran dan kompleksitasnya yaitu dengan menghapus *token-type embedding*, *pooler*, mengurangi jumlah lapisan *encoder* menjadi 6 lapisan, serta mengurangi jumlah parameter yang dilatih menjadi 66 juta parameter. Model DistilBERT juga dilatih dengan menggunakan *batch* yang sangat besar, menerapkan *dynamic masking* dan menghapus tujuan dari *next sentence prediction*. Meskipun model DistilBERT merupakan model yang lebih ringan daripada model BERT. Model ini tetap dilatih pada korpus yang sama dengan model BERT sehingga tetap mempertahankan kualitas dan kemampuan model dalam memahami bahasa (Sanh dkk., 2019). Arsitektur model DistilBERT dapat diilustrasikan pada Gambar 14.



Gambar 14. Arsitektur Model DistilBERT (Adel dkk., 2022).

Arsitektur model ekstraksi fitur berbasis DistilBERT bekerja dengan mengonversi *input* berupa urutan kata menjadi vektor-vektor *embedding*. Pada langkah awal model DistilBERT menerima *input* dalam bentuk teks yang telah diubah menjadi token. Setiap urutan kata selalu diawali dengan token khusus [CLS] sedangkan token [SEP] ditempatkan di akhir kalimat. Setiap kata dalam urutan dipetakan ke dalam vektor *embedding* yang merepresentasikan informasi spesifik dari masing-masing token. Model DistilBERT menggunakan *transformer encoder* untuk mempelajari makna kontekstual untuk setiap kata. *Transformer encoder* dilengkapi dengan mekanisme *self-attention* yang memungkinkan model mempertimbangkan hubungan antarkata dalam kalimat secara keseluruhan. Mekanisme *self-attention* menghasilkan *contextual embedding* untuk setiap kata kemudian *embedding* tersebut digabungkan menjadi satu vektor untuk merepresentasikan informasi semantik dari keseluruhan teks. Kemudian vektor semantik yang dihasilkan dimasukkan ke dalam lapisan *fully connected* sehingga menghasilkan vektor akhir berukuran d (jumlah *neuron* dalam lapisan), vektor ini merupakan representasi final dari *input* teks, selanjutnya lapisan klasifikasi ditempatkan pada akhir model ekstraktor fitur untuk menyempurnakan DistilBERT yang telah dilatih sebelumnya (Adel dkk., 2022). Proses *feature extraction* dari model DistilBERT dapat diilustrasikan pada Gambar 15.



Gambar 15. Proses *Feature Extraction* Model DistilBERT (Adel dkk., 2022).

2.12 Evaluasi Kinerja Model

Evaluasi kinerja model merupakan tahapan penting dalam klasifikasi yang bertujuan untuk mengukur kemampuan model yang telah dilatih dalam memprediksi data baru. Salah satu metode yang umum digunakan untuk mengevaluasi kinerja model klasifikasi adalah *confusion matrix*. *Confusion matrix* adalah tabel yang digunakan untuk menampilkan jumlah data uji yang berhasil diklasifikasikan dengan benar maupun salah. Metode ini cukup efektif untuk mengukur kinerja suatu sistem klasifikasi. Karena dengan menggunakan *confusion matrix*, kinerja suatu sistem klasifikasi dapat dilihat secara detail sehingga dapat mengidentifikasi letak kesalahan dalam proses klasifikasi (Nurhidayat & Dewi, 2023).

Pada dasarnya *confusion matrix* merupakan tabel berdimensi 2x2 yang terdiri dari kolom yang mewakili kelas prediksi (*predicted classes*) dan baris yang mewakili kelas sebenarnya (*actual classes*) yang direpresentasikan seperti pada Gambar 16. (Markoulidakis dkk., 2021):

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Gambar 16. *Confusion Matrix Klasifikasi Biner* (Markoulidakis dkk., 2021).

Pada *confusion matrix* terdapat empat atribut sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang dapat dijelaskan (Romadloni dkk., 2022):

1. *True Positive* (TP) adalah jumlah data positif yang diprediksi dengan benar.
2. *True Negative* (TN) adalah jumlah data negatif yang diprediksi dengan benar.
3. *False Positive* (FP) atau error tipe 1 adalah jumlah data negatif yang diprediksi sebagai data positif.
4. *False Negative* (FN) atau error tipe 2 adalah jumlah data positif yang diprediksi sebagai data negatif.

Pada kasus klasifikasi *multiclass*, *confusion matrix* untuk klasifikasi biner tidak berlaku sepenuhnya. *Confusion matrix* pada klasifikasi *multiclass* memiliki dimensi $N \times N$, dimana N adalah jumlah kelas yang berbeda seperti $C_0, C_1, C_2, \dots, C_N$. Nilai-nilai *confusion matrix* pada klasifikasi *multiclass* diperoleh dengan menggabungkan beberapa metrik yang digunakan dalam klasifikais biner. *Confusion matrix* klasifikasi *multiclass* dapat disajikan pada Gambar 17 (Markoulidakis dkk., 2021).

		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	...	$C_{N,N}$

Gambar 17. *Confusion Matrix* klasifikasi *Multiclass* (Markoulidakis dkk., 2021).

Confusion matrix juga dapat digunakan sebagai dasar pendekatan peluang, seperti yang diusulkan oleh Teorema Bayes. Teorema Bayes adalah teorema yang digunakan untuk menghitung peluang bersyarat, yaitu peluang suatu kejadian berdasarkan informasi atau kejadian lain yang telah terjadi. Pada konteks klasifikasi, Teorema Bayes digunakan untuk menghitung peluang bahwa suatu data termasuk dalam kelas tertentu berdasarkan hasil prediksi model dengan memanfaatkan informasi dari *confusion matrix*. Secara matematis Teorema Bayes dapat dirumuskan seperti pada Persamaan (19) (Lynch & Bartlett, 2019).

