

**OPTIMALISASI *FINE-TUNING* MODEL T5 MENGGUNAKAN ADAMW
UNTUK PERINGKASAN TEKS BERITA BAHASA INDONESIA
DENGAN EVALUASI ROUGE DAN *COSINE SIMILARITY***

Skripsi

Oleh

**NI MADE RHEA ADNYANI
NPM. 2117031030**



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

2025

ABSTRACT

OPTIMIZING FINE-TUNING OF THE T5 MODEL USING ADAMW FOR INDONESIAN NEWS TEXT SUMMARIZATION: AN EVALUATION WITH ROUGE AND COSINE SIMILARITY

By

Ni Made Rhea Adnyani

According to Statista (2022), about 68% of Indonesian society actively accesses online news every day. The large amount of digital information makes it hard for readers to find important content, especially because news texts are often long and complex. Automatic text summarization is an important solution to present information in a short, fast, and meaningful way. This study improves the T5 model using the AdamW optimizer through a fine-tuning process to enhance the quality of Indonesian news summaries. The dataset used is IndoSum with 9387 entries. The data is split into 90% for model building and 10% for testing. The model data is further divided into 80% training and 10% validation. The research process includes preprocessing steps such as feature selection, token reconstruction, text cleaning, and lowercasing. Then, the embedding and fine-tuning of the T5 model are done by adjusting batch size, learning rate, weight decay, and number of epochs to get the best performance. The model is evaluated using ROUGE and cosine similarity metrics. The results show a clear improvement compared to past studies. The ROUGE and cosine similarity scores show that the model's summaries are relevant to the references, both in structure and meaning.

Keywords: IndoSum, T5 Model, AdamW Optimizer, Fine-Tuning, ROUGE, Cosine Similarity, Automatic Text Summarization.

ABSTRAK

OPTIMALISASI *FINE-TUNING* MODEL T5 MENGGUNAKAN ADAMW UNTUK PERINGKASAN TEKS BERITA BAHASA INDONESIA DENGAN EVALUASI ROUGE DAN *COSINE SIMILARITY*

Oleh

Ni Made Rhea Adnyani

Menurut data Statista (2022), sekitar 68% dari total masyarakat Indonesia secara aktif mengakses berita daring setiap hari. Banyaknya informasi yang tersedia secara digital menyebabkan pembaca kesulitan dalam menyaring informasi yang relevan, terutama karena teks berita cenderung panjang dan kompleks. Peringkasan teks otomatis menjadi solusi penting untuk menyajikan informasi secara ringkas, cepat, dan tetap bermakna. Penelitian ini mengoptimalkan model T5 dengan AdamW *optimizer* melalui proses *fine-tuning* untuk meningkatkan kualitas ringkasan teks berita berbahasa Indonesia. Dataset yang digunakan adalah *IndoSum* sebanyak 9387 data. Data dibagi menjadi 90% untuk model dan 10% untuk pengujian. Data model terdiri dari 80% data *training* dan 10% data *validation*. Proses penelitian mencakup tahap *preprocessing* seperti seleksi fitur, rekonstruksi *token*, pembersihan teks, dan *lowercasing*. Setelah itu, dilakukan proses *embedding* dan *fine-tuning* model T5 dengan pengaturan *batch size*, *learning rate*, *weight decay*, dan jumlah *epoch* yang disesuaikan untuk memperoleh kinerja terbaik. Evaluasi model dilakukan dengan menggunakan metrik ROUGE dan *cosine similarity*. Hasil penelitian menunjukkan adanya peningkatan signifikan dibandingkan kajian terdahulu. Nilai metrik ROUGE dan *cosine similarity* yang diperoleh menunjukkan bahwa ringkasan yang dihasilkan model cukup relevan dengan referensi, baik dari sisi struktur maupun makna.

Kata-kata kunci: *IndoSum*, Model T5, AdamW *Optimizer*, *Fine-Tuning*, ROUGE, *Cosine Similarity*, Peringkasan Teks Otomatis.

**OPTIMALISASI *FINE-TUNING* MODEL T5 MENGGUNAKAN ADAMW
UNTUK PERINGKASAN TEKS BERITA BAHASA INDONESIA
DENGAN EVALUASI ROUGE DAN *COSINE SIMILARITY***

NI MADE RHEA ADNYANI

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam



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

2025

Judul Skripsi : **OPTIMALISASI *FINE-TUNING* MODEL T5 MENGGUNAKAN ADAMW UNTUK PERINGKASAN TEKS BERITA BAHASA INDONESIA DENGAN EVALUASI ROUGE DAN *COSINE SIMILARITY***

Nama Mahasiswa : **Ni Made Rhea Adnyani**

Nomor Pokok Mahasiswa : **2117031030**

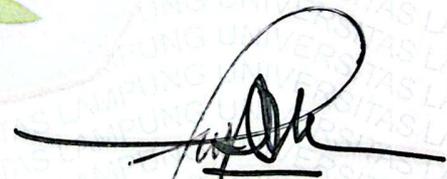
Program Studi : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing


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


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

2. Ketua Jurusan Matematika


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

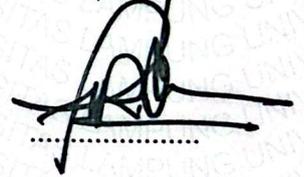
MENGESAHKAN

1. Tim penguji

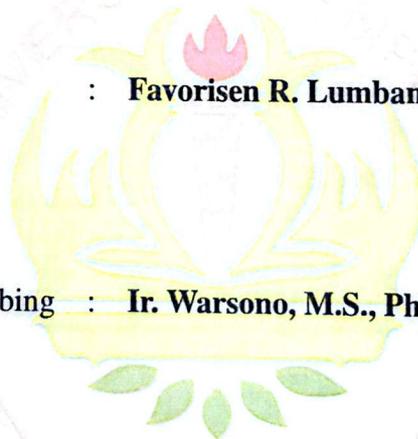
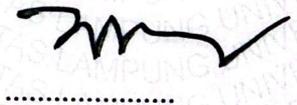
Ketua : Dr. Dian Kurniasari, S.Si., M.Sc.



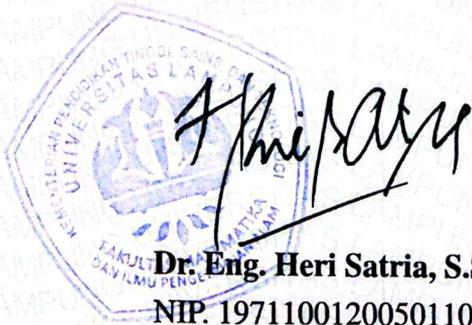
Sekretaris : Favorisen R. Lumbanraja, M.Si., Ph.D



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



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



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

Tanggal Lulus Ujian Skripsi: 13 Juni 2025

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : Ni Made Rhea Adnyani
Nomor Pokok Mahasiswa : 2117031030
Jurusan : Matematika
Judul Skripsi : **Optimalisasi *Fine-tuning* Model T5 Menggunakan ADAMW untuk Peringkasan Teks Berita Bahasa Indonesia dengan Evaluasi ROUGE dan *Cosine Similarity***

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

Bandar Lampung, 13 Juni 2025

Penulis.


METERAI
TEMPEL
SCAMX332255965
Ni Made Rhea Adnyani

RIWAYAT HIDUP

Penulis memiliki nama lengkap Ni Made Rhea Adnyani yang lahir di Lampung Tengah pada tanggal 23 April 2003. Penulis anak kedua dari dua bersaudara dari pasangan Bapak I Wayan Supandia dan Ibu Ni Made Sudiartini.

Penulis memulai pendidikan di Taman Kanak-kanak (TK) PT. GPM 1/9 pada tahun 2007-2009, lalu melanjutkan Sekolah Dasar di SDS 02 GPM dari tahun 2009-2015. Kemudian, penulis menempuh pendidikan Sekolah Menengah Pertama di SMP GPM pada tahun 2015-2018, dan melanjutkan ke jenjang Menengah Atas di SMAS *Sugar Group* pada tahun 2018-2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN).

Selama menjalani pendidikan di Universitas Lampung, penulis aktif dalam organisasi. Pada tahun 2022 sampai 2023, penulis menjadi bagian dari UKM HINDU UNILA. Pada Desember 2023 hingga Februari 2024, penulis melaksanakan Kerja Praktik (KP) di Dinas Perhubungan Kota Bandar Lampung dan pada Juni sampai Agustus 2024, sebagai salah satu bentuk pengabdian kepada masyarakat penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Labuhan Ratu 3, Kecamatan Labuhan Ratu, Kabupaten Lampung Timur.

KATA INSPIRASI

”Jangan takut melangkah. Tuhan bersamamu di setiap langkahnya.”
(Bhagavad Gita 18.66)

”Kamu lebih kuat dari rasa takutmu.”
(Bhagavad Gita 6.5)

”Lakukan yang terbaik, tanpa takut hasilnya.”
(Bhagavad Gita 2.14)

”Hadapi semuanya langsung di muka, apapun yang terjadi tidak apa, setiap hari ku bersyukur melihatmu berselimut harapan, berbekal cerita”
(Baskara Putra)

”Hidup bukan saling mendahului, bermimpilah sendiri-sendiri”
(Baskara Putra)

”Tugas kita bukanlah untuk berhasil, tugas kita adalah untuk mencoba karena di dalam mencoba itulah kita menemukan kesempatan untuk berhasil”
(Buya Hamka)

”Lakukan apa yang kau mau, sekarang saat hatimu bergerak, jangan kau larang, hidup ini tak ada artinya, maka kau bebas mengarang maknanya seorang!”
(Baskara Putra)

”Kegagalan bukan menjadi akhir untuk berputus asa, tetapi menjadi pembelajaran menuju keberhasilan”
(Ni Made Rhea Adnyani)

PERSEMBAHAN

Puji syukur kepada Tuhan Yang Maha Esa atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, penulis persembahkan rasa terima kasih kepada:

Ayah, Ibu dan Kakak

Terima kasih kepada orang tua tercinta atas segala doa, dukungan, dan pelajaran hidup yang tak ternilai. Ucapan terima kasih yang sama disampaikan kepada kakak tersayang atas semangat, perhatian, dan teladan yang selalu menjadi penguat dalam setiap langkah perjalanan yang penulis hadapi.

Dosen Pembimbing dan Pembahas

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

Sahabat-sahabatku

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasinya, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

Almamater Tercinta

Universitas Lampung

SANWACANA

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Optimalisasi *Fine-Tuning* Model T5 Menggunakan AdamW untuk Peringkasan Teks Berita Bahasa Indonesia dengan Evaluasi ROUGE dan *Cosine Similarity*" dengan baik, lancar, dan tepat waktu.

Pada proses penyusunan skripsi ini, penulis banyak menerima bantuan, arahan, dukungan moral, maupun semangat dari berbagai pihak. Oleh karena itu, dengan segala hormat dan rasa terima kasih yang mendalam, penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc., selaku Pembimbing I, yang telah banyak meluangkan waktu untuk memberikan arahan, bimbingan, motivasi, dan dukungan kepada penulis sejak awal hingga skripsi ini terselesaikan.
2. Bapak Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D., selaku Pembimbing II, yang juga memberikan banyak arahan dan masukan berharga selama proses bimbingan.
3. Bapak Ir. Warsono, M.S., Ph.D., selaku Penguji, atas kritik, saran, dan evaluasi yang membangun sehingga penyusunan skripsi ini menjadi lebih baik.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si., selaku Dosen Pembimbing Akademik dan Ketua Jurusan Matematika FMIPA Universitas Lampung, atas bimbingan dan arahnya selama masa studi.
5. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Seluruh dosen dan staf Jurusan Matematika FMIPA Universitas Lampung yang telah memberikan ilmu, bantuan, dan pelayanan terbaik selama perkuliahan.
7. Orang tua tercinta, Bapak I Wayan Supandia dan Ibu Ni Made Sudiartini, yang telah menjadi sumber kekuatan melalui kasih sayang tulus, doa yang tiada henti, serta perhatian dan dukungan yang selalu hadir. Segala keterbatasan tidak pernah menjadi halangan dalam memberikan yang terbaik, hingga penulis menyelesaikan studi dan meraih gelar sarjana.

8. Kakak tersayang, I Wayan Eska Adnyana, atas segala dukungan, nasihat, dan perhatian yang selalu memberi semangat dan menjadi teladan bagi penulis.
9. Dewi, Lola, Salma, dan Shelvi. Terima kasih atas kebersamaan yang penuh warna, tawa, dan semangat yang selalu menguatkan selama perjalanan kuliah ini. Semoga langkah kita selalu menuju masa depan yang cerah.
10. Teman-teman sepembimbingan, Nabila, Lusi, Yulina, Anastasia, Dina, Fathan, Adin, Erwin, Anggy, Mey, Maya, Andi, Dita, Sherina dan Ariz atas kebersamaan, diskusi, dan semangat yang saling menguatkan selama proses menyelesaikan skripsi.
11. Seluruh teman Jurusan Matematika FMIPA UNILA angkatan 2021. Terima kasih atas persahabatan, dukungan, dan kebersamaan yang telah terjalin selama masa kuliah. Semoga silaturahmi dan ilmu yang kita raih dapat bermanfaat di masa depan.
12. Kepada sahabat terdekat, Madin, yang selalu menjadi tempat berbagi suka dan duka, tempat curhat, dan sumber semangat saat penulis merasa lelah. Terima kasih atas kehadirannya yang begitu berarti dalam perjalanan ini.
13. Seluruh pihak terkait yang telah membantu menyelesaikan skripsi ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan, sehingga masukan dan saran yang membangun sangat diharapkan untuk penyempurnaan ke depan. Semoga skripsi ini dapat memberikan manfaat bagi pembaca maupun bagi pengembangan ilmu pengetahuan ke depannya.

