OPTIMALISASI *FINE-TUNING* MODEL BIOBART MENGGUNAKAN ADAMW *OPTIMIZER* DALAM PERINGKASAN TEKS OTOMATIS PADA ARTIKEL ILMIAH BIOMEDIS DENGAN EVALUASI ROUGE DAN BERTSCORE

Skripsi

Oleh

MEILINA RISDIANTI
NPM. 2217031070

MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRACT

OPTIMIZATION OF BIOBART MODEL FINE-TUNING USING ADAMW OPTIMIZER IN AUTOMATIC TEXT SUMMARIZATION OF BIOMEDICAL SCIENTIFIC ARTICLES WITH ROUGE AND BERTSCORE

By

Meilina Risdianti

In general, health and biomedical articles have technical and specialized content that makes it difficult for the general public or non-experts to understand and filter information in articles that tend to be long and complex. The increasing number and complexity of biomedical literature, especially in long scientific documents, makes NLP (Natural Language Processing) an important approach in decision making. NLP is capable of making long texts more concise, relevant, and informative through automatic text summarization. In this study, the BioBART model was optimized using the AdamW Optimizer with a fine-tuning process to improve the quality of summaries of long biomedical articles by utilizing the PMC (PubMed Central) dataset, which is known to have a high level of heterogeneity in terms of document length, article structure variation, and terminology complexity. A dataset of 10,000 articles, with 90% for the model and 10% for testing. Of the 9,000 model data, 75% is training data and 25% is validation data. The research process consisted of a preprocessing stage involving text cleaning, normalization, and the application of a sliding window to handle long documents. Then, the BioBART model was fine-tuned by exploring hyperparameters to obtain the best model configuration. Model evaluation was performed using the ROUGE and BERTScore metrics. The results of the study show that the fine-tuned and hyperparameter-optimized BioBART model is capable of producing better and more competitive summaries than the pretrained BioBART model, both lexically and semantically, despite the complexity of long documents and heterogeneity in the PMC dataset.

Keywords: PMC, Biomedical Articles, NLP, BioBART Model, AdamW Optimizer, Fine-Tuning, ROUGE, BERTScore, Automatic Text Summarization.

ABSTRAK

OPTIMALISASI *FINE-TUNING* MODEL BIOBART MENGGUNAKAN ADAMW *OPTIMIZER* DALAM PERINGKASAN TEKS OTOMATIS PADA ARTIKEL ILMIAH BIOMEDIS DENGAN EVALUASI ROUGE DAN BERTSCORE

Oleh

Meilina Risdianti

Secara umum, artikel kesehatan dan biomedis memiliki konten teknis dan spesialis yang menyebabkan pembaca non-ahli kesulitan memahami dan menyaring informasi dalam artikel yang cenderung panjang dan kompleks. Meningkatnya jumlah dan kompleksitas literatur biomedis, khususnya pada dokumen ilmiah yang panjang (*long documents*), menjadikan NLP (*Natural Language Processing*) sebagai pendekatan dalam pengambilan keputusan. NLP mampu menjadikan teks panjang menjadi lebih ringkas, relevan, dan informatif melalui *automatic text summarization*. Dalam penelitian ini, model BioBART dioptimalkan menggunakan AdamW *Optimizer* dengan *fine-tuning* untuk meningkatkan kualitas ringkasan teks artikel biomedis panjang dengan memanfaatkan dataset PMC (*PubMed Central*) yang dikenal memiliki tingkat heterogenitas tinggi dari sisi panjang dokumen, variasi struktur artikel, dan kompleksitas terminologi. Data sebanyak 10000 artikel, dengan 90% untuk model dan 10% pengujian. Dari 90% data model dibagi menjadi 75% data *training* dan 25% data *validation*. Proses penelitian terdiri dari tahap *preprocessing* yaitu pembersihan teks, normalisasi, serta penerapan *sliding window* untuk menangani dokumen panjang. Kemudian proses *fine-tuning* model BioBART dengan eksplorasi *hyperparameter* untuk memperoleh konfigurasi model terbaik. Evaluasi model dilakukan menggunakan metrik ROUGE dan BERTScore. Hasil penelitian menunjukkan bahwa model BioBART hasil *fine-tuning* dan optimalisasi hyperparameter mampu menghasilkan ringkasan yang lebih baik dan kompetitif dibandingkan model BioBART *pretrained*, baik secara leksikal maupun semantik, meskipun dihadapkan pada kompleksitas dokumen panjang dan heterogenitas pada dataset PMC.

Kata-kata kunci: PMC, Artikel Biomedis, NLP, Model BioBART, AdamW *Optimizer*, *Fine-Tuning*, ROUGE, BERTScore, *Automatic Text Summarization*.

OPTIMALISASI *FINE-TUNING* MODEL BIOBART MENGGUNAKAN ADAMW *OPTIMIZER* DALAM PERINGKASAN TEKS OTOMATIS PADA ARTIKEL ILMIAH BIOMEDIS DENGAN EVALUASI ROUGE DAN BERTSCORE

MEILINA RISDIANTI

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam

MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

Judul Skripsi	:	**OPTIMALISASI *FINE-TUNING* MODEL BIOBART MENGGUNAKAN ADAMW *OPTIMIZER* DALAM PERINGKASAN TEKS OTOMATIS PADA ARTIKEL ILMIAH BIOMEDIS DENGAN EVALUASI ROUGE DAN BERTSCORE**
Nama Mahasiswa	:	**Meilina Risdianti**
Nomor Pokok Mahasiswa	:	**2217031070**
Program Studi	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. Dian Kurniasari, S.Si., M.Sc.
NIP 196903051996023001

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, S.kom., M.Si., Ph.D.** .....

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

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

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

Tanggal Lulus Ujian Skripsi: 12 Maret 2026

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama	:	**Meilina Risdianti**
Nomor Pokok Mahasiswa	:	**2217031070**
Jurusan	:	**Matematika**
Judul Skripsi	:	**Optimalisasi *Fine-Tuning* Model BioBART Menggunakan AdamW *Optimizer* Dalam Peringkasan Teks Otomatis Pada Artikel Ilmiah Biomedis Dengan Evaluasi ROUGE Dan BERTScore**

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

Bandar Lampung, 12 Maret 2026

Penulis

Meilina Risdianti

NPM. 2217031070

RIWAYAT HIDUP

Penulis memiliki nama lengkap Meilina Risdianti yang lahir di OKU Timur Provinsi Sumatera Selatan pada tanggal 14 Mei 2004. Penulis merupakan anak pertama dari empat bersaudara dari pasangan Bapak Mujiyanto dan Ibu Mariyem.

Penulis mulai menempuh pendidikan di Taman Kanak-kanak Pendidikan Anak Usia Dini (TK-PAUD) Ratu Ibu pada tahun 2008-2010, dan dilanjutkan di Sekolah Dasar di SD Negeri 1 Tugu Harum dari tahun 2010-2016. Penulis melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 1 Belitang pada tahun 2016-2019, kemudian melanjutkan ke jenjang menengah atas di SMA Negeri 1 Belitang pada tahun 2019-2022. Pada tahun 2022, penulis menjadi salah satu mahasiswa jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN).

Selama menjalani studi di perguruan tinggi, penulis telah terlibat aktif dalam kegiatan organisasi kemahasiswaan dengan menjadi bagian dari UKM KOPMA UNILA pada tahun 2022-2024, kemudian bergabung pada bagian Staff Bidang Administrasi UKM KOPMA UNILA pada tahun 2023-2024. Pada Desember 2024 hingga Februari 2025, penulis melaksanakan Kerja Praktik (KP) di Dinas Komunikasi Informatika dan Statistika (DISKOMINFOTIK) Provinsi Lampung. Serta, pada Juni sampai Agustus 2025, penulis melaksanakan Kuliah Kerja Nyata (KKN) sebagai bentuk pengabdian penulis kepada masyarakat di Kelurahan Enggal, Kecamatan Enggal, Kota Bandar Lampung.

KATA INSPIRASI

”Jangan pernah merasa tertinggal, setiap orang punya proses dan rezeki-Nya masing-masing.”
(Q.S Maryam : 4)

”Hidup bukan tentang hari ini dan besok aja, hidup adalah tentang perjalanan panjang tapi perjalanan itu pasti ada banyak sekali kejutan-kejutan baik bagi mereka yang sabar menjalaninya.”
(Ustadz Hanan Attaki)

”Maka tunggulah hari yang sangat indah itu. Jangan berputus asa dulu, jangan nyerah dulu, jangan berhenti berharap dulu, katakan saja ’gapapa deh, gapapa deh’ walaupun harus teriak, nangis, terluka, gapapa ya Allah, gapapa ya Allah. Sampai Allah bilang udah ya sekarang udah selesai, aku ganti sekarang episode berikutnya kamu akan dapatkan semua kebaikan di langit dan di bumi, sampai kamu sendiri terheran-heran dan gabisa membendungnya.”
(Ustadz Hanan Attaki)

”Tanggung jawab apapun yang Allah berikan kepadamu, Allah sendiri yang akan membantu menyelesaikannya.”
(Meilina Risdianti)

PERSEMBAHAN

Dengan mengucap Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, saya persembahkan rasa terima kasih saya kepada:

Ayah dan Ibuku Tercinta

Terimakasih kepada orang tua saya atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih juga kepada adik-adik saya, yang menjadi semangat bagi kakakmu ini agar bisa menjadi contoh yang baik untuk kalian.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, meluangkan waktu, memberikan motivasi, arahan, serta ilmu yang berharga.

Seluruh Manusia Baik

Terimakasih kepada orang-orang baik yang telah memberikan semangat, motivasi, doa, dukungan, dan bersedia menjadi pendengar atas segala keluh kesah saya.

Almamater Tercinta
Universitas Lampung

SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul ”Optimalisasi Fine-Tuning Model BioBART Menggunakan AdamW *Optimizer* dalam Peringkasan Teks Otomatis pada Artikel Ilmiah Biomedis dengan Evaluasi Metrik ROUGE dan BERTScore” dengan baik dan lancar serta tepat pada waktu yang telah ditentukan. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, banyak pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku Pembimbing I sekaligus dosen pembimbing akademik yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga skripsi ini dapat terselesaikan.
2. Bapak Favorizen R. Lumbanraja, M.Si., Ph.D. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga skripsi ini dapat terselesaikan.
3. Bapak Ir. Warsono, M.S., Ph.D. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

6. Bapak yang selalu kuat untuk anaknya, Ibu yang selalu menyediakan tangan dan telinganya, adik-adik yang selalu memberikan sumber semangat, serta seluruh keluarga yang senantiasa memberikan doa, dukungan, dorongan, motivasi, serta semangat kepada penulis selama ini.
7. Sahabat penulis sejak SMA di antaranya Panca Diana Nurwani, Indri Kusuma Wardani, Dyta Kharisma Putri, dan Anisa Azzahra. Terimakasih sudah berdiri berada di antara perjalanan hidup saya.
8. Terkhusus kepada Restian Maharani, Indah Istiani, dan Zetira Marshanda Putri yang menjadi best support system selama penulis menjalankan perkuliahan di Bandar Lampung.
9. Rekan seperjuangan skripsi, di antaranya Nadia Ghassani, Khusni Sinta Rodiah, Fadillah Pinasti, Anita Caroline, Erin Elfitriani, Ahmad Rizki Munandar, Oja Widiyatama, Fatur Rozak, dan Benaya. Terimakasih sudah bersama-sama menyelesaikan ini.
10. Teman-teman seperjuangan Jurusan Matematika angkatan 2022.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung, 12 Maret 2026

Meilina Risdianti

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

Gambar Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang dan Masalah

Menurut Setiaji dan Pramudho (2022), banyak para pakar menerbitkan berbagai jurnal, makalah *symposium* dan karya *preprint* dalam teknologi informasi berbasis edisi elektronik dan dalam berbagai macam bidang keilmuan termasuk jurnal kesehatan. Jurnal kesehatan itu sendiri merupakan jurnal ilmiah yang menyajikan artikel yang relevan dengan isu-isu kesehatan, kebidanan, keperawatan, kesehatan klinis dan sosial berupa artikel hasil penelitian, artikel review, literatur atau artikel laporan lapangan yang di tujukan sebagai sarana publikasi dan sarana berbagi riset dalam pengembangan di bidang kesehatan. Artikel kesehatan terutama biomedis memiliki bentuk yang sangat teknis dan panjang, sehingga sulit dipahami oleh non-ahli dan memerlukan waktu yang cukup lama terutama bagi para pembuat rekomendasi kebijakan di bidang kesehatan. Peringkasan teks otomatis dapat menjadi hal yang sangat krusial dan semakin mendapat perhatian karena berpotensi dalam menyediakan informasi ilmiah yang lebih singkat dan mudah dipahami oleh non-ahli maupun pelaku medis (Phan *et al*., 2023).

Peringkasan teks otomatis atau *automatic text summarization* merupakan cabang penting dalam ilmu *Natural Language Processing* (NLP) yang meringkas dokumen panjang menjadi versi yang lebih ringkas dan lebih mudah dipahami oleh pembaca (Hartawan *et al*., 2024). Salah satu model NLP yang digunakan dalam mengatasi tugas NLP kompleks seperti peringkasan teks otomatis di antaranya yaitu model BART, sebuah model transformer yang inovatif yang mampu memahami dan merangkum informasi kompleks secara efektif, mampu mengubah teks panjang menjadi ringkasan yang lebih ringkas, relevan, dan informatif (Widiantoro & Sanjaya, 2024). Mengingat sifat artikel kesehatan dan biomedis yang di anggap komplek, maka model BioBART sebagai bagian dari BART dasar digunakan sebagai model yang telah dilatih pada korpus teks biomedis sehingga sering kali

dijadikan sebagai pilihan optimal untuk tugas-tugas biomedis.

Meskipun model bahasa besar seperti BioBART yang telah dilatih dalam korpus biomedis telah menunjukkan peningkatan performa dalam tugas generalisasi bahasa alami biomedis, keterbacaan ringkasan yang dihasilkan masih perlu di tingkatkan terutama dari sudut pandang audiens non-ahli. Adanya proses *fine-tuning* yang substansial dan tidak selalu mampu mentransfer pengetahuan secara efektif ke domain target, maka diperlukan penetapan hiperparameter menggunakan data validasi (Karotia & Susan, 2024). Dalam proses optimisasi, diperkenalkan *optimizer* AdamW sebagai variasi dari *optimizer* Adam yang memisahkan *weight decay* dari proses pembaruan parameter sehingga lebih stabil dan efektif dalam mencegah *overfitting* (Mahajaya *et al*., 2024).

Dalam penerapan tugas-tugas sebuah model, perlu adanya evaluasi sebagai pengukur sejauh mana model mampu bekerja secara optimal. Metrik ROUGE dan BERTScore merupakan metrik evaluasi yang umum digunakan dalam tugas NLP. Metrik ROUGE sebagai pengukur kesamaan n-gram dan keselarasan struktur antara hasil ringkasan terhadap ringkasan referensi. Sedangkan BERTScore digunakan sebagai pengukur kedekatan makna antara ringkasan refrensi dan ringkasan yang dihasilkan model (Aulia *et al*., 2025).

Berbagai penelitian telah mengimplementasikan model BioBART dalam meringkas teks artikel biomedis. Penelitian Phan, Tran, dan Trieu (2023) menggunakan model BioBART-*large* dan FactorSum untuk tugas *lay summarization* dengan dua pendekatan yaitu *explicit* dan *implicit key information selection* dengan perolehan hasil evaluasi pada pendekatan *explicit* yaitu ROUGE-1 0,4592 (PLOS) dan 0,4875 (eLife), ROUGE-2 0,1476 (PLOS) dan 0,1409 (eLife), ROUGE-L 0,4147 (PLOS) dan 0,4599 (eLife), dengan BERTScore 0,6196 (PLOS) dan 0,6218 (eLife). Sedangkan pada pendekatan implicit key diperoleh nilai ROUGE-1 0,4933 (PLOS) dan 0,5007 (eLife), ROUGE-2 0,1726 (PLOS) dan 0,1285 (eLife), ROUGE-L 0,4503 (PLOS) dan 0,4702 (eLife), dengan BERTScore 0,6388 (PLOS) dan 0,6231 (eLife). Penelitian Yuan et all (2022), menunjukkan bahwa *pretraining* untuk domain biomedis pada BioBART memberikan peningkatan kinerja yang konsisten di bandingkan BART standar. Pada masing-masing dataset diperoleh nilai ROUGE-1 61,07 (iClinic), 46,67 (HealthCareMagic), 30,12 (MEDIQA-QS), 32,90 (MEDIQA-MAS), 18,97 (MEDIQA-ANS), dan 53,75 (MeQSum), nilai ROUGE-2 masing-masing sebesar 48,47 (iClinic), 26,03 (HealthCareMagic),

11,28 (MEDIQA-QS), 11,28 (MEDIQA-MAS), 7,46 (MEDIQA-ANS), dan 36,50 (MeQSum), nilai ROUGE-L masing-masing sebesar 59,42 (iClinic), 44,11 (HealthCareMagic), 27,44 (MEDIQA-QS), 29,26 (MEDIQA-MAS), 16,77 (MEDIQA-ANS), dan 51,27 (MeQSum), dan nilai BERTScore masing-masing 0,941 (iClinic), 0,918 (HealthCareMagic), 0,898 (MEDIQA-QS), 0,861 (MEDIQA-MAS), 0,850 (MEDIQA-ANS), dan 0,929 (MeQSum). Penelitian oleh (Karotia & Susan, 2024) menggunakan BioBART-*large* sebagai model utama untuk peringkasan *lay summary* dokumen biomedis panjang, dengan tiga model pembanding yaitu T5, BART, dan PEGASUS. Evaluasi dilakukan pada dataset PLOS dan eLife dalam kompetisi BioLaySumm 2023. Hasil penelitian menunjukkan bahwa BioBART-*large* unggul dibandingkan model pembanding pada sebagian besar metrik ROUGE. Contohnya pada dataset eLife, BioBART-*large* mencapai nilai ROUGE-1 sebesar 0,4343, ROUGE-2 sebesar 0,1043, dan ROUGE-L sebesar 0,3871.