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (19)$$

Keterangan:

$P(A|B)$ = Peluang terjadinya kejadian A , dengan syarat B telah terjadi

$P(B|A)$ = Peluang terjadinya kejadian B , dengan syarat A telah terjadi

$P(A)$ = Peluang kejadian A

$P(B)$ = Peluang kejadian B

Peluang bersyarat $P(A|B)$ didefinisikan sebagai Persamaan (20).

$$P(A|B) = \frac{P(B \cap A)}{P(B)} \quad (20)$$

$P(A \cap B)$ merupakan peluang kejadian A dan B terjadi secara bersamaan.

Berdasarkan definisi 20, Teorema Bayes dapat dijabarkan seperti pada Persamaan (21).

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)} = \frac{P(B \cap A)}{P(A)} \quad (21)$$

Beberapa metrik evaluasi yang umum digunakan untuk mengukur kinerja model klasifikasi adalah akurasi, presisi, sensitivitas, spesifisitas, *F1-score*, serta *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC) (Ramírez-Arias dkk., 2022). Untuk klasifikasi *multiclass* diperlukan metrik evaluasi yang dapat memberikan rata-rata dari akurasi, presisi, sensitivitas, spesifisitas, *F1-score*, dan ROC-AUC dari setiap kelas (Markoulidakis dkk., 2021).

1. Akurasi

Akurasi digunakan untuk mengukur seberapa baik model dalam mengklasifikasikan dengan benar. Secara matematis, akurasi dapat dituliskan pada Persamaan (22).

$$Akurasi_{avg} = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N C_{i,j}} \quad (22)$$

2. Presisi

Presisi menunjukkan seberapa akurat model dalam memprediksi data positif yang diprediksi benar positif dengan keseluruhan data yang diprediksi positif. Secara matematis, presisi dapat dituliskan pada Persamaan (23).

$$Presisi \text{ kelas } C_i = PPV(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (23)$$

Presisi juga dapat dihitung dengan menggunakan konsep peluang bersyarat dalam Teorema Bayes, sehingga dapat dituliskan seperti pada Persamaan (24).

$$\begin{aligned} Presisi \text{ kelas } C_i &= P(\text{Aktual Positif}(C_i) | \text{Prediksi Positif}(C_i)) \\ &= \frac{P(\text{Aktual Positif}(C_i) \cap \text{Prediksi Positif}(C_i))}{P(\text{Prediksi Positif}(C_i))} \end{aligned} \quad (24)$$

Rata-rata presisi dari setiap kelas dapat dihitung menggunakan Persamaan (25).

$$Presisi_{avg} = \frac{1}{N} \sum_{i=1}^N PPV(C_i) = \frac{1}{N} \sum_{i=1}^N Presisi\ kelas(C_i) \quad (25)$$

3. Sensitivitas

Sensitivitas adalah perbandingan antara jumlah data positif yang diprediksi benar positif dengan keseluruhan data positif yang sebenarnya. Sensitivitas menggambarkan keberhasilan model dalam mengidentifikasi dengan benar data yang sebenarnya positif. Secara matematis, sensitivitas dapat dituliskan pada Persamaan (26).

$$Sensitivitas\ kelas\ C_i = TPR(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (26)$$

Sensitivitas juga dapat dihitung dengan menggunakan konsep peluang bersyarat dalam Teorema Bayes, sehingga dapat dijabarkan pada Persamaan (27).

$$\begin{aligned} Sensitivitas\ kelas\ C_i &= P(\text{Prediksi Positif}(C_i) | \text{Aktual Positif}(C_i)) \\ &= \frac{P(\text{Prediksi Positif}(C_i) \cap \text{Aktual Positif}(C_i))}{P(\text{Aktual Positif}(C_i))} \end{aligned} \quad (27)$$

Rata-rata sensitivitas dari setiap kelas dapat dihitung menggunakan Persamaan (28).

$$Sensitivitas_{avg} = \frac{1}{N} \sum_{i=1}^N TPR(C_i) = \frac{1}{N} \sum_{i=1}^N Sensitivitas\ kelas(C_i) \quad (28)$$

4. Spesifisitas

Spesifisitas adalah perbandingan antara jumlah data negatif yang diprediksi benar negatif dengan keseluruhan data negatif yang sebenarnya. Sensitivitas menggambarkan keberhasilan model dalam mengidentifikasi dengan benar data yang sebenarnya negatif. Secara matematis, spesifisitas dapat dituliskan pada Persamaan (29).

$$Spesifisitas\ kelas\ C_i = TNR(C_i) = \frac{TN(C_i)}{TN(C_i) + FP(C_i)} \quad (29)$$

Spesifisitas juga dapat dihitung dengan menggunakan konsep peluang bersyarat

dalam Teorema Bayes, sehingga dapat dituliskan seperti pada Persamaan (30).

$$\begin{aligned} \text{Spesifisitas kelas } C_i &= P(\text{Prediksi Negatif}(C_i) | \text{Aktual Negatif}(C_i)) \\ &= \frac{P(\text{Prediksi Negatif}(C_i) \cap \text{Aktual Negatif}(C_i))}{P(\text{Aktual Negatif}(C_i))} \end{aligned} \quad (30)$$

Rata-rata spesifisitas dari setiap kelas dapat dihitung menggunakan Persamaan (31).

$$\text{Spesifisitas}_{\text{avg}} = \frac{1}{N} \sum_{i=1}^N \text{TNR}(C_i) = \frac{1}{N} \sum_{i=1}^N \text{Spesifisitas kelas}(C_i) \quad (31)$$

5. *F1-Score*

F1-Score adalah metrik yang membandingkan rata-rata *precision* dan *recall* yang dibobotkan. *F1-Score* digunakan untuk mengevaluasi kinerja model dalam memprediksi data positif dengan benar dan mengidentifikasi sebagian besar data positif yang sebenarnya. Semakin besar nilai *F1-Score* maka semakin baik kinerja model klasifikasi. Secara matematis, *F1-Score* dapat dituliskan pada Persamaan (32) dan (33).

$$\text{F1-Score kelas } C_i = F_1(C_i) = 2 \times \frac{\text{TPR}(C_i) \times \text{PPV}(C_i)}{\text{TPR}(C_i) + \text{PPV}(C_i)} \quad (32)$$

$$\text{F1-Score}_{\text{avg}} = 2 \times \frac{\text{TPR}(\text{Macro}) \times \text{PPV}(\text{Macro})}{\text{TPR}(\text{Macro}) + \text{PPV}(\text{Macro})} \quad (33)$$