Bandar Lampung, 13 Juni 2025

Ni Made Rhea Adnyani

DAFTAR ISI

Halaman

DAFTAR ISI	xiii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
I PENDAHULUAN	1
1.1 Latar Belakang dan Masalah	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
II TINJAUAN PUSTAKA	4
2.1 Penelitian Terkait	4
2.1.1 Penelitian Pertama (Itsnaini dkk., 2022)	5
2.1.2 Penelitian Kedua (Purnama dan Utami, 2023)	6
2.1.3 Penelitian Ketiga (Bahari dan Dewi, 2024)	6
2.2 <i>Natural Language Processing</i> (NLP)	7
2.3 <i>Text Mining</i>	8
2.4 Tokenisasi	9
2.5 <i>Word Embedding</i>	10
2.6 Peringkasan Teks Otomatis	10
2.7 <i>Machine Learning</i> (ML)	11
2.8 <i>Deep Learning</i>	13
2.9 Fungsi Aktivasi	14
2.9.1 Fungsi Aktivasi Sigmoid	14
2.9.2 Fungsi Aktivasi Hiperbolik Tangen (tanh)	15
2.9.3 Fungsi Aktivasi <i>Rectified Linear Unit</i> (ReLU)	16
2.9.4 Fungsi Softmax	16
2.10 <i>AdamW Optimizer</i>	17

2.11	<i>Hyperparameter</i>	18
2.12	Transformer	19
2.12.1	<i>Encoder dan Decoder</i>	20
2.12.2	<i>Attention</i>	21
2.12.3	<i>Scaled Dot-Product Attention</i>	21
2.12.4	<i>Multi-Head Attention</i>	21
2.12.5	<i>Position-Wise Feed-Forward Networks</i>	23
2.12.6	<i>Embedding dan Fungsi Softmax</i>	23
2.12.7	<i>Positional Encoding</i>	24
2.13	<i>Text-to-Text Transfer Transformer (T5)</i>	24
2.14	<i>Fine-Tuning</i>	25
2.15	Evaluasi Model	26
2.15.1	<i>Recall-Oriented Understudy for Gisting Evaluation (ROUGE)</i>	26
2.15.2	<i>Cosine Similarity</i>	27
III	METODOLOGI PENELITIAN	29
3.1	Waktu dan Tempat Penelitian	29
3.1.1	Tempat Penelitian	29
3.1.2	Waktu Penelitian	29
3.2	Data Penelitian	30
3.2.1	Data	30
3.2.2	Alat	32
3.3	Metode Penelitian	34
IV	HASIL DAN PEMBAHASAN	37
4.1	<i>Input Data</i>	37
4.2	<i>Preprocessing Data</i>	37
4.2.1	Seleksi Fitur	38
4.2.2	Rekonstruksi <i>Token</i>	38
4.2.3	Reduksi <i>Noise</i>	39
4.2.4	<i>Lowercasing</i>	39
4.3	<i>Splitting Data</i>	40
4.4	<i>Embedding T5</i>	40
4.5	Konfigurasi Pelatihan	44
4.6	<i>Fine-tuning T5</i>	45
4.7	<i>Generated Summary</i>	48
4.8	Evaluasi Model	50

4.8.1	<i>Recall Oriented Understudy for Gisting Evaluation (ROUGE)</i>	50
4.8.2	<i>Cosine Similarity</i>	60
4.9	Perbandingan dengan penelitian sebelumnya	62
V	KESIMPULAN DAN SARAN	67
5.1	Kesimpulan	67
5.2	Saran	68
	DAFTAR PUSTAKA	69

DAFTAR TABEL

Tabel	Halaman
1 Penelitian Terkait Implementasi Model T5 dalam Meringkas Teks	4
2 Dataset <i>IndoSum</i>	31
3 Atribut yang dibutuhkan	38
4 Perbandingan Teks Sebelum dan Sesudah Diolah Rekonstruksi <i>Token</i>	38
5 Perbandingan Teks Sebelum dan Sesudah Diolah Reduksi <i>Noise</i>	39
6 Perbandingan Teks Sebelum dan Sesudah Diolah <i>Lowercasing</i>	39
7 Hasil Pembagian Dataset <i>IndoSum</i>	40
8 Proses <i>Embedding</i> T5	42
9 Konfigurasi Pelatihan Model T5	45
10 <i>Train Loss</i> dan <i>Val Loss Epoch 5</i>	47
11 Hasil <i>Generated Summary</i>	49
12 Artikel dan Ringkasan 1	49
13 Artikel dan Ringkasan 5	50
14 Contoh Ringkasan baris 1	51
15 Tokenisasi <i>Unigram</i> Ringkasan 1	52
16 Tokenisasi <i>Bigram</i> Ringkasan 1	54
17 Hasil Evaluasi ROUGE dari Ringkasan 1 (PM)	58
18 Hasil Evaluasi ROUGE dari Ringkasan 1	58
19 Hasil Evaluasi ROUGE Data Uji	59
20 Hasil <i>Cosine Similarity</i>	60
21 Vektor Biner Ringkasan 1	61
22 Perbandingan Metode dan Hasil Penelitian dengan Penelitian Terdahulu	63

DAFTAR GAMBAR

Gambar	Halaman
1. Struktur <i>Deep Learning</i>	13
2. Fungsi sigmoid	15
3. Fungsi Tanh	15
4. Fungsi ReLU	16
5. Fungsi softmax	17
6. Arsitektur Transformer	20
7. <i>Multi-Head Attention</i>	22
8. <i>Positional Encoding</i>	24
9. Arsitektur T5	25
10. Diagram Alir Penelitian.	36
11. <i>Distribusi Kata</i>	41
12. Grafik <i>Loss</i> dengan <i>Weight Decay</i> 0,01.	46
13. Grafik <i>Loss</i> dengan <i>Weight Decay</i> 0,05.	46
14. Grafik <i>Loss</i> dengan <i>Weight Decay</i> 0,001.	47
15. <i>Perbandingan Perhitungan Nilai ROUGE</i>	58
16. <i>Hasil Nilai ROUGE</i>	59
17. Perbandingan ROUGE antar Penelitian.	66

BAB I

PENDAHULUAN

1.1 Latar Belakang dan Masalah

Menurut Statista (2022), lebih dari 90 juta pengguna internet di Indonesia, atau sekitar 68% dari total populasi, secara aktif mengakses berita daring setiap hari. Berita daring telah menjadi sumber informasi utama di Indonesia dengan berbagai situs seperti CNN Indonesia, Kompas.com, Tribunnews.com, dan Liputan6.com yang menyediakan berita terbaru dan mudah diakses masyarakat. Pembaca kerap kesulitan dalam memilih dan menyaring informasi yang relevan karena banyaknya berita daring yang tersedia. Selain itu, teks berita yang panjang dan kompleks memerlukan waktu cukup lama untuk dibaca, sehingga mengganggu kenyamanan pembaca, terutama saat menggunakan perangkat digital seperti *smartphone* atau komputer (Ong dkk., 2021). Permasalahan ini semakin penting mengingat meningkatnya kebutuhan masyarakat akan akses informasi yang cepat dan efisien, khususnya dalam membaca dan memahami berita daring (Laksana dkk., 2022).

Peringkasan otomatis teks berita atau *automatic text summarization* dapat menjadi solusi efektif. Cabang penting dalam ilmu *Natural Language Processing* (NLP) ini bertujuan merepresentasikan teks panjang dengan mengompres informasi sehingga lebih cepat dipahami dan dibaca oleh pengguna (Joshi dkk., 2019). Peringkasan memungkinkan pengguna memahami berita hanya dengan membaca ringkasan yang memuat bagian terpenting. Oleh sebab itu, diperlukan model yang mampu melakukan peringkasan teks secara otomatis.

Model berbasis transformer merupakan salah satu pendekatan terdepan dalam peringkasan teks karena kemampuannya memahami konteks secara mendalam dan menghasilkan ringkasan berkualitas tinggi. Di antara model berbasis transformer, *Text-to-Text Transfer Transformer* (T5) menjadi pilihan unggul. Dikembangkan oleh Google, model T5 dirancang dengan pendekatan *text-to-text*

yang memungkinkan semua tugas NLP direpresentasikan sebagai transformasi teks. Model T5 telah digunakan secara luas dalam tugas seperti penerjemahan, klasifikasi teks, dan peringkasan, serta terbukti efektif untuk teks berbahasa Indonesia (Itsnaini dkk., 2023). Model ini tidak hanya mengekstrak kalimat penting dari teks asli, tetapi juga menyusun ulang informasi menjadi format yang lebih singkat, relevan, dan mudah dipahami.

Berbagai penelitian telah mengaplikasikan model T5 untuk meringkas teks berita bahasa Indonesia. Penelitian Itsnaini (2022) melaporkan hasil evaluasi memadai menggunakan model *pre-trained t5-base-indonesian-summarization-cased* pada teks berbahasa Indonesia yang ditunjukkan dari capaian skor ROUGE-1 sebesar 0,68, ROUGE-2 sebesar 0,61, dan ROUGE-L sebesar 0,65. Purnama dan Utami (2023) menguji penggunaan T5 pada dataset *IndoSum* dengan tiga skenario berbeda dalam tahap *preprocessing*, yaitu menggunakan *stemming* dan *stopword removal*, hanya menggunakan *stemming*, serta tanpa menggunakan kedua tahapan tersebut. Skenario yang hanya menggunakan *stemming* menghasilkan nilai ROUGE-1 tertinggi yaitu sebesar 0,17866 dibandingkan skenario lainnya. Penelitian Bahari dan Dewi (2024) menggunakan dataset *IndoSum* dengan *preprocessing* tiga tahap, yaitu *lower casing*, *filtering*, dan tokenisasi. Penelitian ini membandingkan dua *vocab* model T5, yakni *T5-default* dan *T5-id*. Hasil penelitian menunjukkan *T5-id* dengan *Adam optimizer* dan *learning rate* sebesar 1×10^{-4} merupakan model terbaik, dengan capaian nilai ROUGE-1 sebesar 0,61 dan ROUGE-2 sebesar 0,51.

Beberapa aspek perlu mendapat perhatian lebih lanjut, khususnya pengaturan *hyperparameter* saat *fine-tuning* yang belum banyak dieksplorasi secara mendalam. Penggunaan *optimizer* yang lebih optimal, seperti *AdamW optimizer*, yang dikenal lebih stabil dan cocok untuk model berbasis transformer seperti T5, masih jarang diterapkan dalam penelitian sebelumnya. Selain itu, evaluasi kinerja model yang hanya mengandalkan metrik ROUGE sederhana dianggap kurang memadai. Penggunaan metrik yang lebih komprehensif, seperti *precision*, *recall*, dan *F1-score*, juga masih jarang dilakukan. Penambahan metrik *cosine similarity* penting untuk menilai kemiripan makna antara ringkasan hasil model dan ringkasan referensi. Penggabungan metrik ROUGE yang komprehensif dan *cosine similarity* dapat menghasilkan penilaian yang lebih akurat dari segi struktur dan makna.

Penelitian ini bertujuan mengeksplorasi penerapan model T5 untuk peringkasan

teks berita bahasa Indonesia pada dataset *IndoSum*, dengan fokus pada *fine-tuning* dan peningkatan *hyperparameter* melalui penggunaan *AdamW optimizer*, serta evaluasi menggunakan metrik ROUGE yang komprehensif dan metrik *cosine similarity*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut dapat dirumuskan beberapa masalah sebagai berikut:

1. Optimalisasi model T5 melalui proses *fine-tuning* dengan *AdamW optimizer* untuk meningkatkan kualitas peringkasan teks berita berbahasa Indonesia.
2. Evaluasi kualitas ringkasan dengan penerapan metrik ROUGE (*precision*, *recall*, dan *F1-score*) serta *cosine similarity* untuk menilai kesesuaian struktur dan makna antara ringkasan model dan ringkasan referensi.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis proses *fine-tuning* model T5 melalui pengaturan *hyperparameter* dengan menggunakan *AdamW optimizer* terhadap kualitas ringkasan teks berita berbahasa Indonesia.
2. Mengevaluasi kinerja model T5 dalam menghasilkan ringkasan teks berita berbahasa Indonesia dengan menggunakan metrik ROUGE (*precision*, *recall*, dan *F1-score*) serta metrik *cosine similarity*.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Menambah wawasan mengenai pengembangan model T5 yang dioptimalkan dengan *AdamW optimizer* dalam peringkasan teks berita berbahasa Indonesia.
2. Menambah referensi dan dasar evaluasi komprehensif melalui metrik ROUGE dan *cosine similarity*, yang dapat digunakan untuk mendukung penelitian selanjutnya dalam bidang peringkasan teks otomatis.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian-penelitian yang relevan dijadikan sebagai acuan untuk memperdalam pemahaman mengenai metode yang akan diterapkan dalam penelitian ini. Selain itu, penelitian-penelitian tersebut juga berfungsi sebagai pembandingan antara hasil penelitian yang telah ada dengan penelitian yang akan dilakukan. Penelitian tersebut ditampilkan pada Tabel 1 berikut:

Tabel 1. Penelitian Terkait Implementasi Model T5 dalam Meringkas Teks

Penelitian	Data	Metode	Hasil
<i>Abstractive Text Summarization using Pre-Trained Language Model "Text-To-Text Transfer Transformer (T5)"</i> (Itsaini dkk., 2022)	Dataset <i>IndoSum</i> , digunakan 9387 dari 18774 data, dibagi menjadi: <i>train</i> 90% (8448 data), <i>test</i> 10% (939 data), dan <i>validation</i> 10% dari <i>train</i> (845 data)	<i>Preprocessing: Tokenization</i> , menambahkan prefiks <i>summary</i> pada artikel. <i>Modeling: pre-trained T5-base</i> dan <i>fine-tuning T5</i>	ROUGE-1: 0,68, ROUGE-2: 0,61, ROUGE-L: 0,65

Penelitian	Data	Metode	Hasil
Implementasi Peringkasan Dokumen Berbahasa Indonesia Menggunakan Metode <i>Text To Text Transfer Transformer</i> (T5) (Purnama dan Utami, 2023)	Dataset <i>IndoSum</i> dengan data <i>train</i> pada <i>fold</i> 1 dengan jumlah data sebanyak 5000 data	<i>Preprocessing: Lowercase, String punctuation, Tokenisasi.</i> Skenario: 1. <i>Stemming</i> dan <i>stopwords</i> 2. <i>Stemming</i> saja 3. Tanpa <i>stemming</i> dan <i>stopwords</i> <i>Modeling: pre-trained dan fine-tuning T5</i>	ROUGE-1: 0,17866
Peringkasan Teks Otomatis Abstraktif Menggunakan Transformer pada Teks Bahasa Indonesia (Bahari dan Dewi, 2024)	Dataset <i>IndoSum</i> dengan jumlah 19 ribu data, dibagi menjadi <i>train</i> 80% dan <i>test</i> 20%	<i>Preprocessing: Lowercasing, Filtering, Tokenization.</i> <i>Hyperparameter: Dropout 0,1 Optimizer Adam Learning rate: 1×10^{-3}, 1×10^{-4}, 1×10^{-5} Epoch 3.</i> <i>Modeling: pre-trained T5-id, T5-default dan fine-tuning T5</i>	ROUGE-1: 0,61, ROUGE-2: 0,51

2.1.1 Penelitian Pertama (Itsnaini dkk., 2022)

Penelitian pertama menggunakan model T5 untuk melakukan peringkasan teks pada dataset *IndoSum*. Penelitian ini menggunakan 9387 data berita berbahasa Indonesia yang tersedia dalam *IndoSum*, yang dibagi menjadi 90% data pelatihan dan 10% data pengujian. Selain itu, data validasi diambil sebesar 10% dari data pelatihan. Dengan demikian, jumlah data pelatihan adalah 8448, data pengujian sebanyak 939, dan data validasi sebanyak 845. Model "*cahya/t5-base-indonesian-summarization-cased*" digunakan sebagai model pra-pelatihan. Model T5-base ini diterapkan pada bagian *encoder* dalam bentuk teks berita berbahasa Indonesia. Hasil penelitian menunjukkan bahwa nilai

evaluasi rata-rata ROUGE-1 sebesar 0,68, ROUGE-2 sebesar 0,61, dan ROUGE-L sebesar 0,65.

2.1.2 Penelitian Kedua (Purnama dan Utami, 2023)

Penelitian kedua menggunakan model T5 untuk meringkas teks berita berbahasa Indonesia pada dataset *IndoSum*. Peneliti merancang tiga skenario berbeda dalam tahap *preprocessing* data, yaitu skenario pertama mengimplementasikan *stemming* dan *stopwords removal*. Skenario kedua hanya menggunakan *stemming*, sedangkan skenario ketiga tidak menerapkan *stemming* maupun *stopwords removal*. Pengujian dilakukan pada *fold* pertama dari dataset *IndoSum* dengan menggunakan 500 *paragraph* pertama, bertujuan untuk mempercepat waktu pelatihan data. Berdasarkan hasil evaluasi model, skenario kedua yang hanya menggunakan *stemming* menghasilkan nilai ROUGE-1 tertinggi sebesar 0,17866, dibandingkan dengan skenario pertama dan ketiga. Meskipun demikian, perbedaan nilai tersebut relatif kecil jika dibandingkan dengan skenario ketiga yang tidak menggunakan *stemming* maupun *stopwords removal*, yaitu sebesar 0,17568.

2.1.3 Penelitian Ketiga (Bahari dan Dewi, 2024)

Penelitian ketiga menggunakan model T5 untuk melakukan peringkasan teks pada dataset *IndoSum*. Pada penelitian ini, proses *preprocessing* yang dilakukan meliputi beberapa tahap, yaitu pertama, melakukan *lowercasing* untuk menyeragamkan huruf menjadi huruf kecil, kedua, melakukan *filtering* untuk menyaring teks yang memenuhi pola *regex*, dan ketiga, melakukan tokenisasi menggunakan *SentencePiece* untuk mengubah teks menjadi bagian-bagian kecil (*token*). Setelah proses *preprocessing* selesai, data dibagi menjadi dua set, yaitu *train set* yang digunakan untuk pelatihan model dan *test set* yang digunakan untuk mengukur kinerja model. Pelatihan model dilakukan menggunakan *pre-trained* model T5 dengan parameter sebagai berikut: $d_{\text{model}} = 512$, enam *layer encoder*, delapan *attention head*, $d_{\text{kv}} = 64$, dan $P_{\text{dropout}} = 0,1$. Komposisi data terdiri atas 80% *train set* dan 20% *test set*. *Optimizer* yang digunakan adalah Adam dengan konfigurasi $\beta_1 = 0,9$, $\beta_2 = 0,999$, $\epsilon = 1 \times 10^{-8}$, serta *learning rate* sebesar 1×10^{-3} , 1×10^{-4} , dan 1×10^{-5} . Hasil penelitian menunjukkan bahwa T5-*id* dengan menggunakan Adam *optimizer* dan *learning rate* sebesar 1×10^{-4} merupakan model terbaik dalam meringkas teks pada dataset *IndoSum*, dengan nilai ROUGE tertinggi, yaitu

ROUGE-1 sebesar 0,61 dan ROUGE-2 sebesar 0,51.

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) atau pemrosesan bahasa alami merupakan subbidang dari kecerdasan buatan yang berfokus menggali hubungan kompleks antara komputer dan bahasa manusia. Tujuan utama NLP adalah untuk memahami, menganalisis, dan menghasilkan informasi dalam bentuk teks dengan cara meniru kemampuan manusia (Nurwanda dkk., 2024).

Natural Language Processing (NLP) dapat diterapkan dalam berbagai bidang, seperti penerjemahan mesin, deteksi spam Email, ekstraksi informasi, ringkasan, dan tanya jawab. Berikut ini adalah penjelasan beberapa penerapan NLP beserta pekerjaan yang relevan yang telah dilakukan dalam bidang tersebut (Khurana dkk., 2022):

1. Penerjemahan Mesin (*Machine Translation*)

Penerjemahan mesin membantu mengatasi masalah bahasa dengan menerjemahkan kalimat dari satu bahasa ke bahasa lain menggunakan teknik statistik dan *deep learning*, seperti pada *Google Translate*. Tantangannya adalah agar terjemahan tetap mempertahankan makna dan struktur bahasa yang benar.

2. Kategorisasi Teks

Kategorisasi teks merupakan proses pengolahan data dalam jumlah besar, seperti dokumen dan laporan, untuk dikelompokkan ke dalam kategori yang telah ditentukan sebelumnya. Contoh penerapannya antara lain adalah sistem klasifikasi berita, filter spam pada email, dan pengelolaan tiket masalah. Filter spam menggunakan berbagai metode, seperti pemeriksaan konten atau penerapan aturan tertentu, guna mengidentifikasi email yang tidak diinginkan.

3. Penyaringan Spam

Penyaringan spam menggunakan teknik kategorisasi teks dan pembelajaran mesin seperti *Naïve Bayes*, *Support Vector Machines*, dan *Decision Trees*. Model yang sering digunakan untuk menyaring spam adalah Multivariat Bernoulli dan Multinomial, yang menganalisis kata-kata dalam email dan frekuensi penggunaannya.

4. Ekstraksi Informasi

Ekstraksi informasi bertujuan untuk menemukan informasi penting dalam

teks, seperti nama, tempat, atau tanggal, untuk membantu merangkum data dan mempermudah pencarian. Teknik yang digunakan termasuk *POS tagging*, *Chunking*, dan *Stemming*. Hasil ekstraksi ini digunakan untuk berbagai tujuan, seperti membangun database atau menyusun ringkasan.

5. Ringkasan

Ringkasan teks membantu mengurangi kelebihan informasi. Ada dua jenis ringkasan yaitu satu dokumen atau banyak dokumen. Ringkasan ini bisa berdasarkan konten umum atau fokus pada pertanyaan tertentu, dan dapat dilakukan dengan atau tanpa pelatihan sebelumnya. Ringkasan ini sangat berguna untuk memahami informasi penting dari banyak data.

6. Sistem Dialog

Sistem dialog digunakan dalam berbagai aplikasi dunia nyata, mulai dari memberikan dukungan hingga melakukan tindakan tertentu. Pada sistem dialog dukungan, dibutuhkan kesadaran konteks, sementara untuk melakukan tindakan, tidak memerlukan kesadaran konteks yang mendalam. Sistem dialog ini telah berkembang dari aplikasi kecil, seperti sistem teater rumah, hingga memungkinkan interaksi manusia dengan robot melalui bahasa alami, contohnya asisten Google, Cortana, Siri, dan Alexa.

Berbagai model NLP telah dikembangkan untuk menangani tugas-tugas bahasa alami dengan pendekatan serta keunggulan masing-masing. Salah satu model terbaru yang merevolusi berbagai tugas dalam NLP adalah T5. Model T5 mengubah setiap tugas NLP menjadi permasalahan *text-to-text*, yang berarti seluruh masukan dan keluaran direpresentasikan dalam bentuk teks. Pendekatan ini memungkinkan model untuk menangani beragam tugas NLP, mulai dari penerjemahan, ekstraksi informasi, hingga peringkasan, dalam satu kerangka kerja yang konsisten (Itsnaini dkk., 2023). Berkat kemampuannya yang serbaguna, T5 mampu mengatasi berbagai tantangan NLP secara lebih efisien dan fleksibel, serta dapat diterapkan dalam berbagai aplikasi yang melibatkan pemahaman dan pengolahan bahasa alami.

2.3 Text Mining

Text mining adalah metode yang menggunakan teknik komputasi dan pembelajaran mesin untuk menganalisis dan memproses teks secara otomatis. Metode ini bertujuan untuk mengekstrak informasi penting dari teks, seperti mengenali topik, menemukan kata kunci atau entitas, dan mengidentifikasi sentimen atau emosi.

Proses *text mining* memanfaatkan berbagai pendekatan, seperti pemrosesan bahasa alami (*natural language processing*), analisis korpus, dan algoritma pembelajaran mesin untuk mendukung analisis data teks (Amna dkk., 2023).

Selain itu, *text mining* juga mengintegrasikan konsep linguistik komputasi, seperti tokenisasi, *stemming*, dan analisis N-gram, untuk menghasilkan informasi yang relevan. Teknik ini banyak diterapkan di berbagai sektor, termasuk pemasaran, penelitian, dan pemerintahan, untuk keperluan seperti klasifikasi teks, pengelompokan data (*clustering*), serta pengelolaan sumber daya linguistik. Kemampuan *text mining* dalam mengelola dan menganalisis teks secara otomatis menjadikannya berperan penting dalam berbagai aplikasi karena dapat meningkatkan efisiensi dan menghasilkan wawasan yang bermanfaat.

2.4 Tokenisasi

Tokenisasi dalam NLP memecah teks menjadi unit-unit kecil yang disebut *token*. *Token-token* ini digunakan untuk membentuk *n-gram*, yaitu urutan dari *n token* yang berurutan (Alsharman dkk., 2020). *Unigram* adalah metode yang memisahkan teks menjadi *token* individual, seperti kata "Saya", "suka", "belajar". Metode ini berguna untuk analisis dasar seperti menghitung frekuensi kata, namun tidak melihat hubungan antar kata. Sementara itu, *bigram* memperhatikan pasangan dua kata berurutan, seperti "Saya suka" dan "suka belajar". *Bigram* membantu memahami hubungan antar kata dalam kalimat.

Selain *n-gram*, metode seperti *Longest Common Subsequence (LCS)* digunakan untuk menganalisis kesamaan urutan *token* dalam teks. LCS menemukan subsekuens *token* yang paling panjang dan sama di antara dua atau lebih teks, tanpa harus berurutan secara langsung. Contohnya, antara kalimat "Saya suka belajar NLP" dan "Belajar NLP itu menyenangkan," LCS akan mengidentifikasi subsekuens seperti "belajar NLP" sebagai bagian yang sama. Metode ini berguna untuk tugas seperti perbandingan teks, deteksi plagiarisme, atau analisis kesamaan konteks secara global.

2.5 Word Embedding

Word embedding adalah teknik untuk mengubah kata menjadi vektor angka yang merepresentasikan makna kata secara efisien dalam dimensi rendah, biasanya 300 angka yang secara efektif menggambarkan makna kata tersebut (Corcoran dkk., 2021). Metode ini mengatur kata-kata dengan makna serupa ke dalam pola spasial yang mirip, sehingga mempermudah analisis hubungan antar kata. *Word embedding* telah terbukti meningkatkan kinerja tugas NLP seperti analisis sentimen dan pengenalan nama.

Pada model berbasis transformer, seperti *Text-To-Text Transfer Transformer (T5)*, *embedding* digunakan untuk mengonversi *token* (potongan kata atau *subword*) menjadi representasi numerik sebelum diproses lebih lanjut. Tidak seperti *word embedding* tradisional yang fokus pada kata utuh, T5 mengoptimalkan *embedding* dengan pendekatan berbasis *subword* menggunakan metode seperti *SentencePiece*. Pendekatan ini memungkinkan T5 menangkap pola hubungan antara bagian-bagian kata yang lebih kecil sehingga memudahkan model untuk menangani variasi bentuk kata (infleksi) dan bahasa yang kompleks. Cara tersebut membuat T5 mampu memproses berbagai tugas NLP, seperti terjemahan, analisis sentimen, dan pengenalan entitas, meskipun untuk bahasa yang memiliki sumber daya terbatas.

