

**KLASIFIKASI SENTIMEN ULASAN PENILAIAN OBAT  
DENGAN METODE *BIDIRECTIONAL ENCODER REPRESENTATIONS  
FROM TRANSFORMER (BERT)***

**(Skripsi)**

**Oleh**

**YAZID ZINEDINE HDIANA**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

## **ABSTRACT**

### **SENTIMENT CLASSIFICATION OF DRUG ASSESSMENT REVIEWS USING THE BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMER (BERT) METHOD**

**By**

**YAZID ZINEDINE HDIANA**

In this era, highly accessible health products such as medicines sold freely through pharmacies and e-commerce platforms have caused individuals who may have limited experience in the health sector to rely on user reviews to understand the types of drugs to treat certain conditions. These reviews are not only beneficial for individuals but also contribute to improving the quality of medicines produced by healthcare professionals. The problem of unstructured data that exists in reviews in the form of text data can be overcome with an innovative approach called BERT. BERT is a transformer-based model that uses pre-training to improve contextual understanding of text. This research evaluates BERT's performance in sentiment analysis using the BERT BASE Uncased pre-training model. The dataset used consists of user reviews of medicines available on the Kaggle platform. This research involves hyperparameter tuning to identify optimal classification models and uses two data labeling methods, namely rating-based labeling and textblob-based labeling. Experimental results show that the BERT BASE Uncased model achieves extraordinary accuracy, reaching a peak of 98% with certain hyperparameters in textblob-based data labeling. In addition, BERT BASE fine-tuning results for rating-based data classification achieved 97% accuracy.

**Keywords :** Sentiment Classification, Deep Learning, Transformers, BERT

## ABSTRAK

### KLASIFIKASI SENTIMEN ULASAN PENILAIAN OBAT DENGAN METODE *BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMER (BERT)*

Oleh

YAZID ZINEDINE HDIANA

Pada era ini, produk kesehatan sangat mudah diakses seperti obat-obatan yang dijual secara bebas melalui apotek dan platform *e-commerce* menyebabkan individu yang mungkin memiliki pengalaman dalam bidang kesehatan yang terbatas mengandalkan ulasan pengguna untuk memahami jenis obat dalam mengatasi kondisi tertentu. Ulasan ini tidak hanya bermanfaat bagi individu tetapi juga berkontribusi terhadap peningkatan kualitas obat yang diproduksi oleh para profesional kesehatan. Permasalahan data tidak terstruktur yang ada pada ulasan dalam bentuk data teks dapat diatasi dengan pendekatan inovatif yang disebut BERT. BERT adalah model berbasis *transformer* yang menggunakan *pre-training* untuk meningkatkan pemahaman teks kontekstual. Penelitian ini mengevaluasi kinerja BERT dalam analisis sentimen menggunakan model *pre-training* BERT BASE *Uncased*. *Dataset* yang digunakan terdiri dari ulasan pengguna terhadap obat-obatan yang tersedia di platform Kaggle. Penelitian ini melibatkan penyetulan *hyperparameter* untuk mengidentifikasi model klasifikasi optimal dan menggunakan dua metode pelabelan data, yaitu pelabelan berbasis *rating* dan pelabelan berbasis *textblob*. Hasil eksperimen menunjukkan bahwa model BERT BASE *Uncased* mencapai akurasi yang luar biasa, mencapai puncak 98% dengan *hyperparameter* tertentu dalam pelabelan data berbasis *textblob*. Selain itu, hasil *fine-tuning* BERT BASE untuk klasifikasi data berbasis *rating* mencapai akurasi 97%.

**Kata Kunci :** Klasifikasi Sentimen, *Deep Learning*, *Transformers*, BERT

**KLASIFIKASI SENTIMEN ULASAN PENILAIAN OBAT  
DENGAN METODE *BIDIRECTIONAL ENCODER REPRESENTATIONS  
FROM TRANSFORMER (BERT)***

Oleh  
**YAZID ZINEDINE HDIANA**  
**2017031071**

**Skripsi**

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA MATEMATIKA**

Pada

Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

Judul Skripsi : **KLASIFIKASI SENTIMEN ULASAN  
PENILAIAN OBAT DENGAN METODE  
BIDIRECTCIONAL ENCODER  
REPRESENTATIONS FROM  
TRANSFORMER (BERT)**

Nama Mahasiswa : **Yazid Zinedine Hadiana**

Nomor Pokok Mahasiswa : **2017031071**


Jurusan : **Matematika**

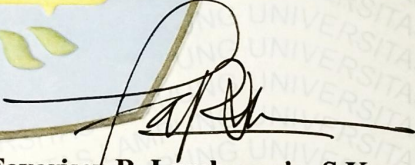
Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



Bandar Lampung, 31 Mei 2024

**MENYETUJUI**  
1. **Komisi Pembimbing**

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

  
**Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D.**  
NIP. 198301102008121002

2. **Ketua Jurusan Matematika**

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

**MENGESAHKAN**

1. Tim Penguji

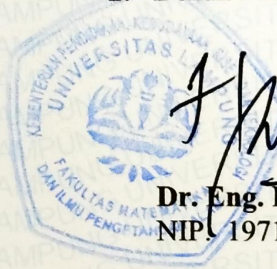
Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.** .....

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: **21 Mei 2024**

## PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama : **Yazid Zinedine Hdiana**  
Nomor Pokok Mahasiswa : **2017031071**  
Jurusan : **Matematika**  
Judul Skripsi : **Klasifikasi Sentimen Ulasan Penilaian Obat  
dengan Metode *Bidirectional Encoder  
Representations From Transformer (BERT)***

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah karya penulisan ilmiah Universitas Lampung.

Bandar Lampung, 31 Mei 2024  
Penulis



**Yazid Zinedine Hdiana**  
**NPM. 2017031071**

## **RIWAYAT HIDUP**

Yazid Zinedine Hdiana lahir di Bekasi pada 8 September 2002. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Hadian dan Ibu Iyet Hayatul.

Penulis memulai perjalanan pendidikannya di Taman Kanak Budi Mulya, Desa Wanasari, Kecamatan Cibitung, Kabupaten Bekasi pada tahun 2007-2008. Pendidikan dasar berhasil diselesaikan di MIS PINK 08, Desa Wanasari, Kecamatan Cibitung, Kabupaten Bekasi pada tahun 2014. Kemudian penulis melanjutkan ke jenjang pendidikan menengah pertama di SMPN 2 Sukaratu, Desa Gunung Sari, Kecamatan Sukaratu, Kabupaten Tasikmalaya dari tahun 2014 hingga 2017. Pilihan karier penulis membawanya ke bidang kejuruan elektronika industri, dimana ia menyelesaikan pendidikan di SMKS Mitra Industri MM2100 pada tahun 2017-2020. Tidak berhenti di situ, penulis melanjutkan studi ke jenjang perguruan tinggi di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, setelah berhasil lolos dalam Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) pada tahun 2020.

Selama menjadi mahasiswa, penulis aktif di beberapa kegiatan sebagai berikut:

1. Pada tahun 2021 penulis menjadi pengurus HIMATIKA Unila dan menjabat sebagai anggota Bidang Eksternal.
2. Pada tahun 2022 penulis menjadi pengurus HIMATIKA Unila dan menjabat sebagai kepala Bidang Eksternal.



3. Pada tahun 2022 penulis mengikuti kegiatan lomba Satria Data di bidang lomba *Big Data Challenge* untuk mengolah data BPJS Kesehatan dengan metode *Random Forest Classifier*.
4. Pada tahun 2023 penulis menjalankan kerja praktikum di Badan Pusat Statistik (BPS) Kabupaten Bekasi dan memegang amanah sebagai petugas *entry data REGSOSEK*.
5. Pada tahun 2023 penulis diamanahkan sebagai asisten dosen mata kuliah Pengantar *Data Mining*.
6. Pada tahun 2023 penulis berperan aktif pada kegiatan sertifikasi Jurusan Matematika oleh ASIIN (*Accreditation in Engineering Computer Sciencies Natural Sciences Mathematics*)
7. Pada tahun 2023 penulis menjalankan kegiatan Kuliah Kerja Nyata (KKN) di Desa Kelungu, Kecamatan Kota Agung Pusat, Kabupaten Tanggamus.
8. Pada tahun 2023 penulis mengikuti kegiatan Satria Data di bidang lomba *Big Data Challenge* untuk mengolah data gambar pelat nomor dengan metode *Optical Character Recognition (OCR)*.
9. Pada tahun 2023 penulis mengikuti kegiatan studi *independent* Kampus Merdeka di Bangkit Academy led by Google, Tokopedia, Gojek, & Traveloka.
10. Pada tahun 2023 penulis diamanahkan sebagai asisten dosen mata kuliah Fungsi Kompleks dan Eksplorasi Data.

## **PERSEMBAHAN**

Dengan Mengucapkan rasa syukur atas segala puji dan kehadiran Allah SWT yang telah melimpahkan nikmat serta hidayah-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Serta tak lupa juga selawat serta salam selalu tercurahkan kepada junjungan Nabi Muhammad SAW. Dengan ketulusan, penulis mempersembahkan rasa terima kasih ini kepada:

### **Ayah dan Ibu**

Setiap kata dalam skripsi ini terukir dari keteguhan dan disiplin yang kalian tanamkan dalam diriku. Sosok yang tegar menjadi mercusuar yang menerangi jalanku saat badai keraguan menerjang. Terima kasih atas setiap teguran dan dorongan yang membantuku bangkit dari kegagalan. Kasih sayang kalian bagaikan samudra yang luas, menyelimutiku dengan rasa aman dan nyaman. Doa-doa kalian yang tulus menjadi kekuatan tak terhingga yang membantuku melewati setiap rintangan. Terima kasih atas setiap pengorbanan dan tetes air mata yang kau korbankan demi kebahagiaanku. Skripsi ini bukan hanya hasil jerih payahku, melainkan buah dari kolaborasi cinta dan pengorbanan kalian. Setiap lembar kertasnya memuat jejak kasih sayang, dedikasi, dan dukungan yang tak ternilai. Terima kasih Ayah dan Ibu atas semua yang telah kalian berikan. Skripsi ini adalah persembahan terindah untuk dua insan terhebat yang selalu mewarnai hidupku dengan cinta dan kasih sayang. Semoga Allah SWT membalas semua kebaikan kalian dengan pahala yang berlimpah.

## **Dosen Pembimbing dan Pembahas**

Skripsi ini adalah titik balik dalam perjalanan akademik saya dan semuanya tidak akan mungkin terjadi tanpa dosen pembimbing dan dosen pembahas. Terima kasih atas kesediaan untuk membimbing saya, atas waktu yang diluangkan untuk saya, atas kesabaran dan ketelitian yang ditunjukkan kepada saya. Semoga Allah SWT senantiasa melimpahkan rahmat dan keberkahan kepada dosen pembimbing dan dosen pembahas, serta membalas semua kebaikan dosen pembimbing dan dosen pembahas dengan pahala yang berlimpah.

## **Teman-temanku**

Skripsi ini saya persembahkan dengan rasa pertemanan yang tulus dan penuh rasa syukur. Di tengah perjalanan akademis yang penuh rintangan, kalian hadir sebagai sumber kekuatan dan semangat. Tawa dan canda kalian menjadi pelarian di kala penat, dan dukungan kalian menjadi suntikan energi saat rasa lelah mulai mendera. Skripsi ini bukan hanya pencapaian individu, melainkan bukti kekuatan pertemanan kita. Kita telah belajar, berjuang, dan bertumbuh bersama. Terima kasih telah menjadi bagian dari perjalanan ini, dan mari kita terus melangkah maju bersama meraih mimpi-mimpi kita.

**Almamater Tercinta Universitas Lampung**

## **KATA INSPIRASI**

*"Dan katakanlah, Berjalanlah di muka bumi dan perhatikanlah bagaimana Allah memulai penciptaan, kemudian Allah SWT. Menciptakan kembali. Sesungguhnya Allah SWT. Mahakuasa atas segala sesuatu."*

**(QS. Al-Ankabut: 20)**

*"Dan yang berjihad (berusaha keras) dengan sungguh-sungguh, maka mereka itulah yang bersyukur kepada Allah."*

**(Q.S. Al-Hajj: 78)**

*"Kegigihan mengalahkan bakat ketika bakat tidak mau bekerja keras."*

**(Calvin Coolidge)**

*"Always Positive Thinking"*

**(Penulis)**

## SANWACANA

Puji syukur kehadirat Allah SWT, atas segala rahmat dan karunianya sehingga penulis dapat menyelesaikan skripsi dengan judul “Klasifikasi Sentimen Ulasan Penilaian Obat dengan Metode *Bidirectional Encoder Representations from Transformers* (BERT)”.

Penulis menyadari bahwa selama proses penulisan skripsi ini masih jauh dari kata sempurna. Dalam menyelesaikan skripsi ini, penulis mendapat dukungan, bimbingan dan bantuan dari beberapa pihak. Oleh karena itu, penulis berterima kasih kepada :

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku dosen pembimbing 1 yang senantiasa memberikan arahan, bantuan, motivasi, dan saran kepada penulis dalam menyusun skripsi ini.
2. Bapak Favorisen R. Lumbanraja, S.Si., M.Sc., Ph.D. selaku dosen pembimbing 2 yang senantiasa memberikan arahan dan saran kepada penulis dalam menyusun skripsi ini.
3. Bapak Ir. Warsono, M.S., Ph.D. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun selama proses penyusunan skripsi.
4. Bapak Dr. Ahmad Faisol, S.Si., M.Sc. selaku dosen pembimbing akademik.
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, staff, karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

8. Bapak, Ibu, adik-adik, beserta keluarga besar yang selalu memberikan semangat dukungan, dan doa kepada penulis.
9. Teman-teman seperjuangan pimpinan HIMATIKA FMIPA Unila 2022 yang telah memberikan semangat.
10. Teman-teman dari bidang eksternal HIMATIKA FMIPA Unila 2022 yang telah menemani.
11. Teman-teman satu kelompok KKN Desa Kelungu yang telah memberi dukungan.
12. Teman-teman satu bimbingan (Arif, Harum, Muhtarom, Nanda, Wais, Ziyad) yang telah memberikan dukungan selama skripsi.
13. Adzra, Calvin, yunda Gusti, dan Margeliza yang telah menemani.
14. Seluruh pihak terkait yang membantu dalam menyelesaikan skripsi ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari skripsi ini masih jauh dari kata sempurna dan masih terdapat banyak kekurangan baik dalam penyajian maupun teknik penulisan. Oleh sebab itu, saran dan kritikan yang membangun senantiasa penulis harapkan demi menyempurnakan skripsi ini.

Bandar Lampung, 31 Mei 2024

Penulis

**Yazid Zinedine Hdiana**

NPM. 2017031071

## DAFTAR ISI

	Halaman
<b>DAFTAR TABEL</b> .....	<b>iv</b>
<b>DAFTAR GAMBAR</b> .....	<b>vi</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang dan Masalah .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
<b>II. TINJAUAN PUSTAKA</b> .....	<b>6</b>
2.1 Penelitian Terkait .....	6
2.1.1 Penelitian Pertama (Ardiansyah dkk., 2023).....	8
2.1.2 Penelitian Kedua (Braja dan Kodra, 2023) .....	9
2.1.3 Penelitian Ketiga (Mass dkk., 2021).....	10
2.1.4 Penelitian Keempat (Rendragraha dkk., 2021) .....	11
2.2 Analisis Sentimen.....	12
2.3 <i>Natural Language Processing</i> .....	12
2.4 <i>Text Mining</i> .....	13
2.5 <i>Imbalance Data</i> .....	14
2.6 <i>Word Embedding</i> .....	15
2.7 <i>Splitting Data</i> .....	16
2.9 Fungsi Aktivasi.....	18
2.10 <i>Deep Learning</i> .....	22
2.11 <i>Adam Optimizer</i> .....	23
2.12 <i>Transformer</i> .....	24
2.12.1 <i>Attention</i> .....	26
2.12.2 <i>Scaled Dot-Product Attention</i> .....	26
2.12.3 <i>Multi Head Attention</i> .....	27

2.12.4	<i>Position-Wise Feed-Forward Networks</i> .....	28
2.12.5	<i>Positional Encoding</i> .....	28
2.13	<i>Bidirectional Encoder Representations from Transformer</i> .....	29
2.13.1	<i>Pre-training BERT</i> .....	31
2.13.2	<i>Fine-tuning BERT</i> .....	33
2.14	Evaluasi Kinerja Model.....	34
<b>III.</b>	<b>METODELOGI PENELITIAN</b> .....	<b>38</b>
3.1	Waktu dan Tempat Penelitian .....	38
3.2	Data dan Alat.....	39
3.2.1	Data .....	39
3.2.2	Alat.....	41
3.3	Metode Penelitian.....	42
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN</b> .....	<b>45</b>
4.1	Proses <i>Input Data</i> .....	45
4.2	<i>Preprocessing Data</i> .....	46
4.3	<i>Labeling Data</i> .....	47
4.4	<i>Balancing Data</i> .....	57
4.5	<i>BERT Embedding</i> .....	58
4.6	<i>Splitting Data</i> .....	60
4.7	<i>Fine-tuning BERT</i> .....	61
4.8	Evaluasi Model.....	65
4.9	<i>Benchmarking</i> dengan Penelitian Terdahulu.....	72
<b>V.</b>	<b>PENUTUP</b> .....	<b>74</b>
5.1	Kesimpulan.....	74
5.2	Saran .....	75
	<b>DAFTAR PUSTAKA</b> .....	<b>76</b>
	<b>LAMPIRAN</b>	



## DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terkait Klasifikasi Sentimen dengan BERT .....	6
2. Sebaran Ulasan Tiap <i>Rating</i> .....	40
3. Sampel Data <i>Training Drugs Review</i> .....	40
4. Sampel Data Validasi <i>Drugs Review</i> .....	41
5. Sampel Data <i>Testing Drugs Review</i> .....	41
6. Perbandingan Teks Sebelum dan Sesudah Diolah .....	47
7. Distribusi Label Lama Pada Label Baru Berdasarkan <i>Rating</i> .....	48
8. Distribusi Label Lama Pada Label Baru Berdasarkan <i>Textblob</i> .....	49
9. Kelas <i>Rating</i> yang Telah Diterapkan <i>Label Encoding</i> .....	55
10. Kelas <i>Textblob</i> yang Telah Diterapkan <i>Label Encoding</i> .....	55
11. Sebaran Kelas Berdasarkan <i>Rating</i> .....	56
12. Sebaran Kelas Berdasarkan <i>Textblob</i> .....	56
13. Sebaran Kelas <i>Rating</i> Setelah ROS .....	58
14. Sebaran Kelas <i>Textblob</i> Setelah ROS .....	58
15. Proses BERT <i>Embedding</i> .....	59
16. Hasil Pembagian Data Berdasarkan <i>Rating</i> .....	61
17. Hasil Pembagian Data Berdasarkan <i>Textblob</i> .....	61
18. Kombinasi Parameter BERT .....	62
19. Hasil Klasifikasi BERT .....	69

20. Hasil Klasifikasi <i>Rating</i> Pada Data <i>Testing</i> .....	70
21. Hasil Klasifikasi <i>Textblob</i> Pada Data <i>Testing</i> .....	71
22. <i>Benchmarking</i> Hasil Penelitian .....	73

## DAFTAR GAMBAR

Gambar	Halaman
1. Ilustrasi Jaringan Saraf Tiruan (Aggarwal, 2018).....	17
2. Fungsi Aktivasi <i>Sigmoid</i> (Sharma dkk., 2020) .....	19
3. Fungsi Aktivasi <i>ReLU</i> (Sharma dkk., 2020) .....	20
4. Fungsi Aktivasi <i>GELU</i> (Lee, 2023) .....	21
5. Arsitektur Model <i>Transformer</i> (Vaswani dkk., 2017) .....	25
6. <i>Scaled Dot-Product Attention</i> (Vaswani dkk., 2017).....	27
7. Arsitektur BERT, Open AI GPT, dan ELMo (Devlin dkk., 2019) .....	29
8. Representasi <i>Input</i> BERT (Devlin dkk., 2019) .....	30
9. <i>Framework</i> BERT (Devlin dkk., 2019) .....	31
10. <i>Fine-tuning</i> BERT (Devlin dkk., 2019) .....	34
11. <i>Confusion Matrix Multiclass</i> (Markoulidakis dkk., 2021).....	35
12. <i>Flowchart</i> Metode Penelitian .....	43
13. Data yang Telah Digabung.....	45
14. Data yang Telah Diseleksi Variabel.....	46
15. Data yang Telah Dilakukan Pelabelan .....	48
16. <i>Wordcloud</i> Kata Pada Kelas <i>Rating</i> 1 .....	49
17. <i>Wordcloud</i> Kata Pada Kelas <i>Rating</i> 2 .....	50
18. <i>Wordcloud</i> Kata Pada Kelas <i>Rating</i> 3 .....	50
19. <i>Wordcloud</i> Kata Pada Kelas <i>Rating</i> 4.....	51