6. *Receiver Operating Characteristic (ROC)* dan *Area Under the Curve (AUC)*

Receiver operating characteristic merupakan metode yang menyajikan grafik *true positive rate* (TPR) atau sensitivitas terhadap *false positive rate* (FPR) dalam ukuran 2 dimensi dengan garis horizontal adalah nilai *false positive* dan garis vertikal adalah *true positive*. *False Positive Rate* adalah risiko salah mengidentifikasi kejadian negatif sebagai positif. *False Positive Rate* secara matematis direpresentasikan pada Persamaan (34) (Figueiredo, dkk., 2018).

$$\text{FPR} = 1 - \text{Spesifisitas} = 1 - \text{TNR} \quad (34)$$

Kurva ROC digunakan untuk mengevaluasi seberapa baik model dapat membedakan antara kelas positif dan negatif. Kurva ini membandingkan kinerja model dengan pemilihan kelas secara acak, yang direpresentasikan oleh garis diagonal dari titik (0,0) ke (1,1). Model dengan kinerja lebih baik akan memiliki

kurva ROC di atas garis ini, sementara model yang kurang efektif cenderung berada di bawahnya (Markoulidakis dkk., 2021).

Area Under the Curve adalah metrik yang menunjukkan luas area di bawah kurva ROC. Nilai AUC berada di antara 0 hingga 1, semakin tinggi nilai AUC maka semakin baik kinerja suatu algoritma. Analisis ROC juga dapat diterapkan dalam klasifikasi *multiclass*, tetapi kompleksitas meningkat seiring bertambahnya jumlah kelas. Misalnya, dalam masalah klasifikasi dengan N kelas, salah satu metode penerapan analisis ROC adalah dengan membuat ROC chart untuk setiap kelas, di mana kelas yang dipilih dianggap sebagai kelas positif dan gabungan kelas lainnya dianggap sebagai kelas negatif. Hal ini menyebabkan N diagram ROC terbentuk. Selanjutnya, AUC makro dihitung dengan mengambil rata-rata dari nilai AUC tiap kelas. Secara matematis, nilai AUC dapat dituliskan pada Persamaan (35) dan (36) (Powers, 2020).

$$\begin{aligned} AUC \text{ kelas } C_i &= \frac{TPR(C_i) - FPR(C_i) + 1}{2} \\ &= \frac{TPR(C_i) + TNR(C_i)}{2} \end{aligned} \quad (35)$$

$$AUC_{\text{avg}} = \frac{1}{N} \sum_{i=1}^N AUC(C_i) \quad (36)$$

2.13 Uji-t

Uji-t merupakan suatu metode analisis statistik yang digunakan untuk menentukan apakah terdapat perbedaan yang signifikan secara statistik antara dua kelompok atau populasi. Ada beberapa jenis uji-t yang umum digunakan, salah satunya adalah uji-t berpasangan atau *paired t-test*, yaitu metode pengujian hipotesis yang diterapkan pada data yang saling berhubungan atau berpasangan. Umumnya, uji ini digunakan ketika satu objek penelitian dikenai dua perlakuan berbeda (Montolalu & Langi, 2018). Berikut adalah tahapan pengujian hipotesis menggunakan uji-t berpasangan.

1. Hipotesis:

H_0 : Tidak terdapat perbedaan rata-rata yang signifikan antara dua kelompok ($\mu_1 - \mu_2 = 0$)

H_1 : Terdapat perbedaan rata-rata yang signifikan antara dua kelompok ($\mu_1 - \mu_2 \neq 0$)

2. Taraf Signifikansi

$$\alpha = 5\% = 0.05$$

3. Statistik Uji:

Secara matematis statistik uji untuk uji-t berpasangan dapat dituliskan pada Persamaan (37).

$$t = \frac{\bar{d}}{\frac{Sd}{\sqrt{n}}} \quad (37)$$

Standar deviasi dijabarkan pada Persamaan (38).

$$Sd = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (38)$$

Keterangan:

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 $T_{hitung} > T_{tabel}$.

Tidak tolak H_0 jika $p - value > \alpha = 0.05$ atau $T_{hitung} < T_{tabel}$.

5. Kesimpulan:

Jika $p - value < \alpha = 0.05$ atau $T_{hitung} > T_{tabel}$ maka H_0 ditolak, artinya terdapat perbedaan rata-rata yang signifikan antara dua kelompok.

Jika $p - value > \alpha = 0.05$ atau $T_{hitung} < T_{tabel}$ maka H_0 tidak ditolak, artinya tidak terdapat perbedaan rata-rata yang signifikan antara dua kelompok.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilakukan melalui studi literatur yang bertempat di Jurusan Matematika, Fakultas Matematika, dan Ilmu Pengetahuan Alam, Universitas Lampung. Berlokasi di Jalan Prof. Dr. Soemantri Brojonegoro No.1, Gedong Meneng, Bandar Lampung.

3.1.2 Waktu Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2024/2025 tepatnya bulan Agustus 2024. Proses pengerjaan terbagi menjadi tiga tahap. Tahap pertama melakukan pengumpulan, pemahaman terhadap literatur dan menentukan tema penelitian. Setelah menentukan tema penelitian, akan dilakukan pengumpulan data terkait. Tahap kedua merupakan tahap pengerjaan program yang dimulai dari *pre-processing* data, augmentasi data dan pemodelan klasifikasi menggunakan model DistilBERT. 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 diperoleh dari Kaggle <https://www.kaggle.com/datasets/suchintikasarkar/>