2.6 Peringkasan Teks Otomatis

Peringkasan teks otomatis merupakan cabang penting dalam ilmu *natural language processing (NLP)* yang bertujuan untuk merepresentasikan teks yang panjang dengan cara mengompres informasi sehingga dapat dengan cepat dipahami dan dibaca oleh pengguna. Peringkasan teks memungkinkan pengguna untuk mudah memahami suatu berita hanya dengan membaca ringkasan yang memuat bagian penting dalam sebuah berita (Joshi dkk., 2019).

Peringkasan teks otomatis kini banyak digunakan untuk membantu meringkas informasi dalam berbagai bidang sehingga memudahkan pekerjaan dan menghemat waktu. Kehadiran teknologi ini mempermudah kita mendapatkan inti dari informasi yang panjang dengan hanya menampilkan bagian-bagian yang relevan. Teknologi ini memiliki manfaat luas di berbagai sektor, mulai dari akademik, media, hingga layanan publik, karena mampu meringkas isi dokumen secara efektif tanpa

mengorbankan makna utama.

Menurut Gusra dkk. (2024), peringkasan teks memiliki dua pendekatan yaitu sebagai berikut:

1. Peringkasan Teks Ekstraktif

Peringkasan teks ekstraktif berarti informasi atau kalimat penting diekstraksi dari file teks yang diberikan atau dokumen asli. Pendekatan peringkasan teks ekstraktif menggunakan fitur linguistik atau statistik untuk memilih kalimat informatif yang berguna. Pendekatan ekstraktif umumnya lebih sederhana karena hanya memilih bagian-bagian penting dari teks asli dan menyusunnya kembali, tanpa perlu memperhatikan tata bahasa atau pemilihan kata yang tepat.

2. Peringkasan Teks Abstraktif

Peringkasan teks abstraktif akan mencoba memahami file *input* atau file asli dan menghasilkan kembali *output* dalam beberapa kata dengan mengidentifikasi konsep utama dari file *input*. Pendekatan ini menghasilkan ringkasan dengan memahami konteks teks asli menggunakan NLP untuk membuat versi yang lebih ringkas. Hasil ringkasan dari metode ini umumnya merupakan parafrase dari teks sumber, dan bisa mengandung kata-kata baru yang tidak ada dalam teks aslinya.

2.7 *Machine Learning* (ML)

Machine Learning adalah cabang dari Kecerdasan Buatan (*Artificial Intelligence/AI*) yang memungkinkan sistem untuk belajar secara mandiri dan meningkatkan kemampuannya berdasarkan pengalaman tanpa campur tangan manusia (Rachna dkk., 2021). Pemanfaatan ML memungkinkan perangkat lunak menghasilkan prediksi yang lebih akurat tanpa memerlukan pengkodean secara langsung.

Konsep utama ML berfokus pada pengembangan algoritma yang dapat memproses data sebagai *input* dan, melalui analisis statistik, menghasilkan prediksi sebagai *output*. Saat ini, ML telah mendapatkan popularitas yang luas dan diterapkan dalam berbagai bidang, seperti pengenalan objek, identifikasi pola, pemrosesan teks, serta berbagai penelitian lainnya. Tujuan utama ML adalah mengajarkan komputer untuk memanfaatkan data guna menyelesaikan masalah tertentu dengan lebih efisien.

Terdapat beberapa tipe *machine learning* dalam menjalankan tugasnya (Shaveta,

2023) yang dijabarkan sebagai berikut:

1. *Supervised Learning*

Supervised learning (pembelajaran terawasi) adalah teknik pembelajaran mesin di mana sistem dilatih menggunakan data berlabel untuk memprediksi hasil. Algoritma ini belajar dari data yang sudah diketahui hasilnya untuk memetakan masukan ke keluaran yang diinginkan. Setelah pelatihan, sistem dapat memberikan prediksi pada data baru dan memperbaiki model berdasarkan perbandingan hasil prediksi dengan hasil yang benar. *Supervised Learning* dapat dibagi menjadi dua kategori utama yaitu klasifikasi yang digunakan untuk variabel keluaran kategorikal seperti deteksi spam, dan regresi yang digunakan untuk memprediksi variabel kontinu seperti cuaca atau tren pasar. Beberapa algoritma yang sering digunakan dalam kedua kategori ini termasuk *Random Forest*, *Decision Trees*, *Logistic Regression*, *Linear Regression*, dan *Support Vector Machines*.

2. *Unsupervised Learning*

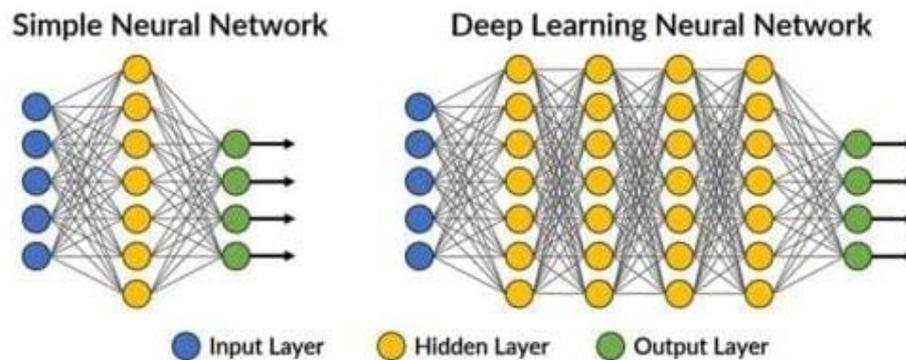
Unsupervised learning (pembelajaran tanpa pengawasan) adalah teknik pembelajaran mesin di mana sistem memproses data tanpa label atau kategori dan mencoba menemukan pola atau struktur tersembunyi dalam data tersebut. Tujuannya adalah untuk mengelompokkan data menjadi fitur atau objek yang serupa. Teknik ini tidak memiliki hasil yang sudah ditentukan sebelumnya dan digunakan ketika data tidak terklasifikasi. Pembelajaran tanpa pengawasan dibagi menjadi dua algoritma utama yaitu *clustering* yang mengelompokkan data berdasarkan kesamaan, dan *association* yang mencari hubungan antar item dalam dataset, seperti dalam analisis Market Basket di mana orang yang membeli roti sering membeli mentega atau selai.

3. *Reinforcement Learning*

Reinforcement learning (pembelajaran penguatan) adalah metode di mana agen belajar melalui umpan balik berupa penghargaan untuk tindakan yang benar dan hukuman untuk yang salah. Agen berinteraksi dengan lingkungan untuk menemukan cara terbaik memperoleh penghargaan, dengan tujuan meningkatkan kinerja. Pembelajaran ini mengandalkan *trial-and-error* dan penghargaan yang diberikan setelah beberapa waktu untuk menentukan tindakan terbaik. Contohnya, anjing robot yang belajar menggerakkan lengannya melalui proses ini.

2.8 Deep Learning

Deep Learning merupakan bagian dari *Machine Learning* yang menggunakan (*Artificial Neural Network*) dengan struktur yang mendalam dan kompleks. Teknologi ini dirancang untuk memungkinkan komputer atau sistem belajar secara mandiri dari data, mengenali pola-pola yang rumit, serta melaksanakan tugas atau mengambil keputusan tanpa memerlukan instruksi pemrograman eksplisit. Terinspirasi oleh cara kerja *neuron* dalam otak manusia, *deep learning* memanfaatkan ANN yang terdiri dari lapisan-lapisan pemrosesan dalam *deep learning* terdiri dari *input layer* untuk menerima data, *hidden layers* untuk mempelajari pola melalui perhitungan matematis dan fungsi aktivasi, serta *output layer* untuk menghasilkan hasil akhir. Setiap lapisan bertugas mengekstraksi fitur dari data mentah, sehingga memungkinkan pemahaman yang lebih mendalam terhadap hubungan yang kompleks dalam data (Sorin dkk., 2020) yang ditampilkan pada Gambar 1.



Gambar 1. Struktur *Deep Learning* (Nurhaki dan Yahfizham, 2024).

Deep learning memiliki berbagai arsitektur untuk menangani data dan tugas tertentu. *Convolutional Neural Network* (CNN) dirancang untuk data seperti gambar, menggunakan lapisan konvolusi untuk mengekstraksi fitur, *pooling* untuk mengurangi dimensi, dan *fully connected* untuk prediksi. Sementara itu, data sekuensial seperti teks ditangani oleh *Recurrent Neural Network* (RNN) yang memanfaatkan *hidden state* untuk menyimpan informasi urutan, meskipun sering menghadapi masalah *vanishing gradient*. Masalah ini diatasi oleh *Long Short-Term Memory* (LSTM) melalui penggunaan sel memori dan mekanisme gerbang. Selanjutnya, transformer merevolusi pemrosesan data sekuensial dengan *self-attention*, yang memungkinkan analisis seluruh *input* secara bersamaan,

sehingga lebih efisien dan akurat dibandingkan RNN maupun LSTM, khususnya dalam tugas NLP.

2.9 Fungsi Aktivasi

Fungsi aktivasi adalah elemen penting dalam jaringan saraf tiruan yang berfungsi untuk mengaktifkan atau menonaktifkan *neuron*. Fungsi ini memungkinkan jaringan saraf tiruan mengenali pola data nonlinier, karena *output* yang dihasilkan cenderung tidak linier. Proses kerjanya melibatkan pengubahan data nonlinier menjadi linier pada suatu rentang tertentu. Karakteristik jaringan saraf tiruan ditentukan oleh bobot dan hasil dari penerapan fungsi aktivasi pada *input-output* (Pasaribu dkk., 2020). Adapun fungsi aktivasi yang umum digunakan yaitu:

2.9.1 Fungsi Aktivasi Sigmoid

Fungsi aktivasi sigmoid adalah fungsi nonlinier yang mengubah nilai menjadi rentang antara 0 dan 1. Ketika nilai *input* mendekati 0, perubahan kecil pada *input* dapat menghasilkan perubahan signifikan pada *output*. Namun, gradien pada area ini cenderung sangat kecil, sehingga penting untuk memperhatikan inisialisasi bobot *neuron* pada fungsi sigmoid (Ivan Purnomo, 2022). Fungsi sigmoid dirumuskan pada Persamaan (1).

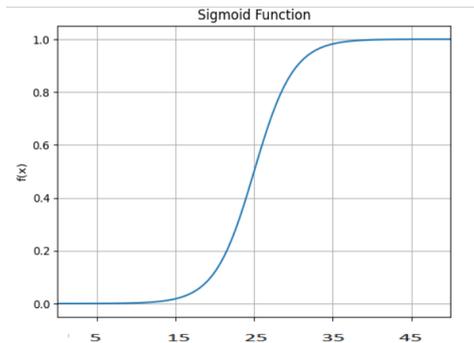
$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

Keterangan:

e = Bilangan Euler (sekitar 2,71828).

x = Input ke fungsi sigmoid.

Fungsi sigmoid dapat digambarkan dalam bentuk grafik yang ditunjukkan pada Gambar 2.



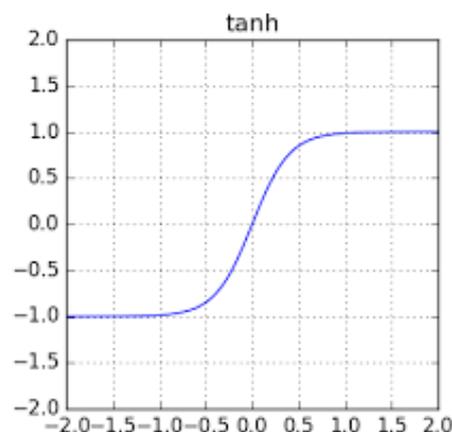
Gambar 2. Fungsi sigmoid (Wibawa, 2016).

2.9.2 Fungsi Aktivasi Hiperbolik Tangen (tanh)

Fungsi aktivasi tanh mirip dengan fungsi sigmoid. Jika digambarkan pada diagram keduanya membentuk huruf "S". Akan tetapi, nilai fungsi tanh berada pada rentang -1 hingga 1 (Ivan Purnomo, 2022). Fungsi tanh didefinisikan pada Persamaan (2) sebagai berikut:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{\sinh(x)}{\cosh(x)} \quad (2)$$

Fungsi tanh berpusat pada nol, sehingga tidak perlu khawatir tentang perubahan besar pada *output* berdasarkan beberapa *input*, meskipun masih menjadi masalah bahwa gradien fungsi semakin kecil saat mendekati 0, hal ini ditunjukkan pada Gambar 3.



Gambar 3. Fungsi tanh (Wibawa, 2016).

2.9.3 Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

Fungsi aktivasi ReLU merupakan fungsi nonlinier dimana fungsi ini tidak mengaktifkan semua *neuron* secara bersamaan. *Neuron* hanya akan dinonaktifkan ketika *output* dari transformer linier kurang dari 0 (Taqiyuddin dkk., 2023). Fungsi ReLU ditunjukkan pada Persamaan (3).

$$f(x) = \max(0, x) \quad (3)$$

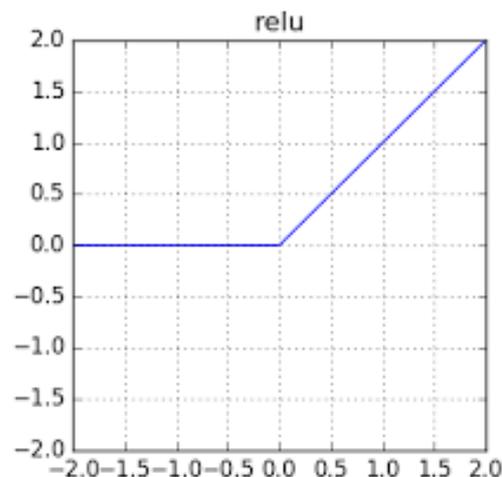
Keterangan

$f(x)$ = Fungsi ReLU (*Rectified Linear Unit*).

x = Input ke fungsi ReLU.