Sebagian besar penelitian peringkasan teks artikel biomedis masih mengandalkan dataset kompetisi seperti PLOS dan eLife yang memiliki struktur artikel yang relatif konsisten dan abstrak yang secara eksplisit dibentuk sebagai ringkasan dokumen untuk para non-ahli atau *lay summary*, sehingga model cenderung dievaluasi pada kondisi yang terkontrol dan kurang merepresentasikan keragaman artikel ilmiah biomedis di dunia nyata. Sedangkan PubMed *Central* (PMC) merupakan repositori artikel biomedis berskala besar dengan variasi panjang dokumen, struktur penulisan, dan kompleksitas terminologi yang lebih beragam. Meskipun PMC menyediakan data yang merepresentasikan kondisi nyata publikasi artikel ilmiah biomedis, pemanfaatannya dalam peringkasan teks otomatis masih terbatas khususnya dalam optimalisasi *fine-tuning* pada model transformer berbasis domain seperti BioBART.

Meskipun BioBART telah terbukti unggul dalam berbagai tugas *biomedical summarization*, namun sebagian besar penelitian masih berfokus pada perbandingan model tanpa melakukan optimalisasi menyeluruh, seperti penyetelan *hyperparameter* yang sistematis pada fungsi *learning rate, dropout, batch size*, dan *weight decay*, serta pemilihan *optimizer* yang lebih optimal seperti AdamW *optimizer*. Selain itu, evaluasi kinerja model umumnya hanya mengandalkan metrik berbasis n-gram seperti ROUGE dan BLEU tanpa melibatkan penilaian berbasis kesamaan semantik maupun kombinasi metrik yang lebih komprehensif seperti BERTScore yang mampu menilai kemiripan makna antara ringkasan yang dihasilkan terhadap ringkasan referensi pada artikel melalui *metric precision, recall*, dan *f1-score*.

Oleh sebab itu, penelitian ini bertujuan untuk mengeksplorasi model BioBART melalui proses *fine-tuning* dan eksplorasi *hyperparameter* dengan AdamW *optimizer* pada dataset PMC (*PubMed Central*), serta mengevaluasi hasilnya menggunakan metrik ROUGE dan BERTScore.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut maka dirumuskan beberapa masalah sebagai berikut:

1. Mengimplementasikan model BioBART pada tugas *abstractive summarization* dengan AdamW *optimizer* menggunakan dataset PMC (*PubMed Central*).
2. Peningkatan performa model BioBART pada tugas *abstractive summarization* melalui *fine-tuning* dengan AdamW *optimizer* menggunakan dataset PMC (*PubMed Central*).
3. Evaluasi kualitas ringkasan dengan menerapkan metrik ROUGE dan BERTScore untuk mengukur kesamaan kata dan makna antara ringkasan yang dihasilkan model terhadap ringkasan referensi (*abstract*).

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu:

1. Menganalisis pengaturan *hyperparameter* dalam proses *fine-tuning* model BioBART dengan menggunakan AdamW *optimizer* terhadap kualitas ringkasan teks artikel biomedis.
2. Mengevaluasi kemampuan model BioBART dalam menghasilkan ringkasan teks artikel biomedis menggunakan dataset PMC (*PubMed Central*) melalui metrik ROUGE dan BERTScore.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu:

1. Menambah wawasan mengenai penggunaan model BioBART dalam *text summarization* pada dataset PMC (*PubMed Central*) yang dioptimalkan dengan AdamW *optimizer*.
2. Menambah referensi evaluasi melalui metrik ROUGE dan BERTScore, yang dapat digunakan untuk mendukung penelitian selanjutnya dalam bidang *automatic text summarization*.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian-penelitian terkait dijadikan sebagai acuan dan menambah pemahaman mengenai metode yang akan digunakan dalam penelitian ini. Penelitian-penelitian tersebut ditampilkan dalam Tabel 1.

Tabel 1. Penelitian Terkait Implementasi Model BART dan BioBART dalam Peringkasan Teks

No.	Penelitian	Data	Metode	Hasil
1.	*"VBD-NLP at BioLay Summ Task 1: Explicit and Implicit Key Information Slelction for Lay Summarization on Biomedical Long Documents"* (Phan *et al*., 2023)	**Dataset PLOS**: 26291 data ; 24773 data latih, 1376 data validasi, dan 142 data uji. **Dataset eLife**: 4729 data; 4346 data latih, 241 data validasi, dan 142 data uji.	***Preprocessing*** : *Explicit selection*, *Tokenisation*, Truncation. ***Hyperparameter***: Learning Rate 5e-5, *Gradient Acumulation* : 4, *Batch Size* : 2, *Max Training Iteration* : 50.000 steps, *Generation max length* : 512, *Generation num beams* : 4, *Max sourch length* : 1024, 1 GPU. ***Optimizer*** : AdamW. **Model**: BioBART, LED, BRIO, dan FactorSum. Metrik: Relevance (ROUGE-1, ROUGE-2, ROUGE-L, BERTScore), Readibility (FKGL, DCRS), Factuality (BARTScore).	**BioBART+Lead PLOS**; R-1:0,4592, R-2:0,1476, R-L:0,4147, BERTScore:0,6196. **BioBART+Lead eLife**; R-1:0,4875, R-2:0,1409, R-L:0,4599, BERTScore:0,6218. **BioBART+key PLOS**; R-1:0,4933, R-2:0,1726, R-L:0,4503, BERTScore:0,6388. **BioBART+key eLife**; R-1:0,5007, R-2:0,1285, R-L:0,4702, BERTScore:0,6231.

No.	Penelitian	Data	Metode	Hasil
2.	*"BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model"* (Yuan *et al*., 2022)	MeQSum: 1000, iCliniq: 31064, HealthCare Magic: 226405, MEDIQA-ANS: 38892, MEDIQA-QS: 1150, MEDIQA-MAS: 1234	***Preprocessing*** : Tokenisasi, *Truncation*, *corruption (text infilling)*, *Masking configuration*. ***Hyperparameter***: *Learning rate : 3e-5*, *Batch Size* : 32, *Max input length* : 1024, *Max Target Length* : 256, *Number of epoch* : 10, *Beam Search* : 5, *Label Smoothing* : 0.1, *Weight Decay* : 0.01, *Gradient clipping*: 1.0. ***Optimizer*** : AdamW. **Model** : BioBART. **Evaluasi**: ROUGE, BERTScore.	**iClinic**; R-1/2/L: 61.07/48.47/59.42, BERTScore: 0.941. **HealthCare Magic**; R-1/2/L: 46.67/26.03/44.11, BERTScore: 0.918. MEDIQA-QS; R-1/2/L: 30.12/11.28/27.44, BERTScore: 0.898. **MEDIQA- MAS**; R-1/2/L: 32.90/11.28/29.26, BERTScore: 0.861. **MEDIQA- ANS**; R-1/2/L: 18.97/7.46/16.77, BERTScore: 0.850. **MeQSum**; R-1/2/L: 53.75/36.50/51.27, BERTScore: 0.929.
3.	*"BioLay-AK-SS at BioLaySumm: Domain Adaptation by Two-Stage Fine-Tuning of Large Language Models used for Biomedical Lay Summary Generation"* (Karotia & Susan, 2024)	**Dataset PLOS**: 26291 data ; 24773 data latih, 1376 data validasi, dan 142 data uji. **Dataset eLife**: 4729 data; 4346 data latih, 241 data validasi, dan 142 data uji.	***Hyperparameter*** : *Learning Rate* : 5e-5, *Batch size* : 16, *patience* : 3, *Max input length* : 1024, *Max output length* : 512 (eLife) dan 200 (PLOS), *Beam search* : 4.. **Model** : BART, PEGASUS, BioBART. **Evaluasi**: ROUGE-1/2/L dan BERTScore.	**eLife**; R-1:0,4343, R-2:0,1043, R-L:0,3589, BERTScore:0,8308. **PLOS**; R-1:0,4248, R-2:0,1420, R-L:0,3839, BERTScore:0,8508.

No.	Penelitian	Data	Metode	Hasil
4.	*"Bidirectional and Auto-Regressive Transformer (BART) For Indonesian Abstractive Text Summarization"* (Hartawan *et al*., 2024)	Berita Liputan6 dengan 193883 data latih, 10972 data validasi, dan 10972 data uji.	***Preprocessing*** : Tokenisasi, *Truncation*, *Extracting important information*. ***Hyperparameter*** : *Learning rate* : 0.0001, *Per device train batch size* : 8, *Per device evaluation batch size* : 4, *Epoch* : 5, *Warmup steps* : 500, *Weight decay* : 0.01, *Logging steps* : 100, *Checkpoint saving* : 6.250/step. ***Optimizer*** : AdamW. **Model** : BART. **Model pelatihan** facebook/bart -large-cnn. **Evaluasi** : ROUGE*Score*	ROUGE-1; recall:52,28, precision:29,16, dan f1-score:37,19. ROUGE-2; recall:21,92, precision:10,41, dan f1-score:14,03. ROUGE-L; recall:48,07, precision:26,39, dan f1-score:33,85.

2.1.1 Penelitian Pertama (Phan *et al*., 2023)

Penelitian pertama menggunakan FactorSum, BioBART, LED, dan BRIO untuk melakukan peringkasan teks pada domain biomedis. Keempat model dibandingkan kinerjanya pada dataset *Public Library og Science* (PLOS) dan eLife yang disediakan oleh penyelenggara kompetisi BioLaySum2023. Terdapat 26391 data article pada dataset PLOS, yang dibagi menjadi 93,87% data *training*, 5,59% data *validation*, dan 0,54% data *testing*. Dengan demikian, jumlah data pelatihan sebanyak 24773, data validasi sebanyak 1476, dan data pengujian sebanyak 142. Sementara itu, untuk data eLife sebanyak 4729 data dengan 91,88% atau 4346 data pelatihan, 5,1% atau 241 data validasi, dan 3% atau 142 data pengujian. Dari seluruh model yang digunakan, FactorSum + BioBART adalah *pretrained* model utama dengan konfigurasi *training* dan *generation* yaitu menggunakan AdamW *optimizer*, *learning rate* 5e-5, *batch size* 2 karena keterbatasan komputasi, *gradient accumulation* setiap 4 iterasi, *generation max length* 512, *num beams* 4, *max source length* 1024, dan *max target length* 490 untuk data PLOS serta 512 pada data eLife.

Hasil penelitian pertama menunjukkan bahwa, pada dataset PLOS model ExSum(key)+BioBART mendekati skor terbaik pada metrik ROUGE-1, ROUGE-2, ROUGE-L, dan DCRS metrik dengan nilai 0,4933, 0,1726, 0,4503, dan 11,6746. Model ExSum(Lead)+BioBART unggul pada FKGL dengan nilai 14,6584. Model FS-12k (FactorSum+BioBART) menghasilkan BERTScore dan BARTScore

tertinggi dengan nilai 0,8611 dan -1,3312, menunjukkan ringkasan yang paling faktual dan semantik paling mendekati referensi. Sedangkan pada dataset eLife, model ExSum(key)+BioBART terbaik pada ROUGE-L dengan nilai 0,4702, sedangkan ExSum(lead)+BioBART unggul pada ROUGE-2 dengan nilai 0,1409. Model ExSum(key)+LED memberikan hasil terbaik untuk DCRS dan BARTScore dengan nilai 8,0722 dan -2,0177. FS-9k (FactorSum+BioBART) unggul pada ROUGE-1, BERTScore, dan FKGL dengan nilai 0,5009, 0,8520, dan 9,9070 sehingga menunjukkan keseimbangan terbaik antara relevansi, kemiripan semantik, dan keterbacaan.

2.1.2 Penelitian Kedua (Yuan *et al*., 2022)

Penelitian kedua menggunakan model BioBART-*base* dan BioBART-*large* dengan model pembanding yaitu model BART. Dataset yang digunakan terbagi menjadi dua bagian utama yaitu dataset untuk *pretraining* dan dataset untuk *fine-tuning*. Untuk dataset *pretraining* bersumber dari PubMed *abstract* sekitar 41GB task abstrak artikel biomedis. Untuk dataset *fine-tuning* dan evaluasi berbeda pada setiap tugas. Pada *Dialogue system tasks* bersumber dari CovidDialog berisi 603 konsultasi medis dengan total 1232 *utterances*. Pada *Abstractive Summarization tasks* digunakan beberapa dataset medis, yaitu dataset iClinic dengan 31062 sampel, dataset HealthCareMagic dengan 226405 sampel, dataset MeQSum dengan 1000 pasang Q-summary, dataset MEDIQA-ANS dengan 38166 data pelatihan, 174 data validasi, dan 552 data pengujian, dataset MEDIQA-QS dengan 1000 data pelatihan, 50 data validasi, dan 100 data pengujian, serta dataset MEDIQA-MAS dengan 1104 data pelatihan, 50 data validasi, dan 80 data pengujian. Pada *Entity Linking tasks* digunakan 5 dataset yaitu, dataset MedMentions dengan 4392 PubMed *abstract*, dataset BC5CDR dengan 1500 artikel PubMed, dataset NCBI *Disease Corpus* dengan 793 abstrak penyakit dari PubMed, dataset COMETA dengan 20000 mention dari Reddit, serta dataset AskAPatient dengan 8662 frasa dari media sosial. Pada *Named Entity Recognition* (NER) *tasks* digunakan beberapa dataset juga, yaitu dataset ShARe13, ShARe14, CADEC, dan GENIA dengan 2000 abstrak. *Hyperparameter pretraining* yang digunakan dalam penelitian kedua ini yaitu *batch size* 2,560, *learning rate* awal 1e-4 untuk BART-*base* dan BART-*large*, *warm-up ratio* 0,02, tokenisasi menggunakan *vocabulary* BART asli tidak mengubah tokenizer, panjang input maksimum 512 token, *masking rate* 30% token, dan *optimizer DeepSpeed*. Sedangkan *hyperparameter fine-tuning* yaitu, *learning rate*

5e-5 untuk BART-*base* dan 1e-5 untuk BART-*large*, *epochs* 20, *loss-function* dengan *negative log-likelihood*, dan *beam search* 5. Terakhir saat *generated summarization*, menggunakan *learning rate* 5e-5 untuk BART-*base* dan 1e-5 untuk BART-*large*, *epochs* 6, dan *beam search* 5 tanpa length penalty. Evaluasi yang digunakan dalam penelitian ini yaitu menggunakan metrik ROUGE dan BERTScore.

Hasil penelitian kedua dalam berbagai tugas menunjukkan bahwa pada *Dialogue System tasks* pada dataset CovidDialogue, BioBART unggul pada ROUGE-2, ROUGE-L, dan BLEU dengan nilai 13,79, 26,96, dan 12,05 dibanding BART. Pada *Summarization Tasks* pada dataset MeQSum, BioBART unggul pada seluruh ROUGE-1/2/L dan BERTScore dengan nilai 55,61/38,11/53,15 dan 0,9333, pada dataset MEDIQA-MAS BioBART juga unggul pada ROUGE-1, ROUGE-L, dan BERTScore dengan nilai 32,90, 29,26, dan 0,861, sedangkan pada dataset *HealthCareMagic* dan iClinic menunjukkan performa sebanding dengan BART. Pada *Entity Linking tasks* unggul pada seluruh recall@1 dan recall@5 dengan seluruh datasetnya. Terakhir Pada NER *tasks* seluruh dataset juga unggul dibanding BART terutama pada dataset ShARe13, ShARe13, dan GENIA.

2.1.3 Penelitian Ketiga (Karotia & Susan, 2024)

Penelitian ketiga menggunakan BioBART-*large* untuk melakukan peringkasan teks artikel panjang pada domain biomedis sebagai model utama, dengan T5, BART, dan PEGASUS sebagai pembanding. Keempat model dibandingkan kinerjanya pada dataset *Public Library og Science* (PLOS) dan eLife yang disediakan oleh penyelenggara kompetisi BioLaySum2023. Terdapat 26391 data *article* pada dataset PLOS, yang dibagi menjadi 93,87% data *training*, 5,59% data *validation*, dan 0,54% data *testing*. Dengan demikian, jumlah data pelatihan sebanyak 24773, data validasi sebanyak 1476, dan data pengujian sebanyak 142. Sementara itu, untuk data eLife sebanyak 4729 data dengan 91,88% atau 4346 data pelatihan, 5,1% atau 241 data validasi, dan 3% atau 142 data pengujian. Seluruh model dilatih menggunakan konfigurasi *hyperparameter* yang sama yaitu AdamW *optimizer* dengan *batch size* sebesar 16 dan *learning rate* 5e-5, serta mekanisme *early stopping* dengan *patience* 3. Panjang maksimum input dibatasi hingga 1024 *token*. Pada generalisasi ringkasan, jumlah beam ditetapkan sebesar 4 dengan penalty 2. Panjang target disesuaikan dengan karakteristik dataset yaitu 180-200 untuk PLOS dan 350-512 untuk eLife.
Hasil penelitian menunjukkan bahwa, BioBART-*large* memberikan performa terbaik

dibandingkan model pembanding pada sebagian besar metrik ROUGE. Pada dataset eLife, BioBART-*large* mencapai nilai ROUGE-1 sebesar 0,4343 yang lebih tinggi dibandingkan T5 dan PEGASUS. Sementara itu, nilai ROUGE-2 dilaporkan sebesar 0,1043 dan ROUGE-L sebesar 0,3589 yang menunjukkan kemampuan model dalam menangkap kesamaan n-gram dan struktur ringkasan referensi. Pada dataset PLOS BioBART-*large* juga menunjukkan performa yang konsisten meskipun selisih performanya tidak relatif besar. Hasil ini menunjukkan bahwa pretraining BioBART pada korpus biomedis memberikan keuntungan leksikal pada tugas peringkasan *lay summarzation*. Namun performa yang diperoleh masih sangat dipengaruhi oleh strategi pemotongan dokumen, sehingga potensi kehilangan konteks semantik dari artikel ilmiah panjang tetap menjadi keterbatasan utama.