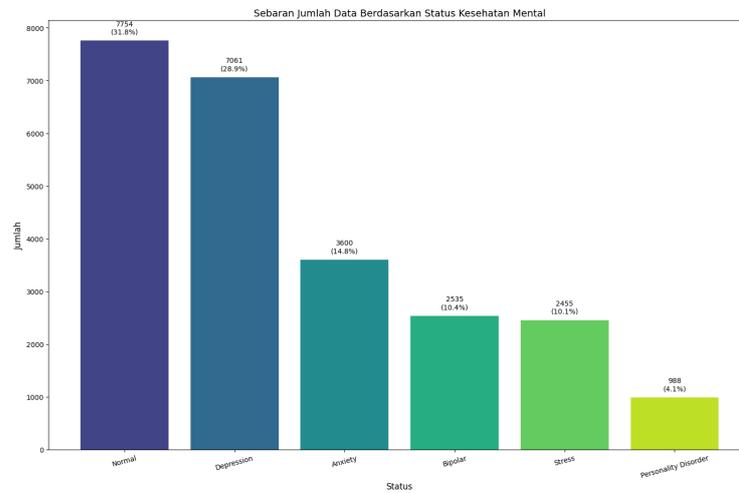
sentiment-analysis-for-mental-health. Data tersebut merupakan data teks berbahasa Inggris yang terakhir di perbaharui pada bulan Juni 2024 oleh Suchintika Sarkar. Data ini merupakan kumpulan status kesehatan mental yang digabungkan dari berbagai sumber. Kumpulan data tersebut bersumber dari *3k conversations dataset for chatbot, depression reddit cleaned, human stress prediction, predicting anxiety in mental health data, mental health dataset bipolar, reddit mental health data, students anxiety and depression dataset, suicidal mental health dataset, suicidal tweet detection dataset*, yang dibuat menjadi 7 label yaitu, *anxiety, bipolar, depression, normal, stress, suicidal, dan personality disorder*. Pada penelitian ini, data dengan label *suicidal* tidak digunakan karena *suicidal* atau pikiran untuk bunuh diri bukan merupakan gangguan mental yang berdiri sendiri, melainkan gejala atau dampak dari gangguan mental lain. Penghapusan label *suicidal*, diharapkan dapat membuat model lebih fokus dalam mengklasifikasikan jenis gangguan mental yang sudah terdefinisi secara jelas, sehingga kinerja model juga dapat meningkat.

Dataset ini memiliki 3 atribut/variabel yaitu *unique_id, statement, dan status*. *Unique_id* adalah pengidentifikasi unik untuk setiap baris, *statement* adalah data tekstual atau pernyataan yang diposting, *status* adalah status kesehatan mental yang ditandai pada pernyataan. Data yang digunakan dalam penelitian ini adalah data kualitatif dengan total jumlah pengamatan sebanyak 24.393 pengamatan. Variabel yang akan digunakan untuk klasifikasi hanya variabel *statement* dan *status* saja. Data ini termasuk data *supervised learning* karena sudah memiliki label.

Sebaran jumlah statement berdasarkan status kesehatan mentalnya dapat disajikan pada Tabel 3 dan Gambar 18.

Tabel 3. Sebaran Jumlah Data Berdasarkan Status Kesehatan Mental

Status	Jumlah	Persentase (%)
<i>Normal</i>	7754	31.8
<i>Depression</i>	7061	28.9
<i>Anxiety</i>	3600	14.8
<i>Bipolar</i>	2535	10.4
<i>Stress</i>	2455	10.1
<i>Personality Disorder</i>	988	4.1



Gambar 18. Sebaran Jumlah Data Berdasarkan Status Kesehatan Mental

Kemudian sampel dari kumpulan data yang digunakan dalam penelitian ini disajikan dalam Tabel 4.

Tabel 4. Sampel Data Penelitian

	Unnamed:0	statement	status
0	1	could you imagine angel eva fight against tita...	Depression
1	2	i am on week on prozac and experiencing no rea...	Depression
2	3	i hope so. i have to go to the bathroom.	Normal
3	4	I feel like I never stood a chance I was made...	Depression
4	5	but of course the document with the important ...	Normal
...
24389	24390	A''Fuck you bitch, I can make your life hell and...	Stress
24390	24391	IYou've only been playing for a while, don't be...	Normal
24391	24392	My mom [54F] has catered to his every need sin...	Normal
24392	24393	[Kinda Gross] Hey what is this in my throat, I...	Anxiety

3.2.2 Alat

Peralatan yang digunakan dalam menunjang penelitian ini adalah:

a. Perangkat Keras (*Hardware*)

Perangkat yang digunakan dalam penelitian ini adalah laptop dengan merk

VivoBook 14 ASUS Laptop X407UAR dengan 64-bit *operating system*, *x64-based processor*. Spesifikasi *hardware* perangkat tersebut adalah sebagai berikut:

- *Processor*: Intel® Core™ i3-7020U CPU @ 2.30GHz (4 CPUs), 2.3GHz
- *RAM*: 4 GB

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk mendukung penelitian ini adalah:

1. Sistem Operasi

Windows 10 Pro 64-bit, adalah versi sistem operasi *Windows* 10 profesional yang menggunakan prosesor 64-bit. *Windows* versi ini memiliki beberapa kelebihan, yaitu lebih cepat, efisien, serta lebih responsif ketika menjalankan beberapa program secara bersamaan, sehingga memungkinkan kinerja prosesor lebih optimal.

2. *Library Pandas 2.2.2*

Pandas adalah *library Python* yang digunakan untuk pengolahan dan analisis data (Hermanto dkk., 2022).

3. *Library NumPy 1.26.4*

NumPy adalah *library Python* yang digunakan untuk proses komputasi numerik. *NumPy* menyediakan objek array multidimensi serta berbagai fungsi matematika yang dapat dioperasikan pada array tersebut (Hermanto dkk., 2022).

4. *Library Matplotlib 3.7.5*

Matplotlib adalah *library Python* yang digunakan untuk memvisualisasikan data. *Matplotlib* menyediakan berbagai fungsi dan fitur untuk membuat berbagai jenis grafik, plot, serta bentuk visualisasi data lainnya (Hermanto dkk., 2022).

5. *Library Seaborn 0.12.2*

Seaborn adalah *library Python* dikembangkan di atas *Matplotlib* dan digunakan untuk visualisasi data statistik. *Seaborn* menawarkan fitur dan gaya visual yang lebih beragam untuk menghasilkan grafik yang menarik dan mudah dipahami (Hermanto dkk., 2022).