$\max(0, x)$ = Fungsi nilai maksimum antara 0 dan x .

Fungsi ReLU dapat ditampilkan dalam bentuk grafik pada Gambar 4.



Gambar 4. Fungsi ReLU (Wibawa, 2016).

2.9.4 Fungsi Softmax

Fungsi aktivasi softmax berperan dalam mengubah keluaran mentah jaringan saraf menjadi sebuah vektor probabilitas yang merepresentasikan distribusi probabilitas di antara berbagai kelas *input* (Taqiyuddin dkk., 2023). Pada arsitektur *deep learning*, fungsi softmax memiliki peran penting sebagai fungsi aktivasi pada lapisan *output*. Fungsinya adalah mengambil vektor nilai real dari lapisan sebelumnya dan menormalisasikannya sehingga menghasilkan probabilitas yang

dapat dipahami. Persamaan yang menggambarkan fungsi ini dituliskan pada Persamaan (4).

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (4)$$

Keterangan

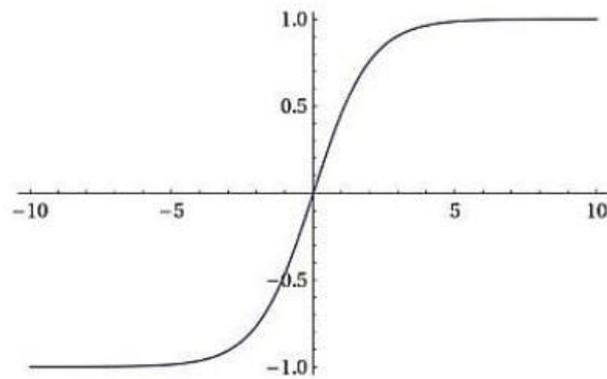
x_i = Nilai *input* untuk kelas i .

e = Bilangan Euler (sekitar 2,71828).

n = Jumlah total kelas.

$\sum_{j=1}^n e^{x_j}$ = Penjumlahan dari e^{x_j} , untuk setiap kelas j dari 1 sampai n .

Fungsi softmax digambarkan dalam bentuk grafik pada Gambar 5.



Gambar 5. Fungsi softmax (Purwitasari dan Soleh, 2023).

2.10 AdamW Optimizer

AdamW adalah modifikasi dari *optimizer* Adam yang mengatasi kelemahan dalam penanganan *weight decay* pada Adam. Pada Adam, *weight decay* sering diterapkan sebagai penalti pada fungsi *loss* dan dicampur langsung dengan perhitungan gradien, sehingga efektivitas regularisasi menjadi kurang optimal. Pada AdamW, *weight decay* tidak dicampur dengan estimasi gradien, melainkan diterapkan secara langsung untuk memperkecil besarnya bobot, yaitu parameter model yang menentukan kekuatan pengaruh setiap *input* terhadap *output* (Mahajaya dkk., 2024).

Formulasi pembaruan dalam AdamW dijabarkan pada Persamaan (5), (6), (7), (8) dan (9) sebagai berikut:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot g_t \quad (5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (6)$$

$$\hat{m}_t = \frac{m_{t-1}}{1 - \beta_1^t} \quad (7)$$

$$\hat{v}_t = \frac{v_{t-1}}{1 - \beta_2^t} \quad (8)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} - \alpha \cdot \omega d \cdot \theta_t \quad (9)$$

Keterangan:

m_t = estimasi momen pertama (*mean*) dari gradien.

v_t = estimasi momen kedua (variansi tak terpusat) dari gradien.

g_t = gradien pada waktu t .

β = koefisien peluruhan eksponensial untuk estimasi momen.

θ = parameter yang akan dioptimalkan.

θ_{t+1} = Parameter model setelah pembaruan.

α = *learning rate*.

ϵ = h konstanta yang sangat kecil untuk mencegah pembagian dengan nol.

ωd = *weight decay*.

2.11 Hyperparameter

Hyperparameter merupakan variabel di luar model yang dinilainya tidak dapat diprediksi langsung dari data, sehingga perlu ditentukan sebelum pelatihan dimulai. Penentuan *hyperparameter* ini penting untuk menghasilkan prediksi yang optimal (Hikmaturokhman dkk., 2022). Beberapa komponen utama yang harus diperhatikan dalam proses konfigurasi model pelatihan meliputi:

1. Fungsi Aktivasi untuk mengatur apakah *neuron* dalam jaringan saraf tiruan akan diaktifkan, sehingga memengaruhi keluaran jaringan.
2. Fungsi *Loss* digunakan untuk mengukur kesalahan model selama pelatihan yang nantinya menjadi acuan untuk memperbaiki parameter model.
3. *Optimizer* berfungsi memperbaharui bobot jaringan berdasarkan data pelatihan untuk meningkatkan kinerja model.

Selain itu, terdapat beberapa *hyperparameter* utama yang berperan dalam

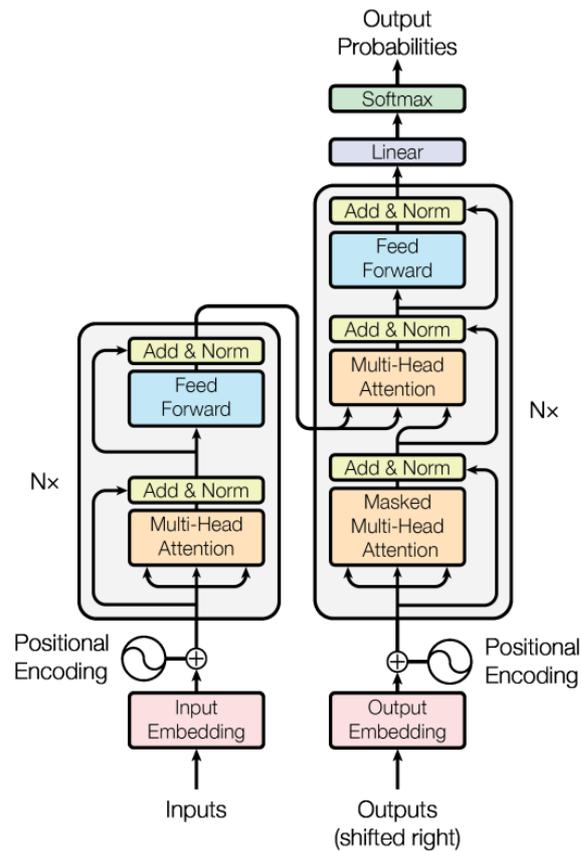
pengaturan kinerja model, di antaranya:

- Jumlah Lapisan Tersembunyi
Meningkatkan kapasitas model dalam mengenali pola-pola yang kompleks.
- Jumlah *Neuron* pada Tiap Lapisan Tersembunyi
Mempengaruhi kemampuan jaringan untuk mempersentasikan data. Jika terlalu banyak *neuron* akan menyebabkan *overfitting*.
- Ukuran *Batch*
Menentukan jumlah data yang dilatih dalam setiap iterasi.
- Jumlah *Epoch*
Menggambarkan jumlah siklus penuh pelatihan pada seluruh data.
- *Learning Rate*
Menentukan seberapa besar parameter model diperbaharui dalam setiap langkah optimisasi.
- Parameter Regulasi
Digunakan untuk mengurangi risiko *overfitting* dengan mengontrol kompleksitas model.

Pemilihan *hyperparameter* yang tepat sangat berpengaruh terhadap keberhasilan pelatihan model. Kombinasi komponen yang optimal dapat meminimalkan kesalahan yang mendekati target yang diharapkan (Hikmaturokhman dkk., 2022).

2.12 Transformer

Transformer adalah arsitektur jaringan saraf tiruan yang revolusioner dalam NLP, sebagai bagian dari pengembangan *deep learning*. Berbeda dengan model sebelumnya seperti RNN dan LSTM, transformer mengandalkan mekanisme *self-attention*, yang memungkinkan pemrosesan *input* secara paralel dan lebih efisien. Diperkenalkan oleh Vaswani dan rekannya pada 2017 dalam artikel *Attention is All You Need*, model ini menggunakan pendekatan *encoder-decoder* yang dilatih secara *end-to-end*, meningkatkan kinerja dalam memahami konteks global dalam teks. Arsitektur model ini diilustrasikan pada Gambar 6.



Gambar 6. Arsitektur Transformer (Vaswani dkk., 2017).

2.12.1 Encoder dan Decoder

Encoder terdiri dari enam lapisan identik, masing-masing dengan dua sub-lapis yaitu *multi-head self-attention* dan jaringan *feed-forward* posisi tertentu. Setiap sub-lapis menggunakan koneksi residual diikuti dengan *layer normalization*. Output dari setiap sub-lapis memiliki dimensi $d_{\text{model}} = 512$.

Decoder juga terdiri dari enam lapisan, tetapi ditambahkan sub-lapis ketiga yang melakukan *multi-head attention* pada *output encoder*. Seperti *encoder*, digunakan koneksi residual dan *layer normalization*, serta modifikasi pada sub-lapis *self-attention* untuk mencegah posisi memperhatikan posisi yang lebih lanjut. Ini memastikan prediksi untuk posisi i hanya bergantung pada *output* yang diketahui dari posisi sebelumnya.

2.12.2 Attention

Fungsi *Attention* adalah mekanisme yang memetakan sebuah *query* bersama dengan kumpulan pasangan *key-value* menjadi sebuah *output*, di mana *query*, *key*, *value*, dan *output* direpresentasikan dalam bentuk vektor. Hasil akhir (*output*) diperoleh melalui penjumlahan berbobot (*weighted sum*) dari elemen-elemen *value*, dengan bobot yang ditentukan oleh tingkat kesesuaian (*similarity*) antara *query* dan *key*.

2.12.3 Scaled Dot-Product Attention

Scaled Dot-Product Attention terdiri dari *Queries* dan *Keys* yang memiliki dimensi d_k , serta *Values* yang memiliki dimensi d_v . Produk skalar dihitung antara *query* dan semua *key*, kemudian dibagi dengan $\sqrt{d_k}$, dan fungsi softmax diterapkan untuk menghasilkan bobot yang diterapkan pada *values*. Dalam praktiknya, *Scaled Dot-Product Attention* dihitung untuk sejumlah *query* secara bersamaan yang disusun dalam matriks Q . *Keys* dan *values* juga disusun dalam matriks K dan V . Matriks keluaran dihitung menggunakan rumus yang ditunjukkan pada Persamaan (10).

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (10)$$

Keterangan:

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

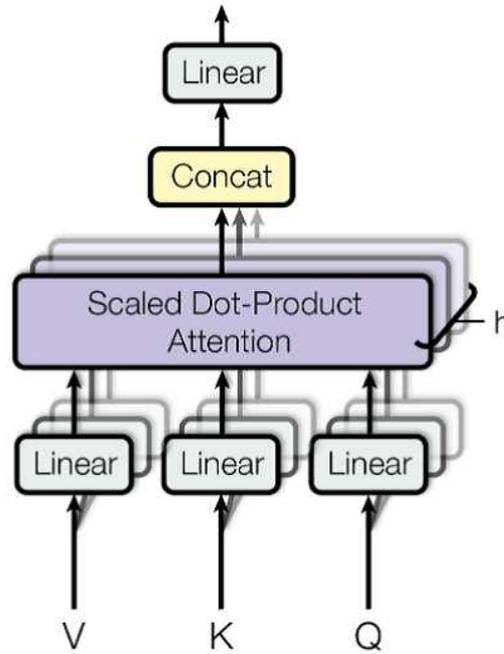
K = *Key*, representasi elemen untuk mencocokkan relevansi.

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

d_k = Dimensi dari vektor *key*, digunakan untuk normalisasi.

2.12.4 Multi-Head Attention

Multi-Head Attention memproyeksikan *queries*, *keys*, dan *values* secara linier sebanyak h kali dengan proyeksi linier yang dipelajari ke dimensi d_k , dan d_v masing-masing. Pada setiap versi proyeksi dari *queries*, *keys*, dan *values*, dilakukan fungsi *Attention* secara paralel yang menghasilkan *output values* berdimensi d_v . *Output* ini kemudian digabungkan dan diproyeksikan kembali untuk menghasilkan nilai akhir. Bagian *Multi-Head Attention* disajikan pada Gambar 7.



Gambar 7. *Multi-Head Attention* (Vaswani dkk., 2017).

Multi-Head Attention memungkinkan model untuk secara bersamaan memperhatikan informasi dari subruang representasi yang berbeda pada posisi yang berbeda yang ditunjukkan pada Persamaan (11) dan (12).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \quad (11)$$

$$\text{head}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i}) \quad (12)$$

dimana proyeksi-proyeksi tersebut adalah matriks parameter:

$$W_{Q_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}, \quad W_{K_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}, \quad W_{V_i} \in \mathbb{R}^{d_{\text{model}} \times d_v}, \quad W_O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$$

Keterangan:

Q = *Query*.

K = *Key*.

V = *Value*.

head_i = *Output* dari mekanisme *attention* pada kepala ke- i .

W_{Q_i} = Matriks parameter untuk memproyeksikan *query* pada kepala ke- i .

W_{K_i} = Matriks parameter untuk memproyeksikan *key* pada kepala ke- i .

W_{V_i} = 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*.

2.12.5 Position-Wise Feed-Forward Networks

Selain sub-lapisan *Attention*, setiap lapisan pada *encoder* dan *decoder* juga dilengkapi dengan jaringan *feed-forward* yang sepenuhnya terhubung, yang diterapkan pada setiap posisi secara terpisah dan identik. Jaringan ini terdiri dari dua transformasi linier dengan aktivasi ReLU diantara keduanya yang ditunjukkan pada Persamaan (13).

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (13)$$

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*).

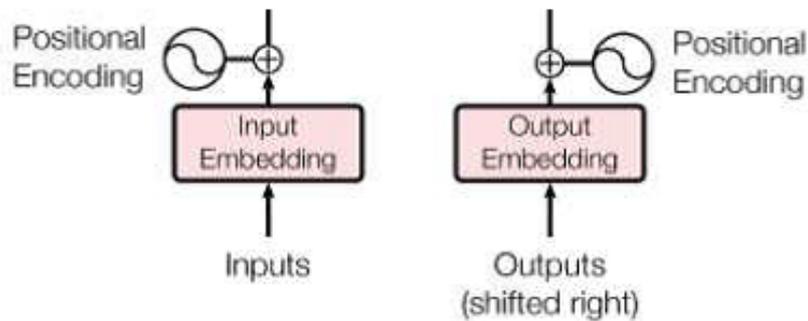
Meskipun transformasi linier yang diterapkan adalah sama untuk setiap posisi, parameter yang digunakan berbeda untuk setiap lapisan. Hal ini juga dijelaskan sebagai dua konvolusi dengan ukuran kernel 1. Dimensi *input* dan *output* adalah $d_{\text{model}} = 512$, sementara dimensi lapisan tengahnya adalah $quadd_{ff} = 2048$.

2.12.6 Embedding dan Fungsi Softmax

Mirip dengan model-transduksi urutan lainnya, digunakan *embedding* yang dipelajari untuk mengonversi *token input* dan *output* menjadi vektor dengan dimensi d_{model} . Selain itu, digunakan transformasi linier yang dipelajari dan fungsi softmax untuk mengonversi *output* dari *decoder* menjadi probabilitas *token* berikutnya yang diprediksi. Matriks bobot yang sama digunakan antara dua lapisan *embedding* dan transformasi linier sebelum *softmax*. Pada lapisan *embedding*, bobot tersebut dikalikan dengan $\sqrt{d_{\text{model}}}$.

2.12.7 Positional Encoding

Positional Encoding digunakan untuk memberikan informasi tentang posisi relatif atau absolut dari *token* dalam sekuens *input* karena model tidak memiliki mekanisme rekursif dan konvolusi secara alami. *Positional Encoding* ditambahkan pada *embedding input* di bagian bawah tumpukan *encoder* dan *decoder*. *Positional encoding* memiliki dimensi yang sama dengan d_{model} seperti pada *embedding* sehingga keduanya dapat dijumlahkan. Biasanya, *positional encoding* dihasilkan menggunakan fungsi sinus dan cosinus dengan frekuensi berbeda yang memungkinkan model mengenali posisi relatif antar *token*. Posisi *positional encoding* diilustrasikan pada Gambar 8.



Gambar 8. *Positional Encoding* (Vaswani dkk., 2017).

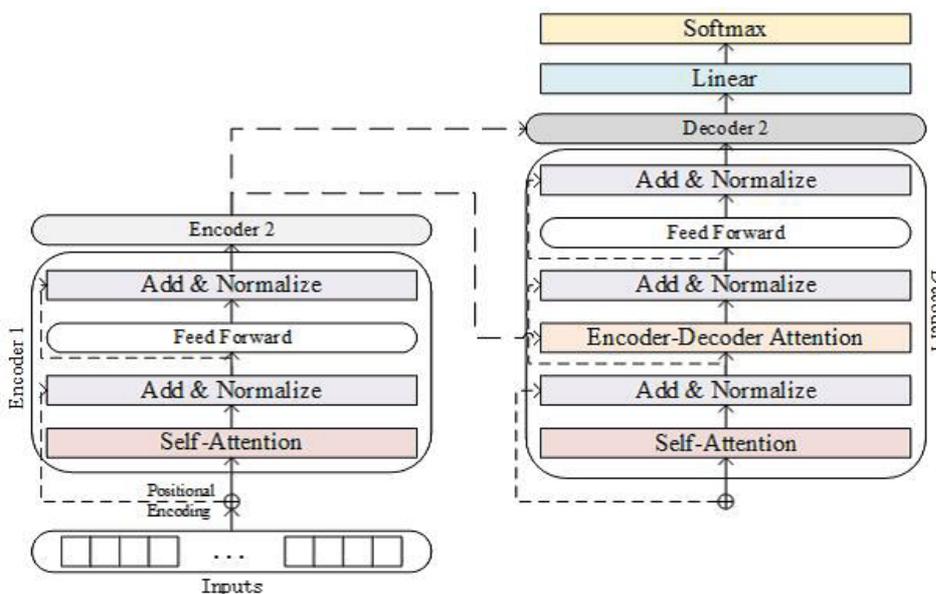
2.13 Text-to-Text Transfer Transformer (T5)

Text-to-Text Transfer Transformer (T5) adalah *framework* yang dibangun di atas arsitektur populer seperti *Bidirectional Encoder Representations from Transformers* (BERT) dan *Generative Pre-trained Transformer* (GPT), dengan pendekatan pembelajaran transfer berbasis teks-ke-teks. Model T5 merupakan model transformer *end-to-end* yang menggunakan teks sebagai *input* dan menghasilkan teks yang telah diolah sebagai *output*. Model ini telah berhasil diterapkan pada berbagai tugas NLP, seperti terjemahan mesin, ringkasan teks, penjawaban pertanyaan, dan analisis sentimen (Itsaini dkk., 2023).

Model T5 menggunakan *pre-training* untuk mempelajari pola bahasa umum dari dataset besar dan *fine-tuning* untuk menyesuaikan model dengan tugas spesifik. Pada tahap *fine-tuning*, prefiks tugas ditambahkan pada *input*, dan bobot model

disesuaikan menggunakan dataset spesifik, sementara arsitektur *encoder-decoder* tetap tidak berubah. Pendekatan ini memanfaatkan pengetahuan *pre-training* untuk meningkatkan efisiensi dan kinerja pada tugas tertentu (Wang dkk., 2023).

Arsitektur model T5 terdiri dari struktur *encoder-decoder* yang dirancang untuk mengolah data dalam format teks. *Encoder* memproses input teks dengan mekanisme *self-attention*, *add & normalize*, dan *feed-forward*, serta menambahkan *positional encoding* untuk mempertimbangkan urutan data. *Decoder* menghasilkan output secara *autoregresif* dengan mekanisme *self-attention encoder-decoder attention* untuk menghubungkan hasil dari *encoder* dan *feed forward* untuk transformasi tambahan. Lapisan akhir *decoder* menggunakan fungsi softmax untuk menghasilkan probabilitas *token* keluaran. Dengan pendekatan ini, T5 mampu menangani berbagai tugas NLP dalam format *text-to-text* secara efisien. Arsitektur model T5 dapat diilustrasikan pada Gambar 9.



Gambar 9. Arsitektur T5 (Wang dkk., 2017).

2.14 Fine-Tuning

Fine-tuning adalah salah satu metode dalam *transfer learning* yang digunakan untuk meningkatkan kinerja model pada tugas tertentu dengan cara menyesuaikan kembali parameter model yang telah dilatih sebelumnya (*pre-trained model*) (Pradana dkk., 2024). Teknik ini memungkinkan penggunaan kembali pengetahuan

yang telah dipelajari oleh model dari dataset berskala besar, kemudian disesuaikan agar relevan dengan tugas baru yang memiliki cakupan lebih sempit atau data yang lebih sedikit.

Proses *fine-tuning* biasanya dimulai dengan mengambil model yang sudah dilatih pada dataset besar dan umum, seperti data Wikipedia atau *Common Crawl*. Selanjutnya, lapisan akhir model yang bertugas melakukan prediksi diubah agar sesuai dengan jenis *output* yang diinginkan, misalnya klasifikasi label atau ringkasan teks. Setelah itu, model dilatih ulang (biasanya hanya sebagian lapisan) menggunakan dataset target. Pada tahap ini, parameter seperti bobot dan bias diperbarui agar sesuai dengan pola dalam data yang baru.

Fine-tuning sangat berguna karena tidak memerlukan pelatihan model dari awal (*training from scratch*), sehingga menghemat waktu dan sumber daya komputasi, sekaligus menghasilkan kinerja yang tinggi meskipun dengan data yang terbatas.

2.15 Evaluasi Model

Ringkasan berita yang didapat dari model T5 perlu dilakukan evaluasi untuk menentukan seberapa baik ringkasan yang dihasilkan melalui model T5. Model ini menggunakan *ROUGE scores* dan *cosine similarity* sebagai evaluasi model untuk menilai kesesuaian isi ringkasan dengan teks aslinya secara keseluruhan.

2.15.1 *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE)

Recall-Oriented Understudy for Gisting Evaluation atau ROUGE merupakan kumpulan metrik yang digunakan untuk menilai rangkuman teks otomatis dan terjemahan mesin dalam NLP (Lin, 2004). *ROUGE scores* terdiri dari *presicion*, *recall* dan *F1-score* untuk ROUGE-N dengan nilai N adalah 1, 2, atau L. Variabel N bermakna jumlah N kata berurutan yang sama dari kedua dokumen dibandingkan, yaitu ringkasan metode T5 dan ringkasan manusia. N bernilai 1 menghitung *word-1-gram* (*unigram*), dan N bernilai 2 menghitung *word-2-gram* (*bigram*) yang terdapat di dalam kedua dokumen. Sementara itu, variabel-N yang bernilai L adalah *Longest Common Subsequent* (LCS), yaitu kata-kata berurutan terpanjang yang ada pada kedua dokumen yang dibandingkan. Menurut Idhafi dkk. (2023) persamaan untuk menghitung *ROUGE scores* adalah sebagai berikut:

1. Recall

Recall adalah metode untuk mengukur jumlah prediksi yang relevan dengan cara menghitung jumlah kata yang sama, baik *unigram*, *bigram*, atau LCS dibagi keseluruhan kata pada ringkasan manusia yang dirumuskan pada Persamaan (14), (15), dan (16).

$$\text{ROUGE-1 recall} = \frac{\text{jumlah unigram yang sama}}{\text{keseluruhan kata di ringkasan manusia}} \quad (14)$$

$$\text{ROUGE-2 recall} = \frac{\text{jumlah bigram yang sama}}{\text{keseluruhan kata di ringkasan manusia}} \quad (15)$$

$$\text{ROUGE-L recall} = \frac{\text{LCS (Longest Common Subsequent)}}{\text{keseluruhan kata di ringkasan manusia}} \quad (16)$$

2. Precision

Precision adalah metode untuk mengukur jumlah yang diprediksi relevan dengan cara menghitung jumlah kata yang sama, baik *unigram*, *bigram*, atau LCS dibagi keseluruhan kata pada ringkasan sistem yang dirumuskan pada Persamaan (17), (18), dan (19).

$$\text{ROUGE-1 precision} = \frac{\text{jumlah unigram yang sama}}{\text{keseluruhan kata di ringkasan mesin}} \quad (17)$$

$$\text{ROUGE-2 precision} = \frac{\text{jumlah bigram yang sama}}{\text{keseluruhan kata di ringkasan mesin}} \quad (18)$$

$$\text{ROUGE-L precision} = \frac{\text{LCS (Longest Common Subsequent)}}{\text{keseluruhan kata di ringkasan mesin}} \quad (19)$$

3. F1-Score

F1-Score adalah metode untuk mengukur nilai rata-rata hermonok (*harmonic mean*) antara *recall* dan *precision* yang dirumuskan pada Persamaan (20).

$$\text{F1-Score} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} \quad (20)$$

2.15.2 Cosine Similarity

Cosine Similarity merupakan metode yang digunakan untuk mengukur seberapa mirip dua teks dengan merepresentasikan masing-masing sebagai vektor dalam ruang berdimensi-n. Dalam konteks ini, setiap kata dalam teks dianggap sebagai dimensi, dan frekuensi kemunculan kata tersebut menjadi nilai dalam vektor. Kesamaan antara dua teks dihitung dengan mencari nilai kosinus dari sudut antara

kedua vektor tersebut (Fiarni dkk., 2019). Persamaan yang menggambarkan *cosine similarity* ditunjukkan pada Persamaan (21).

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \cos(\theta) = \frac{\sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (21)$$

dengan:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i \quad (\text{perkalian titik/dot product dari } \mathbf{x} \text{ dan } \mathbf{y})$$

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n \mathbf{x}_i^2} \quad (\text{panjang/norma dari vektor } \mathbf{x})$$

$$\|\mathbf{y}\| = \sqrt{\sum_{i=1}^n \mathbf{y}_i^2} \quad (\text{panjang/norma dari vektor } \mathbf{y})$$

Nilai *cosine similarity* berkisar antara 0 hingga 1, dimana 1 menunjukkan bahwa kedua teks tersebut sangat mirip atau identik, dan 0 menunjukkan bahwa kedua teks sama sekali tidak mirip (Fiarni dkk., 2019).

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilakukan secara studi pustaka di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang berlokasi 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 2024/2025, tepatnya pada bulan September 2024, dan terdiri dari tiga tahap. Tahap pertama melakukan pengumpulan dan pemahaman literatur dari sumber jurnal dan buku serta menentukan tema penelitian. Setelah menentukan tema penelitian, dilakukan pencarian data terkait. Pada tahap ini memerlukan waktu kurang lebih 14 minggu. Tahap kedua melakukan proses pengolahan data menggunakan program, seperti *preprocessing* data dilakukan selama 4 minggu. Setelah itu, peneliti melakukan *splitting* data, *embedding* data, *fine-tuning* T5, *generated summary*, dan evaluasi kinerja model. Pada tahap ini dibutuhkan waktu kurang lebih 11 minggu. Tahap terakhir adalah penyusunan hasil pengerjaan dan analisis model dalam bentuk draf laporan, yang kemudian dipresentasikan dalam seminar hasil kurang lebih 5 minggu.

3.2 Data Penelitian

3.2.1 Data

Data yang digunakan dalam penelitian ini adalah dataset *IndoSum* yang diambil dari *website* Kaggle dengan *link* sebagai berikut: <https://www.kaggle.com/datasets/linkgish/indosum>. Dataset *IndoSum* memiliki jumlah data ada sebanyak 18774 data. Pada penelitian ini, dataset tersebut hanya digunakan sebanyak 9387 data karena disebabkan oleh keterbatasan sumber daya dalam pelaksanaan penelitian. Meskipun demikian, keterbatasan tersebut tidak berdampak pada proses evaluasi dalam pemodelan, sehingga tujuan penelitian tetap dapat dicapai.

Dataset ini berformat *jsonl* dan terdiri dari beberapa atribut, yaitu *category* (menunjukkan kategori berita), *gold labels* (label ringkasan emas untuk peringkasan ekstraktif), *id* (identitas unik setiap berita), *source* (sumber berita), *source url* (URL dari berita), *paragraphs* (berisi paragraf teks dari berita), dan *summary* (ringkasan berita yang dibuat oleh manusia).

Pada penelitian dataset dibagi menjadi dua yaitu data model 90% dan data pengujian sebesar 10%. Data model dibagi lagi menjadi data *training* sebesar 80% dan data *validation* sebesar 10%. Berikut ini struktur *dataset IndoSum* yang ditampilkan pada Tabel 2.

Tabel 2. Dataset *IndoSum*

	Category	Gold Labels	ID	Paragraphs	Source	Source URL	Summary
0	Tajuk Utama	[[False, True],[True, True], [False, False, F...]]	1501893029 -lula-kamal-dokter -ryan-thamrin- saki...	[[[Jakarta, ,, CNN, Indonesia, -, -, Dokter, ...]]]]	CNN Indonesia	https://www.cnnindonesia.com/hiburan/20170804123456-234-123456	[[Dokter, Lula, Kamal, yang, merupakan, selebr...]]
1	Teknologi	[[False, False, False, False], [False, True, T...]]	1509072914-dua -smarthpne- zenfone-baru- tawarka...	[[[Selfie, ialah, salah, satu, tema, terpanas, ...]]]]	dailysocial.id	https://dailysocial.id/post/dua-smartphone-zen...	[[Asus, memperkenalkan, ZenFone, generasi, ...]]
2	Hiburan	[[True], [True], [False], [False], [False], [False], [Fal...]]	1510613677- songsong-visit- 2020-bengkulu- perkua...	[[[Jakarta, ,, CNN, Indonesia, -, -, Dinas, Pa...]]]]	cnn indonesia	https://www.cnnindonesia.com/gaya-hidup/201711...	[[Dinas, Parowosata, Provinsi, Bengkulu, kempa...]]
...
9386	Olahraga	[[True], [True, True], [False], [False], [Fals...]]	1517509800- manchester-city- mantap-di-puncak -kl...	[[[INGGRIS, -, Manchester, City, kian, nyaman, ...]]]]	poskotanews	http://poskotanews.com/2018/02/01/bekap-westb...	[[Manchester, City, kian, nyaman, di, puncak, ...]]

3.2.2 Alat

Alat yang digunakan dari penelitian ini terbagi menjadi dua sebagai berikut:

a. Sistem Operasi

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

b. Menggunakan *Notebook Kaggle* dengan *language python 3.10.14*, *accelerator GPU P100*

Notebook kaggle adalah platform berbasis web yang memungkinkan pengguna untuk menulis, menjalankan, dan berbagi kode *python* dalam format *Jupyter Notebook*. *Python 3.10.14* adalah versi *python* yang digunakan, menawarkan fitur baru, perbaikan bug, dan peningkatan performa. *Accelerator GPU P100* memberikan akses ke GPU Tesla P100 untuk mempercepat komputasi, terutama dalam pelatihan model *deep learning*, mengurangi waktu pelatihan dibandingkan menggunakan CPU biasa (Uslu, 2022).

c. *Library Pandas 2.2.3*

Pandas merupakan *library python* yang berfungsi untuk membaca, membersihkan, memanipulasi, dan menganalisis data secara efisien. *Library* ini mendukung berbagai format data termasuk *jsonl*, yang dapat dikonversi ke dalam struktur *DataFrame* untuk memudahkan proses analisis dan penyajian informasi pada data (Hermanto dkk., 2022).

d. *Library NumPy 1.26.4*

NumPy merupakan *library python* yang berfungsi untuk melakukan komputasi numerik secara efisien. *NumPy* menghadirkan struktur data *array* multidimensi yang mendukung pelaksanaan operasi matematika secara cepat. *Library* ini digunakan untuk melakukan perhitungan statistik, operasi matematika, serta pengolahan data numerik (Hermanto dkk., 2022).

e. *Library Matplotlib 3.7.5*

Matplotlib adalah *library python* yang digunakan untuk membuat visualisasi data dalam bentuk grafik dan plot. *Matplotlib* memungkinkan pembuatan berbagai jenis grafik seperti garis, batang, sebar, dan histogram secara fleksibel dan dapat disesuaikan untuk keperluan analisis data maupun presentasi (Hermanto dkk., 2022).

f. *Library Seaborn 0.12.2*

Seaborn adalah *library Python* berbasis *Matplotlib* untuk visualisasi data statistik. *Library* ini menyediakan fitur yang kaya untuk menghasilkan grafik yang informatif dan menarik, seperti heatmap dan diagram batang, guna

mendukung analisis data secara mendalam (Hermanto dkk., 2022).

g. *Library TensorFlow 2.18.0*

TensorFlow adalah *library open source* yang dikembangkan oleh tim *Google Brain* untuk mendukung komputasi numerik serta pengembangan model *machine learning* dan *deep learning* skala besar. *Library* ini menyediakan API berbasis *Python* sebagai antarmuka *front-end* untuk membangun dan menjalankan model secara efisien dan optimal pada berbagai perangkat keras, sekaligus mempercepat proses pelatihan melalui pemanfaatan kemampuan komputasi paralel pada CPU, GPU, dan TPU (Muharram dan Suryadi, 2022).

h. *Library Transformers 4.51.1*

Transformers merupakan *library Python* yang dikembangkan oleh *Hugging Face* dan menyediakan ribuan model AI pra-latih (*pretrained*) yang bersifat *open-source*. *Library* ini memungkinkan pengguna untuk mengunduh, menyesuaikan, dan menjalankan model untuk berbagai tugas, seperti pemrosesan bahasa alami (NLP), visi komputer, pengolahan audio, hingga pembelajaran multimodal. Penggunaan model pra-latih ini membantu pengguna menghemat biaya, mempercepat proses pelatihan, serta memberikan fleksibilitas lebih dalam membangun dan menyesuaikan model sesuai kebutuhan (Hoffman, 2024).

i. *Library Datasets 3.5.0*

Library Datasets dari *Hugging Face* berperan dalam mempersiapkan dan mengelola data agar siap digunakan dalam proses pelatihan model pembelajaran mesin. *Library* ini memungkinkan konversi data dari *Pandas DataFrame* dan dirancang untuk menangani dataset berukuran besar secara efisien melalui fitur seperti *memory-mapping* dan *caching* otomatis. Data yang telah diolah dapat langsung dimanfaatkan untuk proses *transfer learning* menggunakan model-model *pre-trained* seperti BERT, GPT, atau T5, agar dapat disesuaikan dengan tugas spesifik seperti pembuatan ringkasan teks (Khanna, 2021).

j. *Library tqdm 4.67.1*

Tqdm adalah *library python* yang berfungsi untuk menampilkan progres bar selama proses iterasi (*loop*), tugas berdurasi panjang, atau operasi yang memerlukan waktu eksekusi yang lama. Fitur ini memungkinkan pemantauan proses secara *real-time* dengan menyajikan informasi seperti jumlah langkah yang telah diselesaikan, kecepatan pemrosesan, serta estimasi waktu yang tersisa (Farnschlader, 2024).

k. *Library Rouge Score 0.1.2*

Rouge Score adalah *library Python* yang dirancang untuk menghitung metrik

ROUGE (*Recall Oriented Understudy for Gisting Evaluation*). Fungsi utamanya adalah mengevaluasi kesamaan antara teks prediksi (*prediction*) dan teks referensi (*ground truth*) berdasarkan *unigram*, *bigram*, dan *longest common subsequence* (LCS) (Mamdouh, 2023).

1. *Library Scikit Learn* 1.2.2

Scikit-learn adalah *library Python* yang digunakan untuk pembelajaran mesin dan analisis data. *Scikit-learn* menyediakan berbagai algoritma untuk klasifikasi, regresi, klustering, reduksi dimensi, serta alat bantu untuk praproses data dan evaluasi model (Jain, 2024).

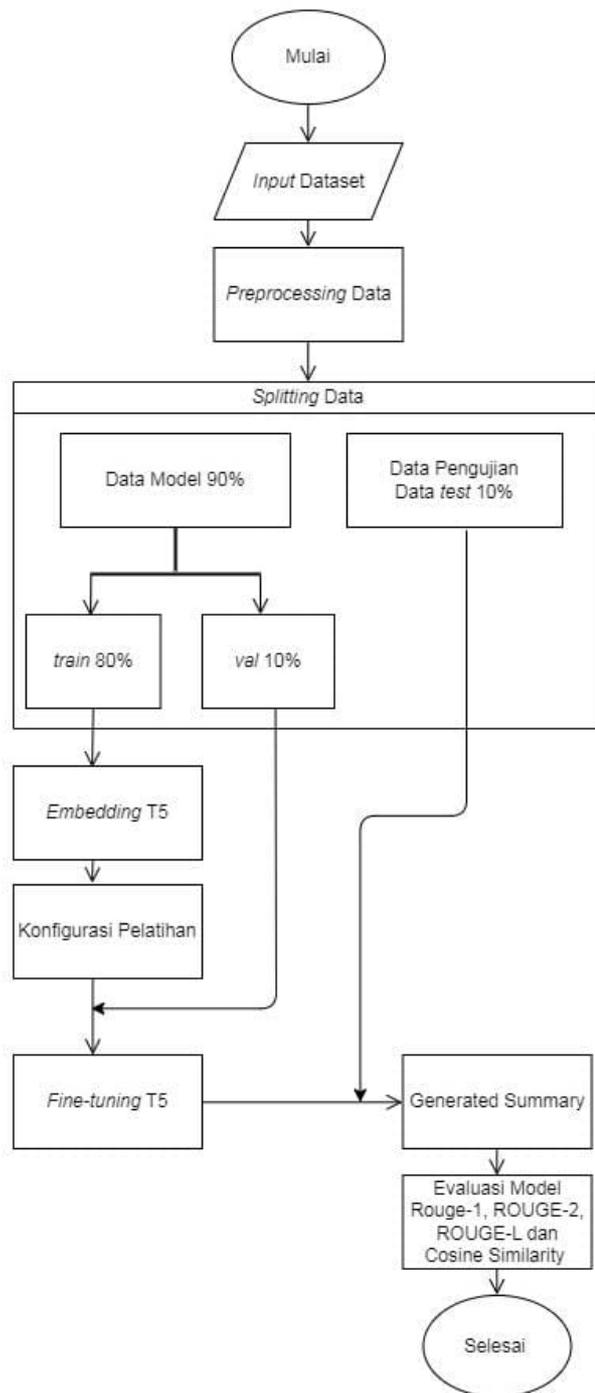
3.3 Metode Penelitian

Terdapat delapan tahapan dalam melakukan ringkasan teks berita berbahasa Indonesia dengan model *fine-tuning* T5 dan diilustrasikan pada Gambar 10.

1. Tahap pertama adalah melakukan *input* dataset *IndoSum*. Pada *notebook Kaggle*, proses ini dilakukan melalui bagian *input* yang terletak di sebelah kanan atas tampilan *notebook*. Dataset *IndoSum* sudah tersedia di *Kaggle* dan dapat langsung ditambahkan tanpa perlu diunggah secara manual. Dataset ini terdiri dari 18774 data, yang dibagi menjadi dua bagian untuk keperluan penelitian ini sebanyak 9837 data. Setelah proses pembagian selesai, data ditampilkan dalam bentuk tabel.
2. Tahap kedua, melakukan *preprocessing* pada dataset *IndoSum*. Pada tahap ini dilakukan beberapa langkah: Pertama melakukan seleksi fitur merupakan tahap pemilihan atribut (variabel) yang digunakan. Atribut yang digunakan pada penelitian ini adalah *paragraphs* dan *summary*. Selain dari dua atribut tersebut, atribut lainnya dihapus. Kedua melakukan rekonstruksi *token* merupakan proses menggabungkan *token* yang terpisah untuk memastikan makna teks tetap utuh. Ketiga melakukan reduksi *noise* merupakan proses menghilangkan spasi berlebih atau tanda baca yang tidak penting. Terakhir, melakukan *Lowercasing* merupakan proses mengubah semua huruf dalam teks menjadi huruf kecil untuk konsistensi dan efisiensi pemrosesan model.
3. Tahap ketiga, melakukan *splitting* data yang terdiri dari data model sebesar 90% dan data pengujian sebesar 10%. Data model dibagi menjadi dua yang terdiri dari data *training* sebesar 80% dan data *validation* sebesar 10% dari data *training*.
4. Tahap keempat, melakukan *embedding* menggunakan model T5 pada data.

Proses *embedding* T5 mengubah teks menjadi *token ID* (angka) menggunakan *sentencepiece tokenizer*, yang membagi teks menjadi *subword*. Pada tahap ini diterapkan *padding* dan *truncation* untuk menyesuaikan panjang *input*. Hasilnya, teks siap diproses untuk *fine-tuning* T5 dengan bantuan *attention mask* agar model dapat fokus pada bagian penting dari *input*.

5. Tahap kelima, melakukan pengaturan konfigurasi pelatihan pada data *training*. Pada tahap ini dilakukan penyesuaian model dan *hyperparameter* yang digunakan untuk pelatihan. Model yang dipakai adalah *T5 base indonesian summarization cased*. Pengaturan *hyperparameter* meliputi penentuan ukuran *batch size* sebesar 8, *epoch* sebanyak 3 dan 5, *dropout* sebesar 0,1 dan 0,2, serta tiga kombinasi *learning rate* yaitu 1×10^{-4} , 1×10^{-5} , dan 1×10^{-6} . Tahap ini juga mencakup pengaturan *optimizer* AdamW dengan *weight decay* sebesar 0,01.
6. Tahap keenam, melakukan *fine-tuning* model menggunakan data model. Pada tahap ini ditampilkan nilai *train loss*, *validation loss*, serta waktu yang diperlukan selama proses *fine-tuning*. Nilai *train loss* dan *validation loss* disajikan dalam bentuk grafik garis untuk memperlihatkan perkembangan selama pelatihan.
7. Tahap ketujuh, setelah melakukan model *fine-tuning*, langkah selanjutnya adalah menghasilkan *generated summary* (ringkasan yang dihasilkan) pada data *testing*. Pada tahap ini, model T5 yang telah dilatih akan digunakan untuk menghasilkan ringkasan otomatis dari teks *input* pada data pengujian. Model memproses teks *input* dari data pengujian dan menghasilkan ringkasan prediksi yang relevan berdasarkan pelatihan yang telah dilakukan.
8. Tahap kedelapan, melakukan evaluasi model dari hasil *generated summary* dengan perhitungan nilai ROUGE *recall*, *precision*, dan F1-score yang terdiri dari ROUGE-1, ROUGE-2, dan ROUGE-L. Selain itu, melakukan perhitungan metrik *cosine similarity*.
9. Membuat kesimpulan dari hasil yang telah didapat dari penelitian dan memberi saran untuk penelitian selanjutnya.



Gambar 10. Diagram Alir Penelitian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini menunjukkan bahwa penerapan *optimizer* AdamW dalam proses pelatihan model T5 untuk tugas peringkasan teks berita berbahasa Indonesia memberikan hasil yang cukup baik. Beberapa poin penting yang dapat disimpulkan dari penelitian ini adalah:

1. Kinerja model mengalami peningkatan

Dibandingkan dengan penggunaan *optimizer* Adam pada penelitian sebelumnya, AdamW memberikan hasil evaluasi yang lebih tinggi. Nilai ROUGE-1 meningkat dari 0,61 menjadi 0,7336 dan ROUGE-2 dari 0,51 menjadi 0,6632. Hal ini menunjukkan bahwa AdamW lebih efektif dalam membantu model memahami dan memberikan informasi penting dari teks.

2. Hasil ringkasan yang relevan

Nilai ROUGE-1 *f1-score* sebesar 0,7182, ROUGE-2 sebesar 0,6493, dan ROUGE-L sebesar 0,6861 menunjukkan bahwa model cukup baik dalam mengambil informasi penting dari teks, baik dari kata-kata maupun susunan kalimatnya. Selain itu, nilai *cosine similarity* sebesar 0,69 menunjukkan bahwa ringkasan yang dibuat oleh model memiliki makna yang cukup mirip dengan ringkasan aslinya. Hasil ini menunjukkan bahwa model mampu memahami isi dan konteks teks dengan baik.

Secara keseluruhan, penerapan AdamW pada model T5 terbukti dapat mendukung proses peringkasan teks secara otomatis dengan cukup baik, meskipun masih ada ruang untuk perbaikan dan pengembangan penelitian selanjutnya.

5.2 Saran

Penelitian selanjutnya disarankan memperluas penggunaan dataset yang tidak hanya terbatas pada *IndoSum*, tetapi juga melibatkan dataset lain yang lebih beragam, baik dari segi topik, gaya penulisan, maupun jenis teks, seperti berita internasional, artikel ilmiah, atau opini. Hal ini penting agar model memiliki kemampuan generalisasi yang lebih baik dan tidak hanya unggul dalam satu domain tertentu. Selain itu, penggunaan dataset yang lebih besar dan bervariasi juga dapat membantu meningkatkan kemampuan model dalam memahami struktur kalimat dan konteks yang lebih kompleks.

Pada *hyperparameter*, eksplorasi pengaturan yang lebih optimal, seperti variasi *learning rate*, jumlah *epoch*, *batch size*, serta teknik regularisasi lain selain *weight decay* sangat dianjurkan. Penyesuaian *hyperparameter* yang tepat berpotensi memberikan dampak signifikan terhadap kualitas ringkasan yang dihasilkan, terutama dalam menghindari *overfitting* atau *underfitting*.

Pada sisi model, meskipun T5 menunjukkan hasil yang baik, evaluasi terhadap kinerja model lain yang juga berbasis transformer, seperti BART, Pegasus, atau mT5, yang telah dirancang khusus untuk tugas peringkasan atau bersifat multilingual, sangat disarankan. Perbandingan antar model ini dapat memberikan gambaran yang lebih menyeluruh mengenai model mana yang paling sesuai dengan karakteristik bahasa Indonesia dan jenis teks tertentu. Selain itu, dilakukan analisis secara lebih mendalam terhadap masing-masing kategori berita. Hal ini penting untuk mengidentifikasi topik-topik yang memiliki karakteristik teks tertentu yang dapat memengaruhi kinerja model, seperti tingkat formalisme bahasa atau panjang dokumen.

Penggabungan teknik ekstraktif dan abstraktif secara *hybrid* dapat diarahkan agar menghasilkan ringkasan yang lebih informatif dan natural. Evaluasi model sebaiknya tidak hanya bergantung pada metrik otomatis seperti ROUGE dan *cosine similarity*, tetapi juga melibatkan evaluasi manual dari sisi keterbacaan, koherensi, dan relevansi, agar hasil ringkasan benar-benar sesuai dengan kebutuhan pengguna.

Memperhatikan aspek-aspek tersebut diharapkan dapat menghasilkan model peringkasan teks yang lebih akurat dan relevan .

DAFTAR PUSTAKA

- Alsharman, N., J., dan Pivkina, I. 2020. Generating Summaries Through Unigram and Bigram: Text Summarization. *Int. J. Inf. Technol. Web Eng.*, **15**(1): 64-74.
- Amna, Wahyuddin, Sudipa, I. G. I., Putra, T. A. E., Wahidin, A. J., Syukrilla, W. A., Wardhani, A. K. A., Heryana, N., Indryani, T., dan Santoso, L. W. 2023. *Data Mining*. Sumatera Barat: PT Global Eksekutif Teknologi.
- Bahari, A., dan Dewi, K. E. 2024. Peringkasan Teks Otomatis Abstraktif Menggunakan Transformer Pada Teks Bahasa Indonesia. *Jurnal Ilmiah Komputer dan Informatika*. **13**(1): 83–91.
- Corcoran, P., Palmer, G., Arman, L., Knight, D., dan Spasić, I. 2021. Creating Welsh Language Word Embeddings. *Applied Science*.11, 6896.
- Farnschlader, T. 2024. *Tqdm Python: A Guide With Practical Examples*. Datacamp. <https://www.datacamp.com/tutorial/tqdm-python>. Diakses pada 27 Mei 2025.
- Faturohman, F., Irawan, B., dan Setianingsih, C. 2020. Analisis Sentimen pada BPJS Kesehatan Menggunakan Recurrent Neural Network. *e-Proceeding of Engineering*. **7**(2): 4545–4552.
- Fiarni, C., Maharani, H., dan Calista, N. 2019. Perancangan Sistem Rekomendasi Produk Menggunakan Metode Cosine Similarity dan Content-based Filtering. *IJITEE*. **3**(2): 42-48.
- Ghawi, R., dan Pfeffer, J. 2019. Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity. *Open Computer Science*. **9**(1): 160–180.
- Gusra, T. D., Tan, D. T., Cangniago, R., dan Christanti, V. 2024. Peringkasan Berita Otomatis Berbasis Website dengan Metode Text Rank. *Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika* . **6**(2): 139–144.
- Hermanto, K., Salim, D., Wu, B., Salim, O. R., dan Gunadi, R. B. 2022. Penggunaan *Python* Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa Pandemi. *Journal of Student Development Information System*. **2**(2): 120-133.
- Hikmaturokhman, A., Nafi'ah, H., dan Larasati, S. 2022. Deep Learning Algorithm Models for Spam Identification on Cellular Short Message Service. *Journal of Communications*. **17**(9): 769-776.

- Hoffman, H. 2024. *Hugging Face Transformers: Leverage Open-Source AI in Python*. Real Python. <https://realpython.com/huggingface-transformers/>. Diakses pada 27 Mei 2025.
- Idhafi, Z., Agustian, S., Yanto, F., dan 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.
- Ivan, E. & Purnomo, H.D. 2022. Forecasting Prices of Fertilizer Raw Materials Using Long Short Term Memory. *Jurnal Teknik Informatika (Jutif)*. **3**(6): 1663–1673.
- Jain, K. 2024. *Scikit-learn(sklearn) in Python – the most important Machine Learning tool I learnt last year!*. Analytics Vidya. <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>. Diakses pada 27 Mei 2025.
- Joshi, A., Fidalgo, E., Alegre, E., dan Fernández-Robles, L. 2019. SummCoder: An Unsupervised Framework For Extractive Text Summarization Based On Deep Auto-encoders. *Expert Systems with Applications*. **129**: 200–215.
- 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 27 Mei 2025.
- Laksana, M. D. B., Karyawati, A. E., Putri, L. A. A. P., Santiyasa, I. W., Sanjaya, N. A., dan Kadryanan, I. G. A. G. A. 2022. Text Summarization Terhadap Berita Bahasa Indonesia Menggunakan Dual Encoding. *Jurnal Elektroknik Ilmu Komputer Udayana*. **11**(2): 339–348.
- 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). *Association for Computational Linguistics*. <https://aclanthology.org/W04-1013/>.
- Mamdouh, M. 2023. *Mastering ROUGE Matrix: Your Guide to Large Language Model Evaluation for Summarization with Examples*. DEV. <https://dev.to/aws-builders/mastering-rouge-matrix-your-guide-to-large-language-model-evaluation-for-summarization-with-examples-jjg>. Diakses pada 27 Mei 2025.
- Muhharam, R. F., dan Suryadi, A. 2022. Implementasi artificial intelligence untuk deteksi masker secara realtime dengan tensorflow dan ssdmobilenet Berbasis python. *Jurnal Widya*. **3**(2): 281-290.
- Nurhaki, J., dan Yahfizham, Y. 2024. Studi Kepustakaan: Pengenalan 4 Algoritma Pada Pembelajaran Deep Learning Beserta Implikasinya . *Jurnal Pendidikan Berkarakter*. **2**(1): 270-281.

- Nurwanda, Suarna, N., dan Prihatono, W. 2024. Penerapan NLP (Natural Language Processing) Dalam Analisis Sentimen Pengguna Telegram di Playstore. *Jurnal Mahasiswa Teknik Informatika*. **8**(2): 1841–1846.
- Ong, R. J., Ruchiat, A. R., Mahendra, A. R., Amaliah, I. D., dan Yudistira, N. 2021. Text Summarization Dengan Menggunakan BERT Dengan Data Berita Indonesia. *Jmabura Journal of Probability and Statistics*. **2**(2), 66–78.
- Pasaribu, D. J. M., Kusriani, K., dan Sudarmawan, S. 2020. Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan Bidirectional LSTM dan BERT Embedding. *Jurnal Teknologi Informasi dan Komunikasi*. **10**(1): 9–20.
- Pradana, A. G., Setiadi, D. R. I., dan Mushlik, A. R. 2024. Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao. *Journal of Information System and Application Development*. **2**(1): 1–11.
- Purnama, I., dan Utami, N. D. 2023. Penerapan Model T5 untuk Peringkasan Teks Bahasa Indonesia pada Dataset IndoSum. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **10**(1): 45–59.
- Purwitasari, N. A., dan Soleh, M. 2022. Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Processing. *Jurnal IPTEK*. **6**(1): 14–21.
- Rumaisa, F., Puspitarani, Y., Rosita, A., Zakiah, A., dan Violina, S. 2021. Penerapan Natural Language Preprocessing di Bidang Pendidikan. *Jurnal Inovasi Masyarakat*. **1**(3): 232–235.
- Sharma, O. 2019. Deep Challenges Associated with Deep Learning. Dalam *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects (COMITCon 2019)*, hlm. 72–75.
- Shaveta. 2023. A review on machine learning. *International Journal of Science and Research Archive*. **9**(1): 281–285.
- Sorin, V., Barash, Y., Konen, E., dan Klang, E. 2020. Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review. *Journal of the American College of Radiology*. **17**(5): 639–648.
- Sari, R. M., Hidayat, Y. A., Wibowo, A. (2024). Sentiment analysis of Twitter user comments using Naïve Bayes classifier. *E3S Web of Conferences*. **448**:2-10. <https://doi.org/10.1051/e3sconf/202344802025>.
- Uslu, C. 2022. *Kaggle Datasets Tutorial: Kaggle Notebooks*. Datacamp. <https://www.datacamp.com/tutorial/tutorial-kaggle-datasets-tutorials-kaggle-notebooks>. Diakses pada 27 Mei 2025.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., dan Polosukhin, I. 2017. Attention is all you need. *In Advances in Neural Information Processing Systems*. **30**: 5998–6008.
- Wang, M., Xie, P., Du, Y., dan Hu, X. 2023. T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions. *Applied Sciences*, 13(7111).
- Wibawa, M. S. 2016. Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah *Epoch* Terhadap Performa Jaringan Saraf Tiruan. *Jurnal Sistem dan Informatika*. **11**(1): 1-8.