2.1.4 Penelitian Keempat (Hartawan *et al*., 2024)

Penelitian keempat menggunakan model BART untuk melakukan peringkasan teks berita berbahasa indonesia pada Liputan6 yang terdiri dari dua varian yaitu *Canonical* dan *Xtreme*. Terdapat 215827 data dengan 193883 data *training*, 10972 data *validation*, dan 10972 data *testing* pada kategori *Canonical*, sedangkan untuk kategori *Xtreme* terdapat 202693 data dengan 193883 data *training*, 4948 data *validation*, dan 3862 data *testing*. Dalam penelitian menggunakan model pra-latih *facebook/bart-large-cnn* dan *optimizer* AdamW dengan kombinasi *hyperparameter* utama berupa *learning rate* 0,0001, per-device train batch size 8, per-device eval batch size 4, epoch 5, *warmup steps* 500, *weight decay* 0,01, *logging steps* 100, dan *checkpoint* tiap 6250 steps. Evaluasi model menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L dengan metrik *precision, recall,* dan F1-*Score*.

Hasil penelitian keempat menunjukkan bahwa, berdasarkan nilai F1-Score pada ROUGE-1 sebesar 37,19 menunjukkan bahwa model BART mampu menangkap sebagian besar informasi penting dari teks sumber secara cukup baik, pada ROUGE-2 sebesar 14,03 yang lebih rendah menggambarkan bahwa koherensi antar frasa (bigram) masih dapat ditingkatkan, pada ROUGE-L sebesar 33,85 menunjukkn bahwa urutan kata dalam ringkasan yang dihasilkan relatif konsisten dan sesuai dengan ringkasan referensi. Sehingga membuktikan bahwa model BART memiliki performa baik dan kompetitif untuk tugas *Abstractive Summarization*.

2.2 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan cabang kecerdasan buatan yang fokus pada interaksi antara komputer dan bahasa manusia, yang memungkinkan komputer dapat memahami, menganalisis, dan merespon bahasa manusia dalam berbagai bentuk (Widiantoro & Sanjaya, 2024).

Natural Language Processing (NLP) memiliki dua fase kerja yaitu (Saputra *et al*., 2025):

1. Pemrosesan awal data, tahapan dimana teks di seragamkan bentuk dan formatnya sehingga menjadi data yang dapat diolah untuk hasil kalimat yang lebih alami. Tahap pemrosesan awal data terdiri dari *tokenization* dimana kata-kata dipecah menjadi unit yang lebih kecil untuk di proses, *stopword removal* dimana kata-kata yang tidak memiliki arti dihapus dari teks sehingga tersisa kata-kata unik dengan informasi yang bisa di ambil dari teks, *lemmatization* dan *stemming* dimana kata-kata direduksi ke bentuk dasarnya dengan menghapus kata imbuhannya, dan yang terakhir yaitu *part-of-speech tagging* dimana kata-kata ditandai berdasarkan kelas katanya.

2. Pengembangan algoritma, tahapan pembersihan data yang dilakukan sebelum melakukan analisis dengan menggunakan *rule based algorithm* dan *machine learning based algorithm*.Pada *rule based algorithm*, analisis dan proses data teks dilakukan dengan menerapkan aturan linguistik yang telah di rancang dengan teliti dengan tujuan untuk menangkap struktur khusus dalam teks, mengekstraksi informasi yang relevan, dan melakukan tugas-tugas seperti klasifikasi teks berdasarkan aturan atau pola tertentu. Sedangkan pada *machine learning based algorithm*, komputer belajar melakukan tugas berdasarkan data pelatihan yang diberikan dengan pemrosesan dan pembelajaran berulang, serta menggunakan kombinasi dari *machine learning, neural network*, dan *deep learning*.

Natural Language Processing (NLP) memiliki berbagai terapan aplikasi di antaranya yaitu Chatbot atau virtual assistent merupakan aplikasi yang membuat user bisa seolah-olah melakukan komunikasi dengan komputer, summarization atau ringkasan dari suatu bacaan, translation tools atau menerjemahkan bahasa, pendeteksi spam, analisis sentimen media sosial, dan aplikasi-aplikasi yang memungkinkan komputer mampu memahami instruksi bahasa yang di input oleh user (Nurfiyah & Ramadhandi, 2023).

Natural Language Processing (NLP) dalam beberapa tahun terakhir telah membawa kemajuan signifikan di berbagai bidang, termasuk bidang biomedis. Dalam beberapa tahun terakhir, BERT memanfaatkan *encoder* arsitektur transformer, sedangkan GPT memanfaatkan *decoder* transformer. Selain itu, model *sequence to sequence* seperti BART yang memanfaatkan baik *encoder* maupun *decoder* transformer juga muncul sebagai pendekatan yang kuat dalam berbagai tugas generasi teks (Jahan *et al*., 2023). Adanya model khusus seperti BioBART telah menunjukkan hasil yang menjanjikan di bidang biomedis.Namun model ini memerlukan penyempurnaan menggunakan dataset khusus bidang sehingga mampu mengatasi berbagai tantangan NLP secara lebih efisien dan fleksibel.

2.3 *Text Mining*

Text Mining adalah metode dalam menemukan informasi yang tidak diketahui dengan ekstraksi informasi otomatis dari teks yang tidak terstruktur. *Text mining* mampu mengambil data dalam jumlah besar kemudian di proses untuk mendapatkan informasi yang berguna dari sekumpulan teks (Hermawan *et al*., 2023).

Text mining merupakan teknik yang digunakan untuk menangani tugas klasifikasi, *clustering*, *information retrival*, dan *information extraction*. Langkah umum dalam *text mining* yaitu *text preprocessing* termasuk di dalamnya pemilihan data, klasifikasi dan ekstraksi fitur, mengubah dokumen menjadi perantara sehingga sesuai dengan tujuan pencarian. Selanjutnya yaitu operasi *texs mining* dan *postprocessing*. *Text mining* sudah banyak memunculkan penelitian yang akhirnya membentuk bidang penelitian dan pola aplikasi seperti pengelompokan teks, ekstraksi aturan asosiasi dan analisis tren (Firdaus & Firdaus, 2021).

2.4 *Keyword Extraction*

Keyword Extraction adalah metode ekstraksi yang bertujuan untuk mengekstraksi elemen kunci yang merupakan unit tekstual yang dianggap penting. Suatu elemen dianggap sebagai elemen kunci apabila merepresentasikan dan menjadi deskriptor penting dari konten suatu dokumen (Firoozeh *et al*., 2020). Terdapat berbagai cara untuk menilai pentingnya elemen dan berbagai jenis unit yang menjadi target, mulai

dari token tunggal hingga n-gram atau frasa kata.

Menurut Firoozeh *et al*.,(2020), terdapat dua metode pendekatan dalam *keyword extraction* yaitu *supervised method* dan *unsupervised method*. Pendekatan *supervised* menggunakan data pelatihan sebagai input dan bergantung pada fitur-fitur pelatihan untuk melatih sebuah classifier yang digunakan untuk memprediksi kandidat sebagai sebuah *keyword*. Pendekatan ini menunjukkan kinerja yang menjanjikan dalam mengekstraksi kata kunci, namun kebutuhan akan data pelatihan menjadi salah satu keterbatasan utama. Untuk menghindari kebutuhan data pelatihan, pendekatan *unsupervised* digunakan dengan kandidat kata kunci diberi skor menggunakan berbagai macam teknik yaitu :

1. *Basic Statistical Method*
   Beberapa pendekatan ekstraksi kata kunci secara *unsupervised* hanya menggunakan fitur statistik untuk mengekstraksi kata dan frasa yang paling signifikan dari sebuah teks. Meskipun sederhana, pendekatan ini telah digunakan secara luas dan menjadi tolak ukur dalam beberapa studi dan dijadikan *baseline* untuk membandingkan performa metode lain. TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan salah satu fitur statistik yang dominan. TF-IDF mengukur tingkat kepentingan sebuah term dalam suatu dokumen dengan mempertimbangkan frekuensi kemunculan dalam dokumen serta kelangkaan diseluruh korpus. Term dengan skor TF-IDF tinggi dianggap lebih informatif dan karenanya sering dipilih sebagai kandidat kata kunci dalam pendekatan *unsuperviser keyword extraction*.

2. *Entropic Method*
   Entropic method merupakan pendekatan *unsupervised* yang didasarkan pada asumsi bahwa keunikan sebagai kata kunci (*keyness*) terlihat dari distribusi spasial kemunculan kata dalam teks. Asumsi dasarnya adalah kata-kata yang relevan terhadap topik teks cenderung terkonsentrasi pada area-area tententu, sebaliknya kata-kata yang tidak relevan muncul secara acak diseluruh teks. Namun, tantangan utama pendekatan ini adalah kebutuhan untuk melakukan pemartisian awal terhadap teks, dan kualitas hasil ekstraksi sangat bergantung pada cara pemartisian dilakukan.

3. *Graph-based Method*
   Pada pendekatan ini tujuan utamanya adalah memanfaatkan keterhubungan antar kata dan menangkap sentralisasi dari kata kunci. Secara umu, pendekatannya mencakup pembuatan graf dari elemen-elemen teks dan kemudian

menggunakannya untuk melakukan *clustering* atau *ranking*. Pendekatan yang paling umum yaitu TextRank dengan membuat graf kata yang terdiri dari kata benda dan kata sifat, menggunakan algoritma PageRank untuk menentukan skor setiap simpul, dan menggabungkan simpul-simpul yang berdekatan dalam teks menjadi kata kunci majemuk. Selain itu terdapat TopicRank sebagai perluasan dari TextRank yang mengurutkan topik bukan kata, dan pada sebagian besar benchmark mengungguli pendekatan graf lain dan setara dengan TF-IDF.

2.5 Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit kecil, seperti kata-kata, frasa, atau kalimat yang di sebut token. Tokenisasi dalam NLP membantu dalam analisis, pemrosesan, dan pemahaman teks oleh mesin. Konsep-konsep dalam tokenisasi yaitu (Widiantoro & Sanjaya, 2024):

1. Unit token
   Token berupa kata, frasa, kalimat, atau karakter berdasarkan kebutuhan analisis dan pemrosesan yang dilakukan.

2. Pemisahan
   Teks di pisahkan menjadi token berdasarkan spasi antar kata, tanda baca seperti ”Saya senang belajar NLP.” akan menjadi token [”Saya”, ”senang”, ”belajar”, ”NLP”, ”.”], berdasarkan aturan tertentu atau aturan kata seperti ”Dia sedang berjalan-jalan” akan menjadi token [”Dia”,”sedang”,”berjalan-jalan”], berdasarkan frasa atau kalimat seperti ”Saya belajar NLP. Ini seru!” akan menjadi token [”Saya belajar NLP.”, ”Ini seru!”], dan tokenisasi menggunakan aturan algoritma linguistik khusus.

3. Pembersihan dan Normalisasi
   Tokenisasi dapat melibatkan pembersihan teks baik dari karakter khusus, tanda baca, dan *lowercasting* untuk konsistensi sesuai kebutuhan penelitian.

Menurut Toraman *et al*.,(2023), terdapat beberapa algoritma tokenisasi yang memanfaatkan berbagai fitur linguistik termasuk karakter, frekuensi, maupun gramatikal yang dijelaskan berdasarkan tingkat granularitas di antaranya :

1. *Character-Level*

   Keunggulan tokenisasi karakter adalah dapat digunakan pada bahasa apapun untuk merepresentasikan urutan karakter apapun pada tingkat byte dan memungkinkan pemodelan yang beragam serta mengurangi kebutuhan memori dalam hal ukuran model karena jumlah token dalam kosakatanya sangat terbatas. Sedangkan kekurangannya, model harus menggunakan lebih banyak kapasitas untuk mencapai representasi tingkat lebih tinggi. Contohnya, model harus mempelajari selama pelatihan bahwa hurus ”t” dan ”h” sering muncul dalam bahasa inggris, tidak langsung menyediakan informasi dengan token ”th”. Serta keluaran untuk sebuah urutan akan mengandung jumlah token yang jauh lebih besar sedangkan model bahasa memiliki batas panjang input sehingga dapat mengakibatkan potensi kehilangan informasi.

2. *Byte Pair Encoding* (BPE)

   BPE merupakan *tokenizer* yang sering digunakan untuk model pralatih. Token umumnya berupa subword bergantung pada ukuran kosakatanya karena granularitas BPE berada pada tingkat menengah antara kata dan karakter. Semua kosakata di ekstraksi terlebih dahulu kemudian kosakata dasar dibangun dari seluruh simbol yang muncul dalam kata-kata unik. Simbol-simbol digabungkan berdasarkan frekuensi kemunculan pasangan simbol atau subword yang berurutan sehingga terbentuk kosakata final. BPE bekerja dengan representasi *byte* sehingga kosakata yang dihasilkan dapat mencakup token dari berbagai bahasa dan urutan karakter nonformal.

3. *WordPiece*

   WordPiece mirip dengan BPE yang didasarkan pada penggabungan karakter. Perbedaannya, WordPiece menggabungkan simbol berdasarkan pemaksimalan skor likelihood pemodelan bahasa ketika probabilitas simbol hasil gabungan dibagi dengan probabilitas simbol-simbol individual lebih besar dari pasangan simbol lain.

4. *Morphological-level*

   Keunggululan tokenisasi tingkat morfologi berada pada kemampuannya menangkap urutan karakter yang dapat di interpretasikan secara gramatikal dalam pemodelan dan mempelajari makna berdasarkan sufliks kata. Tetapi, akar kata tidak dipecah lebih lanjut sehingga membentuk himpunan yang besar dan harus dimasukkan kedalam kosakata.

5. *Word-level*

   Tokenizer tingkat kata yaitu memisahkan teks berdasarkan spasi antar kata. Tokenisasi ini tidak memerlukan pelatihan kosakata, karena dapat dilakukan hanya dengan memisahkan teks menggunakan karakter spasi. Kekurangan yang eksplisit yaitu *tokenizer* membutuhkan ukuran kosakata yang lebih besar untuk melakukan tokenisasi terhadap jumlah teks yang sama dibandingkan metode lain.

2.6 *Sliding Windows*

Sliding Windows adalah proses ekstraksi fitur dengan cara menggeser jendela berukuran tetap dan bervariasi sepanjang urutan kata dalam teks. Penggunaan beberapa ukuran sliding window memungkinkan model untuk mempelajari hubungan antar kata yang berurutan atau hubungan kata dengan jarak yang lebih jauh dalam satu kalimat (Yang *et al*., 2020). Sliding window mampu menangkap karakteristik semantik dan statistik teks secara efektif melalui analisis korelasi kata dari berbagai skala konteks.

Sliding windows digunakan untuk menangani masalah keterbatasan panjang iinput pada model transformer dengan memecah teks yang panjang menjadi beberapa window berukuran tetap. Setiap window terbentuk dari pengambilan urutan token sebesar panjang maksimum input model dan digeser secara bertahap dengan ukuran langkap atau slide tertentu sehingga beberapa window saling tumpang tindih. Mekanisme tumpang tindih ini memungkinkan setiap window mempertahankan kesinambungan konteks semantik antarbagian teks.

2.7 *Word Embedding*

Word embedding adalah teknik pembelajaran fitur yang merepresentasikan kata-kata dari kosakata kedalam vektor bilangan rill dalam ruang berdimensi rendah sehingga memungkinkan mesin dapat memahami dan memanipulasi makna kata-kata dalam pemrosesan bahasa alami (Widiantoro & Sanjaya, 2024). *Word embedding* telah tebukti meningkatkan kinerja tugas NLP seperti terjemahan mesin, analisis sentimen, analogi kata, pengenalan entitas bernama, dan kesamaan kata.

Pada model berbasis transformer seperti BioBART yang dibangun melalui *pretraining* lanjutan pada korpus biomedis, *embedding* digunakan sebagai pembangun representasi kata yang bersifat dinamis (*contextual embedding*) yang berbeda dengan pendekatan tradisional seperti *Word2Vec* atau *GloVe* yang menghasilkan *embedding* statis. Sehingga menjadikan BioBART lebih efektif digunakan dalam tugas peringkasan teks otomatis pada artikel kesehatan.

2.8 Peringkasan Teks Otomatis

Peringkasan teks otomatis merupakan bidang *information extraction* dengan meringkas dokumen teks digital dan tetap mempertahankan isi informasinya. *information extraction* adalah ekstraksi informasi secara terstruktur dan otomatis dari dokumen tidak terstruktur atau semi terstruktur. Peringkasan teks otomatis mampu memudahkan pembaca untuk mengetahui inti dari artikel tanpa harus membaca keseluruhan isi artikel dan meluangkan banyak waktu (Husniah *et al*., 2022).

Peringkasan teks memiliki dua pendekatan yaitu peringkasan teks ekstraktif dan peringkasan teks abstraktif. Peringkasan ekstraktif yaitu peringkasan yang dilakukan dengan mengekstraksi sebagian kalimat penting dari dokumen asli. Sedangkan peringkasan abstraktif yaitu peringkasan yang dilakukan dengan membuat dan menyusun kalimat baru yang merupakan kombinasi informasi sehingga ringkasan memiliki kosakata yang bervariasi kemudian menggabungkannya menjadi kalimat yang lebih pendek tanpa kehilangan makna dan informasi penting (Hadwiranto *et al*., 2024).

2.9 *Machine Learning*

Machine learning adalah cabang khusus dari *artificial intelligence* (AI) yang memungkinkan mesin untuk belajar meniru kemampuan manusia secara mandiri dari data dan pengalaman sebelumnya.Dengan memberikan algoritma *machine learning* dengan sejumlah besar data, *machine learning* dapat dilatih untuk menganalisis data, membangun model, dan memprediksi output secara otomatis sehingga dapat memberikan waktu dan biaya yang lebih efisien (Shaveta, 2023).