6. *Library NLTK 3.2.4*

Natural Language Tool Kit (NLTK) adalah *library open source* yang terdiri dari modul, tutorial, dan kumpulan tugas, yang dirancang khusus untuk komputasi linguistik. *NLTK Library* ini mendukung pemrosesan bahasa alami secara simbolik maupun statistik, serta berfokus pada penggunaan korpus. *NLTK* menyediakan berbagai alat pemrosesan bahasa seperti *tokenizer*, *stemmer*, *tagger*, *parser* sintaksis, dan analisis semantik, serta

dilengkapi dengan korpora standar dan alat untuk mengaksesnya korpus tersebut dengan efisien (Rifano dkk., 2020).

7. *Library Transformers* 4.44.2

Transformers adalah *library Python* yang dikembangkan untuk memudahkan penggunaan model bahasa berbasis *transformer* dan memfasilitasi distribusi model yang telah dilatih sebelumnya (Wolf dkk., 2020).

8. *Library Scikit-Learn* 1.2.2

Scikit-Learn adalah *library* yang dikembangkan untuk memberikan kemudahan dalam pengkodean *machine learning* dalam *Python* (Ranjan dkk., 2023).

9. *Library PyTorch* 2.4.0

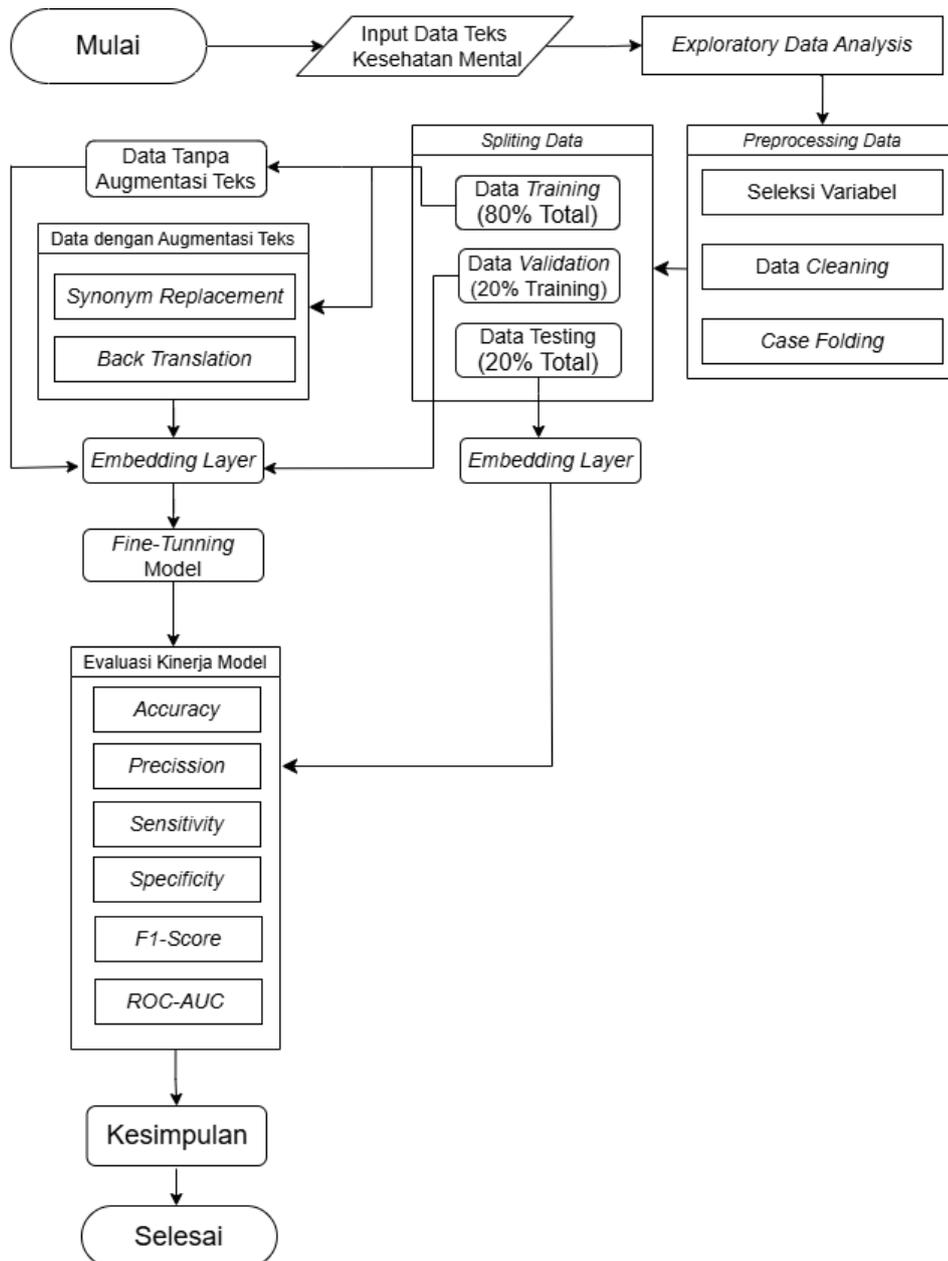
PyTorch adalah *framework open source* berbasis *Python* yang digunakan untuk komputasi tensor dan pembelajaran mesin (Rifano dkk., 2020)(Ranjan dkk., 2023).

10. *Library WordCloud* 1.9.3

WordCloud adalah *library Python* yang digunakan untuk membantu memvisualisasikan kata-kata yang disusun berdasarkan frekuensi kata *Python* (Murthy & Scholar, 2020).

3.3 Metode Penelitian

Penelitian ini membandingkan efektivitas dari teknik augmentasi teks *synonym replacement* dan *back translation* dalam meningkatkan kinerja model klasifikasi DistilBERT pada data kesehatan mental. Penelitian ini dilakukan menggunakan perangkat lunak *python* yang ada di *kaggle* dan *google colab*. Langkah-langkah yang dilakukan dalam penelitian ini diilustrasikan pada Gambar 19.



Gambar 19. *Flowchart* Prosedur Penelitian.