20. <i>Wordcloud</i> Kata Pada Kelas <i>Rating</i> 5 .....	52
21. <i>Wordcloud</i> Kata Pada Kelas Sentimen Negatif.....	52
22. <i>Wordcloud</i> Kata Pada Kelas Sentimen Netral.....	53
23. <i>Wordcloud</i> Kata Pada Kelas Sentimen Positif .....	54
24. Data yang telah Diterapkan <i>Label Encoding</i> .....	54
25. Piechart Sebaran Kelas <i>Rating</i> .....	55
26. Piechart Sebaran Kelas <i>Textblob</i> .....	56
27. Histogram Sebaran Kelas <i>Rating</i> Sebelum dan Setelah ROS .....	57
28. Histogram Sebaran Kelas <i>Textblob</i> Sebelum dan Setelah ROS.....	58
29. Histogram Panjang Kata Data Ulasan .....	59
30. <i>Training</i> dan <i>Validation Loss</i> Berdasarkan <i>Rating</i> .....	63
31. <i>Training</i> dan <i>Validation Accuracy</i> Berdasarkan <i>Rating</i> .....	63
32. <i>Training</i> dan <i>Validation Loss</i> Berdasarkan <i>Textblob</i> .....	64
33. <i>Training</i> dan <i>Validation Accuracy</i> Berdasarkan <i>Textblob</i> .....	64
34. <i>Confusion Matrix Rating</i> .....	65
35. <i>Confusion Matrix Textblob</i> .....	67
36. Histogram <i>Benchmarking</i> Hasil Penelitian .....	72

## I. PENDAHULUAN

### 1.1 Latar Belakang dan Masalah

Pada saat ini produk-produk kesehatan seperti obat-obatan telah menjadi lebih mudah diakses, terutama obat-obatan standar yang dijual secara bebas di apotek dan platform daring, yaitu *e-commerce*. Hal ini membuat individu yang awam dalam bidang kesehatan sering mencari rekomendasi obat untuk kondisi kesehatan yang mereka alami, dan salah satu sumber rekomendasi tersebut adalah ulasan dari individu lain yang telah menggunakan obat tersebut untuk diri mereka sendiri (Ariyulinda, 2018). Melalui ulasan dari pengguna lain, individu dapat memperoleh pemahaman lebih baik tentang jenis obat untuk menangani penyakit yang sedang mereka hadapi melalui penilaian obat tersebut sehingga mereka dapat memilih obat yang sesuai. Kemudian penilaian obat tersebut juga dapat membuat tenaga kesehatan, farmasi dan pihak terkait untuk meningkatkan kualitas obat yang diproduksi.

Analisis sentimen merupakan salah satu bidang penelitian dalam komputasi yang berfokus pada pemeriksaan opini, perasaan, dan emosi dalam teks ulasan. Bidang pemrosesan bahasa alami dan penambangan data, salah satunya analisis sentimen. Tujuan utama dari analisis sentimen adalah untuk mengelola, mengekstrak, merangkum, dan menganalisis informasi dalam teks dengan menggunakan berbagai metode sehingga dapat menyimpulkan informasi dari teks, serta berbagi informasi subjektif tentang kecenderungan emosional dalam teks tersebut (Kurniawan dkk., 2022).

Klasifikasi teks telah menjadi fokus utama dalam penelitian di bidang pemrosesan bahasa alami. Hal ini diakibatkan melonjaknya jumlah posting pengguna di berbagai platform media sosial. Salah satu cabang dari klasifikasi teks adalah analisis sentimen yang bertujuan secara otomatis mengkategorikan emosi atau opini yang terkandung dalam data teks menggunakan metode statistik dan matematika. Tantangan utama yang dihadapi dalam pemrosesan data teks adalah sifat data tidak terstruktur. Data tekstual tidak mengikuti skema atau model tertentu, sehingga memerlukan pendekatan yang inovatif untuk dapat diolah dan dipahami secara efektif (Alwehaibi dkk., 2022).

*Natural Language Processing* (NLP) atau pemrosesan bahasa alami adalah salah satu cabang ilmu di bidang kecerdasan buatan yang memiliki fokus pada pemrosesan bahasa alami. Bahasa alami adalah bentuk komunikasi yang umumnya digunakan oleh manusia ketika berinteraksi satu sama lain (Oyong dkk., 2018). Komputer tidak memiliki kemampuan langsung untuk memahami makna dari kalimat yang diucapkan oleh manusia. Konteks mengklasifikasikan kelas sentimen, komputer belajar dari ulasan responden untuk membentuk model yang dapat mengklasifikasikan ulasan tersebut. Komputer tidak memiliki pemahaman bahasa alami seperti manusia, maka penting untuk menggunakan metode NLP yang dapat mengklasifikasikan ulasan dengan tingkat akurasi yang tinggi. Hal ini bertujuan untuk mencegah kesalahan penafsiran yang dapat berdampak pada pengambilan kebijakan yang salah dalam perusahaan (Pasaribu dkk., 2020). *Feature engineering* merupakan salah satu tahap dalam *preprocessing data* dalam *machine learning* yang bertujuan untuk meningkatkan kinerja model (Nargesian dkk., 2017). Pada saat melakukan *feature engineering* pada data teks dalam *machine learning*, terdapat kendala khusus karena data teks bersifat tidak terstruktur (*unstructured data*).

Pada beberapa tahun terakhir, model BERT telah menjadi model representasi yang luas digunakan dan efisien, mencapai tingkat kinerja terbaru dalam tugas-tugas tingkat kalimat dan *token-level*, mengungguli banyak arsitektur yang dibuat khusus untuk tugas-tugas tersebut. Salah satu representasi dari *transformer*, yaitu BERT

telah diajukan dalam beberapa tahun terakhir untuk mengembangkan berbagai model canggih dalam berbagai tugas NLP seperti pertanyaan dan pemahaman teks alami. Model BERT diciptakan dengan tujuan memberikan representasi mendalam dua arah pada semua lapisan dengan memperhatikan konteks sebelah kiri dan kanan. Prinsip kerja BERT adalah dua tahap, pertama-tama ia dilatih pada teks yang tidak berlabel dalam jumlah besar. Lalu disesuaikan kembali (*fine-tuning*) untuk tugas tertentu dengan menggunakan data yang berlabel (Paul dan Saha, 2022).

Adapun penelitian sebelumnya yang pernah dilakukan terkait analisis sentimen terhadap pelayanan kesehatan berdasarkan ulasan Google Maps menggunakan BERT. Data ulasan yang digunakan pada penelitian kali ini diperoleh dengan cara *scraping* data ulasan penggunaan Google Maps dengan *selenium* dan *beautifulsoup*. Data berjumlah 4299 ulasan dimana 2527 ulasan positif, 239 ulasan netral, dan 1533 ulasan negatif. Penelitian ini memperoleh metrik evaluasi terbaik dengan *accuracy* yang didapat sebesar 85%, *precision* sebesar 77%, *recall* sebesar 74%, dan *F1-score* sebesar 75% (Ardiansyah dkk., 2023).

Penelitian kedua yaitu implementasi *fine-tuning* BERT untuk analisis sentimen terhadap *review* aplikasi *PUBG Mobile* di Google Play Store. Data diberi label berdasarkan *rating* yang telah diperkecil skalanya dan *textblob*. Data berdasarkan *rating* berjumlah 14349 pengamatan dan data berdasarkan *textblob* berjumlah 14341. Hasil klasifikasi berdasarkan *rating* tidak disertakan karena hasil yang kurang baik maka dari itu hanya disajikan hasil klasifikasi berdasarkan *textblob* yang mendapatkan *accuracy* sebesar 94%, *precision* sebesar 94%, *recall* sebesar 93%, dan *F1-score* sebesar 93% (Braja dan Kodar, 2023).

Penelitian ketiga yaitu analisis sentimen *customer review* aplikasi Ruang Guru dengan metode BERT (*Bidirectional Encoder Representations from Transformer*). Data dibagi menjadi 3 kelas, yaitu data positif sebanyak 5254, data negatif sebanyak 167, dan data netral sebanyak 16. Hasilnya diperoleh nilai *accuracy* sebesar 99%,

*precision* sebesar 64,13%, *recall* 60,51% dan *F1-score* sebesar 98,9% (Kusuma dan Yustanti, 2021).

Penelitian keempat yaitu klasifikasi data teks yaitu pendekatan metode *transformer* untuk deteksi bahasa kasar dalam komentar berita *online* di Indonesia. Data tersebut memiliki 3 kelas, yaitu kelas non *offensive* sejumlah 110, normal sejumlah 2789, dan *offensive* sejumlah 285. Hasil dari penelitian ini diperoleh nilai *accuracy* sebesar 54% (Rendragraha dkk., 2021).

Berdasarkan beberapa penelitian di atas terdapat literatur tentang klasifikasi sentimen menggunakan BERT. Hal itu menjadi motivasi untuk melakukan penelitian mengenai klasifikasi sentimen menggunakan BERT berdasarkan sentimen *rating* yang diperkecil skalanya dan *textblob* pada data besar yang berisi ulasan obat. Oleh karena itu penelitian ini akan membahas “Klasifikasi Sentimen Ulasan Penilaian Obat Dengan Metode *Bidirectional Encoder Representations From Transformer* (BERT)”.

## 1.2 Rumusan Masalah

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

1. Bagaimana pemodelan klasifikasi sentimen dengan menggunakan *Bidirectional Encoder Representations from Transformer* (BERT) dalam konteks klasifikasi sentimen ulasan obat berdasarkan *rating* dan *textblob*?
2. Bagaimana performa kinerja model BERT dalam konteks klasifikasi sentimen ulasan obat berdasarkan *rating* dan *textblob*?



### 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun model klasifikasi sentimen dengan *Bidirectional Encoder Representations from Transformer* (BERT) pada klasifikasi sentimen ulasan obat berdasarkan *rating* dan *textblob*.
2. Mengetahui performa kinerja *Bidirectional Encoder Representations from Transformer* (BERT) pada klasifikasi sentimen ulasan obat berdasarkan *rating* dan *textblob* dengan melihat nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

### 1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Membantu dalam melakukan pemodelan klasifikasi sentimen ulasan obat menggunakan BERT berdasarkan *rating* dan *textblob*.
2. Memberikan informasi mengenai kinerja model klasifikasi BERT dalam konteks klasifikasi sentimen ulasan obat berdasarkan *rating* dan *textblob*.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Penelitian yang berkaitan dengan penelitian ini digunakan sebagai bahan acuan dan perbandingan untuk hasil klasifikasi. Topik penelitian yang menjadi acuan adalah klasifikasi sentimen dengan metode BERT. Secara umum, gambaran mengenai beberapa riset yang digunakan dalam penelitian ini terangkum dalam Tabel 1. Tabel 1 berisi penelitian terdahulu, informasi data, metode yang digunakan, dan hasil penelitian terdahulu yang dapat dilihat dari metrik evaluasi seperti *accuracy*(*Acc*), *precision*(*Prec*), *recall*(*Rec*), dan *F1-score*(*F1*).

Tabel 1. Penelitian Terkait Klasifikasi Sentimen dengan BERT

No	Penelitian	Data	Metode	Hasil			
				<i>Acc</i>	<i>Prec</i>	<i>Rec</i>	<i>F1</i>
1.	Analisis Sentimen Terhadap Pelayanan Kesehatan Berdasarkan Google Maps Menggunakan BERT (Ardiansyah dkk., 2023).	Data Ulasan: Positif: 2527 Netral: 239 Negatif: 1533 Total: 4299  Sumber Data: <i>Scraping</i> Data Google Maps	Metode Klasifikasi: IndoBERT BASE	Sentimen 85%	Sentimen 77%	Sentimen 74%	Sentimen 85%

2.	Implementasi <i>Fine-tuning</i> BERT untuk Analisis Sentimen terhadap Review Aplikasi <i>PUBG Mobile</i> di Google Play Store (Braja dan Kodar, 2023).	Data Ulasan: Berdasarkan <i>Rating</i> : Negatif: 4874 Netral: 4849 Positif: 4626 Total: 14349  Berdasarkan <i>Textblob</i> : Negatif: 4360 Netral: 4167 Positif: 5814 Total: 14341  Sumber Data: <i>Scraping</i> Data Google Play Store	Metode Klasifikasi: BERT BASE <i>Multilingual</i> dan IndoBERT BASE P2	<i>Rating</i> 71%  <i>Textblob</i> 94%	<i>Rating</i> -  <i>Textblob</i> 94%	<i>Rating</i> -  <i>Textblob</i> 93%	<i>Rating</i> -  <i>Textblob</i> 93%
3.	Analisis Sentimen <i>Customer Review</i> Aplikasi Ruang Guru dengan Metode BERT ( <i>Bidirectional Encoder Representations from Transformer</i> ) (Kusuma dan Yustanti, 2021).	Data Ulasan: Positif: 5254 Negatif: 167 Netral: 16 Total: 5437  Sumber Data: <i>Scraping</i> Data Google Play Store	Metode Klasifikasi: BERT BASE	Sentimen 99%	Sentimen 64,13%	Sentimen 60,51%	Sentimen 98,9%
4.	Pendekatan Metode <i>Transformer</i> untuk Deteksi Bahasa Kasar dalam Komentar Berita <i>Online</i> Indonesia (Rendragraha dkk., 2021).	Data Ulasan: Non <i>Offensive</i> : 110 Normal: 2789 <i>Offensive</i> : 285 Total: 3184  Sumber Data: <i>Scraping</i> Data Berita Kompas dan Tempo	Metode Klasifikasi: BERT BASE <i>Multilingual</i>	Sentimen 54%	-	-	-

Berikut adalah *resume* penelitian pada Tabel 1:

### 2.1.1 Penelitian Pertama (Ardiansyah dkk., 2023)

Ardiansyah dkk. (2023) melakukan penelitian klasifikasi sentimen terhadap pelayanan kesehatan berdasarkan ulasan Google MAPS menggunakan BERT. Data ulasan yang digunakan pada penelitian kali ini diperoleh dengan cara *scraping* data ulasan penggunaan Google Maps dengan *selenium* dan *beautifulsoup*. Data berjumlah 4299 ulasan dimana 2527 ulasan positif, 239 ulasan netral, dan 1533 ulasan negatif.

Data ulasan dilakukan terlebih dahulu *preprocessing data* seperti *case folding*, *cleaning*, dan *stopword removal* sebelum dibangun model BERT. Setelah itu, *dataset* dibagi dengan rasio 70%:30%:30% berdasarkan data *training*, data validasi, dan data *testing*. Kemudian dilakukan BERT *embedding* seperti menambahkan token khusus di awal [CLS] dan di akhir [SEP]. Selanjutnya menambahkan token *padding* [PAD] untuk memotong *review* menjadi satu panjang konstan. Panjang token BERT pada penelitian ini ditentukan 512 token untuk semua *review*.

Model yang digunakan adalah model *pre-trained* BERT dengan RoBERTa. Model *fine-tuning* pada penelitian ini ditentukan sebagai klasifikasi *BertForSequenceClassification* dan menggunakan IndoBERT BASE. Selain itu, parameter seperti *optimizer* yang digunakan adalah Adam, nilai *learning rate* sebesar  $3 \times 10^{-6}$ , nilai *batch size* 32, nilai *dropout* sebesar 0,3, dan dilatih sebanyak 5 *epoch*. Penelitian ini memperoleh metrik evaluasi terbaik dengan *accuracy* yang didapat sebesar 85%, *precision* sebesar 77%, *recall* sebesar 74%, dan *F1-score* sebesar 75%.

### 2.1.2 Penelitian Kedua (Braja dan Kodra, 2023)

Braja dan Kodra (2023) melakukan penelitian tentang implementasi *fine-tuning* BERT untuk analisis sentimen terhadap *review* aplikasi PUBG *Mobile* di Google Play Store. Data ulasan diperoleh dengan cara *scraping* di Google Play Store. Data berdasarkan *rating* berjumlah 14349 pengamatan dan data berdasarkan *textblob* berjumlah 14341. Data diberi label berdasarkan *rating* yang diperkecil skalanya dan *textblob*. Proses pelabelan data berdasarkan *rating* dilakukan dengan asumsi *rating* 1-2 adalah negatif, *rating* 3 adalah netral, dan *rating* 4-5 adalah positif. Lalu proses pelabelan data berdasarkan *textblob* dilakukan dengan menentukan nilai *polarity* dari *review* mulai dari -1 sampai 1. Nilai yang kurang dari 0 adalah negatif, nilai yang sama dengan 0 adalah netral, dan nilai yang lebih dari 1 adalah positif.

Data dibagi menjadi 80% data *training*, 10% data validasi, dan 10% data *testing*. Selanjutnya melakukan BERT *embedding* seperti menambahkan token khusus di awal [CLS] dan di akhir [SEP]. Kemudian menambahkan token *padding* [PAD] untuk memotong *review* menjadi satu panjang konstan. Panjang maksimum yang ditentukan pada saat BERT *embedding* adalah 80. Model *fine-tuning* pada penelitian ini ditentukan sebagai klasifikasi *BertForSequenceClassification* dan menggunakan IndoBERT BASE P2 dan BERT BASE *Multilingual*. Parameter pada dua model tersebut menggunakan *optimizer* Adam, nilai *learning rate* sebesar 0,00002, nilai *batch size* sebesar 16 dan 32, jumlah *epoch* sebanyak 5, dan nilai *dropout* sebesar 0,3. Hasil dari penelitian ini hanya menunjukkan hasil klasifikasi berdasarkan *textblob* dikarenakan hasil klasifikasi berdasarkan *rating* tidak mendapatkan hasil yang cukup baik. Evaluasi terhadap parameter terunggul dalam penelitian ini dilakukan pada model klasifikasi yang dirancang menggunakan *BertForSequenceClassification*, mengadopsi IndoBERT BASE P2, dan mengatur nilai *batch size* sebesar 32 yang berhasil

mencapai metrik evaluasi dengan *accuracy* yang didapat sebesar 94%, *precision* sebesar 94%, *recall* sebesar 93%, dan *F1-score* sebesar 93%.

### 2.1.3 Penelitian Ketiga (Kusuma dan Yustanti, 2021)

Kusuma dan Yustanti (2021) melakukan penelitian tentang analisis sentimen *customer review* aplikasi Ruang Guru dengan metode BERT. Data ulasan diperoleh dengan cara *scraping* Google Play Store. Data ada 5437 pengamatan. Data dibagi menjadi 3 kelas, diantaranya data positif sebanyak 5254, data negatif sebanyak 167, dan data netral sebanyak 16. Data dilakukan *preprocessing* seperti *case folding* dan *cleansing*. Lalu data diberikan label 0 untuk data negatif, label 1 untuk data positif, dan label 2 untuk data netral. Kemudian diterapkan 3 skema pembagian data *training* dan *testing*, yaitu skema 1 adalah pembagian data *training* sebesar 90% dan data *testing* sebesar 10%, skema 2 adalah pembagian data *training* sebesar 50% dan data *testing* sebesar 50%, dan skema 3 adalah pembagian data *training* sebesar 70% dan data *testing* sebesar 30%. Setelah dilakukan pembagian data *training* dan *testing*, selanjutnya dilakukan BERT *embedding* dengan BERT BASE seperti menambahkan token khusus di awal [CLS] dan di akhir [SEP]. Kemudian menambahkan token *padding* [PAD] untuk memotong *review* menjadi satu panjang konstan.

Model *fine-tuning* pada penelitian ini ditentukan sebagai klasifikasi *BertForSequenceClassification* dan menggunakan BERT BASE *Multilingual*. Kemudian parameter yang ditetapkan oleh peneliti adalah nilai *batch size* sebesar 32, *learning rate*  $1 \times 10^{-5}$ , *epsilon*  $1 \times 10^{-5}$ , dan optimasi Adam. Hasil dari ketiga skema pembagian data *training* dan *testing* untuk analisis sentimen tersebut diperoleh hasil optimal pada penggunaan algoritma BERT dengan metrik evaluasi saat pembagian data *training* sebesar 70% dan *testing* sebesar 30%. Hasilnya dapat dilihat dari metrik

evaluasi seperti nilai *accuracy* sebesar 99%, *precision* sebesar 64,13%, *recall* 60,51% dan *F1-score* sebesar 98,9%.

#### 2.1.4 Penelitian Keempat (Rendragraha dkk., 2021)

Rendragraha dkk. (2021) menguji kinerja BERT untuk deteksi bahasa kasar dalam komentar berita *online* di Indonesia. Data ulasan yang digunakan diperoleh dengan cara *scraping* data komentar yang terdapat pada portal berita *online* Kompas dan Tempo dengan sebuah *scraper web*. Data yang diperoleh sejumlah 3184 komentar. Data tersebut memiliki 3 kelas, yaitu kelas non *offensive* sejumlah 110, normal sejumlah 2789, dan *offensive* sejumlah 285.

Alur penelitian ini dilakukan pengambilan data pada portal berita *online* Kompas dan Tempo dengan sebuah *scraper web*. Setelah mendapatkan kumpulan data, selanjutnya dilakukan *random undersampling* dan *random oversampling* untuk menangani masalah *imbalance data*. Kemudian dilakukan BERT *embedding* seperti menambahkan token khusus di awal [CLS] dan di akhir [SEP]. Kemudian menambahkan token *padding* [PAD] untuk memotong komentar menjadi satu panjang konstan. Untuk panjang token BERT pada penelitian ini ditentukan 250 token untuk semua komentar. Selanjutnya akan dibagi kumpulan data dengan rasio 70%:30% berdasarkan *data training* dan *data testing*. Model *fine-tuning* pada penelitian ini ditentukan sebagai klasifikasi *BertForSequenceClassification* dan menggunakan BERT BASE *Multilingual*. Parameter yang digunakan seperti nilai *learning rate* ditentukan sebesar  $4 \times 10^{-5}$ , nilai *batch size* 6, dan dilatih sebanyak 10 *epoch*. Hasil terbaik dari model yang dibangun pada penelitian ini dapat dilihat dari metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Penelitian ini memperoleh metrik evaluasi terbaik *accuracy* yang didapat sebesar 54%.

## 2.2 Analisis Sentimen

Analisis sentimen atau penambangan opini adalah bentuk aplikasi dari *text mining* yang digunakan untuk mengidentifikasi opini dalam sekelompok data teks terkait suatu peristiwa atau topik tertentu (Li dkk., 2019). Analisis sentimen menjadi aspek kunci dalam pengembangan kecerdasan buatan dari berbagai penelitian yang berfokus pada ekstraksi informasi dari data teks (Cambria dkk., 2017).

Menurut Persi (2019), analisis sentimen atau penambangan opini adalah proses memberikan label opini atau emosional pada teks. Label dapat menunjukkan polaritas, apakah teks tersebut menyatakan opini positif atau negatif. Bentuk paling umum dari analisis sentimen mengasumsikan bahwa teks dokumen menyajikan evaluasi suatu entitas, seperti opini pelanggan yang diungkapkan dalam ulasan produk. Secara kontekstual, analisis sentimen pada dasarnya adalah pengembangan klasifikasi yang memprediksi polaritas dari teks.

Pendekatan untuk analisis sentimen bergantung pada kamus atau kosakata yang memberikan label positif atau negatif pada daftar kata yang telah ditentukan (Persi, 2019). Sebaran kata-kata berlabel ini menentukan sentimen yang diberikan pada suatu dokumen. Sebagai contoh, pertimbangkan ulasan berikut tentang sebuah mesin pembuat kopi yang diposting secara *online*. Ulasan ini mudah diinterpretasikan baik oleh manusia maupun analisis sentimen dasar.

## 2.3 *Natural Language Processing*

*Natural language processing* (NLP) adalah cabang kecerdasan buatan yang berfokus pada interaksi antara manusia dan komputer menggunakan bahasa alami. Tujuan utama dari NLP adalah memungkinkan komunikasi yang lebih mudah dan efektif antara manusia dan komputer, terutama dalam mencari informasi



(Nurhayatunnufus dkk., 2020). Menurut Eisenstein (2018), pemrosesan bahasa alami telah tertanam dalam kehidupan sehari-hari seperti terjemahan mesin otomatis pada *web* dan media sosial, klasifikasi teks untuk menjaga kotak masuk *email* tidak runtuh karena banjir *spam*, mesin pencari beralih dari sekadar pencocokan *string* dan analisis jaringan ke tingkat kecanggihan linguistik yang tinggi, sistem dialog menyediakan cara yang semakin umum dan efektif untuk mendapatkan berbagai informasi dalam beberapa dekade terakhir. Aplikasi yang beragam ini didasarkan pada kumpulan ide yang sama berdasarkan algoritma, linguistik, logika, statistik, dan lain sebagainya.

## 2.4 *Text Mining*

*Text mining* merupakan salah satu bidang penelitian yang sering dianggap sulit. Menyusun informasi yang diperlukan oleh pengguna merupakan tantangan tersendiri. Proses *text mining* memainkan peran penting dalam proses penemuan pengetahuan. Kegiatan ini bertujuan untuk mengekstrak informasi tersembunyi dari data tidak terstruktur hingga semi terstruktur. *Text mining* dapat dianggap sebagai upaya otomatisasi untuk menggali informasi dari berbagai sumber tulisan, sehingga memungkinkan komputer untuk menemukan informasi baru yang sebelumnya tidak diketahui (Gaikwad dkk., 2014). *Text mining* memerlukan pendekatan yang berbeda dibandingkan dengan analisis data numerik biasa dan satu hal yang perlu dilakukan adalah *preprocessing text*.

Menurut Surjandari dkk. (2016), tahap *preprocessing text* dilakukan untuk mengubah data tekstual yang tidak terstruktur menjadi terstruktur. Ada beberapa langkah yang dilakukan dalam preprocessing teks sebagai berikut:

1. Tokenisasi

Tokenisasi adalah langkah dalam menguraikan dokumen menjadi bagian-bagian kata yang disebut sebagai token.

## 2. *Case Folding*

*Case Folding* adalah proses mengubah seluruh huruf besar menjadi huruf kecil pada suatu dokumen dan sebaliknya. Langkah ini juga untuk memastikan hanya huruf a sampai z yang terdapat dalam dokumen.

## 3. *Filtering*

*Filtering* merupakan proses menghilangkan *mention* (@), *hashtag* (#), url, tanda baca, emotikon, dan karakter non abjad.

## 4. *Padding*

*Padding* merupakan proses mengisi barisanan dengan angka nol di bagian awal atau akhir kalimat agar sesuai dengan panjang barisan (Prusa dan Khoshgoftaar, 2017).

## 2.5 *Imbalance Data*

Isu utama *imbalance data* atau ketidakseimbangan data adalah bahwa kelas mayoritas yang memiliki jumlah pola yang besar dapat memengaruhi batasan keputusan klasifikasi, yaitu dengan merugikan kelas minoritas yang hanya memiliki sejumlah pola kecil (Mahmood, 2015). Hal ini berdampak pada akurasi yang tinggi untuk klasifikasi kelas mayoritas dan rendah untuk kelas minoritas. Salah satu solusi umum untuk mengatasi masalah ini adalah dengan mencari keseimbangan jumlah pola di kedua kelas, baik mayoritas maupun minoritas.

Keseimbangan data merupakan suatu metode untuk menyeimbangkan jumlah data antara kelompok mayoritas dan kelompok minoritas dalam kumpulan data. Untuk mengatasi ketidakseimbangan tersebut, sering kali digunakan teknik *resampling*. Menurut Mahmood (2015), terdapat tiga teknik *resampling* utama, yaitu *oversampling*, *undersampling*, dan pendekatan *hybrid* antara *oversampling* dan *undersampling*. Teknik *oversampling* diterapkan dengan menambah jumlah data pada kelompok minoritas hingga mencapai keseimbangan jumlah data dengan kelompok mayoritas. Sebaliknya, teknik *undersampling* digunakan untuk

mengurangi jumlah data pada kelompok mayoritas yang dianggap kurang relevan sehingga mencapai keseimbangan jumlah data antara kelompok mayoritas dan minoritas. Teknik *hybrid oversampling* dan *undersampling* merupakan pendekatan yang menggabungkan kedua teknik *oversampling* dan *undersampling*, disesuaikan dengan kebutuhan dan karakteristik khusus dari kumpulan data yang dimiliki.

## 2.6 Word Embedding

Pada bidang pemrosesan teks atau NLP, pengembangan teknik untuk mengubah kata-kata menjadi vektor merupakan topik penelitian yang terus berkembang. Representasi kata ini memiliki dampak yang sangat penting terhadap performa dan akurasi model pembelajaran yang dibangun (Nurdin dkk., 2020). Proses pengembangan fitur dalam data teks memiliki tantangan tersendiri karena teks bersifat tidak terstruktur.

Menurut Nurdin dkk. (2020) *word embedding* akan memetakan setiap kata dalam dokumen ke dalam *dense* vektor, dimana sebuah vektor merepresentasikan proyeksi kata di dalam ruang vektor. Posisi kata tersebut dipelajari dari teks atau berdasarkan kata-kata di sekitarnya. *Word embedding* ini dapat menangkap makna semantik dan sintaktik kata. *Word embedding* dapat dibuat langsung dari kumpulan data yang dimiliki atau menggunakan *pre-trained word embedding* yang telah tersedia. *Pre-trained word embedding* ini adalah *word embedding* yang telah dilatih menggunakan kumpulan data yang besar pada domain permasalahan tertentu yang dapat digunakan untuk menyelesaikan permasalahan lain yang serupa. Penggunaan *word embedding* ini harus disesuaikan dengan data yang dimiliki (Nurdin dkk., 2020).

Pada tahun 2018, tim peneliti dari Google AI Language memperkenalkan konsep *Bidirectional Encoder Representations from Transformer* (BERT). Model BERT merupakan salah satu pendekatan model yang berbasis pada *transformer* yang

merupakan suatu mekanisme untuk memahami hubungan kontekstual antar kata-kata dalam teks melalui *self-attention mechanism* (Devlin dkk., 2019). Pada *self-attention mechanism*, setiap kata dalam teks dapat saling berinteraksi dan menentukan seberapa besar perhatian yang harus diberikan kepada kata-kata lainnya. Representasi urutan kata dalam suatu kalimat dihasilkan dengan mengaitkan kata-kata yang berbeda dalam urutan yang sama menggunakan mekanisme *encoder* dan *decoder* (Vaswani dkk., 2017). Hal ini memungkinkan pemahaman yang lebih baik tentang konteks dan hubungan antara kata-kata dalam teks.

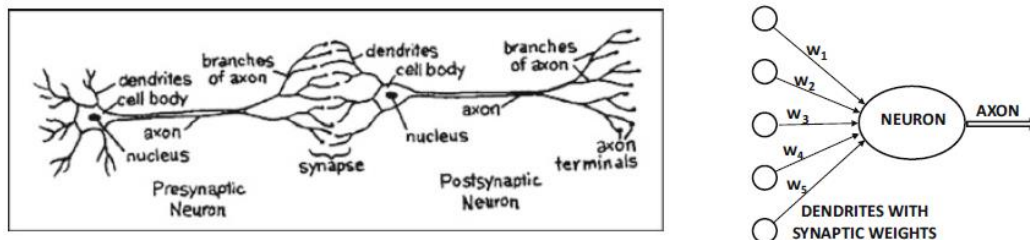
## 2.7 *Splitting Data*

*Splitting* data adalah salah satu proses yang diperlukan untuk menghilangkan atau mengurangi bias pada data *training* dalam model *machine learning*. *Splitting* data dilakukan untuk mencegah algoritma *machine learning* menghasilkan *overfitting* yang dapat berperforma buruk pada data *testing* sebenarnya (Muraina, 2022). Secara umum, kumpulan data akan dibagi menjadi beberapa bagian data lalu dilatih dengan parameter yang berbeda.

Menurut Muraina (2022), Pendekatan tradisional untuk membangun model prediktif yang baik adalah dengan membagi data menjadi tiga bagian: data *training* (untuk penyesuaian model), data validasi (untuk pemilihan model), dan data *testing* (untuk penilaian model akhir). Data *testing* digunakan pada model untuk menentukan akurasi melalui tingkat kesalahannya. Tingkat kesalahan merupakan rasio antara sampel yang diprediksi dengan benar berdasarkan jumlah total sampel dalam kumpulan data pengujian.

## 2.8 Artificial Neural Network

*Artificial Neural Network* (ANN) atau Jaringan Saraf Tiruan (JST) merupakan salah satu metode pembelajaran mesin yang meniru proses pembelajaran yang terjadi dalam organisme biologis (Aggarwal, 2018). Manusia memiliki sistem saraf yang terdiri dari sel-sel yang dikenal sebagai neuron. Neuron ini saling terhubung melalui struktur yang disebut akson dan dendrit. Titik temu antara akson dan dendrit disebut sinapsis, seperti yang diilustrasikan pada Gambar 1. Kekuatan koneksi sinaptik ini seringkali berubah sebagai respons terhadap rangsangan eksternal dan perubahan ini merupakan cara pembelajaran yang terjadi pada organisme yang hidup.



Gambar 1. Ilustrasi Jaringan Saraf Tiruan (Aggarwal, 2018).

Jaringan Saraf Tiruan (JST) adalah model komputasi yang terinspirasi dari mekanisme biologis otak manusia dengan menggunakan unit komputasi yang disebut neuron dan koneksi bobot yang mirip sinaptik (Aggarwal, 2018). Proses perhitungan *input* ke neuron dipengaruhi oleh bobot dan pembelajaran, yaitu dengan memperbarui bobot menggunakan data pelatihan yang mencakup contoh *input-output*. Rangsangan eksternal dari data pelatihan digunakan untuk menyesuaikan bobot dengan tujuan meningkatkan akurasi prediksi. Modifikasi bobot bertujuan untuk mengubah nilai agar menghasilkan prediksi yang lebih tepat pada iterasi berikutnya. Lapisan dalam JST melibatkan sejumlah neuron yang terorganisir dalam struktur tertentu. Neuron-neuron ini terhubung melalui koneksi bobot, mewakili kekuatan sinaptik antar-neuron. Setiap lapisan memiliki peran khusus.

Berikut adalah lapisan yang terdapat pada JST:

1. Lapisan *Input*

Lapisan *input* merupakan tahap awal pada JST yang mengirimkan data ke apisan berikutnya, yaitu lapisan tersembunyi (Aggarwal, 2018).

2. Lapisan Tersembunyi

Peran lapisan tersembunyi memiliki signifikansi penting dalam meningkatkan performa jaringan saraf, terutama dalam menangani tantangan masalah yang kompleks dengan kendala utama pada akurasi dan kompleksitas waktu (Uzair dan Jamil, 2020). Proses penentuan jumlah lapisan tersembunyi dan jumlah neuron pada setiap lapisan beragam. Berdasarkan pengamatan empiris, terungkap bahwa pengurangan jumlah lapisan tersembunyi secara langsung berdampak pada akurasi prediksi, khususnya ketika dihadapkan pada masalah yang kompleks. Sebaliknya, jika jumlah lapisan tersembunyi melebihi jumlah optimal maka kompleksitas waktu meningkat secara signifikan tanpa meningkatkan akurasi prediksi secara proporsional. Oleh karena itu, ketika memilih jumlah lapisan tersembunyi dan neuron perlu dipertimbangkan keseimbangan antara akurasi prediksi dan kompleksitas waktu guna mencapai kinerja yang optimal.

3. Lapisan *Output*

Lapisan *output* adalah lapisan terakhir dalam jaringan yang berhubungan langsung dengan tugas atau masalah yang ditangani dan bertindak sebagai representasi *output* dari *input* yang diberikan kepada jaringan sesuai dengan tujuan tugas (Uzair dan Jamil, 2020). Jumlah neuron pada lapisan ini umumnya disesuaikan dengan jenis tugas yang sedang dijalankan seperti klasifikasi, prediksi, atau tugas lainnya.

## 2.9 Fungsi Aktivasi

Jaringan Saraf Tiruan mengambil inspirasi dari struktur dan fungsi jaringan neuron di otak manusia (Aggarwal, 2018). Hal ini sama seperti informasi yang diproses

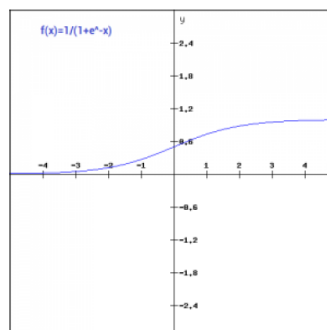
dan diteruskan melalui sambungan *neuro* sinaptik dari satu neuron ke neuron lainnya dalam otak. JST memiliki lapisan sel yang saling terhubung. Proses *output* atau informasi dari lapisan tersembunyi JST mengalir ke lapisan berikutnya, dan akhirnya mencapai lapisan *output* yang menghasilkan nilai *output*. Oleh karena itu, untuk memfasilitasi proses ini nilai *input* ke lapisan tersembunyi mengalami non *linearitas* seiring dengan nilai *output* dari lapisan tersembunyi. Fungsi aktivasi menjadi sangat penting dalam JST karena membantu jaringan untuk memahami dan menangani pemetaan yang bersifat non linier dan kompleks antara *input* dan *output* (Sharma dkk., 2020).

Fungsi aktivasi memiliki peran khusus dalam konteks jaringan saraf tiruan, yaitu berfungsi untuk mengubah sinyal *input* menjadi sinyal *output* yang selanjutnya digunakan sebagai *input* untuk lapisan berikutnya dalam jaringan (Sharma dkk., 2020). Keakuratan prediksi dari jaringan saraf tiruan sangat bergantung pada pengaturan jumlah lapisan yang digunakan dan jenis fungsi aktivasi yang dipilih untuk digunakan. Berikut adalah beberapa fungsi aktivasi yang biasa digunakan:

#### 1. Fungsi Aktivasi *Sigmoid*

Fungsi aktivasi ini seringkali digunakan karena sifatnya yang non linier. Fungsi *sigmoid* merubah nilai ke dalam rentang antara 0 hingga 1. Fungsi aktivasi *sigmoid* dapat didefinisikan pada Persamaan (2.1) sebagai berikut (Sharma dkk., 2020):

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$



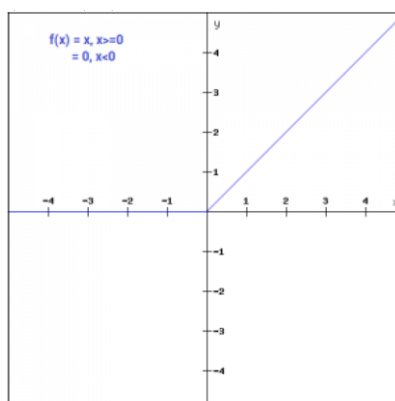
Gambar 2. Fungsi Aktivasi *Sigmoid* (Sharma dkk., 2020).

Fungsi *sigmoid* tidak simetris terhadap nol yang berarti tanda (positif atau negatif) dari semua nilai *output* neuron akan sama (Sharma dkk., 2020). Masalah tersebut dapat diatasi dengan melakukan penskalaan pada fungsi *sigmoid*. Penskalaan ini melibatkan mengalikan hasil fungsi *sigmoid* dengan faktor skala tertentu.

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

Fungsi *ReLU* merupakan fungsi aktivasi non linier yang sering digunakan dalam jaringan saraf. Kelebihan dari penggunaan fungsi *ReLU* adalah bahwa tidak semua neuron diaktifkan secara bersamaan. Hal ini berarti bahwa neuron akan dinonaktifkan hanya ketika nilai *output* dari transformasi linier bernilai nol. Secara matematis, fungsi tersebut dapat didefinisikan pada Persamaan (2.2) sebagai berikut (Sharma dkk., 2020):

$$f(x) = \max(0, x) \quad (2.2)$$



Gambar 3. Fungsi Aktivasi *ReLU* (Sharma dkk., 2020).

Fungsi *ReLU* lebih efisien dibandingkan dengan fungsi aktivasi lainnya yang terletak pada fakta bahwa tidak semua neuron diaktifkan secara bersamaan, melainkan hanya sejumlah tertentu pada satu waktu. Pada situasi tertentu, nilai gradien dapat menjadi nol yang menyebabkan bobot dan bias tidak mengalami pembaruan selama langkah propagasi balik dalam proses pelatihan jaringan saraf.

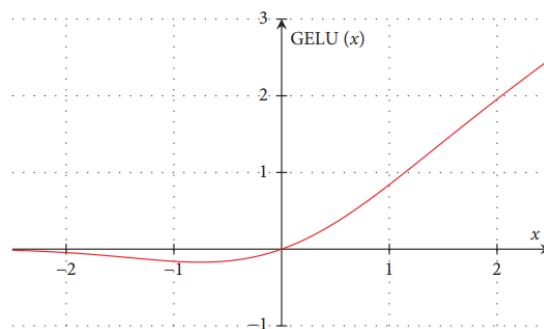


### 3. Fungsi Aktivasi *Gaussian Error Linear Unit (GELU)*

Fungsi aktivasi *GELU* adalah fungsi alternatif dari fungsi aktivasi *ReLU* yang umum digunakan, tanpa mengurangi manfaat yang dimilikinya. Meskipun fungsi *ReLU* dinyatakan sebagai  $\text{ReLU}(x) = \max(0, x)$  memberikan non *linieritas* pada jaringan saraf, namun mengalami ketidakbisaan dalam hal diferensiasi pada titik  $x = 0$ . Ketidakmampuan untuk melakukan diferensiasi ini dapat menyebabkan masalah selama proses optimasi berbasis gradien, seperti dinamika pelatihan yang tidak stabil (Lee, 2023).

Upaya mengatasi permasalahan tersebut, fungsi aktivasi *GELU* dikembangkan sebagai pendekatan yang mulus terhadap fungsi *ReLU*. Pendekatan ini memastikan diferensiasi yang dapat dilakukan di setiap titik, sambil mempertahankan sifat non linier yang penting untuk *deep learning*. Fungsi *GELU* mengambil inspirasi dari fungsi distribusi kumulatif *Gaussian* yang dicirikan oleh sifat kehalusan dan diferensiasi yang melekat. Secara matematis, fungsi tersebut dapat didefinisikan pada Persamaan (2.3) sebagai berikut (Lee, 2023):

$$GELU(x) = 0.5x \left( 1 + \tanh \left( \sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right) \quad (2.3)$$



Gambar 4. Fungsi Aktivasi *GELU* (Lee, 2023).

Fungsi ini menjadi populer pada *deep learning* karena kombinasi sifat-sifatnya yang diinginkan seperti non *linieritas*, diferensiasi, dan *smoothing*.

#### 4. Fungsi Aktivasi *Softmax*

Fungsi *Softmax* adalah gabungan dari beberapa fungsi *sigmoid*. Fungsi *sigmoid* menghasilkan nilai dalam rentang 0 hingga 1 yang dapat diinterpretasikan sebagai probabilitas untuk titik data pada kelas tertentu. Fungsi *softmax* berbeda dengan *sigmoid* yang umumnya digunakan untuk klasifikasi biner, fungsi *softmax* dapat diterapkan pada masalah klasifikasi *multiclass*. Fungsi ini menghasilkan probabilitas untuk setiap titik data terhadap semua kelas individual. Representasi matematisnya dapat dijelaskan pada Persamaan (2.4) sebagai berikut (Sharma dkk., 2020):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \text{ untuk } j = 1, \dots, K \quad (2.4)$$

### 2.10 *Deep Learning*

*Deep learning* adalah bagian dari *machine learning* yang terinspirasi dari cara otak manusia memproses informasi. Berbeda dengan *machine learning* tradisional, *deep learning* tidak bergantung pada aturan yang telah dirancang oleh manusia (Alzubaidi dkk., 2021). *Deep learning* menggunakan kumpulan data yang sangat besar untuk memetakan *input data* ke label-label tertentu. *Deep learning* dibangun dengan menggunakan sejumlah algoritma yang membentuk jaringan saraf tiruan. Setiap lapisan memberikan interpretasi unik terhadap data *input*. Pendekatan *machine learning* konvensional melibatkan serangkaian langkah, termasuk *preprocessing*, ekstraksi fitur, seleksi fitur, pembelajaran, dan klasifikasi. Seleksi fitur memegang peran penting dalam kinerja *machine learning*, tetapi dapat mengakibatkan diskriminasi kelas yang salah jika tidak dilakukan dengan benar. *Deep learning* mengotomatisasi proses pembelajaran fitur untuk berbagai tugas yang membuatnya lebih sederhana. *Deep learning* sangat efektif dalam menggabungkan pembelajaran dan klasifikasi dalam satu langkah.

*Deep learning* membedakan diri dari *machine learning* konvensional dalam metode pembelajarannya. *Deep learning* mengadopsi pendekatan yang lebih adaptif. Pada

*deep learning*, model diberi data yang telah diberi label, lalu secara otomatis mencari pola dan relasi dalam data tersebut (Alzubaidi dkk., 2021). Selama proses pembelajaran, model *deep learning* dapat menyesuaikan diri dengan data baru yang belum pernah dihadapi sebelumnya. Salah satu karakteristik utama *deep learning* adalah penggunaan banyak lapisan dalam jaringan saraf, sehingga sering disebut sebagai “*deep*” *learning*. Kemampuan ini untuk memahami dan menganalisis data yang sangat kompleks menjadikan *deep learning* sangat bermanfaat dalam berbagai aplikasi, termasuk pengenalan wajah, pengenalan suara, analisis teks, pemrosesan gambar, dan bahkan pemecahan masalah yang sangat kompleks seperti mobil otonom. Salah satu contoh algoritma *deep learning* adalah BERT. Model BERT termasuk dalam kategori *deep learning* karena memungkinkannya untuk secara otomatis mengekstraksi fitur-fitur dari data dan memahami pola dalam data teks yang sangat kompleks (Devlin dkk., 2019). Salah satu prinsip dasar *deep learning* dicerminkan dengan kemampuan untuk memproses data yang rumit dan beradaptasi dengan data baru tanpa perlu peraturan yang telah diprogram sebelumnya. Model BERT memiliki banyak lapisan dan *neuron* buatan yang saling terhubung, memungkinkannya menangani tugas analisis bahasa alami dengan sangat akurat, dan menjadikannya salah satu pencapaian penting dalam bidang *deep learning*.

### **2.11 Adam Optimizer**

Pemilihan *optimizer* dalam konteks *deep learning* memiliki dampak signifikan terhadap tingkat akurasi model. *Optimizer* berfungsi untuk mengoptimalkan parameter, seperti bobot dan *learning rate* dalam proses pelatihan model (Syifa dan Dewi, 2022). Salah satu *optimizer* yang sering digunakan dalam *deep learning* adalah *Adam Optimizer* yang dikenal luas karena efektivitasnya. Nama *Adam* sendiri merupakan singkatan dari *Adaptive Moment Estimation*. Keunggulan utama *Adam Optimizer* terletak pada kemampuannya untuk secara otomatis memperbarui bobot dan *learning rate* secara adaptif selama proses pelatihan. Oleh karena itu, saat menggunakan *Adam Optimizer*, nilai *learning rate* dapat berubah secara

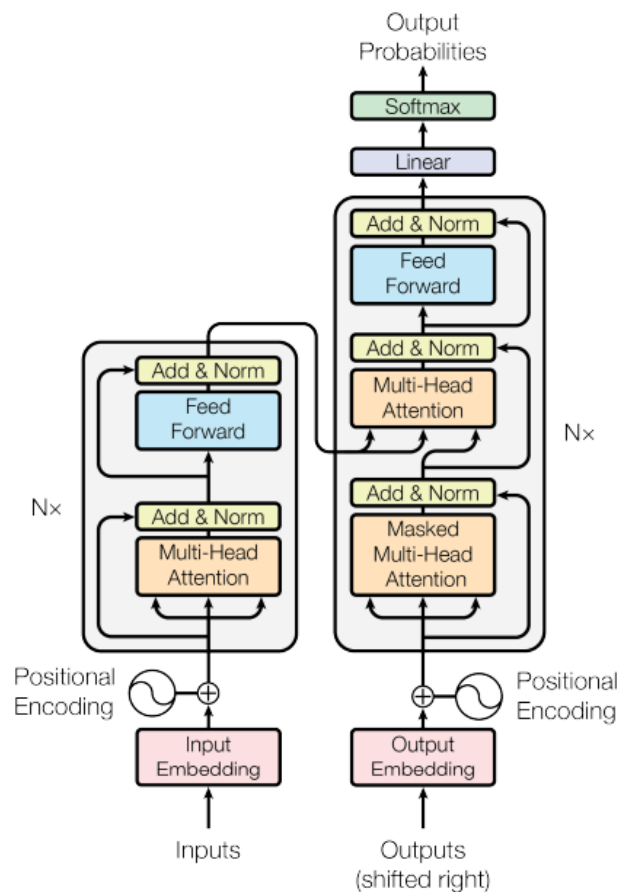
dinamis selama pelatihan berlangsung dan mengoptimalkan kinerja model (Soydaner, 2020).

## 2.12 *Transformer*

Model-model yang memproses data input sekuens umumnya dibangun atas dasar jaringan saraf *recurent* atau konvolusi, melibatkan *encoder* dan *decoder* yang kompleks dengan memanfaatkan *attention mechanism* untuk menghubungkan *encoder* dan *decoder*. Namun model *Transformer* sepenuhnya mengandalkan *attention mechanism* dan tidak menggunakan *recurent* dan konvolusi sama sekali. Pemodelan dengan *recurent* sering memerlukan perhitungan panjang posisi simbol pada urutan *input* dan *output* yang dapat menghambat proses paralelisasi selama pelatihan. *Transformer* yang memiliki *self-attention* sebagai inti utama dapat mengatasi perhitungan panjang posisi simbol pada urutan *input* dan *output* dan memungkinkan paralelisasi yang lebih efisien. *Self-attention* atau *intra-attention* menghubungkan posisi kata yang berbeda dari suatu kalimat untuk menghasilkan representasi kata dari kalimat tersebut. *Transformer* telah berhasil diterapkan dalam berbagai tugas seperti pemahaman bacaan, *summarization* abstraktif, *entailment textual*, dan pembelajaran representasi kalimat. *Transformer* mengeliminasi penggunaan RNN atau konvolusi dalam perhitungan representasi *input* dan *output*-nya (Vaswani dkk., 2017).

*Transformer* memiliki struktur *encoder-decoder*. Peran *encoder* adalah mengubah urutan *input* simbol  $(x_1, \dots, x_n)$  menjadi representasi kontinu  $z = (z_1, \dots, z_n)$ . Selanjutnya, *decoder* memanfaatkan representasi  $z$  untuk menghasilkan urutan *output*  $(y_1, \dots, y_m)$  dengan memproses simbol-simbol secara bertahap. Pada arsitektur *transformer*, *encoder* terdiri dari 6 lapisan identik, masing-masing terdiri dari dua *sub*-lapisan: *multi-head self-attention* dan jaringan *feed-forward* sederhana yang terhubung sepenuhnya. Kemudian *decoder* juga terdiri dari 6 lapisan identik, dengan dua *sub*-lapisan yang sama seperti *encoder*, ditambah satu lapisan *attention*

di antara dua lapisan yang dimodifikasi. (Vaswani dkk., 2017). Satu lapisan *attention* yang dimodifikasi memiliki peran untuk mencegah "mendahului" atau "menghadiri" posisi berikutnya dalam urutan selama proses prediksi. Hal ini membantu meningkatkan kemampuan model untuk memahami dan memproses urutan dengan lebih baik, sehingga hasil prediksi menjadi lebih akurat dan sesuai dengan konteks urutan yang sebenarnya. Walaupun *transformer* memiliki arsitektur *encoder* dan *decoder*, tetapi BERT hanya memanfaatkan lapisan *encoder* dalam strukturnya. Struktur *transformer* dapat diilustrasikan pada Gambar 5 sebagai berikut:



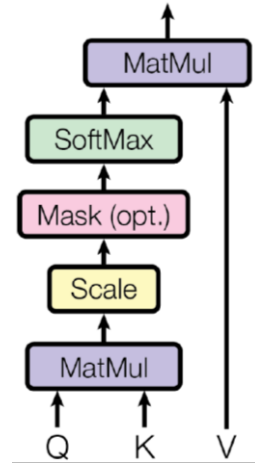
Gambar 5. Arsitektur Model *Transformer* (Vaswani dkk., 2017).

### 2.12.1 Attention

Fungsi *attention* sebagai fungsi yang menghubungkan sebuah *query* dengan sekelompok pasangan *key-value* yang menghasilkan sebuah nilai *output*. Prosesnya melibatkan perhitungan bobot untuk setiap nilai, dimana bobot tersebut ditentukan oleh seberapa cocoknya *query* dengan *key* yang sesuai (Vaswani dkk., 2017). Oleh karena itu, fungsi *attention* memungkinkan *query* untuk mengakses informasi dari *value* dengan tingkat relevansi yang ditentukan oleh kemiripan dengan *key*.

### 2.12.2 Scaled Dot-Product Attention

*Scaled dot-product attention* terdiri dari himpunan vektor *query* dan *key* dengan dimensi  $d_k$ , serta himpunan vektor *value* dengan dimensi  $d_v$ . Himpunan vektor *query*, *key*, dan *value* dapat disusun pada matriks  $Q = [q_1, q_2, \dots, q_n]^T$ ,  $K = [k_1, k_2, \dots, k_n]^T$ , dan  $V = [v_1, v_2, \dots, v_n]^T$ . Secara berurutan proses perhitungan *scaled dot-product attention* melibatkan perkalian skalar antara vektor *query* dan semua vektor *key*, kemudian hasilnya dibagi dengan akar kuadrat dari  $d_k$ . Selanjutnya, diterapkan fungsi *softmax* pada hasil tersebut untuk mendapatkan bobot yang akan diterapkan pada vektor *value* (Vaswani dkk., 2017). Ilustrasi operasi *scaled dot-product attention* dapat dilihat pada Gambar 6 dan perhitungan *scaled dot-product attention* dapat diformulasikan pada Persamaan (2.5).



Gambar 6. *Scaled Dot-Product Attention* (Vaswani dkk., 2017).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.5)$$

### 2.12.3 Multi Head Attention

*Multi-head attention* pada dasarnya merupakan struktur arsitektur yang melakukan fungsi attention sebanyak  $h$  kali secara bersamaan. Proses ini menggunakan matriks  $Q$ ,  $K$ , dan  $V$  yang berbeda, dimana masing-masing matriks merupakan matriks *queries*, *keys*, dan *values*. *Multi-head attention* bertujuan untuk menghasilkan sebanyak  $h$  besaran *attention* yang berbeda untuk setiap kata sehingga mampu menangkap berbagai kemungkinan ketergantungan yang beragam (Vaswani dkk., 2017). *Multi-head attention* dapat diinterpretasikan oleh Persamaan (2.6) dan (2.7) sebagai berikut:

$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2.6)$$

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (2.7)$$

Proyeksi-proyeksi tersebut direpresentasikan sebagai matriks parameter  $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ , dan  $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ .

#### 2.12.4 Position-Wise Feed-Forward Networks

Tiap lapisan dalam *encoder* dan *decoder* pada *transformer* memiliki jaringan *feed-forward* yang *fully connected*. Hal ini diterapkan pada setiap posisi secara terpisah dan identik. Jaringan *feed-forward* ini terdiri dari dua transformasi linear yang dihubungkan oleh fungsi aktivasi *ReLU* (Vaswani dkk., 2017). *Feed-forward* dapat diuraikan pada Persamaan (2.8) sebagai berikut:

$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2 \quad (2.8)$$

#### 2.12.5 Positional Encoding

Model *transformer* tidak memanfaatkan *recurent* dan konvolusi. *Transformer* memasukkan informasi tentang posisi relatif atau absolut token-token dalam urutan menggunakan "*positional encodings*" agar model dapat menggunakan informasi urutan kata dalam urutan kalimat tersebut (Vaswani dkk., 2017). *Positional encodings* ini ditambahkan ke *embedding input* pada dasar setiap tumpukan *encoder* dan *decoder*. Oleh karena itu, *positional encodings* dan *embedding* memiliki dimensi yang sama sehingga keduanya dapat dijumlahkan. *Positional encodings* menggunakan fungsi *sinus* dan *kosinus* yang dapat direpresentasikan pada Persamaan (2.9) dan (2.10) sebagai berikut:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (2.9)$$

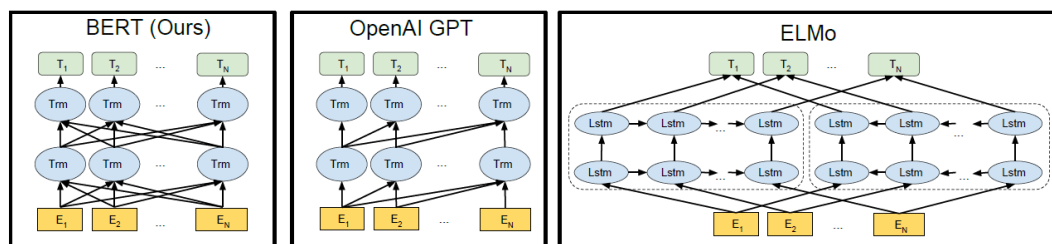
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (2.10)$$

Pada Persamaan (2.9) dan (2.10) terdapat *pos* dan *i*, dimana *pos* adalah representasi posisi dalam urutan dan *i* adalah dimensi. Setiap dimensi dari *positional encoding* dipetakan ke *sinusoid* tertentu. Panjang gelombangnya membentuk suatu progresi geometris yang berkisar antara  $2\pi$  hingga  $10000 \cdot 2\pi$ . Pemilihan fungsi ini dilakukan karena mempermudah model dalam mempelajari *attention* (Vaswani dkk., 2017).



### 2.13 Bidirectional Encoder Representations from Transformer

Tujuan utama BERT adalah melakukan pra-pelatihan representasi mendalam secara *bidirectional* pada teks tanpa label dengan memperhitungkan konteks dari kedua sisi, baik kiri maupun kanan, pada semua lapisan. Model BERT yang telah melalui tahap *pre-training* dapat disesuaikan untuk berbagai tugas seperti klasifikasi hanya dengan menambahkan satu lapisan *output* tambahan tanpa perlu melakukan modifikasi arsitektur yang signifikan. Keunikan BERT terletak pada pendekatan *bidirectional* dalam pelatihan, memungkinkan perhatian pada konteks dari kedua arah kata dalam suatu kalimat. Oleh karena itu, BERT dapat menghasilkan representasi bahasa yang sangat canggih. Ilustrasi perbedaan arsitektur BERT dengan Open AI GPT dan ELMo dapat direpresentasikan pada Gambar 7 (Devlin dkk., 2019).

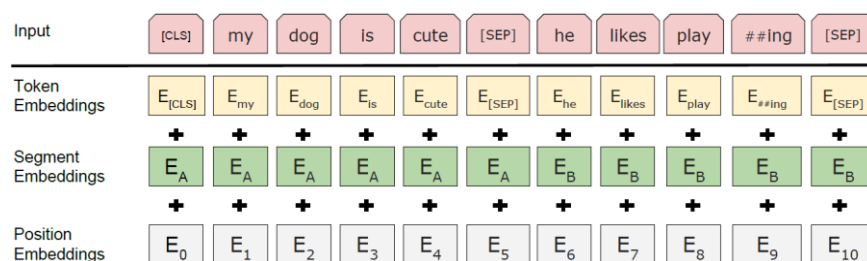


Gambar 7. Arsitektur BERT, Open AI GPT, dan ELMo (Devlin dkk., 2019).

Gambar 7 menampilkan perbedaan dalam arsitektur model pra-pelatihan pada BERT yang menggunakan *transformer* dua arah, OpenAI GPT yang mengadopsi *transformer* kiri ke kanan, dan ELMo yang menggunakan rangkaian LSTM kiri ke kanan dan kanan ke kiri yang dilatih secara independen untuk menghasilkan fitur tugas kompleks. Diantara ketiganya, hanya BERT memperhitungkan informasi dari kata-kata sebelum dan sesudah token saat ini dalam kalimat. Selain perbedaan dalam arsitektur, BERT dan OpenAI GPT memiliki kesamaan dengan menerapkan pendekatan penyempurnaan, sedangkan ELMo memegang pendekatan berbasis fitur (Devlin dkk., 2019).

Struktur utama dari model BERT terdiri dari lapisan *transformer* yang terbatas hingga tahap *encoder*, seperti yang dapat diilustrasikan pada Gambar 8. Model BERT dibedakan menjadi dua ukuran, yaitu BERT BASE dan BERT LARGE. Perbedaan antara keduanya terletak pada jumlah lapisan *encoder*, jumlah *multi-head self-attention*, ukuran tersembunyi, dan jumlah parameter. BERT BASE memiliki 12 lapisan *encoder*, 12 *multi-head self-attention*, 768 ukuran tersembunyi, dan 110 juta parameter. Sementara itu, BERT LARGE memiliki 24 lapisan *encoder*, 16 *multi-head self-attention*, 1024 ukuran tersembunyi, dan 340 juta parameter (Devlin dkk., 2019).

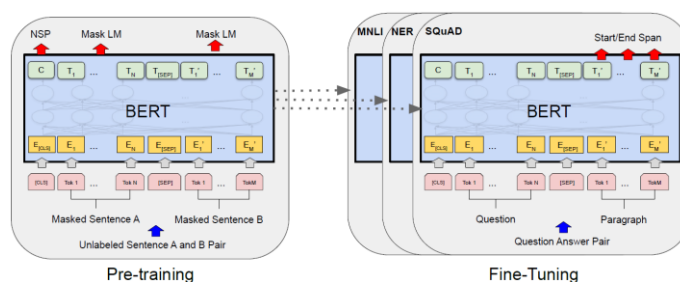
Model BERT memanfaatkan 2 token, yaitu token [CLS] dan [SEP]. Setiap urutan kata diawali dengan token [CLS] yang memiliki fungsi khusus sebagai token klasifikasi. Token [SEP] ditempatkan di akhir kalimat untuk memisahkan antar kalimat. *Segment embedding* diterapkan pada setiap token untuk membedakan kata-kata dari dua kalimat. Selanjutnya, *positional embedding* digunakan untuk memberi tanda posisi setiap token dalam kalimat (Devlin dkk., 2019). Proses *input* ke *encoder* BERT melibatkan kombinasi dari *Token Embedding*, *Segment Embedding*, dan *Positional Embedding*. Representasi *input* untuk suatu token tertentu dibangun dengan menjumlahkan token *embedding*, *segment embedding*, dan *positional embedding*. Representasi ini divisualisasikan pada Gambar 8 sebagai berikut:



Gambar 8. Representasi *Input* BERT (Devlin dkk., 2019).

Model BERT memiliki arsitektur yang terdiri dari *pre-training* dan *fine-tuning* (Devlin dkk., 2019). Arsitektur tersebut dapat dilihat pada Gambar 9. Langkah-

langkah *pre-training* dan *fine-tuning* untuk model BERT melibatkan penggunaan arsitektur yang sama, kecuali pada lapisan *output*.



Gambar 9. *Framework* BERT (Devlin dkk., 2019).

Seluruh arsitektur yang digunakan dalam *pre-training* juga digunakan dalam *fine-tuning*. Parameter dari model *pre-training* yang sama digunakan untuk menginisialisasi model untuk berbagai tugas yang berbeda. Selama proses *fine-tuning*, seluruh parameter disesuaikan dengan cermat (Devlin dkk., 2019).

### 2.13.1 *Pre-training* BERT

Model BERT mengadopsi pendekatan yang berbeda dari model bahasa tradisional yang mempelajari bahasa dari kiri ke kanan atau dari kanan ke kiri saja selama *pre-training* (Devlin dkk., 2019). Sebaliknya, BERT melakukan *pre-training* dengan fokus pada dua tugas yang dapat diselesaikan secara *unsupervised learning*, sebagaimana diperlihatkan pada bagian kiri dari Gambar 9. Kedua tugas tersebut adalah *masked language model* dan *next sentence prediction*.

#### 1. *Masked Language Model*

*Masked language model* pada BERT memberikan pemahaman intuitif bahwa model yang mendalam dan bersifat *bidirectional* memiliki keunggulan dibandingkan dengan model kiri ke kanan atau pendekatan dasar yang menggabungkan keduanya. Model hanya memproses konteks kata dari arah yang ditentukan, tanpa memperhatikan informasi kata dari arah yang berlawanan. Keterbatasan memproses konteks kata dari arah yang ditentukan,

tanpa memperhatikan informasi kata dari arah yang berlawanan berkaitan dengan kemampuan model dalam memproses informasi kata dari kedua arah secara bersamaan. Penggunaan *masked language model* pada model dapat secara simultan melihat konteks dari kedua arah, memungkinkan setiap kata "melihat dirinya sendiri" secara tidak langsung. Pada tugas *masked language model* BERT, sejumlah persentase token *input* dipilih secara acak dan disembunyikan, lalu model bertujuan untuk memprediksi token-token yang disembunyikan tersebut. Pada eksperimen BERT, 15% dari semua token WordPiece dipilih secara acak untuk disembunyikan. BERT mampu memprediksi pada kata-kata yang disembunyikan (Devlin dkk., 2019).

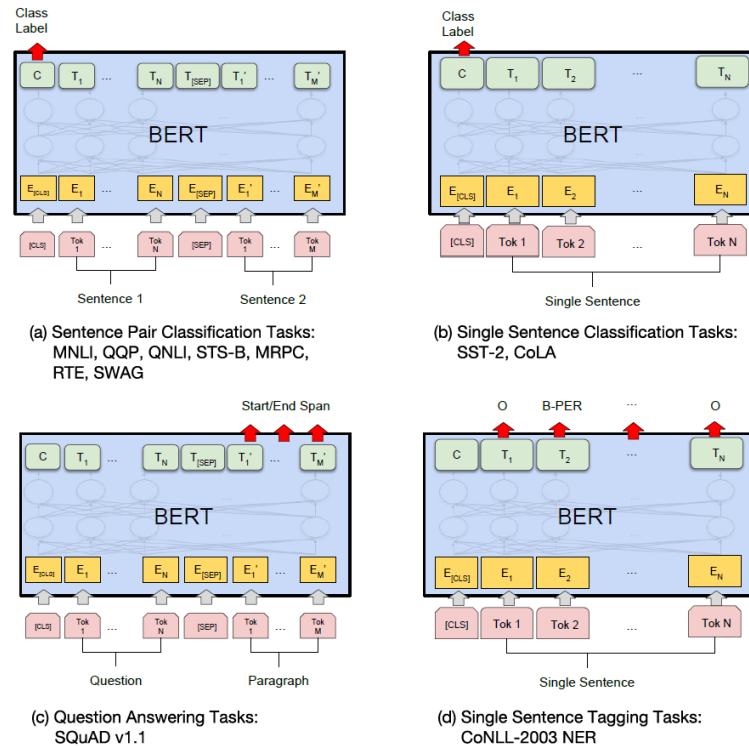
## 2. *Next Sentence Prediction* (NSP)

*Next Sentence Prediction* memiliki signifikansi dalam berbagai tugas khusus seperti *Question Answering* (QA) dan *Natural Language Inference* (NLI) yang memerlukan pemahaman yang mendalam terhadap hubungan antara dua kalimat. Hal ini secara tidak langsung tertangkap oleh pemodelan bahasa konvensional. NSP diimplementasikan sebagai tugas *pre-training* untuk menghasilkan pemahaman model terhadap relasi antarkalimat dalam korpus *monolingual*. Pada tugas NSP, terdapat *pre-training* yang terdiri dari dua bagian kalimat, kalimat A dan B. Pada setengah kasus, kalimat B mengikuti kalimat A secara aktual. Sedangkan pada setengah kasus lainnya, kalimat B dipilih secara acak dari korpus untuk mengikuti kalimat A. Pada tugas ini merupakan representasi akhir dari token khusus [CLS] digunakan untuk memprediksi apakah kalimat B merupakan kalimat berikutnya. Pada tahap ini, model menerima sepasang kalimat sebagai *input* dan dilatih untuk memprediksi apakah kalimat kedua adalah kelanjutan dari kalimat pertama atau tidak. Selama pelatihan, 50% dari pasangan kalimat adalah pasangan sebenarnya, sementara 50% sisanya adalah kalimat kedua yang diambil secara acak dari korpus teks. Meskipun tugas ini sederhana, namun berperan penting dalam membantu model memahami relasi antara dua kalimat (Devlin dkk., 2019).

### 2.13.2 *Fine-tuning* BERT

Proses *fine-tuning* pada BERT menjadi relatif mudah karena adanya mekanisme *self-attention* dalam *transformer*. Mekanisme ini memungkinkan BERT dapat beradaptasi dan memodelkan berbagai tugas dengan menukar *input* dan *output* yang sesuai (Devlin dkk., 2019). Pada aplikasi yang melibatkan pasangan teks, umumnya pola yang diterapkan adalah dengan mengodekan pasangan teks secara independen dan kemudian menerapkan *cross-attention* secara *bidirectional*. Namun model BERT memanfaatkan mekanisme *self-attention* untuk mengintegrasikan dua tahap ini. *Self-attention* menyebabkan pengkodean pasangan teks dilakukan secara efektif dan mencakup *cross-attention bidirectional* antara dua kalimat.

Pada *fine-tuning* untuk setiap tugas, pendekatan yang diambil adalah dengan menghubungkan *input* dan *output* khusus tugas tersebut ke BERT, dan kemudian melakukan *fine-tuning* pada semua parameter dari awal hingga akhir. Pada tahap *input*, kalimat A dan kalimat B dari *pre-training* analog dengan tugas pasangan kalimat dalam perumpamaan, tugas pasangan hipotesis-*premises* dalam *entailment*, tugas pasangan pertanyaan-*passage* dalam *question answering*, tugas klasifikasi teks, tugas penandaan urutan, dan sebagainya (Devlin dkk., 2019). Pada tahap *output*, representasi token disalurkan ke lapisan *output* untuk tugas berbasis token seperti penandaan urutan atau *question-answering*. Sebaliknya, representasi dari token [CLS] disalurkan ke lapisan *output* untuk tugas klasifikasi, seperti analisis sentimen. Ilustrasi arsitektur *fine-tuning* BERT dapat dilihat pada Gambar 10.



Gambar 10. *Fine-tuning* BERT (Devlin dkk., 2019).

Pada saat tahap *fine-tuning* BERT, hampir semua *hyperparameter* yang digunakan pada tahap *pre-training* tetap tidak berubah. Namun, ada beberapa *hyperparameter* yang diubah, seperti *batch size*, *learning rate*, dan jumlah *epoch*. Nilai *dropout probability* selalu pada 0,1. Meskipun setiap tugas pada NLP memiliki *hyperparameter* optimal yang berbeda-beda, hasil yang baik dapat diperoleh dengan menggunakan *batch size* 16 dan 32. Selanjutnya besaran nilai *learning rate* (menggunakan Adam) terdapat pada nilai antara  $5 \times 10^{-5}$ ,  $3 \times 10^{-5}$ , dan  $2 \times 10^{-5}$ . Serta melakukan pelatihan sebanyak 2, 3, atau 4 kali (Devlin dkk., 2019).

## 2.14 Evaluasi Kinerja Model

Pengukuran kinerja model klasifikasi dapat dilakukan menggunakan berbagai jenis metode evaluasi, salah satunya adalah *confusion matrix*. Pada dasarnya *confusion matrix* merupakan tabel berdimensi 2x2 yang terdiri dari baris yang mewakili kelas

prediksi (*predicted classes*) dan kolom yang mewakili kelas sebenarnya (*actual classes*). Pada *confusion matrix* terdapat empat entri utama, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). *True Positive* menggambarkan situasi dimana kelas yang diprediksi adalah positif dan kelas sebenarnya juga positif. *False Positive* terjadi ketika kelas yang diprediksi adalah positif, tetapi kelas sebenarnya adalah negatif. *False Negative* adalah kondisi dimana kelas yang diprediksi adalah negatif, namun kelas sebenarnya adalah positif. *True Negative* terjadi ketika kelas yang diprediksi adalah negatif dan kelas sebenarnya juga negatif (Arifiyanti dkk., 2018).

Pada klasifikasi *multiclass*, metrik yang umumnya digunakan dalam klasifikasi biner tidak sepenuhnya berlaku. *Confusion Matrix* untuk klasifikasi *multiclass* memiliki dimensi  $N \times N$ , dimana  $N$  merupakan jumlah label kelas yang berbeda ( $C_0, C_1, \dots, C_N$ ). Oleh karena itu, definisi TP, TN, FP, dan FN tidak relevan dalam konteks ini. Namun dengan menggabungkan beberapa metrik yang sesuai, dapat dievaluasi kinerja secara keseluruhan dari *confusion matrix*. Metrik-metrik yang dapat diterapkan dalam kasus ini adalah sub-kumpulan dari metrik yang digunakan dalam klasifikasi biner. Beberapa metrik yang relevan dalam kasus klasifikasi *multiclass* meliputi *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik-metrik ini membantu dalam mengevaluasi kinerja model klasifikasi dalam konteks klasifikasi *multiclass*. *Confusion matrix multiclass* disajikan pada Gambar 11 sebagai berikut (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 11. *Confusion Matrix Multiclass* (Markoulidakis dkk., 2021).

Berdasarkan *confusion matrix*, dapat dihitung sejumlah metrik evaluasi yang berguna untuk mengukur kinerja model klasifikasi. Beberapa metrik evaluasi utama meliputi:

1. *Accuracy*

*Accuracy* adalah metrik yang berguna dalam mengevaluasi kinerja model klasifikasi. *Accuracy* dapat mencerminkan persentase prediksi yang tepat yang diberikan oleh model (Yechuri dan Ramadass, 2021). Secara matematis, *accuracy* dapat dijelaskan menggunakan Persamaan (2.11) sebagai berikut (Markoulidakis dkk., 2021):

$$Accuracy = \frac{\sum_{i=0} c_{ii}}{\sum_{i=0} \sum_{j=0} c_{ij}} \quad (2.11)$$

2. *Precision*

*Precision* adalah ukuran yang menunjukkan seberapa akurat model dalam memprediksi hasil positif dari sampel yang diprediksi sebagai positif. Hal ini berguna ketika fokusnya adalah untuk mengurangi jumlah prediksi positif yang salah (Yun, 2021). Secara matematis, *precision* dapat dijelaskan menggunakan Persamaan (2.12) sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2.12)$$

3. *Recall*

*Recall* adalah perbandingan antara jumlah prediksi positif yang benar dengan total data positif yang sebenarnya (Hossin dan Sulaiman, 2015). *Recall* memberikan gambaran tentang sejauh mana model dapat mengidentifikasi dengan benar data positif yang sebenarnya. Secara matematis, *recall* dapat dijelaskan menggunakan Persamaan (2.13) sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \quad (2.13)$$

4. *F1-score*

*F1-score* adalah sebuah metrik yang mencakup *precision* dan *recall* ke dalam satu ukuran tunggal. Hal ini berguna terutama ketika berurusan dengan kumpulan data yang memiliki ketidakseimbangan kelas (Hossin dan Sulaiman, 2015). *F1-score* dapat dihitung menggunakan Persamaan (2.14) sebagai berikut:



$$F1 - score = \frac{2 \times recall \times precision}{(recall + precision)} \quad (2.14)$$

Persamaan (2.14) memadukan *precision* dan *recall* dalam sebuah skor tunggal yang membantu dalam mengevaluasi sejauh mana model dapat memprediksi data positif dengan benar dan mengidentifikasi sebagian besar data positif yang sebenarnya. Semakin tinggi nilai *F1-score*, semakin baik kinerja model klasifikasi tersebut.

### III. METODELOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Waktu dan Tempat Penelitian ini yaitu sebagai berikut:

##### a. Tempat Penelitian

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

##### b. Waktu Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2023/2024, tepatnya pada bulan September 2023. Penelitian ini dibagi menjadi tiga tahap, tahap pertama dilakukan studi literatur dengan topik penelitian yang akan digunakan sebagai referensi dalam penyusunan proposal penelitian. Selanjutnya dilakukan pengumpulan data yang akan digunakan sebagai bahan untuk dilakukan penelitian dan penyusunan draf proposal penelitian. Tahap kedua adalah tahap pengerjaan program mulai dari *input data*, *preprocessing data*, *resampling data*, *word embedding* dengan BERT, pemodelan klasifikasi BERT, *fine-tuning* BERT, dan evaluasi metrik dari model yang telah dibuat. Tahap ketiga adalah penyusunan hasil penelitian dan kesimpulan penelitian.

## 3.2 Data dan Alat

Dalam Penelitian ini dibutuhkan data dan alat sebagai berikut:

### 3.2.1 Data

Data yang digunakan penelitian ini merupakan data teks yang ada di situs Kaggle <https://www.kaggle.com/datasets/mohamedabdelwahabali/drugreview>, dimana data tersebut berupa data teks berbahasa inggris dan terakhir diperbaharui pada November 2023 lalu oleh Mohamed Abdelwahab Ali. Data teks tersebut telah terpisah menjadi data *training*, *testing*, dan validasi. Jumlah data *training* sebanyak 110811 pengamatan. Jumlah data *testing* sebanyak 46108 pengamatan. Jumlah data validasi sebanyak 27703 pengamatan.

Data teks berbahasa inggris yang digunakan masing-masing memiliki 7 atribut/variabel yaitu *uniqueID*, *drugName*, *condition*, *review*, *rating*, *date*, *usefulCount*. *uniqueID* adalah ID unik dari responden, *drugName* adalah nama obat yang digunakan, *condition* adalah kondisi yang dialami responden, *review* adalah ulasan obat yang digunakan responden, *rating* adalah peringkat obat berdasarkan ulasan responden, *date* adalah tanggal responden memberikan ulasan dan peringkat, dan *usefulCount* adalah jumlah responden yang merasakan manfaat yang sama dengan ulasan yang diberikan. Data yang digunakan dalam penelitian ini adalah data kualitatif dengan total jumlah pengamatan sebanyak 184622 pengamatan. Data yang digunakan adalah *review* dan *rating* untuk diklasifikasi. Data ini termasuk data *supervised learning* atau data dengan ciri mempunyai label.

Sebaran ulasan berdasarkan *rating* yang terdapat pada data tersebut disajikan pada Tabel 2 sebagai berikut:

Tabel 2. Sebaran Ulasan Tiap *Rating*

<i>Rating</i>	Frekuensi
1	23876
2	7948
3	7605
4	5862
5	9521
6	7419
7	11122
8	22086
9	32098
10	57085

Kemudian sampel dari kumpulan data yang akan digunakan pada penelitian ini disajikan pada Tabel 3, 4, dan 5 sebagai berikut:

Tabel 3. Sampel Data *Training Drugs Review*

	<b>review</b>	<b>rating</b>
<b>0</b>	"i have used restasis for about a year now and...	2.0
<b>1</b>	"my experience has been somewhat mixed. i have...	7.0
<b>2</b>	"this is my second implanon would not recommen...	1.0
<b>3</b>	"i recommend taking as prescribed, and the bot...	10.0
..	.....	..

Tabel 4. Sampel Data Validasi *Drugs Review*

	<b>review</b>	<b>rating</b>
<b>0</b>	"i've tried a few antidepressants over the yea...	10.0
<b>1</b>	"my son has crohn's disease and has done very ...	8/0
<b>2</b>	"contrave combines drugs that were used for al...	9.0
<b>3</b>	"i have been on this birth control for one cyc...	9.0
..	.....	..

Tabel 5. Sampel Data *Testing Drugs Review*

	<b>review</b>	<b>rating</b>
<b>0</b>	"sober a year 8-25-11. god, aa and campral hav...	10.0
<b>1</b>	"i've been on birth control for a while now du...	4.0
<b>2</b>	"hi, this is an updated experience. \r\r\n\r\r...	8.0
<b>3</b>	"i have been on the ortho evra patch for just ...	8.0
..	.....	..

### 3.2.2 Alat

Adapun alat yang digunakan dalam penelitian ini yaitu sebagai berikut:

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini untuk mendukung dan menunjang pelaksanaan penelitian, yaitu:

1. Processor Intel(R) Core(TM) i5-1135G7 @ 2.40Ghz 2.42Ghz
2. Installed RAM 16 GB

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan pada penelitian ini untuk mendukung dan menunjang penelitian ini, yaitu:

1. Sistem operasi *Windows 11*
2. *Python 3.10.2*

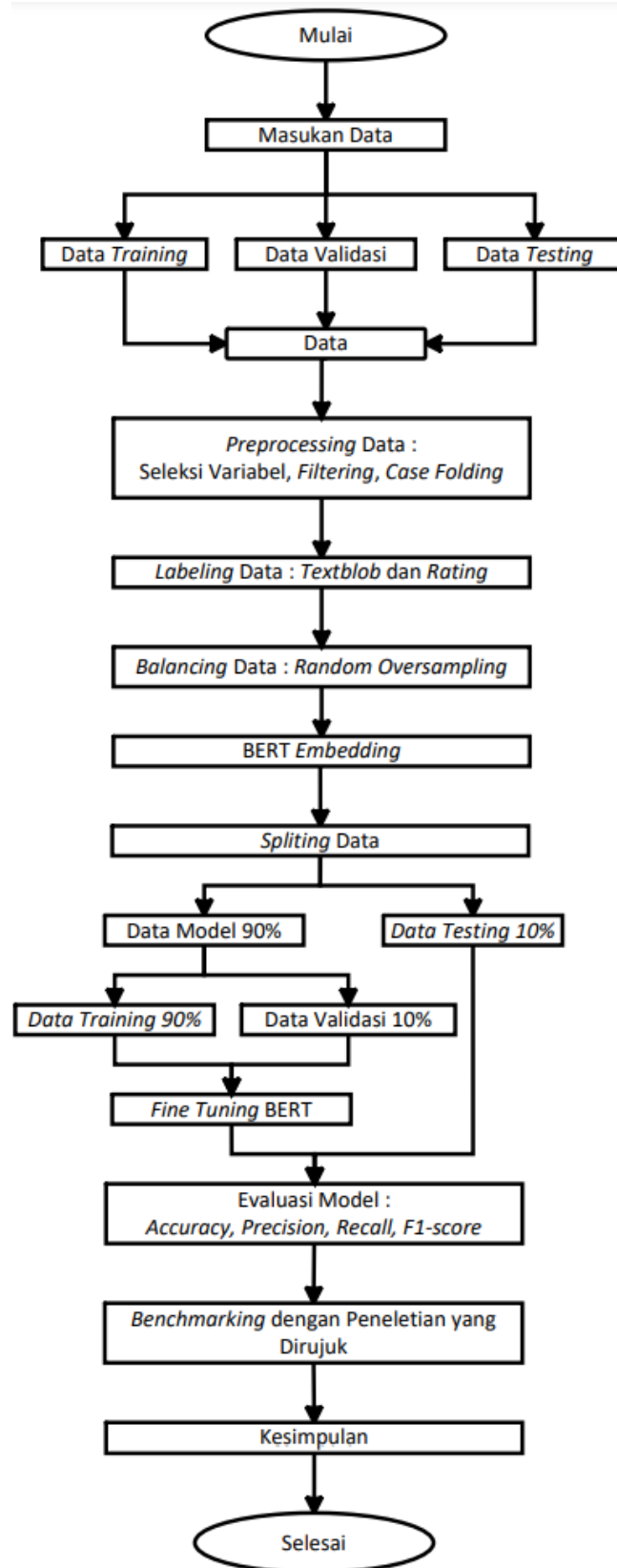
Peneliti menggunakan *website* Kaggle sebagai *notebook Python*. Adapun *package* yang digunakan dalam penelitian ini sebagai berikut:

- a. *pandas* versi 2.0.3: *pandas* adalah *library Python* yang menyediakan struktur data dan alat analisis data yang fleksibel dan efisien.

- b. *numpy* versi 1.23.5: *numPy* adalah *library* untuk komputasi numerik yang mendukung *array* dan matriks besar, serta fungsi matematika yang beroperasi pada *array* tersebut.
- c. *sklearn* versi 1.2.2: *scikit-learn* (*sklearn*) adalah *library machine learning* untuk *Python*.
- d. *matplotlib* versi 3.7.2: *matplotlib* adalah *library* untuk membuat visualisasi grafik 2D dan 3D dalam *Python*.
- e. *nlk* versi 3.2.4: Natural Language Toolkit (NLTK) adalah *library* untuk pemrosesan bahasa alami dalam *Python*.
- f. *imblearn* versi 0.11.0: *imbalanced-learn* adalah *library* untuk menangani masalah kelas tak seimbang dalam *machine learning*.
- g. *torch* versi 2.0.0+cpu: *PyTorch* adalah *framework machine learning* yang berfokus pada arsitektur jaringan saraf.
- h. *transformers* versi 4.33.0: *transformers* adalah *library* yang menyediakan implementasi *state-of-the-art* dari arsitektur *transformer* untuk tugas-tugas seperti pemrosesan bahasa alami.
- i. *wordcloud* versi 1.9.3: *wordcloud* adalah *library* yang digunakan untuk menampilkan visual kata-kata yang sering muncul.

### 3.3 Metode Penelitian

Penelitian ini menggunakan metode BERT untuk melakukan analisis sentimen pada ulasan obat. Ini dilakukan dengan menggunakan perangkat lunak *Python* dengan bantuan teks editor Kaggle. Evaluasi metode BERT dalam klasifikasi sentimen akan didasarkan pada penggunaan *confusion matrix*, dan selanjutnya membangun metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Alur penelitian diilustrasikan pada Gambar 12 sebagai berikut:



Gambar 12. *Flowchart* Metode Penelitian.

Berikut adalah penjelasan dari alur penelitian yang akan dilaksanakan dalam penelitian ini:

1. Melakukan *input* data pada *software Python* dengan Kaggle sebagai tempat menulis kode *Python*.
2. Menyatukan data *training*, validasi, dan *testing*.
3. Melakukan *preprocessing* data dengan menyeleksi variabel yang digunakan. Kemudian dilanjutkan dengan *preprocessing text* seperti membersihkan karakter non alfabet dan *case folding* pada data.
4. Memberi label pada *output* dengan *rating* yang diperkecil skalanya dan *textblob*. Selanjutnya menerapkan *label encoder* pada data yang telah diberi label berdasarkan *rating* dan *textblob*.
5. Menangani *imbalance data* dengan teknik *Random Over Sampling* (ROS).
6. Mengimplementasikan *BERT embedding* menggunakan *BERT BASE Uncased*. Kemudian ditentukan panjang maksimal sebanyak 230 token dari batas maksimal BERT adalah 512 token. Setiap kalimat *review* yang ada di data memiliki jumlah panjang bervariasi. Jika panjang kalimat lebih dari panjang maksimum yang telah ditentukan maka akan dikurangi (*truncate*). Sedangkan jika panjang kalimat kurang dari panjang maksimum akan ditambahkan token [*pad*].
7. Membagi data menjadi 90% data model dan 10% data *testing*. Kemudian 90% data model dibagi menjadi 90% data *training* dan 10% data validasi.
8. Melakukan *fine-tuning* BERT untuk tugas klasifikasi sentimen dengan *BertForSequenceClassification*. Setelah itu dibangun model klasifikasi dengan data *training* dan validasi. Selanjutnya menentukan *batch size*, *dropout*, *learning rate*, *patience*, dan *epoch*.
9. Menguji model BERT dengan data *testing*.
10. Melakukan evaluasi model untuk melihat performa model berupa nilai *accuracy*, *precision*, *recall*, dan *F1-score*.
11. Melakukan *benchmarking* dengan penelitian terdahulu dan membuat kesimpulan.



## V. PENUTUP

### 5.1 Kesimpulan

Kesimpulan dari penelitian ini menunjukkan bahwa penerapan *Bidirectional Encoder Representations from Transformer* (BERT) dalam melakukan klasifikasi terhadap ulasan penilaian obat memberikan hasil yang memuaskan. Beberapa temuan penting yang dapat disimpulkan selama pengerjaan penelitian ini melibatkan:

1. Penggunaan model BERT BASE *Uncased* terbukti sangat efektif untuk tugas *word embedding* dan klasifikasi pada data ulasan penilaian obat berbahasa Inggris. Model ini mampu secara akurat menangkap makna kompleks dan nuansa dalam ulasan, meningkatkan kemampuan pemrosesan bahasa alami dalam konteks klasifikasi penilaian ulasan obat.
2. Model klasifikasi yang dibangun berdasarkan *rating* dan *textblob* ulasan obat berhasil mencapai tingkat akurasi yang signifikan dengan akurasi berturut-turut sebesar 97% dan 98%. Hal ini menandakan bahwa model BERT mampu secara efektif mengenali dan mengklasifikasikan sentimen ulasan penilaian obat.
3. Temuan ini relevan dengan tujuan penelitian untuk klasifikasi sentimen pada ulasan penilaian obat. Penerapan BERT membuktikan kehandalannya dalam menangani data tekstual yang kompleks dan menghasilkan prediksi yang akurat.

Hasil penelitian ini memberikan kontribusi penting dalam pemahaman dan penerapan model BERT untuk tugas klasifikasi sentimen pada ulasan penilaian obat. Implikasi temuan ini dapat dijadikan sebagai landasan untuk pengembangan lebih lanjut dalam bidang analisis sentimen pada domain kesehatan dan farmasi.

## 5.2 Saran

Saran yang diusulkan untuk penelitian selanjutnya mencakup pengembangan lebih lanjut pada penggunaan batasan kata dalam *word embedding* menggunakan BERT agar mendapatkan peningkatan hasil yang lebih optimal. Langkah-langkah yang dapat diterapkan melibatkan pemilihan model BERT *embedding* yang lebih disesuaikan dengan karakteristik data teks. Penyesuaian parameter seperti *batch size*, *learning rate*, *dropout*, *patience*, dan *epoch* juga menjadi kunci dalam mendapatkan model yang lebih baik dalam menjalankan tugas klasifikasi sentimen.

Eksplorasi lebih lanjut memiliki peran penting pada pemilihan model BERT yang sesuai dengan konteks spesifik dari ulasan obat. Penelitian ini dapat mengevaluasi sejumlah model BERT yang ada, termasuk BERT yang telah disesuaikan atau dioptimalkan untuk tugas klasifikasi sentimen pada data ulasan obat. Pengaturan parameter-parameter penting tersebut dapat diuji dan disesuaikan untuk mencapai hasil yang lebih optimal dan akurat.

Selain itu, penelitian selanjutnya dapat melibatkan penerapan klasifikasi BERT menggunakan kumpulan data yang berbeda untuk menguji generalisasi model. Hal ini memungkinkan untuk memvalidasi kehandalan model klasifikasi sentimen pada berbagai domain dan jenis data ulasan.

Selain BERT, penelitian berikutnya dapat mempertimbangkan penerapan model analisis data teks lainnya seperti GPT, RoBERTa, dan metode *transformer* lainnya. Perbandingan kinerja antara berbagai model tersebut dapat memberikan wawasan tambahan terkait efektivitas dan penerapannya pada tugas klasifikasi sentimen pada ulasan penilaian obat.

## DAFTAR PUSTAKA

- Aggarwal, C. 2018. *Neural Networks and Deep Learning*. Springer International Publishing, New York.
- Alwehaibi, A., Bikdash, M., Albogmi, M., & Roy, K. 2022. A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches. *Journal of King Saud University - Computer and Information Sciences*. **34**(8) : 6140–6149.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*. **8** : 1-74
- Ardiansyah, Widagdo, A. S., Qodri, K. N., Saputro, F. E. N., & Rizky, N. A. P. 2023. Analisis sentimen terhadap pelayanan Kesehatan berdasarkan ulasan Google Maps menggunakan BERT. *Jurnal Fasilkom*. **13**(2) : 326–333.
- Arifiyanti, A. A., Pradana, R. M., & Novian, I. F. 2018. Klasifikasi Produk Retour dengan Algoritma Pohon Keputusan C4.5. *Jurnal IPTEK*. **22**(1) : 79–86.
- Ariyulinda, N. 2018. Urgensi Pembentukan Regulasi Penjualan Obat Melalui Media Online. *Jurnal Legislasi Indonesia*. **15**(1) : 37–48.
- Braja, A. S. P., & Kodar, A. 2023. Implementasi Fine-Tuning BERT untuk Analisis Sentimen terhadap Review Aplikasi PUBG Mobile di Google Play Store. *JIMP - Jurnal Informatika Merdeka Pasuruan*. **7**(3) : 120–128.
- Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. 2017. *A Practical Guide to Sentiment Analysis*. Springer International Publishing.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. hlm, 4171-4186
- Eisenstein, J. 2018. *Natural Language Processing*. MIT Press.

- Gaikwad, S. V., Chaugule, A., & Patil, P. 2014. Text Mining Methods and Techniques. *International Journal of Computer Applications*. **85**(17) : 42–45.
- Hossin, M., & Sulaiman, M. N. 2015. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*. **5**(2) : 1–11.
- Kurniawan, B., Aldino, A. A., & Isnain, A. R. 2022. Sentimen Analisis Terhadap Kebijakan Penyelenggara Sistem Elektronik (Pse) Menggunakan Algoritma Bidirectional Encoder Representations From Transformers (Bert). *Jurnal Teknologi Dan Sistem Informasi*. **3**(4) : 98–106.
- Kusuma, R. M. R. W. P., & Yustanti, W. 2021. Analisis Sentimen Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from Transformers). *Journal of Emerging Information System and Business Intelligence (JEISBI)*. **2**(3) : 55–62.
- Lee, M. 2023. Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning. *Journal of Mathematics*. **2023** : 1–13.
- Li, Z., Fan, Y., Jiang, B., Lei, T., & Liu, W. 2019. A survey on sentiment analysis and opinion mining for social multimedia. *Multimedia Tools and Applications*. **78**(6) : 6939–6967.
- Mahmood, A. M. 2015. Class Imbalance Learning in Data Mining - A Survey. *International Journal of Communication Technology for Social Networking Services*. **3**(2) : 17–38.
- Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. 2021. Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem. *Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference*. hlm, 412-419.
- Muraina, I. O. 2022. Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts. *7th International Mardin Artuklu Scientific Researches Conference*. hlm, 496-504
- Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. 2017. Learning feature engineering for classification. *IJCAI International Joint Conference on Artificial Intelligence*. **17** : 2529–2535.
- Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. 2020. Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*. **14**(2) : 74–79.

- Nurhayatunnufus, L., Ridha, M. P., & Maulid, H. 2020. Lappybot : Chatbot Application for Information on Selecting Laptop Using the Natural Language Processing (NLP) Method. *E-Proceeding of Applied Science*. hlm, 2586-2594.
- Oyong, I., Utami, E., & Luthfi, E. T. 2018. Natural language processing and lexical approach for depression symptoms screening of Indonesian twitter user. *Proceedings of 2018 10th International Conference on Information Technology and Electrical Engineering: Smart Technology for Better Society (ICITEE) 2018*. hlm, 359-364.
- Pasaribu, D. J. M., Kusri, K., & Sudarmawan, S. 2020. Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan Bidirectional LSTM dan Bert Embedding. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*. **10**(1) : 9–20.
- Paul, S., & Saha, S. 2022. CyberBERT: BERT for cyberbullying identification: BERT for cyberbullying identification. *Multimedia Systems*. **28**(6) : 1897–1904.
- Persi, U. 2019. Sentiment analysis. *Annual Review of Statistics and Its Application*. **169**(1) : 80–91.
- Prusa, J. D., & Khoshgoftaar, T. M. 2017. Improving deep neural network design with new text data representations. *Journal of Big Data*. **4** (1) : 1-16.
- Rendragraha, A. D., Bijaksana, M. A., & Romadhony, A. 2021. Pendekatan Metode Transformers untuk Deteksi Bahasa Kasar dalam Komentar Berita Online Indonesia. *E-Proceeding of Engineering*. hlm, 3385-3395.
- Sharma, S., Sharma, S., & Athaiya, A. 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*. **4**(12) : 310–316.
- Soydaner, D. 2020. A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*. **34**(13) : 1-27.
- Surjandari, I., Megawati, C., Dhini, A., & Hardaya, I. B. N. S. 2016. Application of text mining for classification of textual reports: A study of Indonesia's national complaint handling system. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. hlm, 1147-1156.

- Syifa, S. A., & Dewi, I. A. 2022. Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru-Paru. *MIND (Multimedia Artificial Intelligent Networking Database) Journal*. **7**(2) : 139–150.
- Uzair, M., & Jamil, N. 2020. Effects of Hidden Layers on the Efficiency of Neural networks. *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*. hlm, 1-6.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., & Polosukhin, I. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems 2017 Conference*. hlm, 5998-6008.
- Yechuri, P. K., & Ramadass, S. 2021. Classification of Image and Text Data Using Deep Learning-Based LSTM Model. *International Information and Engineering Technology Association*. **38**(6) : 1809–1817.
- Yun, H. 2021. Prediction model of algal blooms using logistic regression and confusion matrix. *International Journal of Electrical and Computer Engineering*. **11**(3) : 2407–2413.

## **LAMPIRAN**

## Lampiran 1. Data Training

```
datatrain = pd.read_csv('/kaggle/input/drugreview/drug_review_train.csv')
datatrain
```

Unnamed: 0	patient_id	drugName	condition	review	rating	date	usefulCount	review_length	
0	0	89879	Cyclosporine	keratoconjunctivitis sicca	"i have used restasis for about a year now and...	2.0	April 20, 2013	69	147
1	1	143975	Etonogestrel	birth control	"my experience has been somewhat mixed. i have...	7.0	August 7, 2016	4	136
2	2	106473	Implanon	birth control	"this is my second implanon would not recommen...	1.0	May 11, 2016	6	140
3	3	184526	Hydroxyzine	anxiety	"i recommend taking as prescribed, and the bot...	10.0	March 19, 2012	124	104
4	4	91587	Dalfampridine	multiple sclerosis	"i have been on ampyra for 5 days and have bee...	9.0	August 1, 2010	101	74
...	...	...	...	...	...	...	...	...	...
110806	806	140111	Asenapine	bipolar disorde	"i had tried many medications to level out my ...	8.0	September 14, 2016	18	141
110807	807	103534	Synthroid	underactive thyroid	"had partial thyroid removal with benign tumor...	7.0	October 27, 2014	61	124
110808	808	161014	Risperidone	depression	"my experience with risperdal started when i f...	8.0	September 2, 2015	21	130
110809	809	132424	Ativan	anxiety	"my elderly dad was prescribed a low dose of a...	1.0	March 30, 2009	47	77
110810	810	229396	Lurasidone	schizophrenia	"latuda has been a hero in my battle against s...	10.0	August 12, 2016	15	119

110811 rows x 9 columns

## Lampiran 2. Data Validasi

```
dataval = pd.read_csv('/kaggle/input/drugreview/drug_review_validation.csv')
dataval
```

Unnamed: 0	patient_id	drugName	condition	review	rating	date	usefulCount	review_length	
0	0	191114	Campral	alcohol dependence	"sober a year 8-25-11. god, aa and campral hav...	10.0	September 3, 2011	33	41
1	1	142693	Levonorgestrel	birth control	"i've been on birth control for a while now du...	4.0	August 9, 2017	3	140
2	2	71561	Vraylar	bipolar disorde	"hi, this is an updated experience. VVVVVV...	8.0	August 16, 2016	12	131
3	3	25765	Ethinyl estradiol / norelgestromin	birth control	"i have been on the ortho evra patch for just ...	8.0	September 15, 2013	16	138
4	4	12843	Etanercept	psoriasis	"i have been on enbrel for 7 years and i have ...	9.0	August 5, 2010	9	65
...	...	...	...	...	...	...	...	...	...
27698	7698	176929	Adalimumab	psoriatic arthritis	"after my first and only shot, ten days after ...	1.0	October 18, 2017	1	144
27699	7699	153580	Velivet	birth control	"i have been on velivet for about a year. it h...	6.0	May 29, 2012	1	67
27700	7700	207011	Zarah	birth control	"i have been taking zarah for almost 3 months...	8.0	May 3, 2014	20	127
27701	7701	74831	Kepra	seizures	"i had my first seizure february 2014 and was ...	3.0	November 10, 2015	30	116
27702	7702	104299	Ethinyl estradiol / levonorgestrel	birth control	"i've been on aviane for about 4 weeks. i'm on...	4.0	May 3, 2012	2	67

27703 rows x 9 columns

## Lampiran 3. Data Testing

```
datatest = pd.read_csv('/kaggle/input/drugreview/drug_review_test.csv')
datatest
```

Unnamed: 0	patient_id	drugName	condition	review	rating	date	usefulCount	review_length	
0	0	163740	Mirtazapine	depression	"i've tried a few antidepressants over the yea...	10.0	February 28, 2012	22	68
1	1	206473	Mesalamine	crohn's disease, maintenance	"my son has crohn's disease and has done very ...	8.0	May 17, 2009	17	48
2	2	39293	Contrave	weight loss	"contrave combines drugs that were used for al...	9.0	March 5, 2017	35	143
3	3	97768	Cyclafem 1 / 35	birth control	"i have been on this birth control for one cyc...	9.0	October 22, 2015	4	149
4	4	208087	Zyclara	keratosis	"4 days in on first 2 weeks. using on arms an...	4.0	July 3, 2014	13	60
...	...	...	...	...	...	...	...	...	...
46103	6103	123432	Apri	birth control	"i started taking apri about 7 months ago. my ...	9.0	August 25, 2010	18	86
46104	6104	159999	Tamoxifen	breast cancer, prevention	"i have taken tamoxifen for 5 years. side effe...	10.0	September 13, 2014	43	97
46105	6105	140714	Escitalopram	anxiety	"i've been taking lexapro (escitalopprgram) sl...	9.0	October 8, 2016	11	130
46106	6106	130945	Levonorgestrel	birth control	"i'm married, 34 years old and i have no kids...	8.0	November 15, 2010	7	149
46107	6107	47656	Tapentadol	pain	"i was prescribed nucynta for severe neck/shou...	1.0	November 28, 2011	20	34

46108 rows x 9 columns



## Lampiran 4. Data yang Telah Digabung

```
frames = [datatrain, dataval, datatest]
df=pd.concat(frames)
df
```

Unnamed: 0	patient_id	drugName	condition	review	rating	date	usefulCount	review_length	
0	0	89879	Cyclosporine	keratoconjunctivitis sicca	"i have used restasis for about a year now and...	2.0	April 20, 2013	69	147
1	1	143975	Etonogestrel	birth control	"my experience has been somewhat mixed. i have...	7.0	August 7, 2016	4	136
2	2	106473	Implanon	birth control	"this is my second implanon would not recommen...	1.0	May 11, 2016	6	140
3	3	184526	Hydroxyzine	anxiety	"i recommend taking as prescribed, and the bot...	10.0	March 19, 2012	124	104
4	4	91587	Dalfampridine	multiple sclerosis	"i have been on ampyra for 5 days and have bee...	9.0	August 1, 2010	101	74
...	...	...	...	...	...	...	...	...	...
46103	6103	123432	Apri	birth control	"i started taking apri about 7 months ago. my ...	9.0	August 25, 2010	18	86
46104	6104	159999	Tamoxifen	breast cancer, prevention	"i have taken tamoxifen for 5 years. side effe...	10.0	September 13, 2014	43	97
46105	6105	140714	Escitalopram	anxiety	"i've been taking lexapro (escitaloprogram) sl...	9.0	October 8, 2016	11	130
46106	6106	130945	Levonorgestrel	birth control	"i'm married, 34 years old and i have no kids...	8.0	November 15, 2010	7	149
46107	6107	47656	Tapentadol	pain	"i was prescribed nucynta for severe neck/shou...	1.0	November 28, 2011	20	34

184622 rows x 9 columns

## Lampiran 5. Data yang Telah Diseleksi Variabel

```
df = df.drop(['Unnamed: 0', 'patient_id', 'drugName', 'condition', 'date', 'usefulCount', 'review_length'], axis=1)
df
```

	review	rating
0	"i have used restasis for about a year now and...	2.0
1	"my experience has been somewhat mixed. i have...	7.0
2	"this is my second implanon would not recommen...	1.0
3	"i recommend taking as prescribed, and the bot...	10.0
4	"i have been on ampyra for 5 days and have bee...	9.0
...	...	...
46103	"i started taking apri about 7 months ago. my ...	9.0
46104	"i have taken tamoxifen for 5 years. side effe...	10.0
46105	"i've been taking lexapro (escitaloprogram) sl...	9.0
46106	"i'm married, 34 years old and i have no kids...	8.0
46107	"i was prescribed nucynta for severe neck/shou...	1.0

184622 rows x 2 columns

## Lampiran 6. Perbandingan Teks Sebelum dan Sesudah Diolah

Sebelum <i>Case Folding</i> dan <i>Filtering</i>	Setelah <i>Case Folding</i> dan <i>Filtering</i>
“i was prescribed nucynta for severe neck/shoulder pain. after taking only 2 , 75mg pills i was rushed to the er with severe breathing problems. i have never had any issues with pain medicines before.”	i was prescribed nucynta for severe neckshoulder pain after taking only mg pills i was rushed to the er with severe breathing problems i have never had any issues with pain medicines before

### Lampiran 7. Data yang Telah Di-*Preprocessing* Teks

	<b>review</b>	<b>rating</b>
<b>0</b>	i have used restasis for about a year now and ...	2.0
<b>1</b>	my experience has been somewhat mixed i have b...	7.0
<b>2</b>	this is my second implanon would not recommend...	1.0
<b>3</b>	i recommend taking as prescribed and the bottl...	10.0
<b>4</b>	i have been on ampyra for days and have been s...	9.0
...	...	...
<b>46103</b>	i started taking apri about months ago my brea...	9.0
<b>46104</b>	i have taken tamoxifen for years side effects ...	10.0
<b>46105</b>	ive been taking lexapro escitaploprgram since ...	9.0
<b>46106</b>	im married years old and i have no kids taking...	8.0
<b>46107</b>	i was prescribed nucynta for severe neckshould...	1.0

184622 rows × 2 columns

### Lampiran 8. Data yang Telah Dilakukan Pelabelan

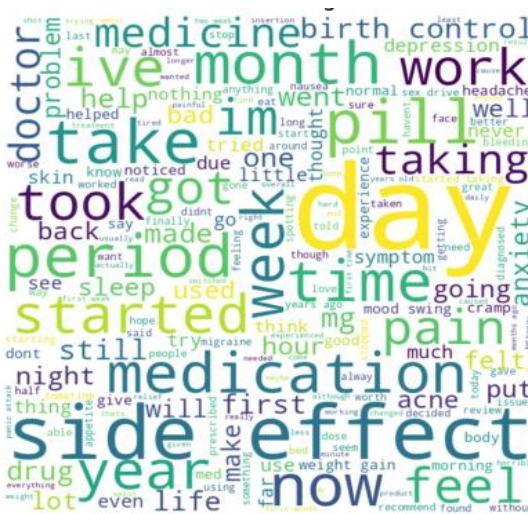
	<b>review</b>	<b>rating</b>	<b>sentimen_textblob</b>
<b>0</b>	i have used restasis for about a year now and ...	1.0	netral
<b>1</b>	my experience has been somewhat mixed i have b...	4.0	positif
<b>2</b>	this is my second implanon would not recommend...	1.0	positif
<b>3</b>	i recommend taking as prescribed and the bottl...	5.0	netral
<b>4</b>	i have been on ampyra for days and have been s...	5.0	positif
...	...	...	...
<b>46103</b>	i started taking apri about months ago my brea...	5.0	netral
<b>46104</b>	i have taken tamoxifen for years side effects ...	5.0	netral
<b>46105</b>	ive been taking lexapro escitaploprgram since ...	5.0	netral
<b>46106</b>	im married years old and i have no kids taking...	4.0	netral
<b>46107</b>	i was prescribed nucynta for severe neckshould...	1.0	netral

184622 rows × 3 columns







Lampiran 16. *Wordcloud* Kata Pada Kelas Sentimen NegatifLampiran 17. *Wordcloud* Kata Pada Kelas Sentimen Netral

Lampiran 18. *Wordcloud* Kata Pada Kelas Sentimen PositifLampiran 19. Data yang Telah Dilakukan *Label Encoder*

	review	rating	sentimen_textblob
0	i have used restasis for about a year now and ...	0	1
1	my experience has been somewhat mixed i have b...	3	2
2	this is my second implanon would not recommend...	0	2
3	i recommend taking as prescribed and the bottl...	4	1
4	i have been on ampyra for days and have been s...	4	2
...	...	...	...
46103	i started taking apri about months ago my brea...	4	1
46104	i have taken tamoxifen for years side effects ...	4	1
46105	ive been taking lexapro escitaploprgram since ...	4	1
46106	im married years old and i have no kids taking...	3	1
46107	i was prescribed nucynta for severe neckshould...	0	1

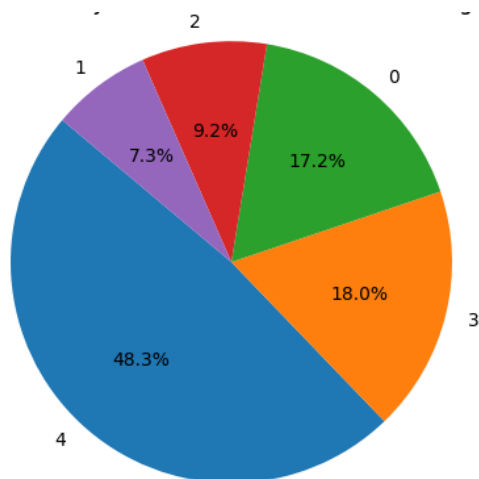
184622 rows × 3 columns

Lampiran 20. Kelas *Rating* yang Telah Diterapkan *Label Encoding*

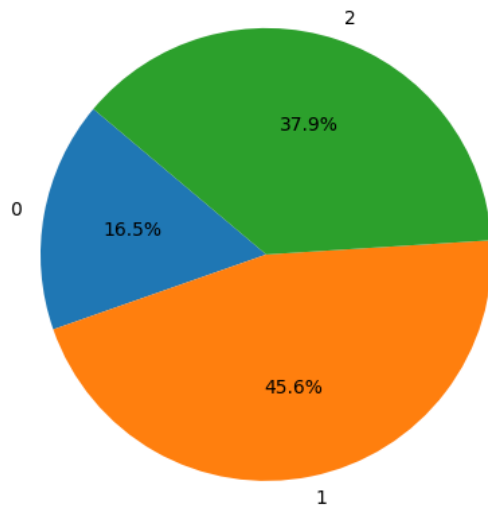
<i>Output</i> Sebelum Diterapkan <i>Label Endoding</i>	<i>Output</i> Setelah Diterapkan <i>Label Endoding</i>
1	0
2	1
3	2
4	3
5	4

Lampiran 21. Kelas *Textblob* yang Telah Diterapkan *Label Encoding*

<i>Output</i> Sebelum Diterapkan <i>Label Endoding</i>	<i>Output</i> Setelah Diterapkan <i>Label Endoding</i>
Negatif	0
Netral	1
Postif	2

Lampiran 22. *Piechart* Sebaran Kelas Berdasarkan *Rating*



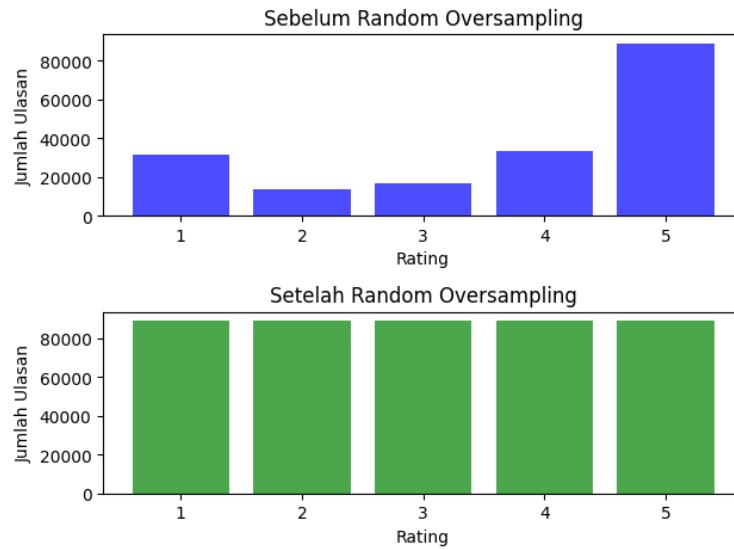
Lampiran 23. Piechart Sebaran Kelas Berdasarkan *Textblob*Lampiran 24. Sebaran Kelas Berdasarkan *Rating*

<i>Output</i>	Frekuensi
0	31824
1	13467
2	16940
3	33208
4	89183

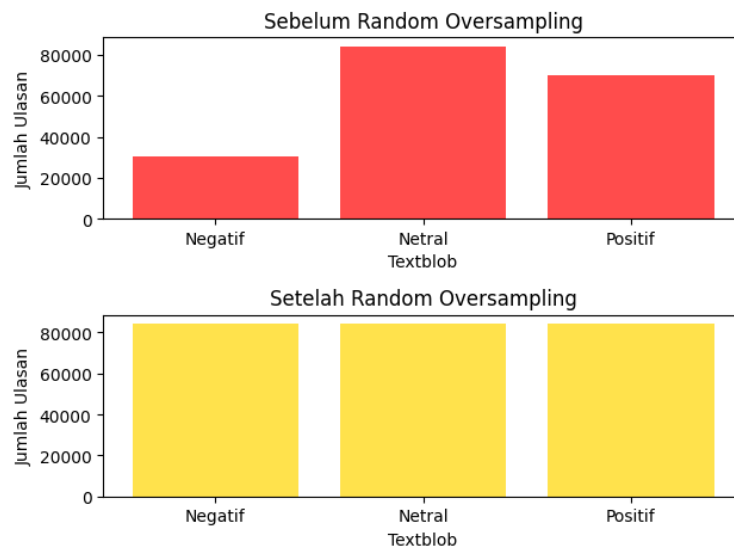
Lampiran 25. Sebaran Kelas Berdasarkan *Textblob*

<i>Output</i>	Frekuensi
0	30423
1	84258
2	69941

Lampiran 26. Histogram Sebaran Kelas Berdasarkan *Rating* Sebelum dan Setelah ROS



Lampiran 27. Histogram Sebaran Kelas Berdasarkan *Textblob* Sebelum dan Setelah ROS



Lampiran 28. Sebaran Kelas Berdasarkan *Rating* Setelah ROS

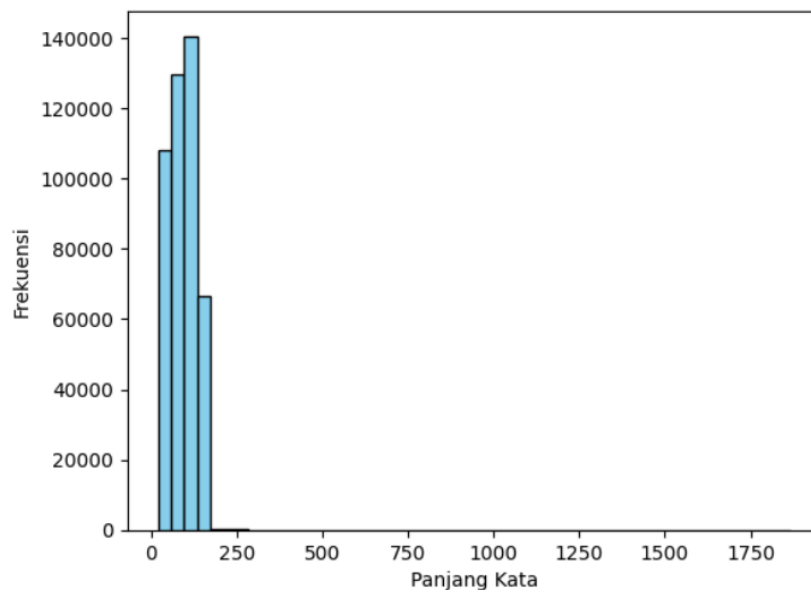
<i>Output</i>	Frekuensi
0	89183
1	89183
2	89183
3	89183
4	89183

92

Lampiran 29. Sebaran Kelas Berdasarkan *Rating* Setelah ROS

<i>Output</i>	Frekuensi
0	84258
1	84258
2	84258

Lampiran 30. Sebaran Panjang Kata Data Ulasan





Lampiran 32. Hasil Pembagian Data Berdasarkan *Rating*

<i>Training Data</i>		<i>Validation Data</i>		<i>Testing Data</i>	
<i>Input IDS</i>	361190	<i>Input IDS</i>	40133	<i>Input IDS</i>	44592
<i>Mask</i>	361190	<i>Mask</i>	40133	<i>Mask</i>	44592
<i>Label</i>	361190	<i>Label</i>	40133	<i>Label</i>	44592

Lampiran 33. Hasil Pembagian Data Berdasarkan *Textblob*

<i>Training Data</i>		<i>Validation Data</i>		<i>Testing Data</i>	
<i>Input IDS</i>	204746	<i>Input IDS</i>	22750	<i>Input IDS</i>	25278
<i>Mask</i>	204746	<i>Mask</i>	22750	<i>Mask</i>	25278
<i>Label</i>	204746	<i>Label</i>	22750	<i>Label</i>	25278

Lampiran 34. *Tensor Dataset Training* Berdasarkan *Rating*

```

Inputs: tensor([[ 101, 2026, 2905, ..., 0, 0, 0],
               [ 101, 4921, 2063, ..., 0, 0, 0],
               [ 101, 2044, 2635, ..., 0, 0, 0],
               ...,
               [ 101, 1045, 2572, ..., 0, 0, 0],
               [ 101, 1045, 2001, ..., 0, 0, 0],
               [ 101, 2023, 14667, ..., 0, 0, 0]])

Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               ...,
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0]])

Labels: tensor([2, 3, 2, 1, 3, 2, 4, 0, 4, 0, 3, 3, 3, 1, 3, 3])

```

### Lampiran 35. *Tensor Dataset Validation* Berdasarkan *Rating*

```

Inputs: tensor([[ 101, 2061, 2005, ..., 0, 0, 0],
               [ 101, 4921, 2063, ..., 0, 0, 0],
               [ 101, 1045, 2001, ..., 0, 0, 0],
               ...,
               [ 101, 1045, 2031, ..., 0, 0, 0],
               [ 101, 1045, 2031, ..., 0, 0, 0],
               [ 101, 2651, 2001, ..., 0, 0, 0]])
Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               ...,
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0]])
Labels: tensor([2, 3, 0, 0, 3, 3, 3, 2, 4, 3, 1, 0, 2, 3, 4, 2])

```

### Lampiran 36. *Tensor Dataset Testing* Berdasarkan *Rating*

```

Inputs: tensor([[ 101, 1045, 2572, ..., 0, 0, 0],
               [ 101, 22597, 2573, ..., 0, 0, 0],
               [ 101, 1045, 2031, ..., 0, 0, 0],
               ...,
               [ 101, 1045, 2001, ..., 0, 0, 0],
               [ 101, 1045, 2288, ..., 0, 0, 0],
               [ 101, 1045, 2001, ..., 0, 0, 0]])
Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               ...,
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0]])
Labels: tensor([0, 1, 3, 2, 3, 4, 4, 0, 0, 2, 4, 3, 4, 1, 3, 2])

```

### Lampiran 37. *Tensor Dataset Training* Berdasarkan *Textblob*

```

Inputs: tensor([[ 101, 2023, 2001, ..., 0, 0, 0],
               [ 101, 1045, 2031, ..., 0, 0, 0],
               [ 101, 1045, 2031, ..., 0, 0, 0],
               ...,
               [ 101, 2106, 2025, ..., 0, 0, 0],
               [ 101, 1045, 2245, ..., 0, 0, 0],
               [ 101, 1045, 2001, ..., 0, 0, 0]])
Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               ...,
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0],
               [1, 1, 1, ..., 0, 0, 0]])
Labels: tensor([2, 2, 2, 0, 2, 0, 2, 1, 0, 2, 2, 1, 2, 0, 0, 2])

```

### Lampiran 38. *Tensor Dataset Validation Berdasarkan Textblob*

```

Inputs: tensor([[ 101, 7632, 6456, ..., 0, 0, 0],
 [ 101, 1045, 2031, ..., 0, 0, 0],
 [ 101, 6919, 4921, ..., 0, 0, 0],
 ...,
 [ 101, 1045, 2031, ..., 0, 0, 0],
 [ 101, 1045, 2001, ..., 0, 0, 0],
 [ 101, 1045, 2165, ..., 0, 0, 0]])
Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 ...,
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0]])
Labels: tensor([1, 1, 2, 2, 0, 1, 2, 2, 0, 2, 0, 1, 2, 2, 1, 1])

```

### Lampiran 39. *Tensor Dataset Testing Berdasarkan Textblob*

```

Inputs: tensor([[ 101, 2033, 1998, ..., 0, 0, 0],
 [ 101, 1045, 2031, ..., 0, 0, 0],
 [ 101, 1045, 9544, ..., 0, 0, 0],
 ...,
 [ 101, 2398, 2091, ..., 0, 0, 0],
 [ 101, 1045, 2572, ..., 0, 0, 0],
 [ 101, 2026, 29610, ..., 0, 0, 0]])
Masks: tensor([[1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 ...,
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0],
 [1, 1, 1, ..., 0, 0, 0]])
Labels: tensor([1, 2, 2, 0, 2, 2, 2, 2, 0, 1, 0, 1, 2, 2, 0, 0])

```

### Lampiran 40. Kombinasi Parameter BERT

<i>Batch Size</i>	16, 32, 64
<i>Learning Rate</i>	$1 \times 10^{-6}$ , $3 \times 10^{-6}$ , $1 \times 10^{-5}$ , $2 \times 10^{-5}$ , $3 \times 10^{-5}$ , $4 \times 10^{-5}$ , $5 \times 10^{-5}$

## Lampiran 41. Model BERT untuk Klasifikasi *Rating*

```

The BERT model has 201 different named parameters.
==== Embedding Layer ====
bert.embeddings.word_embeddings.weight          (30522, 768)
bert.embeddings.position_embeddings.weight      (512, 768)
bert.embeddings.token_type_embeddings.weight    (2, 768)
bert.embeddings.LayerNorm.weight              (768,)
bert.embeddings.LayerNorm.bias                (768,)
==== First Transformers ====
bert.encoder.layer.0.attention.self.query.weight (768, 768)
bert.encoder.layer.0.attention.self.query.bias (768,)
bert.encoder.layer.0.attention.self.key.weight (768, 768)
bert.encoder.layer.0.attention.self.key.bias   (768,)
bert.encoder.layer.0.attention.self.value.weight (768, 768)
bert.encoder.layer.0.attention.self.value.bias (768,)
bert.encoder.layer.0.attention.output.dense.weight (768, 768)
bert.encoder.layer.0.attention.output.dense.bias (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias (768,)
bert.encoder.layer.0.intermediate.dense.weight (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias   (3072,)
bert.encoder.layer.0.output.dense.weight       (768, 3072)
bert.encoder.layer.0.output.dense.bias         (768,)
bert.encoder.layer.0.output.LayerNorm.weight   (768,)
bert.encoder.layer.0.output.LayerNorm.bias     (768,)
==== Output Layer ====
bert.pooler.dense.weight                       (768, 768)
bert.pooler.dense.bias                         (768,)
classifier.weight                              (5, 768)
classifier.bias                                (5,)

```



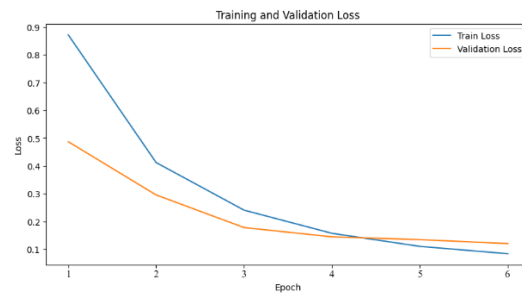
## Lampiran 42. Model BERT untuk Klasifikasi *Textblob*

```

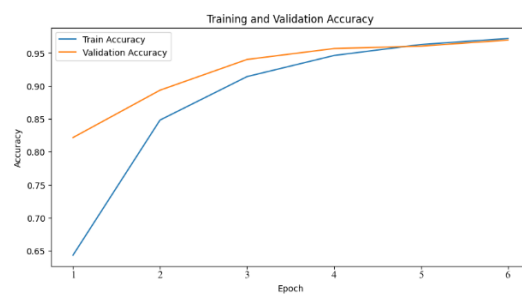
The BERT model has 201 different named parameters.
==== Embedding Layer ====
bert.embeddings.word_embeddings.weight          (30522, 768)
bert.embeddings.position_embeddings.weight      (512, 768)
bert.embeddings.token_type_embeddings.weight    (2, 768)
bert.embeddings.LayerNorm.weight              (768,)
bert.embeddings.LayerNorm.bias                (768,)
==== First Transformers ====
bert.encoder.layer.0.attention.self.query.weight (768, 768)
bert.encoder.layer.0.attention.self.query.bias (768,)
bert.encoder.layer.0.attention.self.key.weight (768, 768)
bert.encoder.layer.0.attention.self.key.bias   (768,)
bert.encoder.layer.0.attention.self.value.weight (768, 768)
bert.encoder.layer.0.attention.self.value.bias (768,)
bert.encoder.layer.0.attention.output.dense.weight (768, 768)
bert.encoder.layer.0.attention.output.dense.bias (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias (768,)
bert.encoder.layer.0.intermediate.dense.weight (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias   (3072,)
bert.encoder.layer.0.output.dense.weight       (768, 3072)
bert.encoder.layer.0.output.dense.bias         (768,)
bert.encoder.layer.0.output.LayerNorm.weight  (768,)
bert.encoder.layer.0.output.LayerNorm.bias    (768,)
==== Output Layer ====
bert.pooler.dense.weight                       (768, 768)
bert.pooler.dense.bias                         (768,)
classifier.weight                              (3, 768)
classifier.bias                                (3,)

```

## Lampiran 43. *Training dan Validation Loss* Berdasarkan *Rating*

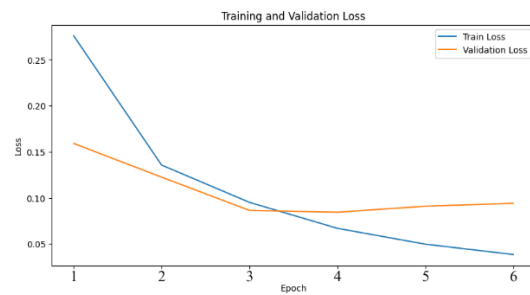
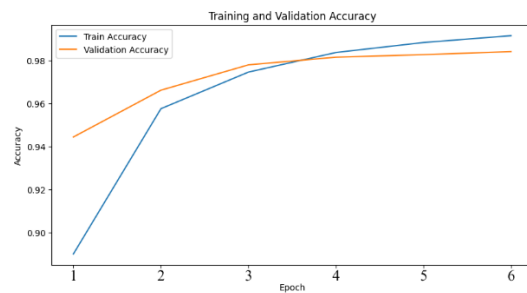


## Lampiran 44. *Training dan Validation Accuracy* Berdasarkan *Rating*



Lampiran 45. Performa Model Klasifikasi BERT Berdasarkan *Rating*

<b><i>Epoch</i></b>	<b><i>Training Loss</i></b>	<b><i>Validation Loss</i></b>	<b><i>Training Accuracy</i></b>	<b><i>Validation Accuracy</i></b>
<b>1</b>	0,8724	0,4872	0,6432	0,8216
<b>2</b>	0,4120	0,2954	0,8481	0,8934
<b>3</b>	0,2406	0,1778	0,9140	0,9400
<b>4</b>	0,1571	0,1444	0,9460	0,9566
<b>5</b>	0,1100	0,1343	0,9627	0,9602
<b>6</b>	0,0835	0,1202	0,9718	0,9694

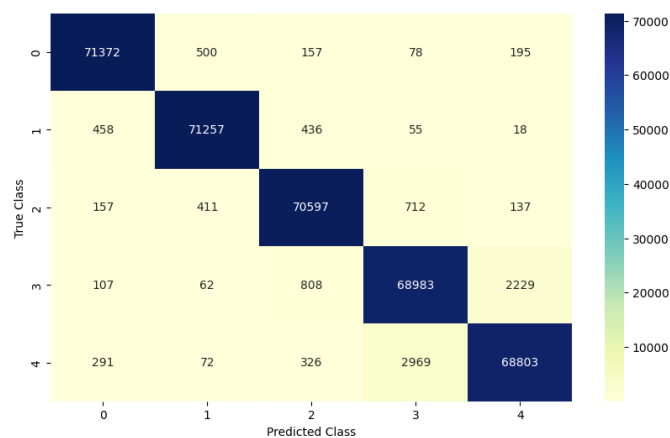
Lampiran 46. *Training dan Validation Loss* Berdasarkan *Textblob*Lampiran 47. *Training dan Validation Accuracy* Berdasarkan *Textblob*

Lampiran 48. Performa Model Klasifikasi BERT Berdasarkan *Textblob*

<i>Epoch</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>
<b>1</b>	0,2759	0,1591	0,8901	0,9444
<b>2</b>	0,1356	0,1225	0,9576	0,9662
<b>3</b>	0,0951	0,0864	0,9746	0,9780
<b>4</b>	0,0670	0,0844	0,9837	0,9815
<b>5</b>	0,0498	0,0910	0,9884	0,9827
<b>6</b>	0,0385	0,0942	0,9915	0,9841

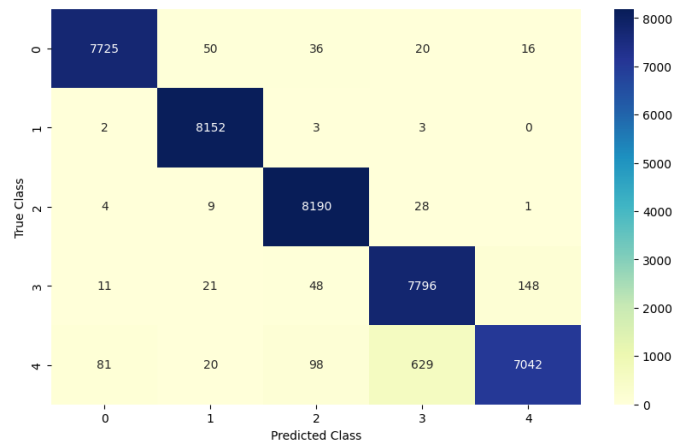
Lampiran 49. Performa Model Klasifikasi BERT *Rating* pada Data *Training*

	precision	recall	f1-score	support
0	0.99	0.99	0.99	72302
1	0.99	0.99	0.99	72224
2	0.98	0.98	0.98	72014
3	0.95	0.96	0.95	72189
4	0.96	0.95	0.96	72461
accuracy			0.97	361190
macro avg	0.97	0.97	0.97	361190
weighted avg	0.97	0.97	0.97	361190

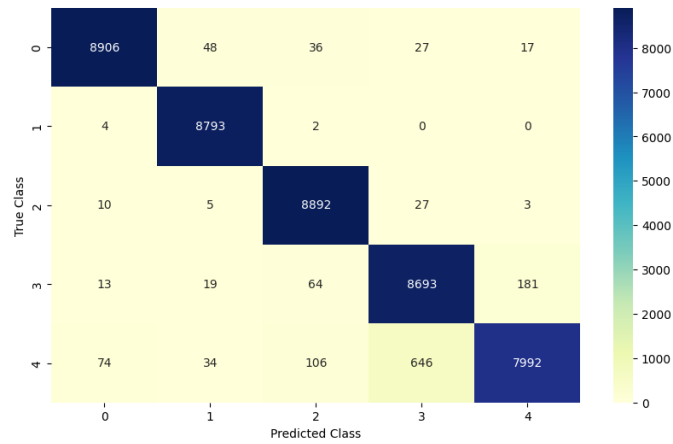
Lampiran 50. *Confusion Matrix* dari Klasifikasi *Rating* pada Data *Training*

Lampiran 51. Performa Model Klasifikasi BERT *Rating* pada Data Validasi

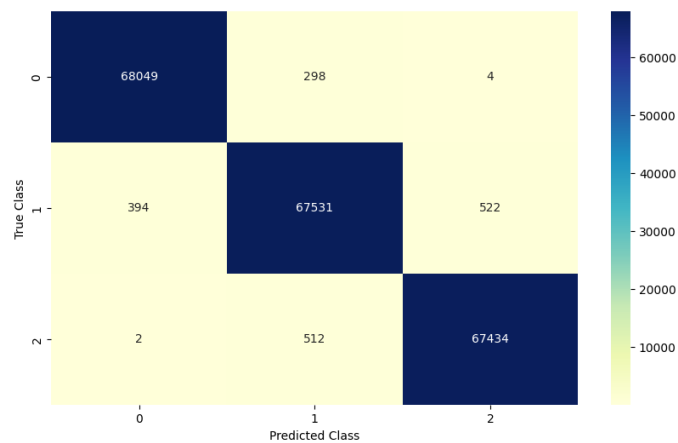
	precision	recall	f1-score	support
0	0.99	0.98	0.99	7847
1	0.99	1.00	0.99	8160
2	0.98	0.99	0.99	8232
3	0.92	0.97	0.94	8024
4	0.98	0.89	0.93	7870
accuracy			0.97	40133
macro avg	0.97	0.97	0.97	40133
weighted avg	0.97	0.97	0.97	40133

Lampiran 52. *Confusion Matrix* dari Klasifikasi *Rating* pada Data ValidasiLampiran 53. Performa Model Klasifikasi BERT *Rating* pada Data *Testing*

	precision	recall	f1-score	support
0	0.99	0.99	0.99	9034
1	0.99	1.00	0.99	8799
2	0.98	0.99	0.99	8937
3	0.93	0.97	0.95	8970
4	0.98	0.90	0.94	8852
accuracy			0.97	44592
macro avg	0.97	0.97	0.97	44592
weighted avg	0.97	0.97	0.97	44592

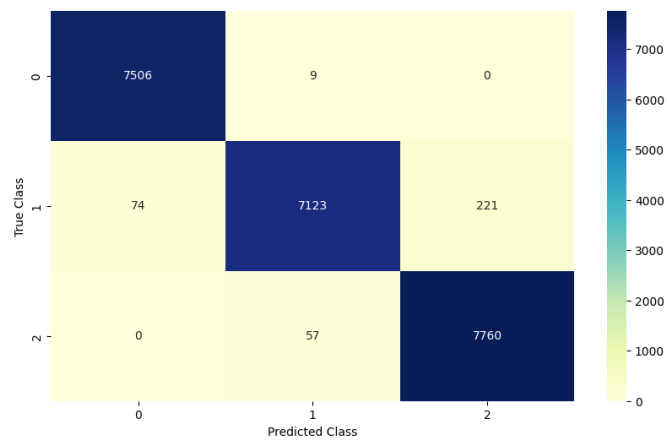
Lampiran 54. *Confusion Matrix* dari Klasifikasi *Rating* pada *Data Testing*Lampiran 55. Performa Model Klasifikasi BERT *Textblob* pada *Data Training*

	precision	recall	f1-score	support
0	0.99	1.00	0.99	68351
1	0.99	0.99	0.99	68447
2	0.99	0.99	0.99	67948
accuracy			0.99	204746
macro avg	0.99	0.99	0.99	204746
weighted avg	0.99	0.99	0.99	204746

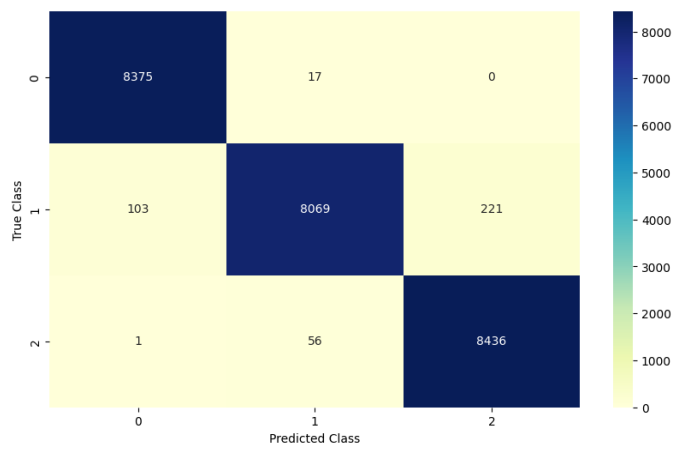
Lampiran 56. *Confusion Matrix* dari Klasifikasi *Textblob* pada *Data Training*

Lampiran 57. Performa Model Klasifikasi BERT *Textblob* pada Data Validasi

	precision	recall	f1-score	support
0	0.99	1.00	0.99	7515
1	0.99	0.96	0.98	7418
2	0.97	0.99	0.98	7817
accuracy			0.98	22750
macro avg	0.98	0.98	0.98	22750
weighted avg	0.98	0.98	0.98	22750

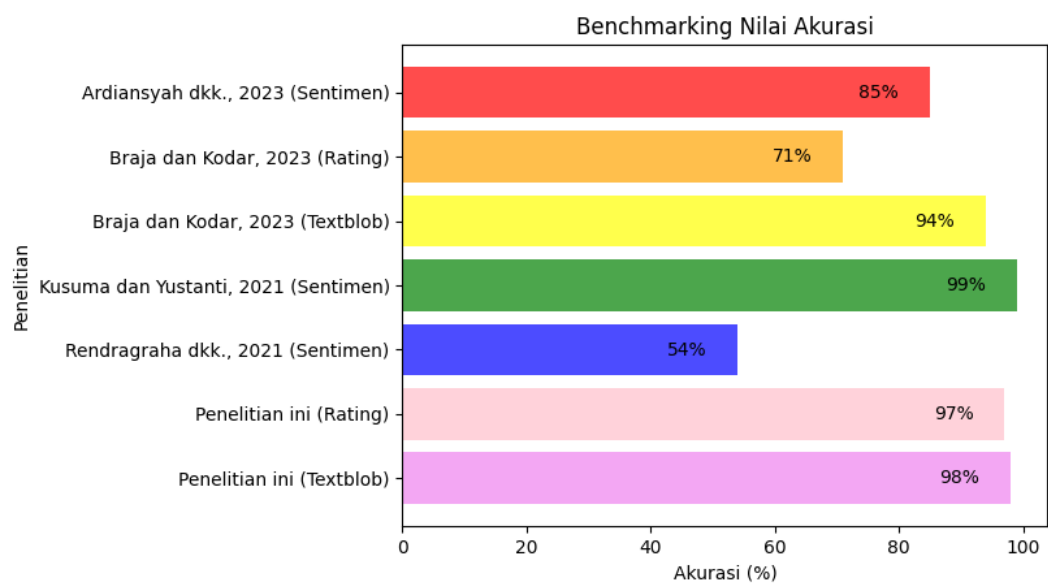
Lampiran 58. *Confusion Matrix* dari Klasifikasi *Textblob* pada Data ValidasiLampiran 59. Performa Model Klasifikasi BERT *Textblob* pada Data *Testing*

	precision	recall	f1-score	support
0	0.99	1.00	0.99	8392
1	0.99	0.96	0.98	8393
2	0.97	0.99	0.98	8493
accuracy			0.98	25278
macro avg	0.98	0.98	0.98	25278
weighted avg	0.98	0.98	0.98	25278

Lampiran 60. *Confusion Matrix* dari Klasifikasi *Textblob* pada Data *Testing*

Lampiran 61. Hasil Klasifikasi BERT

	<i>Accuracy Training</i>	<i>Accuracy Validation</i>	<i>Accuracy Testing</i>
<i>Rating</i>	0,9718	0,9694	0,9705
<i>Textblob</i>	0,9915	0,9841	0,9843

Lampiran 62. Histogram *Benchmarking* Hasil Penelitian

Lampiran 63. *Benchmarking* Hasil Penelitian

No	Penelitian	Hasil			
		<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
1.	Ardiansyah dkk., 2023. Data Ulasan Pelayanan Kesehatan dari Google Maps.	Sentimen 85%	Sentimen 77%	Sentimen 74%	Sentimen 85%
2.	Braja dan Kodar, 2023. Data Ulasan Aplikasi PUBG <i>Mobile</i> dari Google Play Store.	<i>Rating</i> 71%	<i>Rating</i> -	<i>Rating</i> -	<i>Rating</i> -
		<i>Textblob</i> 94%	<i>Textblob</i> 94%	<i>Textblob</i> 93%	<i>Textblob</i> 93%
3.	Kusuma dan Yustanti, 2021. Data Ulasan Aplikasi Ruang Guru dari Google Play Store.	Sentimen 99%	Sentimen 64.13%	Sentimen 60.51%	Sentimen 98.9%
4.	Rendragraha dkk., 2021. Data Komentar Berita <i>Online</i> dari <i>Website</i> Kompas dan Tempo	Sentimen 54%	-	-	-
5.	Penelitian ini. Data Ulasan Obat dari <i>Website</i> Kaggle.	<i>Rating</i> 97%	<i>Rating</i> 97%	<i>Rating</i> 97%	<i>Rating</i> 97%
		<i>Textblob</i> 98%	<i>Textblob</i> 98%	<i>Textblob</i> 98%	<i>Textblob</i> 98%