Secara umum terdapat tiga kategori dalam tugas *machine learning*(Shaveta, 2023), yaitu sebagai berikut:

1. *Supervised Learning*
   Dalam *supervised learning*, sistem dilatih menggunakan data berlabel untuk memahami dan mempelajari masing-masing dataset yang kemudian di amati bagaimana sistem dapat memprediksi hasil secara akurat atau tidak dengan memberikan kumpulan data sampel. Tujuan dari *supervised learning* yaitu pemetaan data input dan output. *Supervised learning* dapat dibagi menjadi dua kategori algoritma yaitu *classification* dan *regression*. *Classification* ketika varabel output bersifat kategorikal atau terdapat dua kelas, dengan beberapa algoritma klasifikasi yaitu *Random Forest, Decision Trees, Logistic Regression, Support Vector Machines*. Serta *Regression* ketika terdapat korelasi antara variabel input dan output, dengan beberapa algoritma populer yaitu *Linear Regression, Regression Trees, Non-Linear Regression, Bayesian Linear Regression*, dan *Polynomial Regression*.

2. *Unsupervised Learning*
   Unsupervised learning adalah pembelajaran dimana komputer mengumpulkan informasi tanpa campur tangan manusia, artinya mesin dilatih dengan kumpulan data yang tidak dilabeli, tidak di klasifikasikan, tidak di kategorikan, dan algoritma harus merespon data secara mandiri. Tujuan dari *unsupervised learning* yaitu untuk memproses ulang data menjadi fitur baru atau kumpulan objek yang terkait. Sistem tidak dapat menjamin bahwa output benar, tetapi menyimpulkan yang seharusnya menjadi hasil berdasarkan dataset. *Unsupervised learning* dapat dibagi menjadi dua kategori algoritma yaitu *clustering* dan *association*. Dengan teknik *clustering*, objek data dikelompokkan berdasarkan ada atau tidaknya kesamaan yang ditemukan melalui analisis *cluster*. Sedangkan *association* digunakan dalam menangkap hubungan antar variabel dalam data yang besar dengan menetapkan kelompok item yang sering muncul bersama *collection*.

3. *Reinforcement in Learning*
   Algoritma *Reinforcement in learning* berinteraksi dengan mengambil tindakan dan mengidentifikasi keberhasilan atau kegagalan. Dua fitur penting dalam *reinforcement in learning* yaitu *trial-and-error learning* dan *delayed rewards* yang membantu mesin dan agen *software* secara otomatis memilih tindakan terbaik untuk meningkatkan kinerja dalam siatuasi tertentu.

2.10 *Deep Learning*

Deep Learning adalah metode *learning* yang memanfaatkan *artificial neural network* berlapis-lapis (*multi layer*) yang dibuat mirip otak manusia, dengan neuron-neuron yang terkoneksi satu sama lain sehingga membentuk jaringan neuron yang sangat rumit. *Deep learning* atau *deep structured learning* atau *hirarchial learning* atau *deep neural* adalah metode *learning* yang memanfaatkan *multiple non-linier transformation* sehingga *deep learning* dapat di pandang sebagai subbidang *machine learning* yang algoritmanya terinspirasi dari *Artificial Neural Networks* (ANN). *Deep learning* memiliki dua jenis pendekatan yaitu *Unsupervised Learning* dan *Hybrid Deep Networks*. *Unsupervised Learning* digunakan ketika label dari variabel target tidak tersedia dan dalam menganalisis polanya korelasi nilai yang lebih tinggi di hitung dari unit yang diamati. Sedangkan *Hybrid Deep Networks* digunakan untuk dapat mencapai hasil yang baik dengan menggunakan *supervised learning* (Raup *et al*., 2022).

Deep Learning memiliki beberapa jenis algoritma dalam menangani data dan tugasnya. *Convolutional neural network* (CNN) atau *ConvNets* digunakan dalam proses data berupa gambar atau deteksi objek. *Recurrent neural network* (RNN) yang biasa digunakan untuk memberi caption pada gambar, menganalisis deret waktu, memproses *natural-language*, hingga mengenali tulisan tangan. RNN memiliki memori internal sehingga dapat menghasilkan koneksi yang membentuk siklus terstruktur dan kemudian memproses output yang berasal dari LSTM untuk dijadikan input. Sedangkan LSTM (*Long Short Term Memory Network*) merupakan jenis RNN yang mampu mempelajari dan mengingat ketergantungan jangka panjang, sehingga mampu mengingat kembali informasi dari waktu lampau. LSTM memiliki struktur yang mirip rantai. Selanjutnya yaitu *Self-Organizing Maps* (SOM) sebagai visualisasi data untuk mengurangi dimensi data melalui jaringan neural buatan yang beroperasi sendiri. Dengan algoritma-algoritma tersebut menjadikan *deep learning* mampu untuk menangani berbagai tugas seperti *virtual assistents*, mobil otomatis, chatbots, penerjemahan, peringkasan, dan berbagai tugas NLP.

2.11 Fungsi Aktivasi

Fungsi aktivasi atau sering disebut fungsi transfer merupakan fungsi yang digunakan dalam menentukan apakah neuron dapat diaktifkan atau tidak dengan menentukan keluaran berdasarkan nilai masukan yang diproses dari lapisan input hingga lapisan output (Kurniawan *et al*., 2024). Terdapat beberapa fungsi aktivasi yang umum digunakan pada arsitektur transformer yaitu:

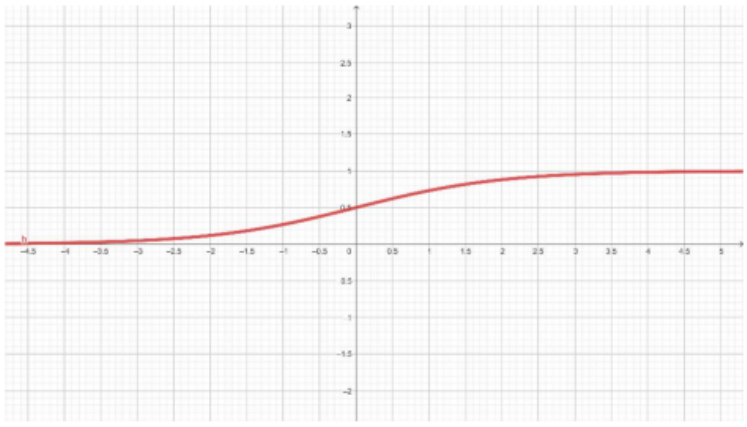
2.11.1 Fungsi Aktivasi Sigmoid

Fungsi aktivasi sigmoid adalah fungsi aktivasi *non-linear* dengan kurva berbentuk ”S” yang mengubah nilai antara rentang 0 hingga 1 berdasarkan besar *input*. Jika *input* positif maka *output* akan mendekati angka 1, sedangkan jika *input* negatif maka *output* akan mendekati angka 0 (Kurniawan *et al*., 2024). Secara matematis, fungsi sigmoid dapat dirumuskan dengan Persamaan (1).

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

Keterangan:
e = Konstanta Euler,
x = Input.

Grafik fungsi aktivasi sigmoid dapat di visualisasikan pada gambar di bawah ini.



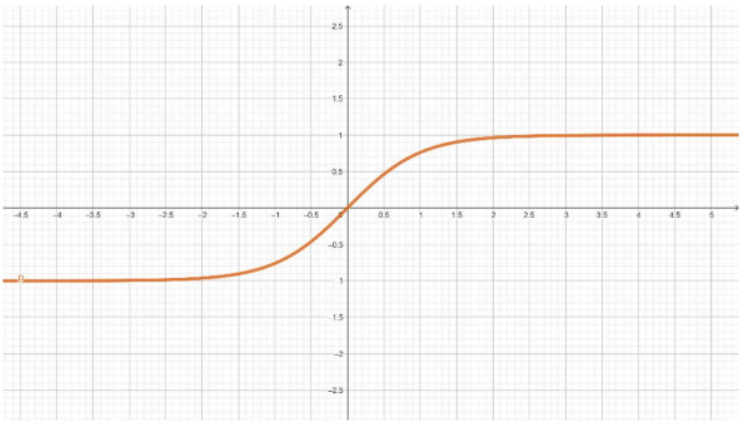
Gambar 1. Grafik Fungsi Aktivasi Sigmoid (Wibawa, 2016).

2.11.2 Fungsi Aktivasi Tanh

Fungsi aktivasi *Tanh* atau tangen hiperbolik merupakan fungsi aktivasi yang mirip dengan fungsi aktivasi sigmoid, tetapi berbeda dalam rentang *output*. Fungsi sigmoid memetakan nilai *input* dalam rentang 0 hingga 1, sedangkan fungsi *Tanh* memetakan nilai *input* dalam rentang -1 hingga 1 dengan berpusat pada nol (Kurniawan *et al*., 2024). Secara matematis, fungsi Tanh dapat dirumuskan dengan Persamaan (2).

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

Grafik fungsi aktivasi tanh dapat di visualisasikan pada gambar di bawah ini.



Gambar 2. Grafik Fungsi Aktivasi Tanh (Wibawa, 2016).

2.11.3 Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

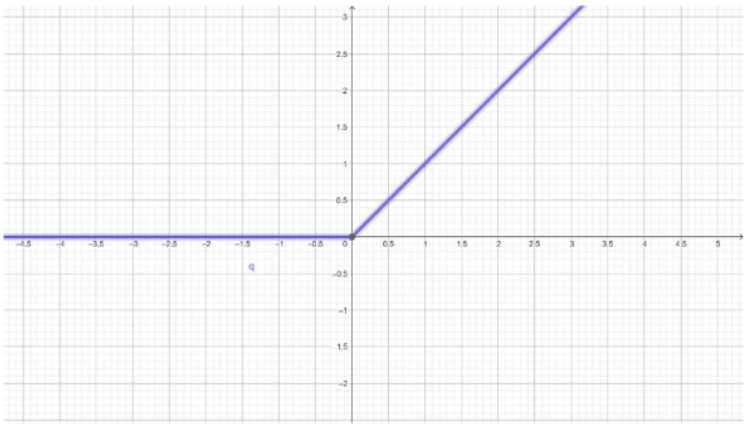
Fungsi ReLU adalah fungsi aktivasi dalam jaringan saraf tiruan yang merupakan bentuk pengembangan dari fungsi linear. Fungsi ReLU akan menghasilkan nilai 0 jika nilai *input*nya kurang atau sama dengan 0 (Wibawa, 2016). Secara matematis, fungsi ReLU dapat dirumuskan dengan Persamaan (3).

$$f(x) = max(0, x) \tag{3}$$

Keterangan:

$$f(x) = \text{Nilai output dari fungsi aktivasi ReLU},$$
$$x = \text{Input},$$
$$max(0, x) = \text{Fungsi nilai maksimum antara 0 dan } x.$$

Grafik fungsi aktivasi ReLU dapat di visualisasikan pada gambar di bawah ini.



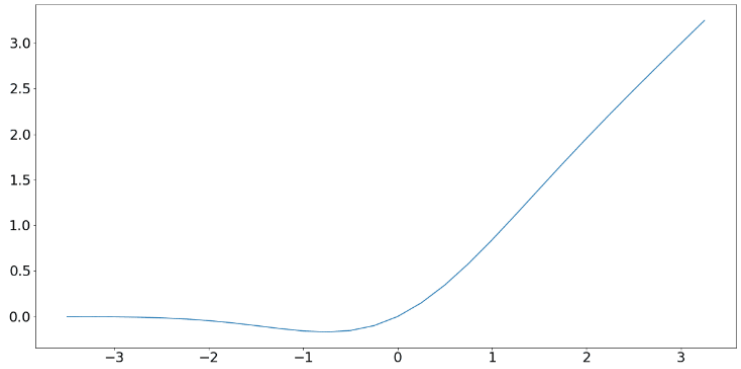
Gambar 3. Grafik Fungsi Aktivasi ReLU (Wibawa, 2016).

2.11.4 Fungsi Aktivasi *Gaussian Error Linear Unit* (GeLU)

Fungsi GeLU merupakan fungsi yang dirancang untuk mengkombinasikan fungsi beberapa fungsi aktivasi termasuk *Sigmoid*, *Tanh*, dan ReLU dengan tujuan meningkatkan kinerja jaringan (Kurniawan *et al*., 2024). Secara matematis, fungsi GeLU dapat dirumuskan dengan Persamaan (4).

$$f(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right) \tag{4}$$

Grafik fungsi aktivasi GeLU dapat di visualisasikan pada gambar di bawah ini.



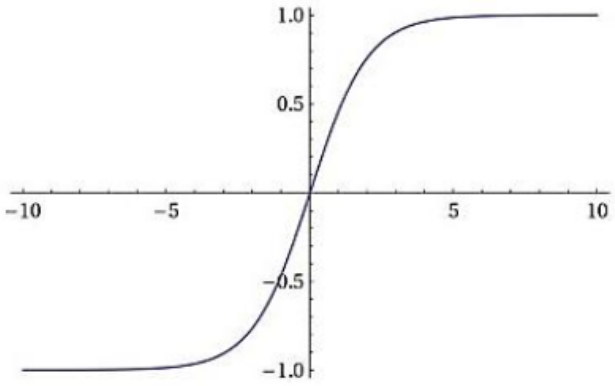
Gambar 4. Grafik Fungsi Aktivasi GeLU (Akil, 2023).

2.11.5 Fungsi Aktivasi Softmax

Fungsi aktivasi softmax adalah fungsi yang digunakan dalam perhitungan probabilitas untuk menentukan klasifikasi *multi class* dengan *output class* yang mempunyai nilai probabilitas yang paling tinggi. Fungsi aktivasi softmax menghasilkan *output* dengan nilai probabilitas antara 0 hingga 1 (Purwitasai & Soleh, 2022). Secara matematis, fungsi softmax dapat dirumuskan dengan Persamaan (5).

$$f(X_i) = \frac{\exp(X_i)}{\sum_{j=0}^{k} \exp(X_j)}, \quad \text{nilai } i = 0, 1, 2, \ldots, k \tag{5}$$

Grafik fungsi aktivasi softmax dapat di visualisasikan pada gambar di bawah ini.



Gambar 5. Grafik Fungsi Aktivasi Softmax (Purwitasai & Soleh, 2022).

2.12 *AdamW Optimizer*

AdamW adalah pembaharuan dari *optimizer* Adam dengan mengatasi kelemahan penggunaan *weight decay* pada Adam. Pada *optimizer* Adam, *weight decay* diterapkan bersamaan dengan perhitungan gradient, sehingga efektivitas regulasi weight decay menjadi tidak optimal karena nilainya terpengaruh oleh perhitungan gradient. Pada *optimizer* AdamW, *weight decay* dipisahkan dengan perhitungan gradient dan diterapkan langsung pada pembobotan model. Pemisahan *weight decay* pada AdamW membuat pengaturan bobot menjadi lebih stabil sehingga model dapat belajar lebih baik , mengurangi *loss*, dan memiliki kemampuan generalisasi yang lebih tinggi dibandingkan Adam (Meng *et al*., 2023).

Formulasi pembaharuan AdamW dijabarkan dengan Persamaan (6), (7), (8), (9), (10), dan (11) berikut:

$$g_t = \nabla_\theta f(\theta_t) \tag{6}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(g_t)^2 \tag{8}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{9}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{10}$$

$$\theta_t' = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{11}$$

$$\theta_{t+1} = \theta_t' - \alpha \lambda \theta_t \tag{12}$$

Keterangan:

θ_t = bobot atau parameter model pada iterasi ke-t,

g_t = gradien dari fungsi kehilangan (*loss*) terhadap parameter θ_t,

$f(\theta_t)$ = fungsi kehilangan (*loss function*) yang dioptimasi,

m_t = estimasi momen pertama,

v_t = estimasi momen kedua dari gradien,

$\hat{m}_t$ = estimasi momen pertama yang telah dikoreksi bias,

$\hat{v}_t$ = estimasi momen kedua yang telah dikoreksi bias,

β = koefisien peluruhan eksponensial untuk estimasi momen,

α = laju pembelajaran (*learning rate*),

ϵ = konstanta kecil untuk mencegah pembagian dengan nol.

λ = *weight decay*,

θ'_t = parameter sementara sebelum diterapkan *weight decay*,

θ_{t+1} = parameter model setelah diperbarui dengan *weight decay*.

2.13 *Hyperparameter*

Hyperparameter adalah parameter yang mengatur proses pelatihan dan struktur model pembelajaran mesin. Berbeda dengan parameter model yang dipelajari selama pelatihan, nilai parameter dalam *hyperparameter* ditetapkan sebelum proses pelatihan dimulai. *Hyperparameter* berperan penting dalam menentukan kinerja model (Ilemobayo *et al*., 2024).

Terdapat beberapa *hyperparameter* utama yang berperan dalam pengaturan kinerja model, di antaranya:

1. *Learning Rate*
   Dalam algoritma optimisasi berbasis gradien, *learning rate* mengatur seberapa besar peubahan model serta menentukan ukuran langkah pada setiap iterasi proses optimisasi.

2. Jumlah *Epoch*
   Menggambarkan jumlah siklus penuh pelatihan pada seluruh data.

3. Ukuran *Batch*
   Menentukan jumlah sampel yang digunakan untuk menghitung gradien pada setiap iterasi.

4. Jumlah Lapisan dan Jumlah Unit Tiap Lapisan Tersembunyi
   Menentukan kapasitas model untuk mempelajari pola-pola yang kompleks.

5. Parameter Regulasi
   Digunakan untuk mengontrol kompleksitas yang diterapkan pada parameter model, sehingga membantu mencegah *overfitting*. Salah satu bentuk parameter yang umum digunakan adalah *weight decay*. Nilai *weight decay* yang terlalu kecil menyebabkan regularisasi kurang efektif sehingga berpotensi mengalami *overfitting*, sedangkan nilai yang terlalu besar dapat menghambat kemampuan

model dalam mempelajari pola data sehingga berpotensi menyebabkan *underfitting*.

Menurut Elgadawi *et al*.,(2021), beberapa teknik umum dalam menentukan *hyperparameter* optimal antara lain :

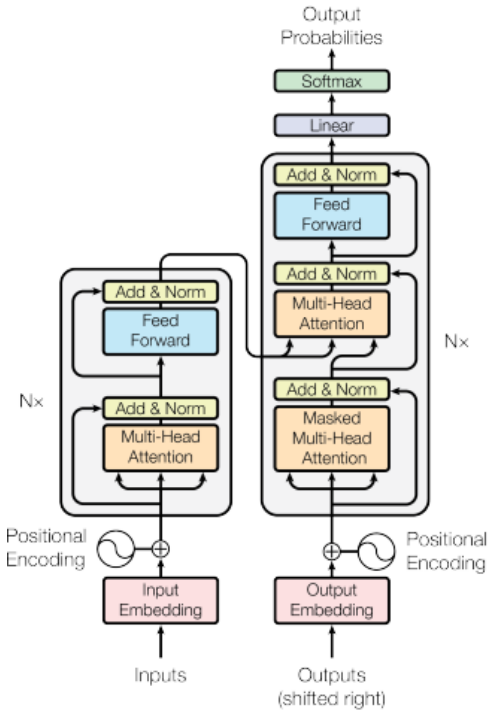
1. *Grid Search*
   Dengan mengevaluasi semua kombinasi *hyperparameter* dalam ruang pencarian yang telah ditentukan. Setiap konfigurasi di uji, kemudian dipilih yang menghasilkan performa terbaik.

2. *Random Search*
   Dengan memilih beberapa kombinasi secara acak dalam ruang pencarian, Meskipun acak, metode ini lebih efisien dari segi waktu dan sering menghasilkan hasil yang mendekati optimal.

3. *Bayesian Optimization*
   Dengan menggunakan pendekatan probabilistik, memanfaatkan hasil iterasi sebelumnya untuk memandu pencarian hyperparameter berikutnya. Pendekatan ini memperlakukan proses tuning sebagai masalah optimisasi fungsi *black-box* dan lebih efisien dibandingkan metode *brufe-force*.

2.14 Transformer

Transformer adalah arsitektur jaringan saraf sederhana yang didasarkan pada attention mechanisms tanpa menggunakan recurrence (RNN) maupun convolutions (CNN). Vaswani dan rekannya pada 2017 dalam artikelnya yang berjudul *Attention is All You Need* memperkenalkan transoformer sebagai model yang menggunakan pendekatan *encoder-decoder* dengan mengandalkan mekanisme *self-attention* dan lapisan *fully connected* pada kedua bagian baik encoder maupun decoder. Arsitektur model ditampilkan pada Gambar 6.



Gambar 6. Arsitektur Model Transformer (Vaswani *et al*., 2017).

2.14.1 *Encoder* dan *Decoder*

Encoder terdiri dari enam lapisan identik, dimana setiap lapisan memiliki dua sub-lapisan yaitu *multi-head self-attention* dan jaringan *feed-forward* posisi tertentu. Setiap lapisan menggunakan koneksi residual dan di ikuti dengan *layer normalization*. Artinya, setiap sub-lapisan menghasilkan *output* LayerNorm dengan Sublayer merupakan fungsi yang di implementasikan oleh sub-lapisan itu sendiri. *Output* dari setiap sub-lapisan berdimensi $d_{\text{model}} = 512$.

Decoder terdiri dari enam lapisan identik. Selain dua sub-lapisan di setiap lapisan encoder, ditambahkan sub-lapisan ketiga yang melakukan *multi-head attention* pada *output encoder*. Serupa dengan *encoder*, digunakan koneksi residual, *layer normalization*, dan memodifikasi sub-lapisan *self-attention* untuk mencegah posisi memperhatikan posisi berikutnya dan memastikan bahwa prediksi untuk posisi i hanya bergantung pada *output* yang diketahui pada posisi sebelumnya.

2.14.2 *Attention*

Fungsi *Attention* merupakan pemetaan antara *query* dan himpunan pasangan *key-value* menjadi suatu *output*, dimana *query, key, value*, dan *output* diformulasikan dalam bentuk vektor. Nilai keluaran (*output*) dihitung sebagai jumlah berbobot dari semua *value*, dengan bobot yang ditentukan dari fungsi kesesuaian (*similarity*) antara *query* dan *key* yang bersesuaian.

2.14.3 *Scaled Dot-Product Attention*

Input Scaled Dot-Product Attention terdiri dari *queries* dan *key* dengan dimensi d_{k}, serta *value* dengan dimensi d_{v}. *Dot product* antara setiap *query* dan semua *key* dihitung menggunakan *attention mecanism*, kemudian dinormalkan dengan membagi hasil dengan $\sqrt{d_{\text{k}}}$, dan fungsi softmax diterapkan untuk memperoleh bobot pada setiap *value*. *Scaled Dot-Product Attention* diperoleh melalui operasi matriks. Dalam penerapannya, sejumlah *query* diperoses secara bersamaan yang disusun dalam matriks Q, sedangkan *key* dan *value* disusun dalam matriks K dan V. Matriks keluaran dihitung menggunakan rumus pada Persamaan (13).

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

Keterangan:

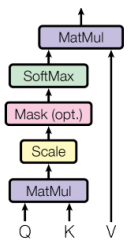
Q = *Query*, representasi elemen input yang akan fokus pada elemen lain,

K = *Key*, representasi elemen untuk mencocokkan relevansi,

V = *Value*, representasi elemen yang mengandung informasi aktual,

$\sqrt{d_{\text{k}}}$ = dimensi dari vektor *key*, digunakan untuk normalisasi.

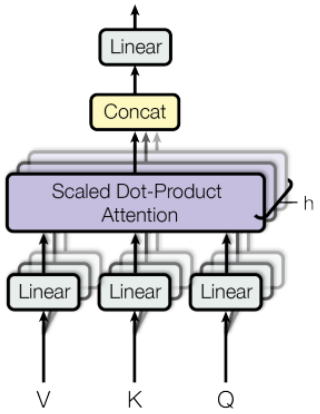
Bagian *Scaled Dot-Product Attention* ditunjukkan pada Gambar 7.



Gambar 7. *Scaled Dot-Product Attention* (Vaswani *et al*., 2017).

2.14.4 *Multi-Head Attention*

Multi-Head Attention memproyeksikan *queries, keys*, dan *values* secara linear sebanyak h kali dengan parameter pembelajaran yang berbeda pada dimensi d_{k} dan d_{v}. Pada setiap hasil proyeksi, fungsi *attention* dijalankan secara paralel untuk menghasilkan *output values* berdimensi d_{v}. Seluruh *output* kemudian digabung dan diproyeksikan kembali untuk membentuk hasil akhir. Bagian *Multi-Head Attention* ditunjukkan pada Gambar 8.



Gambar 8. *Multi-Head Attention* (Vaswani *et al*., 2017).

Pendekatan *Multi-Head Attention* memungkinkan model untuk secara bersamaan memperhatikan informasi dari berbagai subruang representasi dan posisi yang berbeda, ditunjukkan pada Persamaan (14) dan (15).

$$\mathit{MultiHead}(Q, K, V) = \mathit{Concat}(\mathit{head}_1, \ldots, \mathit{head}_h)W^O \tag{14}$$

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

dimana proyeksi-proyeksi tersebut adalah matriks parameter:

$$W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, \quad W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, \quad W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}, \quad W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$$

Keterangan:

$Q = \mathit{Query}$,

$K = \mathit{Key}$,

$V = \mathit{Value}$,

head$_i$ = *Output* dari mekanisme *attention* pada kepala ke-i.

W_i^Q = Matriks parameter untuk memproyeksikan *query* pada kepala ke-*i*,
W_i^K = Matriks parameter untuk memproyeksikan *key* pada kepala ke-*i*,
W_i^V = Matriks parameter untuk memproyeksikan *value* pada kepala ke-*i*,
W^O = Matriks parameter untuk menggabungkan semua *output* kepala,
d_{model} = dimensi model,
d_k = dimensi vektor *query* dan *key*,
d_v = dimensi vektor *value*,
h = jumlah kepala *attention*.

2.14.5 *Position-Wise Feed-Forward Networks*

Selain sub-lapisan *attention*, setiap lapisan pada *encoder* dan *decoder* juga terdapat jaringan *feed-forward* yang terhubung penuh, serta diterapkan secara terpisah dan identik pada setiap posisi. Jaringan ini terdiri dari dua transformer linear dengan fungsi aktivasi ReLU yang dinyatakan dalam Persamaan (16).

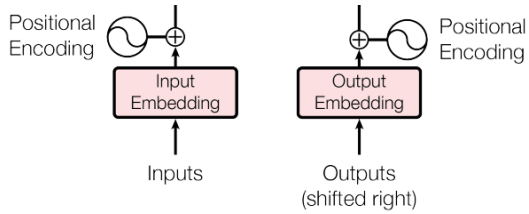
$$FFN(x) = max(0,\, xW_1 + b_1)W_2 + b_2 \tag{16}$$

Keterangan:
x = *input* ke *feed-forward network* (FFN),
W_1, W_2 = matriks bobot lapisan pertama dan kedua,
b_1, b_2 = bias lapisan pertama dan kedua,
$max(0, \cdot)$ = fungsi aktivasi ReLU (*Rectified Linear Unit*).

Meskipun transformasi linear diterapkan sama pada semua posisi, tetapi parameter yang digunakan berbeda untuk setiap lapisan. Hal ini disebut sebagai dua operasi konvolusi dengan ukuran kernel 1. Dimensi *input* dan *output* yaitu $d_model = 512$, dan dimensi lapisan tersembunyi yaitu $d_{ff} = 2048$.

2.14.6 *Positional Encoding*

Positional encoding ditambahkan pada *input embeddings* dibagian dasar *encoder* dan *decoder*. Karena model tidak memiliki mekanisme rekursif dan konvolusi secara alami sehingga model tidak memiliki kemampuan untuk memahami urutan dalam *sekuens input*. Oleh karena itu, *positional encoding* digunakan untuk menambahkan informasi mengenai posisi relatif atau absolut dari setiap token. *Positional encoding* memiliki dimensi yang sama dengan d_model pada *embedding*, sehingga keduanya dapat dijumlahkan. Posisi *Positional Encoding* ditunjukkan pada Gambar 9.



Gambar 9. *Positional encoding* (Vaswani *et al*., 2017).

Positional encoding pada model BART dan BioBART bersifat *Learned Positional Embeddings* bukan menggunakan fungsi *sinus* dan *cosinus* seperti pada transformer asli. Artinya bahwa posisi token pertama, kedua, ketiga, dan seterusnya diwakili oleh vektor yang dipelajari selama *pre-training* dan *fine-tuning*, bukan rumus tetap berbasis sinus dan cosinus seperti pada transformer asli.

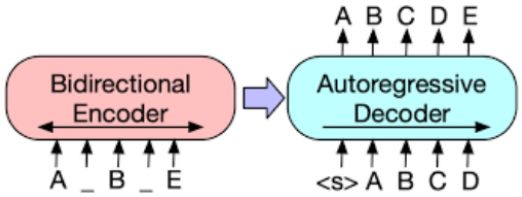
2.14.7 *Embedding* dan Fungsi Softmax

Serupa dengan model-transduksi urutan lainnya, menggunakan *learned embeddings* untuk mengubah token *input* dan *output* menjadi vektor berdimensi d_model. Selain itu, digunakan transformasi linear yang dipelajari dan fungsi softmax untuk mengubah *output decoder* menjadi probabilitas *token* berikutnya. Matriks bobot yang sama digunakan bersamaan pada lapisan *input embedding, output embedding*, dan transformasi linear sebelum softmax. Pada lapisan *embedding*, bobot dikalikan dengan $\sqrt{d_{\text{model}}}$.

2.15 *Bidirectional and Autoregressive Transformer* (BART)

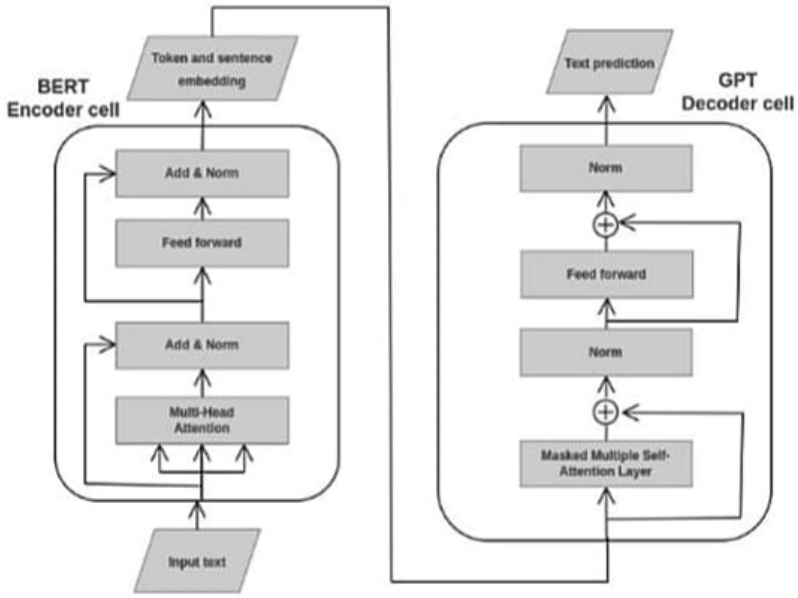
Bidirectional and Autoregressive Transformer (BART) adalah model transformer inovatif di bidang NLP dan *Machine Learning* yang menggabungkan dua arsitektur transformer yaitu *Encoder* BERT (*Bidirectional Encoder Representations from Transformer* dan *Decoder* GPT (Generative Pretrained Transformer. BART memiliki kemampuan dalam menghasilkan *output* teks dua arah, menggabungkan konteks dari kiri ke kanan dan sebaliknya sehingga mampu menghasilkan teks yang lebih bermakna dibandingkan model transformer tradisional. Model ini telah berhasil diterapkan dalam berbagai tugas NLP, seperti peringkasan teks, penerjemahan mesin, dan pembuatan bahasa alami (Hartawan *et al*., 2024).

Pada BART, *encoder* BERT memungkinkan model lebih memahami konteks kalimat, sedangkan *decoder* GPT memungkinkan model mampu menghasilkan teks yang mengalir secara alami dan sesuai konteks. *Input* yang diterima berupa teks yang dikonversi menjadi token yang kemudian dimasukkan kedalam representasi vektor. Selanjutnya, *encoder* akan memproses representasi vektor dari token-token *input* dan menghasilkan representasi tersembunyi untuk setiap token yang kemudian diproses oleh *decoder* dan menghasilkan token *output* secara *autoregressive*. Mekanisme BART dapat ditunjukkan pada Gambar 10.



Gambar 10. Mekanisme BART (Hartawan *et al*., 2024).

Pada *encoder*, *multi-head attention* digunakan dalam memproses teks mentah yang telah ditokenisasi, sedangkan *addititional pre-processing* dilakukan sesuai dengan kebutuhan tugas. *Decoder* GPT menerima *masked embedding* yang dihasilkan oleh *encoder* BERT yang kemudian diteruskan dalam beberapa blok *masked attention* independen. Setiap blok berdasarkan arsitektur dasar *multi-head attention* namun beroperasi secara berurutan. Lapisan ini mempelajari cara mendekode *masked embedding* menjadi token bermakna secara semantik dengan memperhatikan berbagai representasi *input* yang di mask dengan tingkat berbeda. Arsitektur BART dapat ditunjukkan pada Gambar 11.



Gambar 11. Arsitektur BART (Hartawan *et al*., 2024).

2.16 *Biomedical-Bidirectional and Autoregressive Transformer* (BioBART)

BioBART (*Biomedical-Bidirectional and Autoregressive Transformer*) adalah model bahasa generatif *auto-regressive* yang diadaptasi dari model BART ke domain medis. BioBART yang dilatih dalam domain biomedis mampu melampaui kinerja BART dan menetapkan *baseline* kuat dalam berbagai tugas NLG (*Natural Language Generation*) seperti *dialogue system*, *abstractive summarization*, pengaitan entitas, dan *Named Entity Recognition* (NER) generatif (Yuan *et al*., 2022).

Dalam penelitian ini, model BioBART digunakan dalam dua tahap yaitu sebagai model *pretrained* dan model *fine-tuned*. Model BioBART *pretrained* adalah model yang telah dilatih sebelumnya dalam korpus biomedis berskala besar serta digunakan secara langsung tanpa proses pelatihan ulang pada data pelatihan. Model ini berfungsi sebagai *baseline* yang menggambarkan kinerja awal model BioBART dalam tugas peringkasan teks artikel biomedis. Kemudian dilakukan proses *fine-tuning* untuk menyesuaikan parameter model terhadap data dan karakteristik tugas peringkasan.

2.17 *Fine-Tuning*

Fine-Tuning adalah proses penyesuaian lanjutan terhadap *pretrained model* setelah tahap inisialisasi dengan melatih kembali model sehingga parameter-parameter model dapat beradaptasi dengan karakteristik tugas atau domain tertentu. Selain untuk menyesuaikan parameter model pra-latih, *fine-tuning* juga bertujuan untuk mentransfer pengetahuan dari data dengan skala besar sehingga meningkatkan performa model pada kondisi data terbatas (*low-resource*) (Gao *et al*., 2024).

Fine-Tuning dilakukan dalam beberapa tahap. Tahap pertama, inisialisasi model baru menggunakan parameter model pra-latih sehingga membawa pengetahuan awal yang diperoleh dari data berskala besar. Tahap kedua, penyesuaian awal atau *pre-fine-tuning* menggunakan data sumber dari domain atau bahasa target untuk menyesuaikan representasi model terhadap data baru. Terakhir, model dilatih kembali (*fine-tuned*) menggunakan data pelatihan dengan pasangan kalimat sumber dan target seperti dataset PMC dengan pasangan *Article* sebagai kalimat sumber dan *Abstract* sebagai target dengan menggunakan fungsi *cross-entropy loss*, mengikuti prosedur umum *transfer learning*.

Fine-Tuning memiliki sejumlah keunggulan yang menjadikannya sangat berguna dalam *transfer learning*, yaitu memungkinkan penyesuaian model pra-latih agar lebih sesuai dengan karakteristik domain baru, meningkatkan performa model secara signifikan, dan menghemat waktu serta sumber daya komputasi dibandingkan pelatihan dari awal. Sehingga *fine-tuning* dijadikan sebagai pendekatan yang umum digunakan dalam mengadaptasi model besar seperti BioBART terhadap tugas spesifik bidang biomedis.

Fine-Tuning BioBART dikatakan optimal ketika menghasilkan kinerja terbaik berdasarkan metrik evaluasi yang relevan. Kualitas ringkasan dievaluasi dengan ROUGE sebagai metrik kesesuaian dengan ringkasan referensi (Lin *et al*., 2004) dan BERTScore untuk mengukur kesamaan semantik antara ringkasan hasil model dan ringkasan referensi (Zhang *et al*., 2020). Kondisi optimal ditentukan pada epoch dengan nilai evaluasi tertinggi pada data validasi disertai dengan pola konvergensi loss yang stabil tanpa indikasi overfitting, berdasarkan praktik model *deep learning* (Goodfellow *et al*., 2016). Penggunaan AdamW di pilih karena memampuannya menghasilkan pembaharuan parameter yang lebih stabil melalui mekanisme *weight decay* terpisah yang mendukung proses *fine-tuning* lebih optimal.

2.18 Evaluasi Model

Ringkasan artikel yang diperoleh dari model BioBART perlu dilakukan evaluasi untuk menilai seberapa baik ringkasan yang dihasilkan oleh model. Evaluasi dilakukan menggunakan dua metrik evaluasi yaitu ROUGE *scores* dan BERTScore untuk menilai kesesuaian isi ringkasan terhadap teks asli secara keseluruhan.

2.18.1 *Recall-Oriented Understudy for Gisting Evaluastion* (ROUGE)

ROUGE adalah kumpulan metrik yang digunakan untuk menilai kualitas ringkasan teks yang dihasilkan secara otomatis dan terjemahan mesin dalam NLP (Hartawan *et al*., 2024). Pada evaluasi ini, ringkasan yang dihasilkan model dibandingkan dengan ringkasan referensi (*Abstract*) yang dibuat oleh manusia dan menjadi standar kualitas teks ringkasan yang dihasilkan. Terdapat dua varian ROUGE yaitu ROUGE-N dan ROUGE-L dengan ROUGE *Score* terdiri dari *presicion*, *recall*, dan F1-*Score*. ROUGE-N yang mengukur *recall* berdasarkan n-gram atau urutan N kata yang berurutan dalam teks dan ROUGE-L yang fokus pada *Longest Common Subsequent* (LCS) atau kata-kata berurutan terpanjang antara ringkasan teks yang dihasilkan model dan ringkasan teks referensi (Lin *et al*., 2004). Nilai N yang umum digunakan yaitu 1 menghitung *unigram*(ROUGE-1), 2 menghitung *bigram* (ROUGE-2), dan L menghitung kata-kata berurutan terpanjang.

Persamaan yang digunakan untuk menghitung ROUGE *score* adalah sebagai berikut (Idhafi *et al*., 2023):

1. *Recall*
   Recall adalah metode yang digunakan dalam mengukur jumlah prediksi yang relevan dengan cara menghitung jumlah kata yang sama, baik *unigram*, *bigram*, atau LCS dibagi dengan keseluruhan kata pada ringkasan referensi. Persamaan dalam menghitung *recall* dirumuskan dalam Persamaan (17),(18),dan (19).

$$\text{ROUGE-1 } \textit{Recall} = \frac{\text{jumlah } \textit{unigram} \text{ yang sama}}{\text{keseluruhan kata di ringkasan referensi}} \tag{17}$$

$$\text{ROUGE-2 } \textit{Recall} = \frac{\text{jumlah } \textit{bigram} \text{ yang sama}}{\text{keseluruhan kata di ringkasan referensi}} \tag{18}$$

$$\text{ROUGE-L } \textit{Recall} = \frac{\text{LCS}}{\text{keseluruhan kata di ringkasan referensi}} \tag{19}$$

2. *Precision*

   Precision adalah metode yang digunakan dalam mengukur jumlah yang diprediksi relevan dengan cara menghitung jumlah kata yang sama, baik *unigram*, *bigram*, atau LCS dibagi dengan keseluruhan kata pada ringkasan yang dihasilkan model. Persamaan dalam menghitung *recall* dirumuskan dalam Persamaan (20), (21), dan (22).

$$\text{ROUGE-1 } \textit{Precision} = \frac{\text{jumlah } \textit{unigram} \text{ yang sama}}{\text{keseluruhan kata di ringkasan model}} \tag{20}$$

$$\text{ROUGE-2 } \textit{Precision} = \frac{\text{jumlah } \textit{bigram} \text{ yang sama}}{\text{keseluruhan kata di ringkasan model}} \tag{21}$$

$$\text{ROUGE-L } \textit{Precision} = \frac{\text{LCS}}{\text{keseluruhan kata di ringkasan model}} \tag{22}$$

3. F1-*Score*

   F1-*Score* adalah metode yang digunakan dalam mengukur nilai rata-rata hermonik (*harmonic mean*) antara *recall* dan *precision* yang dirumuskan dalam Persamaan (23).

$$\text{F1-}\textit{Score} = 2 \times \frac{precision \times recall}{precision + recall} \tag{23}$$

2.18.2 BERTScore

BERTScore adalah metrik lanjutan berbasis kesamaan yang digunakan dalam mengevaluasi hasil generalisasi teks dengan memanfaatkan *contextual embeddings* dari model BERT. BERTScore menghitung kesamaan semantik antara teks ringkasan referensi dan teks ringkasan target pada tingkat token (Mukherjee *et al*., 2025). Setiap token dalam teks ringkasan referensi dan teks ringkasan target diubah menjadi vektor berdimensi tinggi menggunakan *embedding* BERT, yang masing-masing dilambangkan sebagai *r* untuk token teks ringkasan referensi dan *c* untuk token teks ringkasan target.

Persamaan yang digunakan untuk menghitung BERTScore adalah sebagai berikut (Mukherjee *et al*., 2025):

1. *Recall*

   Recall dari BERTScore adalah metode yang digunakan dalam mengukur seberapa baik setiap token dalam teks ringkasan referensi terwakili dalam teks ringkasan

target. Persamaan menghitung *recall* dirumuskan dalam Persamaan (26).

$$R_{\text{BERT}} = \frac{1}{N_r} \sum_{j=1}^{N_r} \max_{i \in \{1,\ldots,N_c\}} \cos\left(r_j, c_i\right) \tag{26}$$

Keterangan:

N_r = jumlah token pada teks ringkasan referensi,

N_c = jumlah token pada teks ringkasan target,

r_j = vektor embedding token ke-j pada teks ringkasan referensi,

c_i = vektor embedding token ke-i pada teks ringkasan target,

$\cos(\cdot, \cdot)$ = fungsi kesamaan kosinus antar embedding token.

2. *Precision*

   Precision dari BERTScore adalah metode yang digunakan dalam mengukur tentang seberapa baik setiap token dalam teks ringkasan target sesuai dengan token paling mirip dalam teks riingkasan referensi. Persamaan dalam menghitung *precision* dirumuskan dalam Persamaan (27).

$$P_{\text{BERT}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \max_{j \in \{1,\ldots,N_r\}} \cos\left(r_j, c_i\right) \tag{27}$$

   Keterangan:

   N_r = jumlah token pada teks ringkasan referensi,

   N_c = jumlah token pada teks ringkasan target,

   r_j = vektor embedding token ke-j pada teks ringkasan referensi,

   c_i = vektor embedding token ke-i pada teks ringkasan target,

   $\cos(\cdot, \cdot)$ = fungsi kesamaan kosinus antar embedding token.

3. F1-*Score*

   F1-*Score* dari BERTScore adalah metode yang digunakan sebagai rata-rata harmonik atau keseimbangan dari *precision* dan *recall*. Persamaan dalam menghitung F1-*Score* dirumuskan dalam Persamaan (28).

$$F1 - \textit{Score} = \frac{2 \times P_{\text{BERT}} \times R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{28}$$

   Keterangan:

   P_{BERT} : nilai presisi berdasarkan kesamaan semantik antar token,

   R_{BERT} : nilai recall berdasarkan kesamaan semantik antar token.

2.19 Uji Signifikansi Statistik dalam Evaluasi Model NLP

Uji signifikansi statistik digunakan untuk menentukan apakah perbedaan kinerja yang dihasilkan dua model bersifat signifikan secara statistik atau akibat variasi acak data uji. Dalam penelitian bidang NLP dan *Machine Learning* (ML), umumnya perbandingan kinerja model dilakukan menggunakan dataset uji yang sama, sehingga skor hasil metrik evaluasi merupakan data yang saling berpasangan.

Menurut Peyrard *et al*.,(2021), evaluasi NLP sebaiknya dilakukan secara *paired evaluation* karena setiap model diuji diuji pada unit data yang sama, sehingga memungkinkan perbandingan yang lebih adil dan akurat antar model. Goyal *et al*.,(2023) juga menegaskan bahwa penelitian NLP modern perlu melibatkan uji signifikansi statistik untuk memastikan peningkatan performa yang dihasilkan model benar bermakna secara statistik.

Salah satu metode yang umum digunakan adalah *paired t-test*, yaitu uji parametrik yang digunakan untuk menguji apakah terdapat perbedaan rata-rata yang signifikan antara dua kelompok data yang saling berpasangan dengan memperhatikan distribusi selisih mean. Menurut Zhu., (2020) dalam *paired t-test* hipotesis statistik yang diuji dirumuskan sebagai berikut :

H_0 : tidak terdapat perbedaan rata-rata kinerja yang signifikan antara dua model yang dibandingkan.

H_1 : terdapat perbedaan rata-rata kinerja yang signifikan antara dua model yang dibandingkan.

Secara matematis, statistik uji *paired t-test* dirumuskan sebagai berikut:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{29}$$

Keterangan:
$\bar{d}$: rata-rata selisih skor evaluasi antara dua model,
s_d : simpangan baku dari selisih skor evaluasi,
n : jumlah pasangan data,
t : nilai statistik uji *paired t-test*.

Pengujian hipotesis dilakukan pada taraf signifikansi $\alpha = 0{,}05$. Keputusan uji diambil dengan menolak hipotesis nol ketika p-value lebih kecil dari taraf signifikansi. Sehingga, penggunaan *paired t-test* mampu memberikan dasar statistik yang kuat untuk menilai apakah perbedaan kinerja antar model yang dibandingkan bersifat signifikan secara statistik.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilakukan secara studi pustaka di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang beralamat di Jl. Prof. Dr. Sumantri Brojonegoro, No.1, Gedong Meneng, Bandar Lampung.

3.1.2 Waktu Penelitian

Penelitian ini dilaksanakan pada semester ganjil tahun akademik 2025/2026 tepatnya pada bulan September 2025- Maret 2026 yang terdiri dari tiga tahap. Tahap penelitian dapat dilihat pada Tabel 2.

Tabel 2. Waktu Penelitian

Tahap	Kegiatan	2025												2026											
		September				Oktober				November				Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur dan Penentuan Tema Penelitian																												
	Pengumpulan Data																												
	Penyusunan BAB I-III																												
	Seminar Usul Penelitian																												
2	*Exploratory Data Analysis*																												
	Preprocessing Data																												
	Splitting Data 1. Data Pelatihan 2. Data Validasi 3. Data Pengujian																												
	Model BioBART																												
	Evaluasi Kinerja Model																												
3	Penyusunan BAB IV-V																												
	Seminar Hasil Penelitian																												
	Sidang Komprehensif																												

3.2 Data Penelitian

3.2.1 Data

Data yang digunakan dalam penelitian ini adalah dataset PMC (*PubMed Central*) yang merupakan repositori *open-access* yang merupakan bagian dari NLM (*National Library of Medicine*) dan dikelola oleh NCBI (*National Center for Biotechnology Information*) yang mengelola database bioinformatika dan biomedical informatics. Data PMC berisi sekumpulan artikel ilmiah biomedis dalam bentuk *full-text* dengan format filelist.csv, filelist.txt, dan tar.gz. Dataset tersebut dapat di akses melalui link berikut : https://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/oa_noncomm/xml/

Dataset PMC memiliki jutaan data. Pada penelitian ini, dataset tersebut hanya digunakan sebanyak 10.000 data di ambil dengan proses *parsing* dari format tar.gz sebesar 650 MB atau 78.195 data yang diperbaharui terakhir pada tanggal 26 Juni 2025 karena keterbatasan komputasi. Dataset dapat diakses melalui link berikut: https://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/oa_noncomm/xml/oa_noncomm_xml.PMC002xxxxxx.baseline.2025-06-26.tar.gz. Dataset sebanyak 10.000 artikel dengan atribut *Body* yang merupakan isi teks artikel ilmiah lengkap yang didalamnya mencakup bagian seperti *introduction, Methods, Results, Discussion, Conclusion,* dan bahkan sering kali *Supplementary/Reference*, dan *Abstract* yang merupakan ringkasan referensi, dan *Keyword Article* sebagai pendukung tambahan. Karena pada dataset PMC sebagian tidak menyertakan *keyword*, sehingga dalam mendukung evaluasi pada penelitian ini *keyword* article di cari melalui proses *Keyword Extraction* menggunakan perhitungan TF-IDF *keyword* sehingga 10 kata dengan nilai tertinggi sebagai *keyword article*.

Data yang digunakan dibagi menjadi 90% data model dan 10% data pengujian. Dengan 90% data model yaitu sebanyak 9.000 data, artikel dibagi sebanyak 75% atau 6.750 data pelatihan dan 25% atau 2.250 data validasi. Data memiliki rata-rata sebesar 181 kata, minimal sebesar 100 kata, dan maksimal sebesar 1240 kata pada label *Abstract*, serta pada label *Article* memiliki rata-rata 1525 kata, minimal 500 kata, dan maksimal 12210 kata. Dataset hasil pembagian tersebut dapat diakses melalui tautan berikut: https://www.kaggle.com/datasets/indriwardani/hasilsplit. Keterbatasan jumlah data tersebut tidak berdampak signifikan terhadap proses evaluasi pemodelan, sehingga tujuan penelitian tetap dapat dicapai dengan baik.

Berikut ini struktur dataset PMC yang ditampilkan pada Tabel 2, 3, dan 4.

Tabel 2. Dataset Pelatihan

ID	***Body/Article***	***Abstract***	***Keyword Article***
0	Mercury exposure is a known public health risk. Exposure generally occurs through dental...	Exposure to mercury from environmental sources, such as fish consumption, poses potential health ris...	vermont, fish consumption, mercury, burlington, fish, women childbearing, childbearing age,...
1	Traumatic knee injury is a frequently encountered condition with a reported incidence of 5.3...	To evaluate meniscal status change on follow-up MRI after 1 year, prognostic factors and association...	meniscal, follow mri, mri appearance, horns, initial mri, degenerative lesions, tears, ordinal logis...
2	The regulation of mRNA biogenesis and decay plays an important role in the control of gene...	4E-transporter (4E-T) is one of several proteins that bind the mRNA 5 cap-binding protein, eukaryoti..	4e, eif4e, decapping, eif4a, eif4g, ha eif4e, dcp1a, dtdt, bodies, mrna decay
...	...	...	....
6748	For many centuries the treatment of calcaneal fractures has been non-operative and included bandagin...	A large variety of therapeutic modalities for calcaneal fractures have been described in the literat...	calcaneal, distraction, distraction techniques, level iv, calcaneus, sanders, orif, mva, cuboid, tre...
6749	Rice storage proteins are composed of glutelin (acid/alkaline-soluble), prolamin (alcohol-soluble), ...	The α and β subunits of soybean β-conglycinin were expressed in rice seeds in order to improve the n...	transgenic rice, rice seeds, seeds, rice, anti anti, immunoreactions, seeds 10, extractions, daf, ba...

Tabel 3. Dataset Validasi

ID	***Body/Article***	***Abstract***	***Keyword Article***
0	Various liver contrast agents for magnetic resonance imaging (MRI) have been developed to characteri...	This study was designed to compare the diagnostic performance of gadoxetic acid-enhanced magnetic re...	gadobenate dimeglumine, gadobenate, dimeglumine enhanced, enhanced mri, dimeglumine, hepatobiliary, ...
1	A key element in the regulation of cell cell and cell matrix contacts is the tyrosine phosphorylatio...	Most receptor-like protein tyrosine phosphatases (PTPases) display a high degree of homology with ce...	lar, plakoglobin, tpa, al 1993, ionophore, al 1995, proteolytic processing, ptpase, tpa induced, a23...
2	Preferential interactions of proteins with selected lipids are able to drive enrichment of the bilay...	Membrane proteins exhibit different affinities for different lipid species, and protein lipid select...	fret, acceptors, protein lipid, acceptor, fret efficiencies, energy transfer, shells, donor fluoresc...
...	...	...	...
2248	Breast cancer is the most commonly diagnosed cancer in women in North America and Europe, second onl...	PRDX1 was identified as a protein preferentially crosslinked to DNA in estrogen receptor negative bu...	mb 231, mda mb, mcf7 mda, mda, mcf7, 231, linked dna, resolved dimensional, dimensional gels, cispla...
2249	Minimally invasive options in urologic surgery are increasing. The addition of the da Vinci robot sy...	The role of the da Vinci robot is being defined in minimally invasive urologic surgery. Robot-assist...	da vinci, vinci, robot, prostatectomy, robot assisted, bladder neck, radical prostatectomy, vinci ro...

Tabel 4. Dataset Uji

ID	***Body/Article***	***Abstract***	***Keyword Article***
0	Although they do not appear to interact with each other in a yeast two-hybrid assay (The structure o...	DnaD and DnaB are essential DNA-replication-initiation proteins in low-G+C content Gram-positive bac...	hhpred, helix, turn helix, helix turn, unstructured, dna binding, cd, winged, unstructured region, n...
1	Biology is, at its core, the study of systems that evolve by natural selection. The complexity of bi...	Gene regulatory networks exhibit complex, hierarchical features such as global regulation and networ...	global regulators, gene regulatory, stress response, adaptively, global regulation, network dependen...
2	Optimal cell migration requires spatiotemporal feedback between actomyosin contraction, actin polyme...	Both tyrosine-phosphorylated caveolin-1 (pY14Cav1) and GlcNAc-transferase V (Mgat5) are linked with ...	cav1, fa, galectin, fak, mrfp, lattice, gal, α5 integrin, fas, fn
...	...	...	...
998	It is important to improve modern education with its increasing levels of academic achievement among...	Memory is more associated with the temporal cortex than other cortical areas. The two main component...	ges, yoga, mes, verbal memory, spatial memory, memory, increase memory, memory test, verbal, arts
999	New DNA sequencing methods are revolutionising biology, with impacts throughout the pure and applied...	The wide uptake of next-generation sequencing and other ultra-high throughput technologies by life s...	embl ebi, ebi, ena, data resources, embl, pride, elixir, data types, chebi, core data

3.2.2 Alat

Alat yang digunakan dalam penelitian ini yaitu sebagai berikut:

1. Sistem Operasi
   Windows 7 Ultimate 64-bit adalah salah satu versi dari sistem operasi *Windows* 7 yang dirilis oleh Microsoft.

2. *Notebook Kaggle* dengan *language python* 3.11.13, *accelerator* GPU P100
   Notebook Kaggle adalah platform berbasis web tempat pengguna dapat menulis, menjalankan, membagikan, dan berkolaborasi kode *python* dalam format *Jupyter Notebook*. Terdapat berbagai pilihan *Accelerator GPU* seperti GPU P100 yang memberikan akses ke GPU Tesla P100 untuk mempercepat komputasi terutama dalam pelatihan model *deep learning* dibandingkan menggunakan CPU biasa (Turck, 2023). Sedangkan *Python* 3.11.13 merupakan versi *python* yang digunakan dengan menawarkan fitur baru dan peningkatan performa.

3. *Library Pandas* 2.2.3
   Pandas merupakan *Library Python* yang dirancang untuk memudahkan dalam membersihkan, manipulasi, dan analisis data. Salah satu fitur utama pandas yaitu kemampuannya untuk menangani data yang hilang dan melakukan pemfilteran data dengan cepat sehingga pengguna dapat dengan mudah membersihkan, memanipulasi, dan menganalisis data tanpa perlu menulis banyak kode (Candra, 2025).

4. *Library Seaborn* 0.12.2 dan *Matplotlib.pyplot* 3.7.2
   Seaborn dan *Matplotlib* merupakan *Library Python* yang digunakan untuk menciptakan berbagai grafik dan visualisasi dari kumpulan data besar dengan cara yang mudah dan efektif. *Matplotlib* adalah salah satu *library* visualisasi data tertua dan paling populer yang memungkinkan pengguna untuk membuat grafik statis seperti grafik garis, batang, histogram, dan *scatter plot* dengan kontrol penuh terhadap elemen visual dengan menawarkan *flesibilitas* tinggi. *Seaborn* adalah pustaka yang dibangun diatas *Matplotlib* dengan tampilan yang lebih mudah digunakan dan visual yang lebih menarik. *Seaborn* mempermudah pembuatan grafik yang lebih kompleks, seperti distribusi data, hubungan antar variabel, dan pengelompokan data. Kelebihan *Seaborn* yaitu dapat secara otomatis mengatur tampilan grafik agar lebih informatif dan estetik dengan kode yang lebih sedikit dibandingkan *Matplotlib* (Mulyono & Saleh, 2025).

5. *Library re (Regular Expression)* dan *BeautifulSoup* 4.13.4
   Regular Expression merupakan suatu teknik yang digunakan untuk mencocokkan suatu string dengan pola tertentu dalam suatu pencarian. *Regular Expression* memungkinkan sistem untuk mengenali berbagai variasi input, termasuk salah eja, penggunaan karakter khusus, dan format yang tidak konsisten (Meiliyani *et al*., 2025). Sedangkan *BeautifulSoup* merupakan salah satu *library python* yang menyediakan API yang mudah digunakan untuk mengekstrak informasi dari halaman web dengan menggunakan bahasa pemrograman Python. *BeautifulSoup*

menyediakan beberapa fitur yaitu parsing dokumen HTML, navigasi dokumen, seleksi elemen, manipulasi elemen, dan ekstraksi data (Purnomo, 2022). Sehingga kombinasi penggunaan *Regular Expression* dan *BeautifulSoup* memungkinkan pencarian dan ekstraksi data yang lebih fleksibel dan terstruktur dari halaman web atau dokumen semi terstruktur.

6. *Library Transformers* 4.57.1
   Transformer merupakan *library python* yang dikembangkan oleh *Hugging Face* untuk mempermudah akses dalam berbagai model pra-latih berbasis arsitektur transformer seperti BERT, GPT, T5, BART, dan lainnya yang bisa digunakan dalam berbagai tugas NLP (*Natural Language Processing*) seperti peringkasan teks, klasifikasi teks, penerjemahan, dan pembuatan teks (Hoffman, 2024).

7. *Library Datasets* 4.4.2
   Datasets merupakan *library python* yang memudahkan dalam proses pemuatan data dengan dukungan *one-liner data loaders*. Selain itu, *library datasets* memiliki beberapa fitur seperti sistem *memory mapping* berbasis Apache Arrow yang mampu menangani dataset berukuran besar tanpa terkendala RAM, *smart caching* yang mencegah pemrosesan data berulang, memiliki API yang ringan dan cepat bergaya python, serta mendukung interoperabilitas (Khanna, 2021). Sehingga menjadikan *library* ini efisien dan praktis digunakan dalam pengelolan dan analisis data dalam berbagai tugas NLP.

8. *Library Accelerate* 1.11.0
   Accelerate adalah *library* dari Hugging Face yang mampu menyederhanakan kode *PyTorch* dari satu GPU hingga dapat dijalankan di beberapa GPU sehingga mengatasi kompleksitas pada model yang besar. *Accelerate* memungkinkan pengguna yang ingin menulis kode *PyTorch* umum tetap mampu menjalankannya dengan mengurangi beban kerja saat kode dijalankan secara *distributed* (tersebar di beberapa perangkat) (Ball, 2024). Berbeda dengan *PyTorch distributed launch* tradisional yang memerlukan perubahan konfigurasi setiap kali berpindah dari mode satu GPU ke multi-GPU dan sebaliknya.

9. *Library Optuna* 4.6.0
   Optuna adalah *library python* yang digunakan untuk optimisasi *hyperparameter* secara otomatis dan efisien. *Optuna* menyediakan beberapa fitur yang berguna dalam pembelajaran mesin seperti *Smart Samplers* yang membantu proses pencarian *hyperparameter* optimal dengan memilih nilai-nilai yang menjanjikan berdasarkan hasil sebelumnya, *Dynamic Pruning* mampu menghentikan

percobaan yang tidak menjanjikan lebih awal sehingga menghemat sumber daya komputasi, Integrasi mudah, Skalabilitas mendukung komputasi terdistribusi melalui integrasi memungkinkan optimisasi berskala besar, dan *optuna-daskboar* sehingga dapat memeriksa riwayat optimisasi, pentingnya *hyperparameter*, serta hasilnya dalam bentuk grafik dan tabel (Shah, 2024).

10. *Library rouge score* 0.1.2
    Rouge Score adalah *library python* yang digunakan untuk menghitung metrik ROUGE (*Recall Oriented Understudy for Gisting Evaluation*). Fungsinya yaitu mengukur seberapa mirip ringkasan yang dihasilkan komputer terhadap ringkasan referensi buatan manusia menggunakan penilaian berdasarkan *unigram, bigram*, dan *longest common subsequence* (LCS) (Mamdouh, 2023).

11. *Library Numpy* 1.26.4
    Numpy atau *Numerical Python* merupakan *Library Python* yang dirancang untuk memudahkan dalam melakukan operasi numerik yang efisien. Salah satu fitur utama *Numpy* yaitu menyediakan dukungan untuk *array* multidimensi dan berbagai fungsi matematika dan statistika. *Array numpy* dapat menyimpan data dalam bentuk dua dimensi (matriks) atau lebih (Candra, 2025).

12. *Library tqdm* 4.67.1
    Tqdm adalah *library python* yang digunakan untuk menampilkan *progress bar* pada loop dan tugas sehingga memudahkan dalam memantau proses eksekusi secara lebih efisien dan fleksibel seperti menampilkan bilah kemajuan di konsol atau terminal, termasuk statistik seperti jumlah iterasi, waktu yang berlalu, dan perkiraan waktu penyelesaian. Keunggulan *tqdm* tidak hanya pada kemudahan penggunaannya, tetapi juga konsistensinya dalam mendapatkan format output yang informatif (Hynkova, 2025).

13. *Library bert score* 0.3.13
    bert score adalah *library python* yang dirancang untuk menghitung metrik BERTScore. Fungsi utamanya adalah menggunakan *contextual embeddings* untuk menghitung kemiripan semantik antara teks ringkasan target dan teks ringkasan referensi, menggunakan *cossine similarity* antara token embedding ringkasan target dan ringkasan referensi dan mencocokkan token-token untuk menghitung *precision, recall,* dan f1-*score*, menyediakan opsi untuk memakai bobot IDF (*invers document frequency*) agar token yang jarang mendapat bobot lebih besar sehingga evaluasi jadi lebih sensitif terhadap kata penting, serta mampu merescale hasil sehingga lebih mudah dibaca manusia. Sehingga *library bert score* sesuai

untuk digunakan dalam mengevaluasi tugas NLP seperti *summarization* dan *text generation* (Sojasingaranyar, 2024).

3.3 Metode Penelitian

Terdapat sepuluh tahapan dalam melakukan peringkasan teks artikel biomedis dengan model *fine-tuning* BioBART dan diilustrasikan pada Gambar 12.

1. Tahap pertama, melakukan *input datasets* PMC (*PubMed Central*) pada *Notebook Kaggle*. Dataset terdiri dari 649 MB dengan 78.195 data, dan diambil sebanyak 10.000 data untuk kebutuhan penelitian ini.

2. Tahap kedua, melakukan *preprocessing* dengan dilakukan beberapa proses. Pertama, melakukan *cleaning datasets* dengan melakukan penghapusan tag HTML, normalisasi tanda kutip dan simbol umum, pembersihan simbol aneh yang tidak di butuhkan dalam artikel ilmiah, reduksi *noise* atau proses menghilangkan spasi berlebih, memperbaiki spasi di sekitar tanda kurung, menggabung *newline* menjadi spasi sehingga mempermudah model untuk memahami konteks kalimat, dan melakukan *Keyword Extraction* dengan perhitungan TF-IDF *keyword* yang disertai dengan *word tokenization* untuk memperoleh *keyword* setiap artikelnya sebagai penunjang evaluasi tambahan.

3. Tahap ketiga, melakukan *splitting* data yang terdiri 10% data pengujian dan 90% data model. Data model kemudian dibagi menjadi dua yaitu 75% data training dan 25% data validasi.

4. Tahap keempat, tokenisasi menggunakan model BioBART pada ketiga data. Tokenisasi dilakukan dengan mengubah teks artikel dan abstrak menjadi token ID (angka) menggunakan *SentencePiece tokenizer* bawaan model BioBART.

5. Tahap kelima, *sliding windows* yang diterapkan terhadap input token hasil tokenisasi *article* menggunakan *Fixed Sliding Windows* dengan membagi artikel yang panjang menjadi beberapa windows berdasarkan token. Proses menggunakan max input length sebesar 1024 token, dan overlap antar window sebesar 824 token. Setiap windows dipasangkan dengan abstrak artikel asal sebagai target sehingga satu artikel menjadi beberapa pasangan *Article Windows ke n* dan *Abstract*.

6. Tahap keenam, melakukan pengaturan konfigurasi pelatihan pada data *training*. Pada tahap ini dilakukan penyesuaian model dan *hyperparameter* yang digunakan dalam pelatihan. Model yang dipakai yaitu *GanjinZero/biobart-v2-base*. Pengaturan *hyperparameter* meliputi penentuan ukuran *batch size* sebesar 32 dengan pengaturan *Per Device Train Batch Size* dan *Gradient Accumulation Steps*, *num train epoch* 10 dengan *early stopping*, *dropout* sebesar 0,1 hingga 0,2, serta kombinasi *learning rate* antara 1×10^{-6} hingga 5×10^{-5}. Selain itu, tahap ini juga mencakup pengaturan *optimizer* AdamW dengan *weight decay* sebesar 0,01, 0,001, dan 0,05. Selain model BioBART yang akan melalui proses *fine-tuning*, penelitian ini juga menggunakan model BioBART *pretrained* sebagai *baseline*. Model *baseline* digunakan tanpa pelatihan lanjutan dan tidak melalui tahap *hyperparameter* dan digunakan sebagai keperluan perbandingan kinerja pada tahap evaluasi.

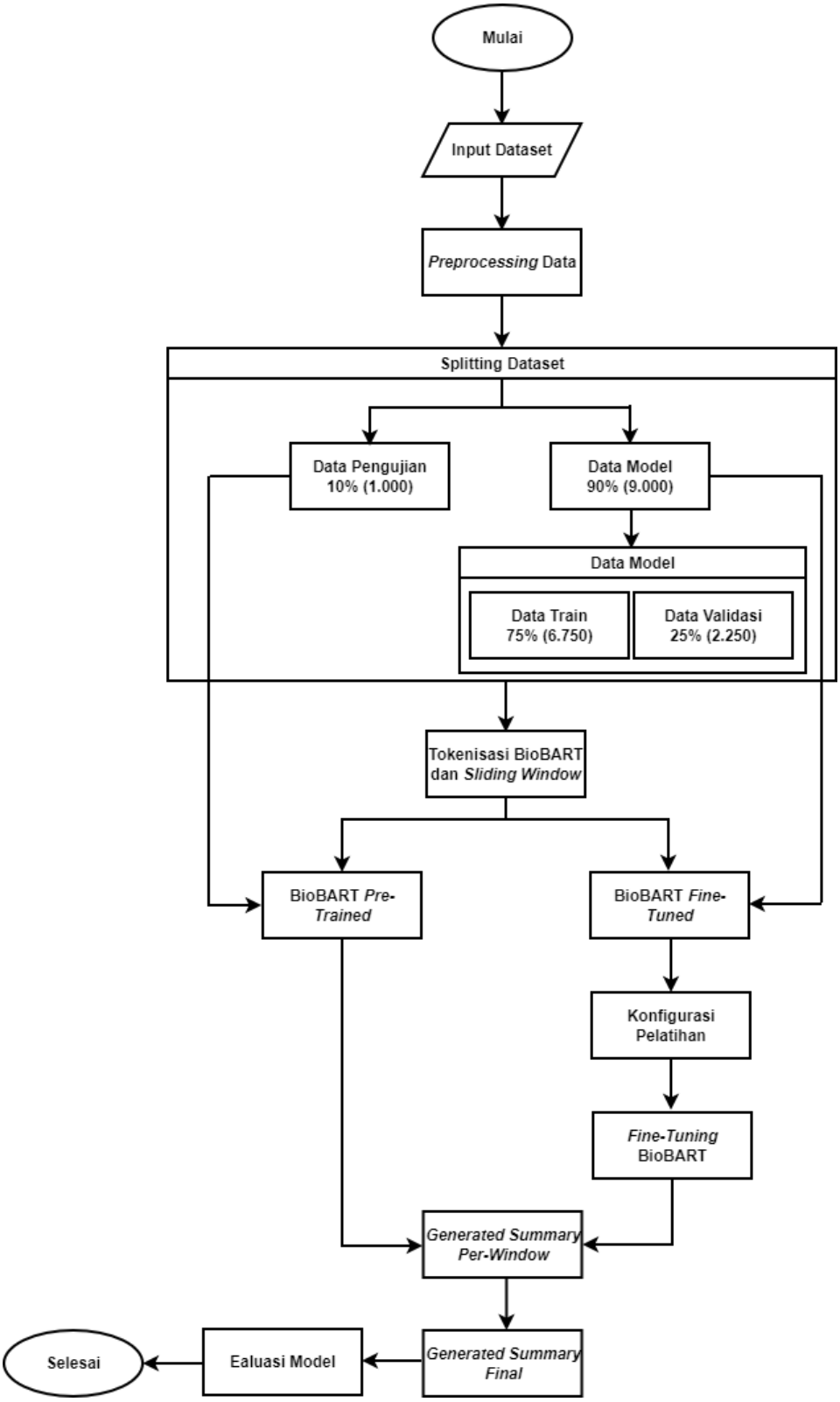
7. Tahap ketujuh, melakukan *fine-tuning* model menggunakan data model. Pada proses ini ditampilkan nilai *train loss*, *vall loss*, serta waktu yang diperlukan selama proses *fine-tuning*. Nilai *train loss* dan *vall loss* disajikan dalam bentuk grafik garis untuk memperlihatkan perkembangan selama pelatihan.

8. Tahap kedelapan, setelah model terbaik di dapatkan dengan proses *fine-tuning*, langkah selanjutnya adalah melakukan *generated summary*. *Generated summary* dilakukan selama dua tahap. Tahap pertama, dilakukan *generated summary* berdasarkan windows pada tiap artikel.

9. Tahap kesembilan, setelah *generated summary* per windows di dapatkan, kemudian hasilnya di satukan berdasarkan asal artikel dari tiap windows. Selanjutnya *generated final* dilakukan dari hasil penggabungan *generated summary* per windows dan menghasilkan satu *summary prediksi final* per artikel.

10. Tahap kesepuluh, melakukan evaluasi model dari hasil *generated summary final* dengan perhitungan nilai ROUGE*Score* dan BERT*Score* dengan tiga metrik *recall, precision,* dan f1-*score*, serta melakukan *keyword extraction* sebagai evaluasi tambahan terhadap hasil ringkasan. Selain evaluasi deskriptif, dilakukan pula uji signifikansi statistik menggunakan *paired t-test* untuk membandingkan kinerja model BioBART *pretrained* dan BioBART *fine-tuned* berdasarkan skor evaluasi per artikel pada dataset uji.

11. Tahap kesebelas, membuat kesimpulan dari hasil yang telah didapat dari penelitian dan memberi saran untuk penelitian selanjutnya.



Gambar 12. Diagram Alur Penelitian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini bertujuan untuk mengoptimalkan *fine-tuning* model BioBART menggunakan AdamW *Optimizer* pada peringkasan teks otomatis artikel ilmiah biomedis menggunakan dataset PubMed *Central* (PMC) dengan evaluasi menggunakan metrik ROUGE dan BERTScore. Berdasarkan hasil eksperimen dan analisis yang telah dilakukan, diperoleh beberapa poin penting yang dapat disimpulkan yaitu:

1. *Fine-tuning* model BioBART menggunakan AdamW *Optimizer* mampu meningkatkan kinerja peringkasan secara signifikan.
   Peningkatan yang konsisten oleh model BioBART *fine-tuned* dibandingkan model BioBART *pretrained* pada seluruh metrik evaluasi menunjukkan bahwa AdamW efektif dalam mengoptimalkan parameter model agar lebih sesuai dengan karakteristik artikel biomedis pada dataset PMC.

2. Kualitas ringkasan secara leksikal tetap terjaga dengan hasil nilai ROUGE yang kompetitif.
   Nilai ROUGE-1, ROUGE-2, maupun ROUGE-L pada *f1-score* menggambarkan bahwa model mampu mempertahankan kata, hubungan antar kata, dan struktur urutan kalimat antara ringkasan referensi dan ringkasan yang dihasilkan model meskipun dihadapkan pada dokumen yang panjang dan lebih kompleks.

3. Keunggulan utama penelitian ini terletak pada kualitas semantik ringkasan.
   Dengan nilai BERTScore pada *f1-score* sebesar 0,8628 telah menunjukkan bahwa ringkasan yang dihasilkan model memiliki kesesuaian makna yang tinggi, dengan mampu mempertahankan konteks ilmiah pada artikel, serta menghasilkan parafrase yang relevan secara semantik.

4. Peningkatan kinerja model terbukti signifikan secara statistik.
   Berdasarkan hasil uji *paired t-test* dengan p-value $< 0,001$ pada seluruh metrik evaluasi, menunjukkan bahwa peningkatan kinerja model BioBART *fine-tuned* terhadap model BioBART *pretrained* signifikan secara statistik, sehingga peningkatan yang dihasilkan tidak hanya bersifat numerik namun juga bermakna.

5. Optimalisasi *fine-tuning* mampu menyeimbangkan antara kualitas leksikal dan semantik.
   Dengan kombinasi nilai ROUGE yang kompetitif dan BERTScore yang tinggi menunjukkan bahwa pendekatan *fine-tuning* yang diimplementasikan berhasil mengelola *trade-off* antara ketepatan leksikal dan pemahaman makna secara efektif.

5.2 Saran

Penelitian selanjutnya diharapkan untuk memperluas penggunaan dataset yang tidak hanya terbatas pada dataset PubMed *Central*, PLOS, maupun eLife, tetapi melibatkan korpus biomedis lain dengan karakteristik artikel yang lebih teknis dan kompleks. Misalnya korpus yang bersumber dari publikasi BioStatistics, *clinical trial* atau artikel metodologis seperti jurnal *statistics in medicine* dapat digunakan sebagai representasi dokumen dengan lebih kepadatan informasi statistik serta kompleksitas terminologi yang tinggi. Hal ini penting agar model memiliki kemampuan generalisasi yang lebih baik meskipun dihadapkan dengan struktur dokumen yang lebih teknis dan kompleks. Selain itu, penggunaan dataset yang lebih besar dan bervariasi membantu meningkatkan kemampuan model dalam memahami struktur kalimat dan konteks yang lebih luas sehingga ringkasan yang dihasilkan menjadi lebih informatif.

Penelitian selanjutnya juga disarankan untuk lebih mengeksplorasi *hyperparameter* secara lebih sistematis, seperti *learning rate*, *weight decay*, *batch size*, dan *warmup ratio* dengan menggunakan pendekatan *grid search* atau *bayesian optimization* memungkinkan untuk menghasilkan konfigurasi yang lebih optimal, khususnya pada dataset dengan tingkat kompleksitas tinggi seperti PMC.

Pada sisi model, penelitian selanjutnya untuk mengeksplorasi arsitektur transoformer seperti Long-T5, LED, atau BigBird yang secara khusus dirancang untuk menangani dokumen panjang dan membandingkan dengan pendekatan *sliding window* dan *two-stage summarization* yang digunakan pada penelitian ini. Perbandingan tersebut diharapkan mampu memberikan gambaran mengenai *trade-off* antara efisiensi komputasi dan kualitas ringkasan yang dihasilkan.

Pada sisi evaluasi, penelitian selanjutnya disarankan untuk tidak hanya mengandalkan metrik otomatis seperti ROUGE dan BERTScore, tetapi juga melibatkan evaluasi manusia (*human evaluation*) untuk menilai aspek keterbacaan dan kesesuaian ringkasan bagi pembaca non-ahli. Pendekatan ini penting karena mampu memastikan bahwa peningkatan nilai metrik semantik juga sejalan dengan kualitas ringkasan secara praktis.

DAFTAR PUSTAKA

Akil, I. 2023. Komparasi Fungsi Aktivasi Neural Network Pada Data Time Series. *Inti Nusa Mandiri*. **18**(1): 7-14. DOI : https://doi.org/10.33480/inti.v18i1.4288.

Aulia, R., Ruslani, A., & Fauzan, M. 2025. Pengembangan Model Transformer untuk Analisis Sentimen dan Ringkasan Ulasan E-commerce Berbahasa Indonesia Abstrak. *SNIV:Seminar Nasional Inovasi Vokasi*. **4**(1): 1213-1224. E-ISSN 2830-0343.

Ball, N. 2024. *Multi-GPU on raw PyTorch with Hugging Face's Accelerate library*. DigitalOcean – Multi-GPU on PyTorch tutorial. https://www.digitalocean.com/community/tutorials/multi-gpu-on-raw-pytorch-with-hugging-faces-accelerate-library. Diakses pada 30 Oktober 2025.

Candra, A.P. 2025. Analisis Data Menggunakan Python: Memperkenalkan Pandas dan NumPy. *Journal of Information System and Education Development*. **3**(1): 11-16. DOI : https://doi.org/10.62386/jised.v3i1.118.

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. 2021. Hyperparameter tuning for machine learning algorithms used for Arabic sentiment analysis. *Informatics*. **8**(4): 79. DOI : https://doi.org/10.3390/informatics8040079

Firdaus, A., & Firdaus, W., I. 2021. Text Mining Dan Pola Algoritma DalamPenyelesaianMasalah Informasi : (Sebuah Ulasan). *Jurnal Jupiter*. **13**(1): 66-78.

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*. **26**(3): 259-291. DOI : https://doi.org/10.1017/S1351324919000457.

Gao, Y., Hou, F., & Wang, R. 2024. A Novel Two-step Fine-tuning Framework for Transfer Learning in Low-Resource Neural Machine Translation. *Findings of the Association for Computational Linguistics: NAACL 2024*. pp. 3214-3224. DOI : https://doi.org/10.18653/v1/2024.findings-naacl.203.

Goodfellow, I., Bengio, Y., & Courville, A. 2016. Deep learning. Cambridge, MA: MIT Press.

Goyal, P., Hu, Q., & Gupta, R. (2023). *Faithful model evaluation for model-based metrics*. In H. Bouamor, J. Pino, & K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7484–7489). Association for Computational Linguistics.

Hadwiranto, M. R., Hamami, F., & Pratiwi, O. N. 2024. Extractive Text Summarization Terhadap Artikel Berita Indonesia Berbasis Machine Learning. e-Proceeding of Engineering. **11**(4): 3941-3947.

Hartawan, G., Sa'adillah Maylawati, D., & Uriawan, W. 2024. Bidirectional And Auto-Regressive Transformer (BART) For Indonesian Abstractive Text Summarization. *JIP (Jurnal Informatika Polinema)*. **10**(4): 535-542. DOI : https://doi.org/10.33795/jip.v10i4.5242.

Hermawan, A., Jowensen, I., Junaedi, J., & Edy, E. 2023. Implementasi Text-Mining untuk Analisis Sentimen pada Twitter dengan Algoritma Support Vector Machine. *JST (Jurnal Sains dan Teknologi)*. **12**(1): 129-137. DOI : https://doi.org/10.23887/jstundiksha.v12i1.52358.

Husniah, F., Agustian, S., & Afrianty, I. 2022. Peringkasan Teks Otomatis Artikel Berbahasa Indonesia Menggunakan Algoritma Textrank. *TEKNOKA National Seminar*. **6**(1): 2502-8782.

Hoffman, H. 2024. *Hugging Face Transformers: Leverage Open-Source AI in Python*. Real Python. https://realpython.com/huggingface-transformers/. Diakses pada 29 Oktober 2025.

Hynkova, K. 2025. *Ultimate guide to tqdm library in Python*. Deepnote. https://deepnote.com/blog/ultimate-guide-to-tqdm-library-in-python. Diakses pada 31 Oktober 2025.

Idhafi, Z., Agustian, S., Yanto, F., & Saafat, N. 2023. Peringkasan Teks Otomatis Pada Artikel Berbahasa Indonesia Menggunakan Metode Maksimum Marginal Relevance. *Jurnal Computer Science and Information Technology*. **4**(3): 609-618. DOI : https://doi.org/10.37859/coscitech.v4i3.6311.

Ilemobayo, J. A., Durodola, O., Alade, O., Awotunde, O. J., Olanrewaju, A. T., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., Ifeanyi, A., Odezuligbo, I. E., & Edu, O. E. 2024. Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*. **26**(6): 388-395. DOI : https://doi.org/10.9734/jerr/2024/v26i61188.

Jahan, I., Laskar, M., Peng, C., & Huang, J. 2023. Evaluation of ChatGPT on Biomedical Tasks: A Zero-Shot Comparison with Fine-Tuned Generative Transformers. Proceedings of the Annual Meeting of the Association for Computational Linguistics. **11**(3): 326-336. DOI : https://doi.org/10.48550/arXiv.2306.04504.

Karotia, A., & Susan, S. 2024. BioLay-AK-SS at BioLaySumm: Domain Adaptation by Two-Stage Fine-Tuning of Large Language Models used for Biomedical Lay Summary Generation. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. (pp. 762-768). DOI : https://doi.org/10.18653/v1/2024.bionlp-1.69.

Khanna, C. 2021. *Use the Datasets library of Hugging Face in your next NLP project*. Towards Data Science. https://towardsdatascience.com/use-the-datasets-library-of-hugging-face-in-your-next-nlp-project-94e300cca850. Diakses pada 30 Oktober 2025.

Kurniawan, K., Ceasaro, B., & Sucipto. 2024. Perbandingan Fungsi Aktivasi Untuk Meningkatkan Kinerja Model LSTM Dalam Prediksi Ketinggian Air Sungai. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*. **10**(1): 134-143. DOI : https://doi.org/10.26418/jp.v10i1.72866.

Lin, C.Y. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. pp. 74-81.

Mahajaya, N.S., Desiana, P., & Ayu, W. 2024. Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB terhadap Model Vision Transformer

pada Klasifikasi Penyakit Paru-paru. *Spinter 2024*. (pp. 818-823). https://spinter.stikom-bali.ac.id/index.php/spinter/article/view/222.

Mamdouh, M. 2023. *Mastering ROUGE Matrix: Your Guide to Large Language Model Evaluation for Summarization with Examples*. DEV Community. https://dev.to/aws-builders/mastering-rouge-matrix-your-guide-to-large-language-model-evaluation-for-summarization-with-examples-jjg. Diakses pada 30 Oktober 2025.

Meiliyani, K., Alfah, A., Awalia, N., & Setiadi,T. 2025. Penggunaan Ekspresi Regular dalam Meningkatkan Efisiensi PencarianProduk pada Aplikasi E-Commerce. *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*. **3**(1): 169-177.

Meng, L. K., Xin, L. J., Yi, H. H., Salam, Z. A. A., & Wei, N. B. 2023. A Machine Learning Approach for Face Mask Detection System with AdamW Optimizer. *Journal of Applied Technology and Innovation*. **7**(3): 25-29.

Mukherjee, A., Hassija, V., Chamola, V., & Gupta, K. K. 2025. A Detailed Comparative Analysis of Automatic Neural Metrics for Machine Translation: BLEURT & BERTScore. *IEEE Open Journal of the Computer Society*. **6**(1): 658-668. DOI : https://doi.org/10.1109/OJCS.2025.3560333.

Mulyono, S., & Saleh, Y.K.P. 2025. Python Untuk Data Science : Analisis Data, Visualisasi, Dan Pembelajaran Mesin. Jakarta : *PT.Media Penerbit Indonesia*.

Peyrard, M., Zhao, W., Eger, S., & West, R. (2021). *Better than average: Paired evaluation of NLP systems*. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. (pp. 2301–2315).

Phan, P., Tran, T., & Trieu, H. 2023. VBD-NLP at BioLaySumm Task 1: Explicit and Implicit Key Information Selection for Lay Summarization on Biomedical Long Documents. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*. pp. 574-578. DOI : https://doi.org/10.18653/v1/2023.bionlp-1.60.

Purnomo, A. 2022. Impementasi Web Scraping Pada OJS Dengan Metode CSS Selector. *RESOLUSI : Rekayasa Teknik Informatika dan Informasi*. **3**(2): 176-191.

Purwitasi, N. A., & Soleh, M. 2022. Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Processing. *Jurnal IPTEK*. **6**(1): 14-21. DOI : https://doi.org/10.31543/jii.v6i1.192.

Nurfiyah, N., & Ramadhandi, N. 2023. Penerapan Metode Natural Language Processing (NLP) Pada Question Answering System Untuk Media Informasi Mahasiswa Universitas Bhayangkara Jakarta Raya. *Journal of Information and Information Security (JIFORTY)*. **4**(2): 175-186. DOI : https://doi.org/10.31599/a24t9m52.

Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, & Zaqiah, Q. Y. 2022. Deep Learning dan Penerapannya dalam Pembelajaran. *JIIP (Jurnal Ilmiah Ilmu Pendidikan)*. **5**(9): 3258-3267.

Saputra, F., Andre, A., & Harefa, K. 2025. Penerapan Metode Natural Language Processing (Nlp) Dalam Implementasi Asisten Virtual Chatbot Dengan Memanfaatkan Api Chatgpt Dan Gradio App. *JORAPI : Journal of Research and Publication Innovation*. **3**(1): 1-15.

Setiaji, B., & Pramudho, P.K. 2022. Pemanfaatan Teknologi Informasi Berbasis Data Dan Jurnal. *Jurnal Inovasi Riset Ilmu Kesehatan*. **1**(3): 166-175. DOI : https://doi.org/10.51878/healthy.v1i3.1649.

Shah, M.D. 2024. *Master Hyperparameter Optimization with Optuna: A Complete Guide [Part 1]*. https://medium.com/@mdshah930/master-hyperparameter-optimization-with-optuna-a-complete-guide-89971b799b0a. Diakses pada 30 Oktober 2025.

Shaveta. 2023. A review on machine learning. *International Journal of Science and Research Archive*. **9**(1): 281-285.

Sojasingaranyar, A. 2024. *BERTScore Explained in 5 minutes*. Medium. https://medium.com/@abonia/bertscore-explained-in-5-minutes-0b98553bfb71. Diakses pada 31 Oktober 2025.

Turck, D. 2023. *How to install Python and enable GPU acceleration*. XDA Developers.

https://www.xda-developers.com/how-install-python-enable-gpu-acceleration/. Diakses pada 31 Oktober 2025.

Toraman, Ç., Yılmaz, E. H., Şahinuç, F., & Özcelik, O. 2023. Impact of Tokenization on Language Models: An Analysis for Turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*. **22**(4): 116. DOI : https://doi.org/10.1145/3578707

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. 2017. Attention Is All You Need. *In Advances in Neural Information Processing Systems*. **30**(1): 5998-6008.

Wibawa, M. S. 2016. Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan. *Jurnal Sistem dan Informatika* **11**(1): 1-8.

Widiantoro, A. D., & Sanjaya, M. R. 2024. *Pengantar Nlp Dan Topik Model Lda*. Palembang : Asosiasi Doktor Sistem Informasi Indonesia. ISBN N: 978-623-10-4853-0.

Yang, Z., Huang, Y., & Zhang, Y. J. 2020. TS-CSW: Text steganalysis and hidden capacity estimation based on convolutional sliding windows. *Multimedia Tools and Applications*. **79**(25): 18293-18316. DOI : https://doi.org/10.1007/s11042-020-08716-w.

Yuan, H., Yuan, Z., Gan, R., Zhang, J., Xie, Y., & Yu, S. (2022). BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model. Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 97-109. DOI : https://doi.org/10.48550/arXiv.2204.03905.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. *In Proceedings of the International Conference on Learning Representations (ICLR)*. DOI : https://doi.org/10.48550/arXiv.1904.09675.

Zhu, Y. (2020). *On the statistical significance testing for natural language processing (Master's thesis)*. University of Washington.