Berdasarkan alur pengerjaan pada Gambar 19, berikut penjelasan setiap tahapannya:

1. Pengerjaan data pada *software python*, dengan *Google Colab* dan *Kaggle* sebagai tempat untuk menulis *code python*.
2. *Preprocessing* pada data teks yang telah diinput. meliputi seleksi variabel dengan menghapus variabel yang tidak diperlukan, Penghapusan *missing values* dan data duplikat, pembersihan karakter non-alfabet, serta penerapan *case folding*.
3. Data yang telah diterapkan *preprocessing* akan dibagi menjadi 3 bagian yaitu

data pelatihan, data validasi, dan data pengujian. *Splitting* data dilakukan dengan rasio 80% dari kumpulan data total untuk data pelatihan dan 20% dari kumpulan data total untuk data pengujian. Pengambilan data validasi dilakukan dengan rasio 20% dari data pelatihan. Data pelatihan digunakan untuk melatih model, data validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan sedangkan data pengujian digunakan untuk menguji seberapa baik model dalam memprediksi data yang belum pernah dilihat.

4. Melakukan augmentasi teks menggunakan teknik *synonym replacement* dan *Back translation* untuk meningkatkan variasi data pelatihan. *Synonym replacement* dilakukan menggunakan sumber sinonim dari *WordNet* dan dikombinasikan dengan algoritma *Lesk* untuk memilih sinonim yang sesuai dengan konteks kalimat. Sedangkan *Back translation* dilakukan dengan menggunakan model terjemahan MarianMT. Teks diterjemahkan ke dalam enam bahasa yaitu Prancis, Spanyol, Jerman, Indonesia, Belanda, dan Swedia.
5. Melakukan *preprocessing* teks kembali pada data hasil augmentasi, *preprocessing* yang dilakukan meliputi penghapusan duplikat, *missing values*, dan kalimat tidak bermakna.
6. Menerapkan *embedding* untuk mengubah setiap teks menjadi vektor numerik yang dapat diproses oleh model menggunakan DistilBERT *embedding*.
7. Melakukan *fine-tuning* DistilBERT untuk klasifikasi status kesehatan mental dengan menggunakan DistilBert *Sequence Classification*. Model dibangun dan dilatih menggunakan data pelatihan dan validasi, selanjutnya menentukan parameter seperti *batch size*, *dropout*, *learning rate*, *weight decay*, dan *epoch*.
8. Menguji model DistilBERT menggunakan data pengujian.
9. Melakukan evaluasi model untuk melihat kinerja model klasifikasi dengan menggunakan nilai akurasi, presisi, sensitivitas, spesifisitas *F1-score*, serta grafik ROC-AUC. metrik evaluasi yang dilakukan bertujuan untuk mengetahui seberapa baik model dapat mengklasifikasikan teks status kesehatan mental.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini telah dilakukan pembangunan dan evaluasi model DistilBERT dengan penerapan augmentasi data menggunakan *back translation* dan *synonym replacement* untuk klasifikasi teks status kesehatan mental. Berdasarkan hasil penelitian, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Pembangunan model klasifikasi:

Model DistilBERT untuk klasifikasi teks status kesehatan mental dibangun melalui beberapa tahapan, di antaranya *pre-processing* data, *splitting* data, augmentasi data menggunakan teknik *back translation* dan *synonym replacement*, dan penyesuaian *hyperparameter* model. Kinerja dari model klasifikasi yang telah dibangun dievaluasi menggunakan beberapa metrik evaluasi, yaitu rata-rata akurasi, rata-rata presisi, rata-rata sensitivitas, rata-rata spesifisitas, rata-rata *F1-score* dan rata-rata AUC berdasarkan *confusion matrix* untuk setiap kelas pada data pengujian. Proses ini dilakukan dengan tujuan untuk menghasilkan model yang optimal dalam mengklasifikasikan teks terkait status kesehatan mental.

2. Efektivitas penggunaan teknik augmentasi teks:

Hasil percobaan menunjukkan bahwa model DistilBERT memiliki kinerja yang cukup baik dalam mengklasifikasikan teks status kesehatan mental. Pada data tanpa augmentasi, model DistilBERT mencapai akurasi sebesar 80%. Setelah dilakukan augmentasi dengan *back translation*, akurasi meningkat menjadi 86%. Sementara itu, model DistilBERT mencapai akurasi tertinggi pada data dengan augmentasi *synonym replacement*, yaitu sebesar 87%. Selain peningkatan akurasi, model dengan augmentasi *synonym replacement* juga menunjukkan rata-rata presisi sebesar 81%, sensitivitas sebesar 82%, spesifisitas 97%, *F1-score* 81%, dan AUC 0,8947. Hal ini menunjukkan bahwa augmentasi *synonym replacement* tidak hanya meningkatkan akurasi, tetapi juga membantu

model mengenali pola pada berbagai kelas dengan lebih baik. Hasil penelitian juga menunjukkan bahwa teknik augmentasi teks memberikan dampak positif terhadap kinerja model DistilBERT. Dibandingkan dengan *back translation*, teknik augmentasi *synonym replacement* terbukti lebih efektif meningkatkan akurasi pada penelitian ini, dengan peningkatan akurasi sebesar 7%, sedangkan *back translation* hanya meningkatkan akurasi sebesar 6%.

3. Uji signifikansi peningkatan kinerja model:

Berdasarkan uji t-berpasangan yang dilakukan, diperoleh nilai *p-value* untuk kedua pasangan berturut-turut adalah 0.0000741317 dan 0.0000693309 yang jauh lebih kecil dari taraf signifikansi 0.05 serta nilai T_{hitung} yang melebihi nilai T_{tabel} , sehingga hipotesis nol ditolak. Dengan demikian, dapat disimpulkan bahwa terdapat perbedaan rata-rata akurasi yang signifikan secara statistik antara model yang dilatih menggunakan data asli dan data hasil augmentasi. Artinya, penggunaan teknik augmentasi terbukti efektif dalam meningkatkan akurasi model dalam tugas klasifikasi status kesehatan mental.

5.2 Saran

Berdasarkan hasil penelitian, beberapa saran dapat diberikan untuk peningkatan kualitas dan keberlanjutan penelitian:

1. Eksplorasi lebih lanjut dengan mempertimbangkan penggunaan model analisis teks yang lebih kompleks, seperti RoBERTa, GPT, XLNet atau model lainnya, untuk mendapat pemahaman yang lebih luas terhadap kinerja model.
2. Melakukan peninjauan kembali terhadap *hyperparameter* model serta menyesuaikan langkah-langkah *pre-processing* dan augmentasi data guna meningkatkan kinerja model dalam melakukan klasifikasi.
3. Mempertimbangkan penggunaan teknik *balancing* data lainnya guna mengatasi ketidakseimbangan kelas, sehingga dapat memberikan wawasan tambahan mengenai metode yang lebih efektif dalam klasifikasi teks status kesehatan mental.
4. Penelitian selanjutnya dapat menerapkan model klasifikasi DistilBERT pada kumpulan data yang berbeda guna menguji kemampuan generalisasi model terhadap berbagai domain teks.

DAFTAR PUSTAKA

- Adel, H., dkk. 2022. Improving Crisis Events Detection Using Distilbert with Hunger Games Search Algorithm. *Mathematics*. **10**(3): 1-22.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M.A., & Farhan, L. .2021. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of big Data*. **8**(1): 1-74.
- Ananda, I. K., Fanani, A. Z., Setiawan, D., & Wicaksono, D. F. 2024. Penerapan Random Oversampling dan Algoritma Boosting untuk Memprediksi Kualitas Buah Jeruk. *Edumatic: Jurnal Pendidikan Informatika*. **8**(1): 282-289.
- Azizah, A. N., Asy'ari, M. F., Prastya, I. W. D., & Purwitasari, D. 2023. Easy Data Augmentation untuk Data yang Imbalance pada Konsultasi Kesehatan Daring. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **10**(5): 1095-1104.
- Azmi, B. N., Hermawan, A., & Avianto, D. 2023. Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver. *JTIM: Jurnal Teknologi Informasi dan Multimedia*. **4**(4): 281-290.
- 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**: 1-13.
- Bucos, M. & Tucudean, G. 2023. Text Data Augmentation Techniques for Fake News Detection in The Romanian Language. *Applied Sciences*. **13**(13): 1-20.
- Cahyani, L. & Arif, M. 2022. Text Mining untuk Pengelompokan Skripsi di Prodi Pendidikan Informatika Universitas Trunojoyo Madura. *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*. **8**(2): 97-108.
- Devlin, J., Chang, M., Lee, K., & Tautanova, K. 2018. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

- Fadilah, N. & Priyanta, S. 2022. Automatic Essay Scoring Using Data Augmentation in Bahasa Indonesia. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. **16**(4): 401-410.
- Fajri, F., Tutuko, B., & Sukemi, S. 2022. Membandingkan Nilai Akurasi BERT dan DistilBERT pada Dataset Twitter. *JUSIFO (Jurnal Sistem Informasi)*. **8**(2): 71-80.
- Fazar, D. M. & Widiastuti, N. I. 2017. Lexical Chain dan Word Sense Disambiguation untuk Peringkasan Artikel Berbahasa Indonesia. *Techno.COM*. **16**(2): 195-207.
- De Figueiredo, 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.
- Gujjar, V., Mago, N., Kumari, R., Patel, S., Chintalapudi, N., & Battineni, G. 2023. A Literature Survey on Word Sense Disambiguation for the Hindi Language. *Information*. **14**(9): 1-25.
- Gumelar, G., Kusriani, & Al Fatta, H. 2023. Kombinasi Algoritma Klasifikasi dengan Algoritma Oversampling untuk Menangani Ketidakseimbangan Kelas pada Level Data. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*. **10**(2): 29-39.
- Hairani, H., Setiawan, N. A., & Adji, T. B. 2016. Metode Klasifikasi Data Mining dan Teknik Sampling SMOTE Menangani Class Imbalance untuk Segmentasi Customer pada Industri Perbankan. *Prosiding Sains Nasional dan Teknologi*, **1**(1): 168-172.
- Haryawan, C. & Ardhana, Y. M. K. 2023. Analisa Perbandingan Teknik Oversampling Smote pada Imbalanced Data. *Jurnal Informatika dan Rekayasa Elektronik*. **6**(1): 73-78.
- Hermanto, K., Salim, D., Wu, B., Salim, O. R., & Gunadi, R. B. 2022. Penggunaan Python untuk Menganalisis Pola Penyebaran Covid-19 di Masa Pandemi. *Journal of Student Development Information System (JoSDIS)*. **2**(2): 120-133.
- Kapusta, J., Drzik, D., Steflovic, K., & Nagy, A. K. S. 2024. Text Data Augmentation Techniques for Word Embeddings in Fake News Classification. *IEEE Access*. **12**: 31538-31550.
- Khairunnisa, S., Adiwijaya, A., & Al Faraby, S. 2021. Pengaruh Text Preprocessing Terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19). *Jurnal Media Informatika Budidarma*. **5**(2): 406-414.
- Kim, J., Lee, J., & Han, J. 2020. A Deep Learning Model for Detecting Mental Illness from User Content on Sossial Media. *Scientific reports*. **10**(1): 1-6.

- Lee, M. 2023. Gelu Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance. arXiv preprint arXiv:2305.12073.
- Lynch, S. M., & Bartlett, B. 2019. Bayesian Statistics in Sociology: Past, Present, and Future. *Annual Review of Sociology*. **45**(1): 47-68.
- Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgaulus, I. 2021. Multi-class Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*. **9**(4): 1-22.
- McCormick, C. & Ryan, N. 2019. BERT Word Embeddings Tutorial. <http://www.mccormickml.com>. Diakses pada 20 Februari 2025.
- Mienye, I. D., & Swart, T. G. 2024. A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information*. **15**(12): 1-45.
- Montolalu, C., & Langi, Y. 2018. Pengaruh Pelatihan Dasar Komputer dan Teknologi Informasi Bagi Guru-Guru dengan Uji-t Berpasangan (Paired Sample T-test). *dCartesian: Jurnal Matematika dan Aplikasi*. **7**(1): 44-46.
- Muhammad, F. F. M., & Pulungan, M. R. M. I. 2022. Perbandingan Kinerja Beberapa Metode Word Embedding pada Bahasa Indonesia (Skripsi). Jurusan Ilmu Komputer FMIPA UGM, Yogyakarta.
- Muraina, I. 2022. Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts, 496-504. international Mardin Artuklu scientific research conference.
- Murthy, K. N., & Scholar, P. G. 2020. Word Cloud in Python. *Complexity International*. **24**(01): 221-226.
- Novac, G. 2023. BERT Embeddings. TinkerD. <https://tinkerD.net/blog/machine-learning/bert-embeddings/>. Diakses Pada 21 Februari 2025.
- Nur, I. M., Ismatullah, & Muntasiroh, L. 2023. Implementasi Adaptive Synthetic Bernoulli Naïve Bayes untuk Mengatasi Imbalance Class Data pada Kasus Mental Health di Indonesia. *Journal of Data Insights*. **1**(1): 10-18.
- Nurhidayat, R., & Dewi, K. E. 2023. Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek. *Komputa: Jurnal Ilmiah Komputer dan Informatika*. **12**(1): 91-100.
- Oktariansyah, I. A., Umbara, F. R., & Kasyadi, F. 2024. Klasifikasi Sentimen untuk Mengetahui Kecenderungan Politik Pengguna X pada Calon Presiden 2024 Menggunakan Metode IndoBERT. *Building of informatics, Technology and Science (BITS)*. **6**(2): 636-648.
- Powers, D. M. 2020. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. arXiv preprint arXiv:2010.16061.

- Prasetyo, V. R., Benarkah, N., & Chrisintha, V. J. 2021. Implementasi Natural Language Processing dalam Pembuatan Chatbot pada Program Information Technology Universitas Surabaya. *Jurnal TEKNIKA*. **10**(2): 114-121.
- Pratiwi, Y. & Yaqin, A. 2022. Klasifikasi Tweet Tidak Senonoh Twitter dengan Naïve Bayes Classifier. *E-JURNAL JUSITI: Jurnal Sistem Informasi dan Teknologi Informasi*. **11**(1): 70-80.
- 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: 14-21.
- 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.
- Rahma, I. A. & Suadaa, L. H. 2023. Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia Studi Kasus: Deteksi Judul Clickbait Dan Komentar Hate Speech Pada Berita Online. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **10**(6): 1329-1340.
- 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.
- Ramírez-Arias, F. J., García-Guerrero, E. E., Tlelo-Cuautle, E., Colores-Vargas, J. M., García-Canseco, E., López-Bonilla, O. R., Galindo-Aldana, G. M., & Inzunza-González, E. 2022. Evaluation of Machine Learning Algorithms for Classification of EEG Signals. *Technologies*. **10**(4): 10-11.
- Rifano, E. J., Fauzan, A. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. Text Summarization pada Berita Bola Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. 2020. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*. **2**(1): 8-17.
- Riyadhi, M. F. 2019. Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering (Studi Kasus: Prodi Sistem Komputer) (Skripsi). Jurusan Sistem Komputer FTIK UNIKOM. Jawa Barat.
- Rivaldi, R. C. & Wismarini, T. D. 2024. Analisis Sentimen pada Ulasan Produk dengan Metode Natural Language Processing (NLP) (Studi Kasus Zalika Store 88 Shopee). *Jurnal Ilmiah Elektronika Dan Komputer*. **17**(1): 120-128.
- Romadloni, P., Kusuma, B. A., & Baihaqi, W. M. 2022. Komparasi Metode Pembelajaran Mesin untuk Implementasi Pengambilan Keputusan dalam Menentukan Promosi Jabatan Karyawan. *JATI (Jurnal Mahasiswa Teknik Informatika)*. **6**(2): 622-628.

- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. 2019. DistilBERT, A Distilled Version of BERT: Small, Faster, Cheaper and Lighter. arXiv preprint arXiv: 1910.01108.
- Sanya, A. D. & Suadaa, L. H. (2022, August). Handling Imbalanced Dataset on Hate Speech Detection in Indonesian Online News Comments. In 2022 10th International Conference on Information and Communication Technology (ICoICT) (pp. 380-385). IEEE.
- Sharma, S., Sharma, S., & Athaiya, A. 2017. Activation Functions in Neural Networks. *Towards Data Sci.* **6**(12): 310-316.
- 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.
- Stahlberg, F. 2020. Neural Machine Translation: A Review. *Journal of Artificial Intelligence Research.* **69**: 343-418.
- Syawanodya, I. 2018. Improvement Of Lemmatization For Indonesian Text Document With Spellchecker (Tesis). Jurusan Teknik Komputer FTE Telkom University.
- 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.
- Tanugeraha, T., Santoso, A. J., & Adithama, S. P. 2023. Pengelompokan Gangguan Kesehatan Mental Mahasiswa yang Sedang Menempuh Skripsi dengan Metode Fuzzy Mamdani. *Jurnal Informatika Atma Jogja.* **4**(1): 77-84.
- 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.
- Vaswani, A., dkk. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems.*
- Vitoasmara, K., Hidayah, F. V., Purnamasari, N. I., Aprilia, R. Y., & Dewi A, L. D. 2024. Gangguan Mental (Mental Disorders). *Student Research Journal.* **2**(3): 57-68.
- Wardani, K. R. R., Suryalim, H., Engel, V. J. L., & Christian, H. 2023. Analisis Pemilihan Optimizer dalam Arsitektur Convolution Neural Network VGG16 dan Inception untuk Sistem Pengenalan Wajah. *JEPIN (Jurnal Edukasi dan Penelitian Informatika).* **9**(2): 186-194.

- Wijayanto, I. R., Cholissodin, I., & Sari, Y. A. 2021. Pengaruh Metode Word Embedding dalam Vector Space Model pada Pemerolehan Informasi Materi IPA Siswa SMP. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 5(3): 950-959.
- Wolf, T., dkk. 2020. Transformers: State-of-the-art natural language processing. hlm. 38-45. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations.