MARIANMT BERBASIS TRANSFORMER UNTUK PENERJEMAHAN BAHASA INGGRIS-INDONESIA DENGAN FINE-TUNING DATA *TECHNOLOGY, EDUCATION, AND DESIGN* (TED) *TALKS*

Skripsi

Oleh

ADINDA PUTRI HERMAWAN
NPM. 2117031086

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

ABSTRACT

TRANSFORMER-BASED MARIANMT FOR ENGLISH-INDONESIA TRANSLATION WITH FINE-TUNING DATA TECHNOLOGY, EDUCATION, AND DESIGN (TED) TALKS

By

Adinda Putri Hermawan

In the era of globalization, the need for accurate automatic translators is increasing to support cross-language communication, especially from English to Indonesian. This study aims to evaluate the effectiveness of fine-tuning on the transformer-based MarianMT model using formal TEDTalks data, and analyze the influence of question marks in question sentences on translation quality. The methods used include data pre-processing, division of the dataset into five scenarios of training proportion and validation, model hyperparameter adjustment, and application of fine-tuning techniques with early-stopping and dropout to avoid overfitting. Evaluation was performed using the BLEU score on the test data. The results show that fine-tuning with in-domain data can improve translation accuracy, indicated by the highest BLEU gain of 32.14. This study also shows that the removal of question marks affects the meaning of translation, especially in short and medium sentences. The 90:10 (training:validation) data proportion scenario produced the best performance with good loss value stability. Thus, a combination of fine-tuning and proper pre-processing settings can produce more accurate and contextually appropriate translations.

Keywords: Neural Machine Translation, Transformer, MarianMT, Fine-Tuning, TEDTalks, Language Formality, BLEU Score

ABSTRAK

MARIANMT BERBASIS TRANSFORMER UNTUK PENERJEMAHAN BAHASA INGGRIS-INDONESIA DENGAN FINE-TUNING DATA *TECHNOLOGY, EDUCATION, AND DESIGN* (TED) *TALKS*

Oleh

Adinda Putri Hermawan

Pada era globalisasi, kebutuhan akan penerjemah otomatis yang akurat semakin meningkat untuk mendukung komunikasi lintas bahasa, khususnya dari bahasa Inggris ke bahasa Indonesia. Penelitian ini bertujuan untuk mengevaluasi efektivitas *fine-tuning* pada model MarianMT berbasis *transformer* menggunakan data TEDTalks yang bersifat formal, serta menganalisis pengaruh tanda tanya dalam kalimat tanya terhadap kualitas terjemahan. Metode yang digunakan meliputi *pre-processing* data, pembagian dataset ke dalam lima skenario proporsi pelatihan dan validasi, penyesuaian *hyperparameter* model, serta penerapan teknik *fine-tuning* dengan *early-stopping* dan *dropout* untuk menghindari *overfitting*. Evaluasi dilakukan menggunakan BLEU *score* pada data uji. Hasil penelitian menunjukkan bahwa *fine-tuning* dengan data *in-domain* dapat meningkatkan akurasi terjemahan, ditunjukkan oleh perolehan BLEU tertinggi sebesar 32,14. Penelitian ini juga menunjukkan bahwa penghapusan tanda tanya memengaruhi makna terjemahan, khususnya pada kalimat pendek dan sedang. Skenario proporsi data 90:10 (pelatihan:validasi) menghasilkan kinerja terbaik dengan stabilitas nilai loss yang baik. Dengan demikian, kombinasi *fine-tuning* dan pengaturan *pre-processing* yang tepat dapat menghasilkan terjemahan yang lebih akurat dan sesuai konteks.

Kata-kata kunci: *Neural Machine Translation*, *Transformer*, MarianMT, *Fine-Tuning*, TEDTalks, Keformalan Bahasa, BLEU *Score*.

MARIANMT BERBASIS TRANSFORMER UNTUK PENERJEMAHAN BAHASA INGGRIS-INDONESIA DENGAN FINE-TUNING DATA *TECHNOLOGY, EDUCATION, AND DESIGN* (TED) *TALKS*

ADINDA PUTRI HERMAWAN

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	:	**MARIANMT BERBASIS TRANSFORMER UNTUK PENERJEMAHAN BAHASA INGGRIS-INDONESIA DENGAN FINE-TUNING DATA *TECHNOLOGY, EDUCATION, AND DESIGN* (TED) *TALKS***
Nama Mahasiswa	:	*Adinda Putri Hermawan*
Nomor Pokok Mahasiswa	:	**2117031086**
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 : **Prof. Dra. Wamiliana, M.A., Ph.D.**

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

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

Tanggal Lulus Ujian Skripsi:**16 Juni 2025**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama	:	**Adinda Putri Hermawan**
Nomor Pokok Mahasiswa	:	**2117031086**
Jurusan	:	**Matematika**
Judul Skripsi	:	**MarianMT Berbasis *Transformer* untuk Penerjemahan Bahasa Inggris-Indonesia dengan *Fine-Tuning* Data *Technology, Education, and Design* (TED) *Talks***

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 degnan ketentuan akademik yang berlaku.

Bandar Lampung, 16 Juni 2025

Penulis,

Adinda Putri Hermawan

RIWAYAT HIDUP

Penulis memiliki nama lengkap Adinda Putri Hermawan lahir pada tanggal 28 Januari 2004. Penulis merupakan anak sulung dari tiga bersaudara, putri dari pasangan Bapak Rahmat Hermawan dan Ibu Titin Kurniasih. Penulis memiliki seorang adik perempuan bernama Andara Putri Hermawan dan adik laki-laki bernama Ghazali Al Buchori Hermawan.

Penulis memulai pendidikan formal dari jenjang PAUD dan taman kanak-kanak pada tahun 2007, kemudian melanjutkan ke pendidikan dasar yang diselesaikan pada tahun 2015. Selanjutnya, penulis menempuh pendidikan di SMP Negeri 1 Purbolinggo dari tahun 2015 hingga 2018 dan melanjutkan ke jenjang menengah atas di SMA Negeri 1 Purbolinggo hingga lulus pada tahun 2021.

Pada tahun 2021 penulis diterima sebagai mahasiswi Program Studi S1 Matematika di Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur SBMPTN. Selama masa studi, penulis aktif mengikuti kegiatan akademik dan turut serta dalam berbagai kegiatan organisasi kemahasiswaan. Pada tahun 2023, penulis dipercaya menjadi Staf Ahli di Dinas Media dan Informasi (MEDINFO) Badan Eksekutif Mahasiswa Fakultas MIPA Universitas Lampung sebagai bentuk kontribusi dalam pengembangan media dan komunikasi internal kampus.

Selama Desember 2023 hingga Februari 2024, penulis menjalani Kerja Praktik (KP) di Badan Pusat Statistik (BPS) Kabupaten Lampung Timur untuk memahami penerapan teori di dunia kerja. Penulis juga mengikuti Kuliah Kerja Nyata (KKN) selama 40 hari di Desa Braja Yekti, Kecamatan Braja Slebah sebagai bentuk kontribusi kepada masyarakat. Melengkapi pengalaman tersebut, penulis mengikuti program Magang dan Studi Independen Bersertifikat (MSIB) Bangkit yang merupakan bagian dari inisiatif Kampus Merdeka dengan fokus pada pengembangan kompetensi di bidang teknologi melalui pembelajaran *machine learning* dan pengerjaan proyek berbasis solusi digital berskala nasional.

KATA INSPIRASI

”Allah tidak membebani seseorang melainkan sesuai dengan kemampuannya.”
(QS. Al Baqarah: 286)

”Dan di atas setiap orang yang berpengetahuan masih ada yang Maha Mengetahui.”
(QS. Yusuf: 76)

”The act of wanting to pursue something maybe even more precious than actually becoming that thing. So I feel like just like being in the process itself is a prize and so you shouldn’t think of it as a hard way and even if you do get stressed out. You should think of it as happy street kind of and so just enjoy while pursuing it because it’s that precious.”
(Mark Lee)

“You’re normal, you know? You’re doing fine. Sometimes you’re doing better. Sometimes you’re doing worse, but at the end it’s you. I just want you to have no regrets, i want you to feel yourself grow and just to love yourself.”
(Mark Lee)

“Never let anyone, including yourself, belittle your dream.”
(Joshua)

”Long story short i survived.”
(Taylor Swift)

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 terimakasih saya kepada:

Ayah dan Mama Tersayang

Dengan segala kerendahan hati, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Ayah dan Mama tersayang atas segala bentuk kasih sayang, pengorbanan, serta doa yang tiada henti mengiringi setiap langkah dalam menempuh pendidikan ini. Pesan Ayah "bisa karena terbiasa", serta pesan Mama "keluhan melemahkan langkah, syukur menguatkan tekad" senantiasa menjadi penyemangat dan pedoman hidup dalam menghadapi berbagai tantangan.

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

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul ”MarianMT Berbasis *Transformer* untuk Penerjemahan Bahasa Inggris-Indonesia dengan *Fine-Tuning* Data *Technology, Education, and Design* (TED) *Talks*” 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 1 yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibu Prof. Dra. Wamiliana, M.A.,Ph.D. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Ibu Dr. Notiragayu, S.SI.,M.SC. selaku dosen pembimbing akademik.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

8. Kedua orang tua, Rahmat Hermawan dan Titin Kurniasih yang senantiasa menjadi sumber semangat, tak henti menyertai dalam doa, memberi dorongan di saat lelah, menyayangi sepenuh hati, serta mencurahkan kasih yang tulus kepada penulis hingga mampu menyelesaikan skripsi ini.
9. Adik-adikku tersayang (Dara dan Al) yang telah menjadi sumber semangat bagi penulis dalam menyelesaikan skripsi ini. Kehadiran kalian mengingatkan penulis akan tanggung jawab untuk menjadi contoh yang baik sebagai seorang kakak. Semoga perjuangan ini bisa menjadi inspirasi dan motivasi bagi kalian dalam menapaki langkah kalian sendiri.
10. Sahabat-sahabatku tersayang (Maya, Dita, Disya, Tasya, Nabila, Eva, Mei, Rahma, Sherina, Lusiana, Fathan, Ica, dan Muti) yang turut mewarnai perjalanan akademik penulis dengan semangat dan kebersamaan, serta menjadi penguat di setiap fase perkuliahan hingga tugas akhir ini terselesaikan.

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, 16 Juni 2025

Adinda Putri Hermawan

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada era globalisasi yang semakin berkembang membuka tuntutan dan kesempatan pada setiap orang untuk berkomunikasi secara internasional. Pentingnya penerjemah otomatis dari bahasa Inggris ke bahasa Indonesia sangat relevan mengingat bahasa inggris merupakan bahasa internasional yang digunakan dalam berbagai bidang (Siregar, 2023). Perkembangan teknologi terhadap *machine translation* (MT) memberikan dampak yang besar dalam pengetahuan karena dengan penerjemah otomatis membantu menjembatani kesenjangan bahasa, memungkinkan lebih banyak orang Indonesia mengakses literatur, penelitian, dan informasi internasional sehingga dibutuhkan mesin terjemahan otomatis yang akurat. Namun, penerjemah otomatis sering menghadapi kesulitan dalam menangkap konteks sosial, khususnya dalam membedakan gaya bahasa formal dan informal. Pada konteks bahasa Indonesia, hal tersebut menjadi penting dikarenakan perbedaan formalitas dapat mempengaruhi makna dan penerimaan pesan. Oleh karena itu, untuk mengatasi permasalahan tersebut dapat dilakukan melalui *fine-tuning* pada dataset spesifik.

Fine-tuning merupakan salah satu hal penting dalam *Natural Language Processing* (NLP) yang secara umum dilakukan pada model yang sudah dilatih (*pre-trained*) dengan data umum (Hafifah, *et al.*, 2021). Salah satu teknik yang biasa digunakan dalam konteks penerjemah bahasa, yaitu *domain-specific fine-tuning* yang dapat menghasilkan model bahasa yang lebih tangguh (Tinn, *et al.*, 2023). *Fine-tuning* muncul karena model *pre-trained* sering kali tidak cukup akurat ketika diterapkan pada domain tertentu. Sebagai contoh, model *pre-trained* yang dilatih dengan dataset umum mungkin kurang tepat dalam menangani bahasa formal di bidang komunikasi profesional. Pada konteks ini, *fine-tuning* memberikan cara untuk mengoptimalkan model tersebut agar lebih relevan dengan domain atau gaya bahasa yang diinginkan. Pada model *pre-trained* dilakukan pelatihan tambahan menggunakan data corpus

yang lebih khusus atau berbeda dari data corpus awal untuk disesuaikan dengan *domain-specific* yang lebih terarah (Hafifah, *et al.*, 2021). Proses *fine-tuning* melibatkan pembaruan parameter model menggunakan dataset yang lebih kecil dan terfokus. Pada teknik ini, model disesuaikan dengan dataset yang spesifik untuk domain atau gaya bahasa tertentu, seperti TEDTalks yang cenderung memiliki struktur formal.

Technology, Education, and Design (TED), pertama kali muncul pada tahun 1984 dan mulai dikenal luas pada 2006 dengan video presentasinya yang dibagikan secara gratis di internet sebagai TEDTalks. Presentasi ini berdurasi maksimal 18 menit dengan menyampaikan gagasan yang dianggap edukatif dan disebarkan secara ringkas, menarik, dan mudah dipahami. Topik yang diangkat pun beragam mulai dari pendidikan, teknologi, hingga isu kemanusiaan. TEDTalks banyak disukai oleh berbagai kalangan karena bentuk pembelajaran digital yang fleksibel dan padat makna. Pendekatan TED dapat diintegrasikan ke dalam pembelajaran sebagai inspirasi dalam pengembangan keterampilan presentasi mahasiswa dan bisa menjadi referensi bagi pendidik dalam menyampaikan informasi secara efektif dan menarik di era digital (Romanelli, *et al.*, 2014).

Perkembangan teknologi dalam bidang NLP terutama dalam MT telah mengalami kemajuan yang signifikan dalam beberapa tahun terakhir. *Natural language processing* memanfaatkan teknologi komputer untuk mempelajari, mengenali, serta memperdalam pemahaman terhadap bahasa manusia, sehingga dapat membantu dalam menerjemahkan teks dari satu bahasa ke bahasa lain (Ghirrid, *et al.*, 2024). Sebelumnya, teknologi MT dikembangkan sebagai perangkat lunak bahasa berbasis aturan, dimana terjemahan dilakukan berdasarkan aturan tata bahasa dan kosakata yang telah ditentukan sebelumnya kemudian bertahap beralih ke pendekatan statistik yang dikenal dengan *statistical machine translation* (SMT) (Rehm, *et al.*, 2018). Namun, seiring perkembangan teknologi *deep learning* membawa perubahan besar dalam bidang MT dengan memperkenalkan pendekatan *neural machine translation* (NMT). Berbeda dengan metode sebelumnya, NMT memanfaatkan jaringan saraf tiruan yang mampu membantu model dalam menangani konteks kalimat panjang sebagai input dengan lebih mendalam (Pertiwi, *et al.*, 2024). Hal ini memungkinkan sistem untuk menghasilkan terjemahan yang lebih akurat dan natural. Inovasi ini membuka jalan bagi perkembangan teknologi penerjemah yang lebih canggih dan mampu menangani kompleksitas bahasa manusia dengan lebih baik.

Salah satu arsitektur yang telah merevolusi NLP adalah *transformer* yang memungkinkan paralelisasi jauh lebih besar dalam pemrosesan data (Baquero, 2023). *Transformer* menjadi arsitekstur tercanggih selama beberapa tahun terakhir, termasuk pada NMT dalam menghasilkan kualitas terjemahan. Implementasi terkenal dari arsitektur *transformer* dalam NMT adalah MarianMT. Model ini dikembangkan oleh Kelompok Riset Teknologi Bahasa Universitas Helsinki (Helsinki-NLP) dengan mengirimkan sekumpulan besar model penerjemah ke pustaka *transformer* yang disebut dengan MarianMT model (Soliman, *et al.*, 2022). Model MarianMT dirancang untuk menerjemahkan banyak bahasa dan telah dilatih dengan OPUS parallel dataset menggunakan pustaka Marian yang mendukung proses pelatihan dan penerjemahan yang lebih cepat dan akurat sehingga memberikan keunggulan signifikan dalam kualitas terjemahan. Meskipun MarianMT telah menunjukkan kinerja yang baik dalam terjemahan umum, *fine-tuning* dengan dataset spesifik dapat meningkatkan kualitas terjemahan pada konteks tertentu. Pendekatan ini memastikan bahwa hasil terjemahan tidak hanya akurat tetapi juga sesuai dengan gaya bahasa yang diharapkan.

Seiring berjalannya waktu, penerapan NMT dengan MarianMT telah dilakukan sebagai penunjang penelitian, seperti penelitian yang dilakukan Miyagawa (2023) yang melakukan penelitian mengenai mesin penerjemah bahasa dengan sumber daya sangat rendah, yaitu bahasa Ainu yang terancam punah di Jepang Utara ke bahasa Jepang dengan menggunakan *fine-tuning* guna menyesuaikan model MarianMT agar mampu menghasilkan terjemahan yang lebih sesuai. Hasil penelitiannya berhasil menunjukkan kelayakan dan kemanjuran MT dalam upaya pelestarian bahasa dengan SacreBLEU 32,90. Kemudian, Dwiastuti (2019) melakukan penelitian mengenai NMT untuk bahasa Indonesia ke bahasa Inggris dan sebaliknya menggunakan Tensor2Tensor (T2T) berbasis transformer dengan memanfaatkan *fine-tunning* dan *back-translation* sebagai *domain adaptation*. Pada penelitiannya diperoleh hasil terjemahan bahasa Inggris ke bahasa Indonesia dengan menggabungkan *in-domain* dan *out-domain* dataset menggunakan *domain adaptation* mendapatkan BLEU sebesar 29,10 dan BLEU sebesar 28,18 untuk terjemahan bahasa Indonesia ke bahasa Inggris. Selain itu, (Maksymenko, *et al.*, 2023) juga membahas bagaimana *fine-tuning* dapat membantu hasil terjemahan yang sesuai untuk terjemahan bahasa Inggris ke bahasa Ukraina dengan memodifikasi MarianMT menggunakan vektor konteks pada hasil penelitiannya diperoleh BLEU sebesar 36,14.

Pemanfaatan model *pre-trained* MarianMT dapat mendukung terjemahan cepat dan akurat dan dapat ditingkatkan kualitasnya melalui *fine-tuning* pada dataset spesifik, seperti data TEDTalks. Data TEDTalks dipilih karena menyajikan gaya bahasa komunikasi sehari-hari yang formal, mudah dipahami dan mencakup berbagai topik mulai dari sains hingga teknologi. Oleh karena itu, penelitian ini akan berfokus pada penerjemahan dari bahasa Inggris ke bahasa Indonesia dengan model MarianMT berbasis transformer dengan *domain-specific fine-tuning* menggunakan data TEDTalks. Pada penelitian ini juga akan dilakukan analisis terhadap pengaruh tanda tanya dalam kalimat tanya untuk mengamati dampaknya terhadap struktur dan makna terjemahan yang dihasilkan. Model diharapkan dapat membedakan karakteristik kalimat tanya dan menghasilkan terjemahan yang sesuai dengan konteks komunikasi profesional dalam skala internasional.

1.2 Rumusan Masalah

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

1. Implementasi model MarianMT berbasis *transformer* dengan *fine-tuning* data TEDTalks untuk mengukur kinerja penerjemahan bahasa Inggris ke bahasa Indonesia dalam mempertahankan tingkat keformalan yang tetap selaras dengan gaya bahasa formal khas TEDTalks menggunakan evaluasi BLEU.
2. Analisis pengaruh keberadaan tanda tanya dalam data pelatihan terhadap hasil terjemahan model MarianMT yang telah melalui proses *fine-tuning*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengevaluasi efektivitas model MarianMT setelah dilakukan penyesuaian parameter dan penerapan teknik *fine-tuning* data TEDTalks, serta menganalisis kualitas hasil terjemahan formal yang dihasilkan berdasarkan evaluasi BLEU pada penerjemah bahasa Inggris ke bahasa Indonesia.
2. Membandingkan model yang dilakukan *fine-tuning* menggunakan data dengan tanda tanya dan model tanpa tanda tanya guna mengidentifikasi sejauh mana perbedaan tahap *pre-processing* tersebut memberikan dampak terhadap kualitas hasil terjemahan.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini:

1. Diperoleh hasil terjemahan formal yang cocok digunakan dalam komunikasi profesional dengan konteks internasional, seperti rapat ataupun presentasi yang dapat membantu memastikan bahwa pesan disampaikan dengan jelas dan tepat, dan mengurangi risiko kesalahpahaman.
2. Memberi informasi bagaimana model menangani kalimat tanya, serta membuka peluang untuk mengembangkan strategi *pre-processing* untuk meningkatkan kualitas penerjemahan teks interogatif.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Analisis Penelitian yang berkaitan dengan penelitian ini digunakan sebagai referensi untuk mengevaluasi kualitas hasil terjemahan menggunakan MT berbasis *transformer*. Secara keseluruhan, ringkasan dari beberapa penelitian yang menjadi referensi dalam penelitian ini terangkum dalam Tabel 1, yang mencakup mengenai penelitian terdahulu terkait NMT berbasis *transformer*.

Tabel 1. Penelitian terkait NMT berbasis *transformer*

No	Penelitian	Data	Metode	Hasil
1.	English Indonesian Neural Machine Translation for Spoken Language Domains Dwiastuti (2019)	Korpus bahasa Inggris-Indonesia OpenSubtitle2018 dengan 9.273.809 baris data dan TEDTalks2017 dengan 111.256 baris data.	Tensor2Tensor (T2T) berbasis *transformer* dengan *Domain Adaptation*	BLEU 29,10 untuk bahasa Inggris ke bahasa Indonesia dan BLEU 28,18 untuk bahasa Indonesia ke bahasa Inggris.
2.	Controllability for English-Ukrainian Machine Translation Based on Specialized Corpora (Maksymenko, *et al.*, 2023)	Subset korpus Opus dataset bahasa Inggris ke bahasa Ukraina dengan jumlah 1.681.349 baris data.	*Transformer* MarianMT dengan memodifikasi semantic dan *stylistic embedding*	BLEU 36.14 pada model yang telah dimodifikasi dan BLEU 34.16 pada model yang tidak dimodifikasi.

No	Penelitian	Data	Metode	Hasil
3.	Machine Translation for Highly Low-Resource Language: A Case Study of Ainu, a Critically Endangered Indigenous Language in Northern Japan Miyagawa (2023)	Sumber teks digital bahasa Ainu dengan 100.000 baris data.	*Transformer* MarianMT menggunakan *early stopping*.	BLEU sebesar 32.90.

2.1.1 Penelitian Pertama (Dwiastuti, 2019)

Dwiastuti (2019) melakukan penelitian mengenai *neural machine translation*(NMT) untuk bahasa Indonesia ke bahasa Inggris dan sebaliknya menggunakan Tensor2Tensor (T2T). Tensor2Tensor adalah salah satu *library open-source* yang berfokus pada penggunaan *transformer*, salah satu arsitektur *deep learning* yang sangat kuat untuk tugas-tugas Seq2Seq seperti terjemahan bahasa. Pada penelitian ini *fine-tunning* dan *back-translation* dimanfaatkan sebagai *domain adaptation*. Dataset yang digunakan dalam pelatihan adalah data OpenSubtitle2018 sebanyak 9.273.809 baris kalimat digunakan sebagai data *out-domain* dan data TEDTalk sebanyak 107.329 baris kalimat digunakan sebagai data *in-domain*. Dataset yang digunakan dalam data validasi adalah Data TEDTalk tst2013 sebanyak 1034 baris kalimat dan TEDTalk tst2014 sebanyak TEDTalk tst2014 sebanyak 878 baris kalimat. Sedangkan, dalam data uji digunakan data TEDTalk tst2015-16 sebanyak 980 baris kalimat dan TEDTalk tst2017-plus-mod sebanyak 1035 baris kalimat.

Penelitian dilakukan menggunakan *transformer* dengan set *hyperparameter "transformer base"*. Model ini terdiri dari 6 lapisan *encoder* dan 6 lapisan *decoder*, dengan ukuran *embedding* (d_{model}) sebesar 512. Pada lapisan *fully-connected*, jumlah unit tersembunyi (d_{ff}) adalah 2048 untuk memastikan kapasitas menangkap hubungan kompleks dalam data. Selain itu, lapisan *multi-head attention* memiliki 8 *head attention* (*h*) dengan ukuran vektor *key/query* (d_k) dan *value* (d_v)

masing-masing sebesar 64, serta menggunakan *learning rate* 0.2, *maximum sequence length* 150, dan ukuran *batch* dengan ukuran 2048 untuk model *fine-tuning*, ukuran 6000 dan 8000 pada *back-translation*. Pada data tidak dilakukan proses tokenisasi dan model dioptimalkan menggunakan *adafactor optimizer*. Pada penelitian ini diperoleh hasil terjemahan bahasa Inggris ke bahasa Indonesia dengan menggabungkan *in-domain* dan *out-domain* dataset menggunakan *domain adaptation* mendapatkan BLEU sebesar 29,10 dan BLEU sebesar 28,18 untuk terjemahan bahasa Indonesia ke bahasa Inggris.

2.1.2 Penelitian Kedua (Maksymenko, *et al*., 2023)

Maksymenko, *et al*., (2023) melakukan penelitian mengenai pengendalian pada penerjemahan mesin bahasa Inggris ke bahasa Ukraina dengan memodifikasi MarianMT arsitektur *encoder-decoder* untuk menggabungkan matriks *embedding* dengan *semantic search embedding* vektor teks asli. Dataset yang digunakan dalam penelitian ini Opus dataset bahasa Inggris ke bahasa Ukraina sebanyak 1.681.349 data sebagai pelatihan sekaligus uji. Penelitian ini menggunakan *transformer* MarianMT sebagai dasar model dengan arsitektur asli 6 lapisan *encoder-decoder*, dimana *encoder* dapat memperoleh 512 token yang mengembalikan matriks *embedding* dengan dimensi 512×512. Pada penelitian ini, modifikasi digunakan Siamese BERT pada format miniLM untuk menangkap fitur teks umum dengan vektor 384 nilai dalam mendeskripsikan domain teks, makna teks, dan gaya pada teks. Vektor tersebut digabungkan pada setiap penyematan token sehingga mendapatkan matriks dengan dimensi 512×896 yang kemudian direduksi menjadi dimensi asli menggunakan lapisan *fully-connected* and aktivasi SELU. Matriks yang telah ditransformasikan digunakan sebagai masukan untuk *decoder* sehingga dengan memodifikasi *semantic* dan *stylistic embedding* dapat mengubah hasil model. Model OpusMT bahasa Inggris ke Ukraina oleh Helsinki NLP digunakan sebagai bobot awal untuk *encoder* dan *decoder* dalam arsitektur yang dimodifikasi. Pada penelitian ini diperoleh hasil BLEU 36.14 pada model yang dimodifikasi dan BLEU 34.16 pada model yang tidak dimodifikasi.

2.1.3 Penelitian Ketiga (Miyagawa, 2023)

Miyagawa (2023) melakukan penelitian mengenai mesin penerjemah bahasa dengan sumber daya sangat rendah, yaitu bahasa Ainu yang terancam punah di Jepang Utara ke bahasa Jepang dengan menggunakan *fine-tuning* guna menyesuaikan model MarianMT agar mampu menghasilkan terjemahan yang lebih sesuai. Data yang digunakan pada penelitian ini adalah 100.000 data yang diperoleh dari beberapa sumber teks digital bahasa Ainu, dimana semua transkip katakana dikonversi ke dalam alphabet romawi. Korpus yang terkumpul kemudian dibersihkan dengan tujuan menghilangkan redundansi dan inkonsistensi dalam kalimat. Setelah itu, membuat token pada data dan mengelompokkannya menjadi pasangan kalimat. Parameter pada model dioptimalkan melalui *learning rate* yang dikombinasikan dengan *early stopping*. Pada penelitian ini, metrik evaluasi BLEU digunakan dalam mengevaluasi kinerja sistem *machine translation*. Hasil penelitian berhasil menunjukan kelayakan *machine translation* dalam upaya pelestarian bahasa dengan hasil BLEU 32,90.

2.2 *Text Mining*

Text mining adalah tahapan yang berfokus pada pengetahuan dengan kumpulan dokumen melalui berbagai alat analisis untuk mengekstraksi informasi yang berguna. Berbeda dari data mining yang bekerja dengan data terstruktur, *text mining* bekerja pada data tidak terstruktur, seperti teks. Proses *text mining* melibatkan identifikasi pola-pola dalam teks dan tahapan *pre-processing* mengubah teks alami menjadi format yang lebih terstruktur (Feldman & Sanger, 2007). Tahapan yang dilakukan dalam *pre-processing text* yang pertama adalah tokenisasi, yaitu membagi bagian-bagian kata menjadi sebuah token. Kemudian, *case folding* dengan mengubah seluruh huruf besar menjadi huruf kecil. *Filtering*, yaitu menghilangkan seluruh mention (@), hastag (#), tanda baca, emotikon, dan karakter non abjad untuk memastikan hanya terdapat huruf pada dokumen. Selanjutnya, *padding* pada *text mining* merupakan proses menambahkan token tertentu ke dalam teks atau urutan data agar semua input memiliki panjang yang sama (Surjandari, *et al.*, 2016).

2.3 *Natural Language Processing* (NLP)

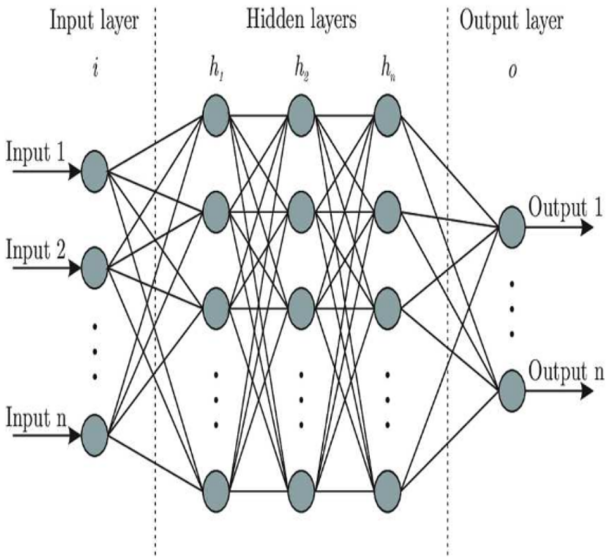
Natural Language Processing merupakan bagian dari *text mining* yang membantu komputer untuk mempelajari, mengenali, dan memahami bahasa manusia dengan data yang kompleks dan tidak terstruktur. Salah satu tugas dari NLP adalah penerjemahan otomatis bahasa yang memungkinkan terciptanya pertukaran kata dalam berbagai bahasa sehingga dapat membantu memberikan wawasan yang luas dalam memahami dunia (Ghirrid, *et al.*, 2024). Berikut merupakan metode yang sering digunakan dalam pemrosesan NLP:

1. Tokenisasi
   Tokenisasi merupakan langkah awal dalam pemrosesan teks, yaitu proses pemisahan kata pada teks menjadi unit yang lebih kecil yang disebut dengan token dan berupa kata atau frasa. Menurut (Kudo, 2018) tokenisasi dengan model sentencePiece merupakan fitur model dengan performa terbaik.
2. *Lemmatization* dan *Stemming*
   Metode ini digunakan untuk mengubah kata-kata menjadi bentuk kata dasar. Pada proses *stemming* dilakukan penghapusan imbuhan dan akhiran kata, sedangkan pada proses *lemmatization* mengubah kata ke bentuk kata dasar berdasarkan kamus kata (Jamaluddin, *et al.*, 2020).
3. Semantik
   Semantik digunakan untuk mempelajari makna suatu kata dan bagaimana makna tersebut berkontribusi dalam membentuk keseluruhan makna dari suatu kalimat (Jamaluddin, *et al.*, 2020).

2.4 Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan metode komputasi dengan cara kerja meniru jaringan saraf manusia yang memiliki kemampuan untuk belajar. Metode ini memecah objek menjadi bagian-bagian kecil yang masing-masing bagian akan terhubung satu sama lain seperti pada neuron sehingga sangat cocok digunakan dalam memecahkan masalah pengenalan pola. Terdapat dua jenis jaringan saraf tiruan, yaitu *supervised* yang mempelajari *input* berdasarkan pengetahuan yang didapat pada pelatihan sebelumnya dan *unsupervised* yang langsung mempelajari *input-input* yang diberikan tanpa pelatihan sebelumnya. Arsitektur jaringan saraf tiruan terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*) (Nugraha, *et al.*, 2013). Berikut adalah ilustrasi

struktur jaringan saraf tiruan yang menunjukkan bagaimana neuron-neuron pada setiap lapisan saling terhubung dan memproses informasi pada Gambar 1.



Gambar 1. Jaringan Saraf Tiruan (Sharma, *et al.*, 2020).

Pada setiap lapisan jaringan saraf tiruan memiliki fungsi aktivasi yaitu, persamaan matematika yang menentukan *output* neuron berdasarkan *input* yang diberikan. Fungsi ini berperan penting dalam arsitektur *deep learning* yang dapat memengaruhi output model, akurasi, dan efisiensi pelatihan (Do, *et al.*, 2021). Seperti pada otak manusia, jaringan saraf tiruan juga terdiri dari beberapa neutron yang akan mentransformasikan informasi melalui sambungan pada neutron lain yang diterima, hubungan ini disebut dengan bobot. Informasi atau *input* tersebut akan diproses oleh suatu fungsi yang menjumlahkan semua nilai-nilai bobot yang datang. Melalui fungsi aktivasi pada setiap neutron, neuron mengirim *output* melalui bobot *output* ke semua neuron yang berhubungan. Informasi dirambatkan melalui lapisan *input* hingga lapisan *output* melalui lapisan lain, yaitu lapisan tersembunyi (*hidden layer*) (Sinuhaji, *et al.*, 2023). Pada jaringan saraf tiruan, fungsi aktivasi dapat berupa *linear* maupun *nonlinier*, dimana setiap lapisan menerapkan fungsi aktivasi yang seragam pada semua neuron (Purwitasari & Soleh, 2022). Beberapa jenis fungsi aktivasi yang umum digunakan dalam jaringan saraf tiruan adalah sebagai berikut:

1. *Rectified Linear Unit* (ReLU)

   Fungsi aktivasi *Rectified Linear Unit* (ReLU) merupakan fungsi non-linier yang mengaktifkan neuron ketika *output* transformasi linier bernilai positif dan menghasilkan nol untuk nilai negatif atau nol (Firmansyah, *et al.*, 2022). Pada penggunaannya, fungsi ReLU dituliskan menggunakan Persamaan (1) dan grafik

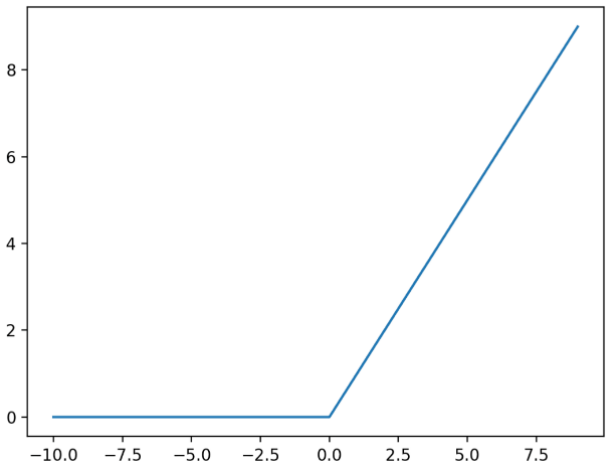
dari fungsi aktivasi ReLU dapat diilustrasikan pada Gambar 2.

$$ReLU(x) = \begin{cases} x, & \text{jika } x > 0 \\ 0, & \text{jika } x \leq 0 \end{cases} \tag{1}$$

Keterangan:

x = Nilai *input*,

$ReLU(x)$ = Hasil *output* fungsi ReLu berupa nilai 0 dan 1.



Gambar 2. Fungsi Aktivasi ReLU (Purwitasari & Soleh, 2022).

2. *Softmax*

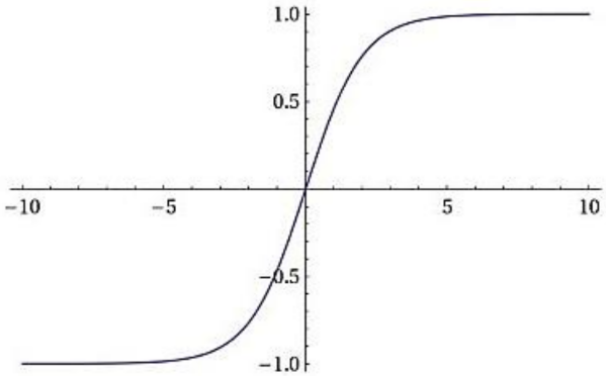
Fungsi aktivasi *softmax* merupakan fungsi yang digunakan untuk menghitung peluang dalam klasifikasi multi-kelas melalui pemilihan kelas dengan peluang paling tinggi sebagai *output*. Nilai peluang yang dihasilkan oleh fungsi ini berada dalam rentang 0 hingga 1 (Purwitasari & Soleh, 2022). Fungsi aktivasi *softmax* dirumuskan pada Persamaan (2) dan grafik dari fungsi aktivasi *softmax* dapat diilustrasikan pada Gambar 3.

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^{k} \exp(x_j)}, \quad i = 0, 1, 2, \ldots, k \tag{2}$$

Keterangan:

$softmax(x_i)$ = Hasil *output* fungsi *softmax*,

x = Nilai *input*.



Gambar 3. Fungsi Aktivasi *Softmax* (Purwitasari & Soleh, 2022).

3. *Sigmoid*

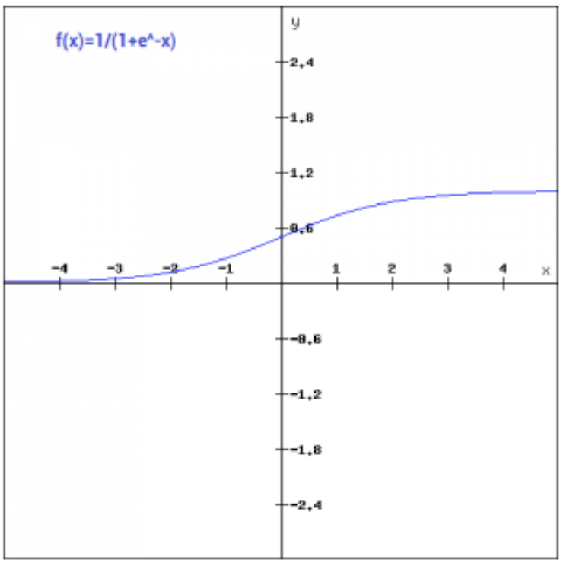
Fungsi aktivasi *sigmoid* merupakan fungsi yang digunakan untuk mengonversi nilai prediksi yang sudah diperoleh ke dalam rentang nilai antara 0 hingga 1 (Aryanti, 2023). Fungsi aktivasi *sigmoid* dirumuskan pada Persamaan (3) dan grafik dari fungsi aktivasi *sigmoid* dapat diilustrasikan pada Gambar 4.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Keterangan:

x = Nilai *input*,

$\sigma(x)$ = Hasil *output* fungsi *sigmoid*.

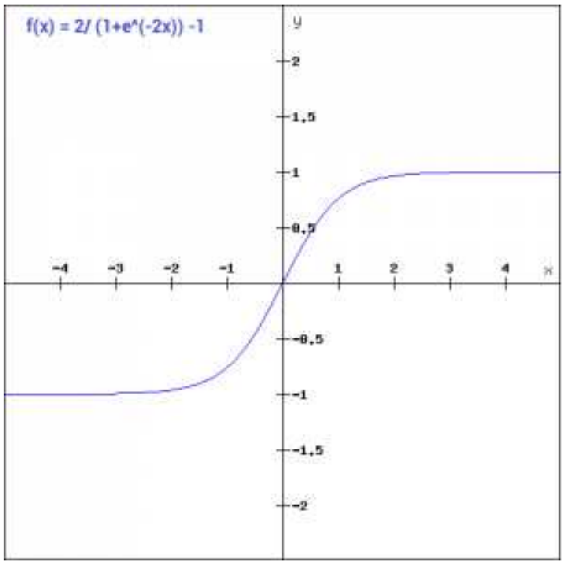


Gambar 4. Fungsi Aktivasi *Sigmoid* (Sharma, *et al.*, 2020).

4. *Hyperbolic tangent* (tanh)
   Fungsi aktivasi tanh merupakan fungsi simetris berbentuk *sigmoid* dengan *output* berada dalam rentang nilai -1 sampai +1 (Firmansyah, *et al.*, 2022). Fungsi aktivasi tanh dirumuskan pada Persamaan (4) dan grafik dari fungsi aktivasi tanh dapat diilustrasikan pada Gambar 5.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4}$$

   Keterangan:
   e^{x} = Nilai eksponsial positif dari nilai input,
   e^{-x} = Nilai eksponsial negatif dari nilai input,
   $tanh(x)$ = Hasil *output* fungsi tanh.



Gambar 5. Fungsi Aktivasi tanh (Sharma, *et al.*, 2020).

2.5 *Deep Learning*

Deep learning adalah cabang ilmu dari *machine learning* yang berfokus pada jaringan saraf tiruan yang merupakan pengembangan lanjutan dari konsep jaringan saraf tiruan tradisional (Pratiwi, *et al.*, 2021). *Deep learning* dapat diartikan sebagai metode pembelajaran mesin yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan neuron untuk menangkap fitur kompleks. Perbedaan utama antara *deep learning* dan pembelajaran mesin tradisional terletak pada jumlah serta kerumitan lapisan neuronnya. Kemampuan *deep learning* dalam memahami pola dari data yang

rumit dan abstrak membuatnya sangat berhasil diterapkan dalam berbagai bidang, seperti pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan lainnya. Selain itu, *Deep learning* dikenal sebagai metode pembelajaran yang didasarkan pada fitur berbentuk hierarki, dimana struktur hierarki tersebut dapat disesuaikan ukurannya sesuai dengan kasus yang sedang diproses.

Berdasarkan hal tersebut memungkinkan algoritma *deep learning* untuk melakukan ekstraksi fitur secara otomatis dari data mentah dengan sangat detail karena proses ekstraksi memanfaatkan struktur yang mendalam. Fitur-fitur yang dieksploitasi oleh *deep learning* biasanya tidak terlihat secara langsung karena pembeda antar kelas data seringkali berada pada tingkat distribusi yang sangat dalam. Oleh karena itu, fitur pada level tinggi perlu ditransformasi terlebih dahulu menjadi fitur pada level rendah agar dapat lebih mudah dipahami oleh mesin pembelajaran. Jika fitur level rendah dapat dengan mudah dikenali oleh *deep learning* maka fitur level tinggi pun akan lebih mudah diekstraksi. Kombinasi inilah yang membuat *deep learning* mampu menghasilkan representasi fitur yang berkualitas dan optimal (Lubis, *et al.*, 2024).

2.6 *Machine Translation* (MT)

Machine Translation (MT) merupakan penerjemahan yang dilakukan dengan menemukan terjemahan dengan peluang maksimum. Misalkan x merepresentasikan bahasa sumber dan y merepresentasikan bahasa target maka berdasarkan Persamaan (5), model akan memilih kalimat target yang memiliki peluang paling tinggi untuk diberikan kalimat sumber x.

$$\hat{y} = \arg\max_{y} P(y \mid x; \theta) \tag{5}$$

Berdasarkan aturan Bayes (Brown, *et al.*, 1993), Persamaan (5) ditulis ulang dalam Persamaan (6).

$$\hat{y} = \arg\max_{y} P(x)P(y; \theta_{\text{lm}})P(x \mid y; \theta_{\text{tm}}) \tag{6}$$

$P(x)$ merupakan konstanta untuk semua kemungkinan terjemahan y, sehingga persamaan tersebut dapat disederhanakan dalam Persamaan (7).

$$\hat{y} = \arg\max_{y} P(y; \theta_{\text{lm}})P(x \mid y; \theta_{\text{tm}}) \tag{7}$$

$P(y; \theta_{\text{lm}})$ merupakan *language model* (LM) yang merepresentasikan seberapa alami suatu kalimat target dalam bahasa target dan $P(x \mid y; \theta_{\text{tm}})$ merupakan *translation model* (TM) yang merepresentasikan sebuah kalimat dalam bahasa sumber dapat diterjemahkan menjadi kalimat dalam bahasa target.

Pendekatan tersebut sulit untuk digeneralisasi karena ketergantungan antar sub-model. Sehingga, model log-linear digunakan melalui pendekatan probabilistik pada Persamaan (8) untuk mengatasi keterbatasan tersebut (Och, *et al.*, 2022).

$$P(y, x \mid \theta) = \frac{\sum_z \exp(\theta \cdot \psi(x, y, z))}{\sum_{y'} \sum_{z'} \exp(\theta \cdot \psi(x', y', z'))} \tag{8}$$

Kemudian, Koehn, *et al.*, (2003) memperkenalkan *phrase-based translation model*, yaitu dengan memecah kalimat menjadi kata atau frasa pendek untuk menangkap pemilihan kata dalam menerjemahkan kalimat. Seleksi aturan transisi dilakukan menggunakan peluang transisi, yang dihitung menggunakan *Maximum Likelihood Estimation* (MLE). Metode MLE bergantung pada jumlah sampel yang cukup besar, sehingga rentan terhadap masalah data *sparsity* dan kurang mampu menangkap semantik yang dalam. Oleh karena itu, pendekatan berbasis *deep learning* menjadi solusi lebih unggul dengan kemampuan menangkap konteks yang lebih baik dalam meningkatkan kualitas terjemahan.

Neural machine translation (NMT) merupakan model penerjemahan menggunakan *deep neural networks* yang secara otomatis memetakan hubungan antara bahasa sumber dan bahasa target langsung dari data, tanpa memerlukan fitur-fitur yang dirancang manual untuk menerjemahkan seperti yang digunakan dalam *Statistical machine translation* (SMT). Model NMT berusaha menemukan peluang kalimat target berdasarkan kalimat sumber dengan memfaktorkan peluang penerjemahan setiap kata dalam konteks kata sebelumnya. Diasumsikan terdapat kalimat sumber dalam bahasa asli $X = \{x_1, \ldots, x_j, \ldots, x_J\}$ dan kalimat target dalam bahasa terjemahan $Y = \{y_1, \ldots, y_i, \ldots, y_I\}$ yang dimodelkan melalui Persamaan (9).

$$P(y \mid x; \theta) = \prod_{i=1}^{I} P(y_i \mid x, y_{<i}; \theta) \tag{9}$$

Menerjemahkan kalimat sumber dan target dengan panjang kalimat yang berbeda, diperlukan mekanisme yang baik agar model tetap bisa menangani konteks dengan baik. *Encoder-decoder* digunakan untuk merepresentasikan kalimat dalam bahasa sumber sebagai vektor dengan panjang tetap (*fixed-length vector*). Vektor ini

kemudian digunakan oleh *decoder* untuk menghasilkan terjemahan dalam bahasa target (Sutskever, *et al.*, 2014). Tujuan utama dalam pelatihan model NMT adalah memaksimalkan *log-likelihood* dari pasangan kalimat sumber dan kalimat target dalam dataset yang direpresentasikan melalui Persamaan (10) dan (11).

$$\hat{\theta} = \arg\max_{\theta} L(\theta) \tag{10}$$

$$L(\theta) = \sum_{i=1}^{I} \log P(y^{(i)} \mid x^{(i)}; \theta) \tag{11}$$

Setelah model NMT selesai dilatih, parameter yang telah dipelajari akan digunakan untuk melakukan penerjemahan. Model akan memilih urutan kata dalam bahasa target yang memiliki peluang tertinggi berdasarkan input dalam bahasa sumber pada Persamaan (12).

$$y = \arg\max_{y} P(y \mid x; \hat{\theta}) \tag{12}$$

Hasil menunjukkan bahwa kualitas terjemahan menurun secara signifikan ketika panjang kalimat sumber meningkat. Sebagai solusi terhadap permasalahan tersebut, Bahdanau, *et al.*, (2015) memperkenalkan mekanisme *Attention* dalam jaringan *encoder*. Mekanisme ini memungkinkan model untuk memilih bagian penting dari kalimat sumber secara dinamis saat menghasilkan kalimat target. Mekanisme *attention* menggunakan *Bi-directional Recurrent Neural Networks* (BRNNs), yaitu menangkap konteks dengan mempertimbangkan informasi dari arah maju (*forward*) dan mundur (*backward*). *Hidden state* dari arah maju dan mundur diperoleh melalui perhitungan yang diformulasikan dalam Persamaan (13) dan (14).

$$\overrightarrow{h_s} = f\left(x^{(s)}, \overrightarrow{h_{s-1}}, \theta\right) \tag{13}$$

$$\overleftarrow{h_s} = f\left(x^{(s)}, \overleftarrow{h_{s-1}}, \theta\right) \tag{14}$$

Hidden state dari arah maju dan mundur digabungkan untuk menangkap informasi dari kedua arah dengan Persamaan (15).

$$h_s = \left[\overrightarrow{h_{s-1}}; \overleftarrow{h_{s-1}}\right] \tag{15}$$

Mekanisme *attention* bekerja dengan menghitung bobot *attention* untuk menentukan seberapa relevan setiap kata dalam kalimat sumber terhadap kata target yang sedang dihasilkan. Perhitungan bobot tersebut ditunjukkan pada Persamaan (16).

$$\alpha_{j,i} = \frac{\exp(a(t_{j-1}, h_i, \theta))}{\sum_{i'=1}^{I+1} \exp(a(t_{j-1}, h_{i'}, \theta))} \tag{16}$$

Bobot perhatian $\alpha_{j,i}$ menentukan sejauh mana kata ke-i dalam bahasa sumber berkontribusi terhadap kata ke-j dalam bahasa target, dengan nilai yang dihitung menggunakan fungsi *softmax* agar totalnya menjadi 1 dan dapat diinterpretasikan sebagai distribusi peluang. Fungsi *alignment score* $a(t_{j-1}, h_i, \theta)$ digunakan untuk mengevaluasi kecocokan antara *hidden state* target sebelumnya t_{j-1} dan *hidden state* sumber h_i, yang kemudian menentukan bobot *attention* untuk setiap kata dalam kalimat sumber. Selain itu, jumlah kata dalam kalimat sumber dinyatakan sebagai $I + 1$, dimana $+1$ berasal dari penambahan token *End-of-Sentence* (EOS) untuk menandai akhir kalimat. Setelah mendapatkan bobot *attention* $\alpha_{j,i}$ maka vektor konteks c_j yang merupakan rata-rata tertimbang dari semua *hidden state* sumber dihitung menggunakan Persamaan (17).

$$c_j = \sum_{i=1}^{I+1} \alpha_{j,i} h_i \tag{17}$$

Vektor konteks c_j merupakan representasi gabungan dari informasi dalam *hidden state* sumber yang relevan untuk menghasilkan kata ke-j dalam bahasa target. Vektor ini dihitung sebagai kombinasi linear dari *hidden state* sumber h_i dengan bobot perhatian $\alpha_{j,i}$ yang menentukan sejauh mana setiap *hidden state* sumber berkontribusi terhadap pembentukan kata target. Berdasarkan kemampuannya menangkap informasi yang paling relevan dari kalimat sumber, vektor konteks c_j memainkan peran penting dalam menentukan kata berikutnya dalam bahasa target. *Hidden state* baru pada *decoder* (t_j) dihitung menggunakan Persamaan (18).

$$t_j = f(y_{j-1}, s_{j-1}, c_j, \theta) \tag{18}$$

Hidden state t_j pada langkah j dalam bahasa target dihitung menggunakan fungsi f yang bergantung pada kata sebelumnya dalam bahasa target y_{j-1}, *hidden state* sebelumnya dari decoder s_{j-1}, vektor konteks c_j yang berisi informasi dari sumber serta parameter model θ.

Model NMT yang menggunakan token kata sebagai *input* membuat model harus menghitung peluang penerjemahan dengan melakukan normalisasi terhadap seluruh kata dalam bahasa target. Akibatnya, perhitungan *log-likelihood* selama pelatihan menjadi sangat mahal secara komputasi, karena model harus mengevaluasi peluang transisi untuk setiap kata dalam target *vocabulary*. Oleh karena itu, digunakan pemrosesan berbasis sub-kata, metode *Byte Pair Encoding* (BPE) yang memecah kata menjadi sub-kata yang lebih kecil. Pendekatan ini tidak hanya

mengurangi kompleksitas komputasi, tetapi juga meningkatkan kemampuan model dalam menangani kata-kata *Out of Vocabulary* (OOV) tanpa mengorbankan kualitas terjemahan (Sennrich, *et al.*, 2015).

Pada pelatihan NMT, metrik standar yang digunakan adalah (MLE). Metode MLE bekerja dengan cara memaksimalkan *log-likelihood* dari data pelatihan untuk menemukan parameter model yang optimal. Metode (MLE) bekerja pada tingkat kata, sedangkan metrik evaluasi dalam MT adalah BLEU yang dihitung pada tingkat kalimat atau korpus. Ketidaksesuaian antara metode pelatihan dan evaluasi ini menyebabkan model yang dioptimalkan dengan MLE tidak selalu menghasilkan terjemahan yang baik berdasarkan metrik evaluasi akhir. Masalah tersebut dapat diatasi dengan mengunakan *Minimum Risk Training* (MRT) pada Persamaan (19) dan (20) sebagai fungsi *loss* dalam pelatihan model NMT.

$$\hat{\theta} = \arg\min_{\theta} R(\theta) \tag{19}$$

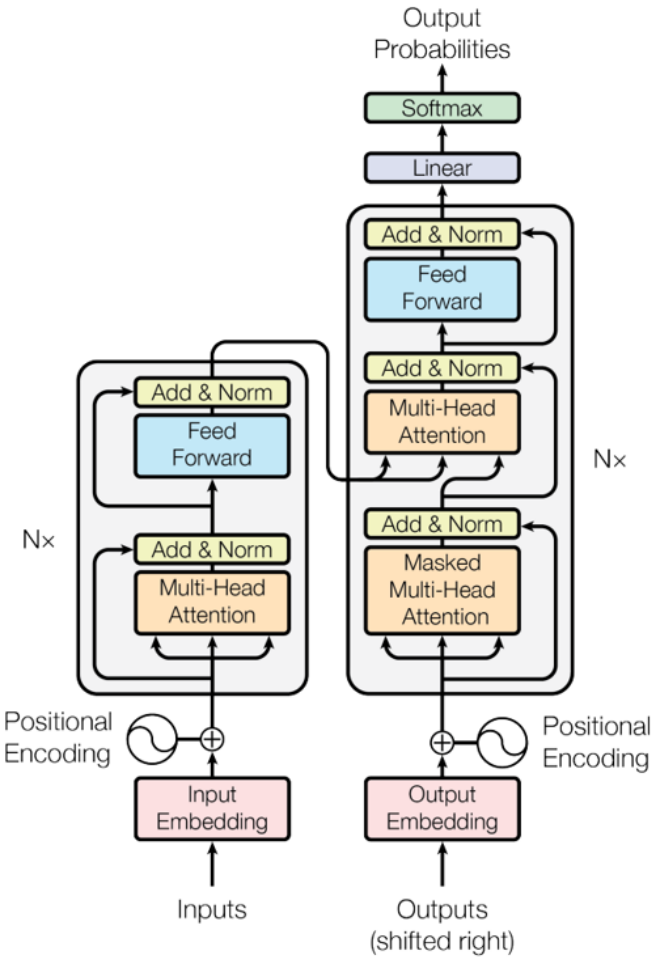
$$R(\theta) = \sum_{s=1}^{S} \sum_{y \in Y(x^{(s)})} P(y \mid x^{(s)}; \theta) \Delta(y, y^{(s)}) \tag{20}$$

Sebagian besar model NMT berbasis *encoder-decoder* awalnya dibangun menggunakan *Recurrent Neural Networks* (RNN) dan variannya seperti *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU). Namun, RNN bekerja secara sekuensial, yang berarti perhitungan setiap langkah bergantung pada langkah sebelumnya sehingga menyebabkan proses pelatihan menjadi lambat dan sulit diparalelisasi (Srivastava, *et al.*, 2018). *Transformer* dapat mengatasi keterbatasan tersebut menggunakan mekanisme *self-attention* untuk memperhatikan setiap kata dalam kalimat secara parallel untuk menangani kalimat yang lebih panjang dengan lebih baik (Firmanto & Sesoca, 2024). Selain itu, *transformer* mampu mempertimbangkan seluruh konteks kalimat secara bersamaan sehingga menghasilkan terjemahan lebih alami dan akurat secara tata bahasa (Zein, 2018). Secara keseluruhan, NMT memberikan hasil yang lebih baik dibandingkan metode sebelumnya, yaitu SMT. *Transformer* menjadi fondasi utama dalam pengembangan model NMT yang memungkinkan penerjemahan yang lebih canggih dan efisien di berbagai domain.

2.7 *Transformer*

Transformer merupakan model yang sepenuhnya mengandalkan *self-attention* untuk menghitung representasi *input* dan *output* tanpa menggunakan jaringan saraf rekuren atau konvolusi yang melibatkan struktur *encoder-decoder* yang kompleks dan menggunakan mekanisme *attention* sebagai penghubung antar kedua komponennya. Perhitungan posisi dalam urutan *input* dan *output* dalam jaringan saraf rekuren sering kali menghambat paralelisasi selama pelatihan. *Self-attention* dalam *transformer* membentuk representasi yang lebih baik dengan mengaitkan berbagai posisi dalam suatu urutan, sehingga tidak lagi membutuhkan konvolusi berurutan dalam perhitungan *input* dan *output*. *Transformer* terbukti mampu mengatasi batasan dalam jaringan saraf rekuren. *Transformer* menggunakan arsitektur *encoder-decoder* untuk menerjemahkan bahasa sumber ke bahasa target. *Encoder* menerima urutan *input* simbol $(x_1, ..., x_n)$ dari bahasa sumber yang berisi informasi penting secara keseluruhan dan mengubahnya menjadi representasi kontinu $z = (z_1, ..., z_n)$. Kemudian, *decoder* memecahkan representasi z tersebut dengan memproses simbol-simbol secara bertahap untuk menghasilkan urutan baru $(y_1, ..., y_m)$ yang berupa terjemahan ke bahasa target (Vaswani, *et al.*, 2017).

Encoder pada arsitektur *transformer* terdiri dari 6 lapisan identik, dimana setiap lapisan memiliki dua sub-lapisan *multi-head-self-attention* dan jaringan *feed-forward* yang terhubung sepenuhnya. Pada setiap dua sub-lapisan digunakan *residual connections* dan dilanjutkan dengan *layer normalization*. Secara sederhana, *LayerNorm*(x+ Sub-lapisan(x)) *output* dari setiap sub-lapisan dihitung dengan menambahkan *input* awal (x) ke hasil sub-lapisan, kemudian hasilnya dilakukan normalisasi. Selanjutnya, *decoder* pada *transformer* juga terdiri dari 6 lapisan identik, setiap lapisan memiliki dua sub-lapisan yang sama seperti *encoder*. Namun di antara dua lapisan tersebut ditambah dengan satu lapisan *attention* untuk mencegah posisi memperhatikan posisi berikutnya dalam urutan selama proses prediksi (Vaswani, *et al.*, 2017). Adapun arsitektur model *transformer* diilustrasikan pada Gambar 6.

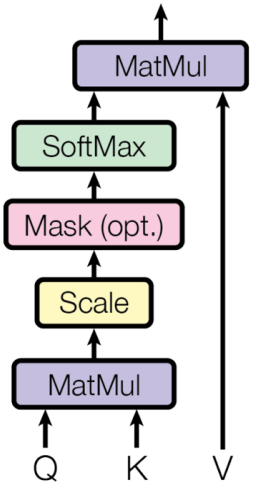


Gambar 6. Arsitektur Model *Transformer* (Vaswani, *et al.*, 2017).

2.7.1 *Scaled Dot-Product Self-Attention*

Attention digunakan untuk menggabungkan sebuah *query* dengan semua pasangan *key-value* dan menghasilkan sebuah *output*. Pada proses perhitungannya, *attention score* dihitung berdasarkan *scaled dot-product* yang terdiri dari himpunan vektor *query* dan *key* dengan dimensi (d_k) dan himpunan vektor value dengan dimensi (d_v). Pada proses ini, *query, key, value*, dan *output* semuanya berbentuk vektor yang disusun pada matriks Q, K, dan V. Kemudian, vektor *query* dengan semua vektor *key* dikalikan dan hasilnya dibagi dengan akar kuadrat dari dimensi *key* untuk menjaga kestabilan nilai. Setelah itu, fungsi *softmax* diterapkan untuk menentukan bagian informasi yang paling relevan. Terakhir, hasil dikalikan dengan *value*

untuk menghasilkan *output* yang mewakili informasi penting yang sesuai dengan *query* (Vaswani, *et al.*, 2017). Adapun arsitektur *Scaled Dot-Product Self-Attention* dapat ditafsirkan pada Gambar 7 dan persamaan *Scaled Dot-Product Self-Attention* dituliskan pada Persamaan (21).



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

$$\textit{Attention}(Q, K, V) = \textit{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{21}$$

Keterangan:

Q = Merepresentasikan ”pertanyaan” tentang relevansi antar elemen input,
K = Merepresentasikan ”kunci” untuk menentukan relevansi,
V = Menyimpan informasi yang akan dikombinasikan,
$QK^\top$ = Mengukur relevansi antar elemen input,
d_k = Dimensi vektor Query (Q) dan Key (K).

2.7.2 *Multi-Head Attention*

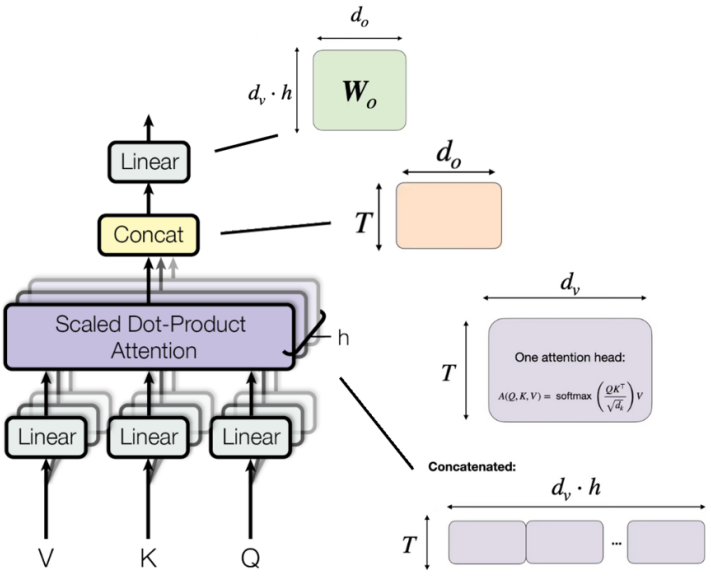
Fungsi *attention* pada *Multi-Head Attention* dilakukan dengan menggunakan matriks *queries*, *keys*, dan *values* yang berbeda sebanyak h kali secara bersamaan dengan tujuan mampu menangkap berbagai kemungkinan yang beragam (Vaswani, *et al.*,

2017). *Multi-head attention* diinterpretasikan melalui Persamaan (22) dan (23).

$$\begin{aligned} head_i &= Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \\ &= softmax\left(\frac{\left(QW_i^Q\right)\left(KW_i^K\right)^\top}{\sqrt{d_k}}\right)\left(VW_i^V\right) \end{aligned} \tag{22}$$

$$MultiHead(Q,K,V) = Concat\left(\text{head}_1,\ldots,\text{head}_h\right)W^O \tag{23}$$

Proyeksi tersebut direpresentasikan sebagai matriks parameter $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$, dan $W_i^O \in \mathbb{R}^{d_{\text{model}} \times d_k}$. Adapun arsitektur Multi-head Attention dapat direpresentasikan pada Gambar 8.



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

2.7.3 *Position-wise Feed-Forward Networks*

Jaringan *feed-forward* terdiri dari dua transformasi linear yang dihubungkan dengan fungsi aktivasi ReLU. *Encoder* dan *decoder* arsitektur *transformer* pada setiap lapisan memiliki sub-lapisan jaringan *feed-forward* yang terhubung sepenuhnya pada posisi terpisah dan identik. Meskipun transformasi sama di berbagai posisi, transformasi tersebut menggunakan parameter yang berbeda dari satu lapisan ke lapisan lainnya (Vaswani, *et al.*, 2017). Interpretasi dari *Feed-foward networks* dapat dilihat pada Persamaan (24).

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{24}$$

Keterangan:

W = Matriks bobot,

b = Vektor bias.

2.7.4 *Positional encoding*

Positional encoding digunakan *transformer* untuk memasukan informasi posisi *relative* atau *absolut* token-token dalam urutan dengan ditambahkan ke *embedding input* pada dasar setiap tumpukan *encoder-decoder*, sehingga *transformer* dapat menggunakan informasi urutan kata walaupun tidak menggunakan rekursif dan konvolusi (Vaswani, *et al.*, 2017). *Positional encoding* menggunakan fungsi sinus dan kosinus dengan frekuensi berbeda dimana *position* (*pos*) merepresentasikan posisi dalam urutan, dan i adalah dimensi yang direpresentasikan dalam Persamaan (25) dan (26).

$$\text{PE}_{(\text{pos},2i)} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{25}$$

$$\text{PE}_{(\text{pos},2i+1)} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{26}$$

Keterangan:

pos = Urutan posisi,

i = Indeks dimensi *embedding*.

2.8 MarianMT

MarianMT merupakan model *pre-trained* yang dirancang khusus untuk menyelesaikan permasalahan terkait NMT dengan menyederhanakan proses penerjemahan dan menjadikannya lebih efisien dan mudah digunakan. MarianMT dikembangkan oleh tim Microsoft Translator yang mampu mencapai kecepatan tinggi dalam pelatihan dan penerjemahan dengan *toolkit* yang ramah sumber daya. Kemudian, Kelompok Riset Teknologi Bahasa Universitas Helsinki (Helsinki-NLP) menjadikan MarianMT sebagai *toolkit* terdepan dalam NMT dengan mengembangkan dan memiliki lebih dari 1000 model bahasa yang telah dilatih

sebelumnya (Soliman, *et al.*, 2022). Selain itu, melakukan *fine-tunning* dengan spesifik domain pada MarianMT dapat dilakukan untuk memungkinkan model melakukan tugas terjemahan dengan hasil bahasa yang sesuai domain meskipun tidak dilatih secara langsung pada domain tersebut. Kemampuan ini memungkinkan MarianMT untuk mengembangkan teknologi terjemahan dengan mudah yang dapat diadaptasi untuk berbagai aplikasi dan kebutuhan.

MarianMT dibangun dan dikembangkan menggunakan arsitektur *transformer*. Menurut (Junczys-Dowmunt, *et al.*, 2018) MarianMT menggunakan arsitektur dengan 6 lapisan *self-attentive* pada *encoder* dan *decoder* dengan masing-masing dilengkapi 8 *head attention* pada setiap lapisan. Pada tahap awal, diberikan kalimat sumber $x = (x_1, x_2, ..., x_n)$ dengan panjang n, setiap token x_i direpresentasikan ke dalam ruang vektor berdimensi d_{model} melalui *embedding* $E(x_i) = e_i \in \mathbb{R}^{d_{\text{model}}}$. $E(x_i)$ merupakan baris ke-x_i matriks *embedding* berukuran $|V| \times d_{\text{model}}$ dengan $|V|$ adalah ukuran kosakata. Setiap token x_i diubah menjadi vektor *embedding* e_i dengan mengambil mengambil baris yang sesuai dengan matriks embedding E. Kemudian, kalimat direpresentasikan sebagai urutan *embedding* $e = (e_1, e_2, e_3)$. Model *transformer* tidak memiliki mekanisme bawaan untuk mengenali urutan token, setiap embedding e_i dijumlahkan dengan *positional encoding* $PE(i)$ yang akan menghasilkan representasi akhir e_i^{final}. *Positional encoding* memiliki dimensi d_{model} yang sama dengan *embedding*, sehingga keduanya dapat dijumlahkan. Setiap posisi $i, PE(i) \in \mathbb{R}^{d_{\text{model}}}$ dan komponen $PE(i)$ dihitung menggunakan Persamaan (25) dan (26) (Vaswani, *et al.*, 2017).

Bagian *encoder* pada MarianMT memproses *input* atau bahasa sumber dan menangkap informasi kontekstualnya dengan mekanisne *self-attention*. Setiap token dalam *input* yang telah direpresentasikan sebagai *embedding* final (e_i^{final}) diproyeksikan menjadi tiga vektor berbeda, yaitu query (Q_i), key (K_i), dan value (V_i). Proyeksi dilakukan melalui operasi matriks menggunakan bobot yang telah dilatih, yaitu W^Q, W^K, and W^V, dengan persamaan $Q_i = W^Q e_i^{\text{final}}$, $K_i = W^K e_i^{\text{final}}$, and $V_i = W^V e_i^{\text{final}}$. Setelah itu, *attention score* dihitung antara setiap pasangan token i dan j melalui operasi *dot product* antara Q_i dan K_j^T, yang kemudian dinormalisasi dengan akar dari dimensi model (d_{model}). Nilai *attention score* diterapkan fungsi softmax untuk menentukan bagian informasi yang paling relevan dan menghasilkan matriks peluang (P_{ij}). Peluang tersebut digunakan untuk mengkombinasikan vektor *value*, melalui *wighted sum* dan menghasilkan *output attention* untuk setiap token. Proses ini diulangi

untuk beberapa head berbeda, sehinggga menghasilkan banyak representasi, yang dituliskan pada Persamaan (23) (Vaswani, *et al.*, 2017). Kemudian, diproses oleh *Feed-Forward Neural Network* (FNN) yang terpisah untuk setiap token dan terdiri dari dua lapisan linear dengan aktivitas ReLU pada Persamaan (24) dimana W_1, W_2 adalah bobot matriks dan b_1, b_2 adalah bias (Vaswani, *et al.*, 2017). Hasil pada proses FNN dijumlahkan dengan *input* asli melalui *residual connection* dan diterapkan *layer normalization* untuk menstabilkan pelatihan. Proses ini diterapkan secara berulang untuk setiap lapisan *encoder*.

Sedangkan, bagian decoder pada MarianMT menghasilkan terjemahan dalam *output* atau bahasa target berdasarkan representasi dari *input* yang sudah diproses sebelumnya. Diberikan kalimat target $y = (y_1, y_2, ..., y_m)$ dengan panjang m diubah menjadi representasi vektor *embedding* $E(y_i) = e_i \in \mathbb{R}^{d_{\text{model}}}$. Setiap *embedding* e_i dijumlahkan dengan *positional encoding* $PE(i)$ yang akan menghasilkan representasi akhir e_i^{final} untuk setiap token. Berbeda dengan *encoder*, pada *decoder* terdapat *Masked Self-Attention* untuk menjamin bahwa model hanya menyatakan token sebelumnya hingga token ke-i untuk menghasilkan token ke-$i+1$. *Attention score* dihitung menggunakan *dot product* antara query (Q_i) dan (K_j) yang telah diskalakan dengan akar dimensi key $\left(\sqrt{d_k}\right)$ jika $j \leq i$. *Attention score* untuk token $j > i \quad$ dan $\quad -\infty$ yang diproses oleh fungsi *softmax*, menghasilkan peluang nol, sehingga model hanya mempertimbangkan token yang valid hingga posisi saat ini untuk menghasilkan *output attention* dan konsistensi dalam prediksi token target (Tan, *et al*., 2020). Mekanisme *encoder-decoder attention* memungkinkan model fokus pada bagian penting kalimat sumber pada saat menghasilkan terjemahan untuk menangkap informasi posisi kata dalam sebuah kalimat yang merupakan hal penting dalam melakukan penerjemahan, baik pada *encoder* maupun *decoder* menggunakan *positional encoding*. *Output encoder* $Z = (z_1, z_2, ..., z_n)$ sebagai *input* untuk vektor K dan V, sedangkan query (Q) berasal dari *output self-attention* pada *decoder*. Output dari *encoder-decoder attention* kemudian dikombinasikan dengan hasil *masked self-attention* melalui mekanisme *residual connection* dan *layer normalisasi*. Representasi akhir dari tahap ini diteruskan ke *feed-forward network* (FFN), dimana setiap token diproses secara *independen* melalui dua lapisan linear yang dipisahkan oleh fungsi aktivasi ReLU. *Output* dari FFN menjadi representasi akhir yang kaya secara kontekstual untuk setiap token target.

Pada langkah terakhir, representasi ini diproyeksikan ke ruang dimensi kosakata melalui lapisan linear, dan distribusi peluang untuk setiap kata target dihitung menggunakan *softmax*. Token target y_i kemudian dipilih berdasarkan argmax dari distribusi ini. Proses ini berulang secara autoregresif hingga semua token dalam bahasa target dihasilkan. *Multi-head attention* memungkinkan *cross attention* pada *decoder* untuk memperhatikan representasi dari berbagai aspek *input* atau bahasa sumber yang telah diproses oleh *encoder* sebelumnya. Pada tahap *fine-tunning, hyperparameter* seperti *dropout rate, learning rate,* dan jumlah *epoch* dapat disesuaikan agar model dapat beradaptasi lebih baik dengan dataset yang spesifik.

2.9 *Early-stopping*

Early-stopping adalah teknik yang digunakan dalam pelatihan model untuk menghentikan proses pelatihan lebih awal pada kondisi tertentu. Teknik *early stopping* diterapkan untuk menghindari *overfitting* atau tidak adanya peningkatan kinerja model pada data sehingga proses *training* akan langsung dihentikan. *Early stopping* bertujuan untuk mencegah model belajar terlalu banyak pada data pelatihan (Utami, *et al.*, 2023). Pada proses pelatihan, *epoch* menjadi salah satu parameter yang penting. *Epoch* adalah parameter yang digunakan dalam menentukan berapa kali model akan dilatih. Pemilihan jumlah *epoch* harus diperhatikan, iterasi pelatihan yang terlalu banyak dapat menyebabkan *overfitting* dan iterasi pelatihan yang terlalu sedikit dapat menyebabkan *underfitting*. Ketika nilai loss secara tiba-tiba naik signifikan, berhenti lebih awal dapat menjadi solusi yang tepat dalam masalah ini (Hidayat & Maki, 2022).

2.10 *Dropout*

Dropout merupakan teknik menghilangkan atau memutuskan jaringan yang masuk dan keluar pada neuron secara acak di lapisan jaringan saraf yang terlihat maupun yang tersembunyi. *Dropout* digunakan untuk menentukan persentasi pembuangan nilai yang dianggap paling buruk dari sejumlah nilai yang dihasilkan pada satu tahap pembelajaran sehingga pada pembuangan ini dapat mengurangi *overfitting* pada model. Pengurangan penggunaan neuron pada jaringan saraf, *dropout* dapat mempercepat proses pelatihan. Nilai *dropout* berkisar antara 0 hingga 1 (Hidayat & Maki, 2022).

2.11 *Fine-tuning*

Pada NLP, *fine-tuning* menjadi salah satu pendekatan penting yang dapat dilakukan dalam model *transformer* untuk meningkatkan kinerjanya. *Fine-tuning* adalah proses melakukan pelatihan menggunakan model *pre-trained* yang telah ada kemudian mengatur model untuk melakukan tugas serupa yang telah disesuaikan. Penggunaan jaringan saraf tiruan yang telah dirancang dan dilatih memungkinkan memperoleh keuntungan dari yang telah dipelajari model tanpa harus mengembangkan dari awal, sehingga tidak melalui banyak pendekatan *trial-error* (Hafifah, *et al.*, 2021). Tujuan dilakukannya *fine-tuning* adalah untuk menyesuaikan fitur khusus pada kumpulan data baru yang dilatih, bukan menimpanya (Haksoro & Setiawan, 2021). Proses *fine tuning* dilakukan untuk mendapatkan hasil yang sesuai dengan yang diinginkan. *Fine-tuning* telah banyak dilakukan pada beberapa penelitian, karena kemudahannya pada pelatihan dan waktu yang lebih sedikit dibandingkan dengan proses *pre-training* (Awalina, *et al.*, 2022).

2.12 Evaluasi BLEU

Metrik evaluasi yang digunakan untuk mengevaluasi teks terjemahan pada *machine translation* secara otomatis adalah BLEU (*Bilingual Evaluation Understudy)*. Metode BLEU *score* berkisar antara 0 dan 100 yang mengukur kesamaan teks yang telah diterjemahkan oleh *machine translation* dengan satu set referensi terjemahan berkualitas tinggi. Nilai 0 mempresentasikan bahwa hasil terjemahan yang telah diterjemahkan oleh *machine translation* tidak tumpang tindih atau tidak ada kesamaan dengan referensi terjemahan sehingga kualitas terjemahan dapat dikatakan rendah. Sedangkan, nilai 100 mempresentasikan bahwa hasil terjemahan yang telah diterjemahkan oleh *machine translation* ada tumpang tindih yang lengkap atau memiliki banyak kesamaan dengan referensi terjemahan sehingga kualitas terjemahan dapat dikatakan tinggi. Skor BLEU dapat diinterpretasikan melalui Tabel 2 yang memberikan pemahaman terarah mengenai tingkat kualitas hasil terjemahan (Primandhika, *et al.*, 2021).

Tabel 2. Interpretasi Skor BLEU dalam Evaluasi Mesin Terjemahan

BLEU *Score*	**Keterangan**
$<$10	Hampir tidak memberikan informasi yang berguna.
10-19	Sulit memahami makna keseluruhan teks.
20-29	Makna utama dapat dipahami, namun terdapat banyak kesalahan tata bahasa.
30-40	Terjemahan berkualitas baik.
40-50	Terjemahan berkualitas tinggi.
$>$ 50	Terjemahan sangat berkualitas melebihi hasil terjemahan manusia.

Berikut merupakan perhitungan manual yang dilakukan untuk memperoleh BLEU *score* pada Persamaan (27).

$$\text{BP} = \begin{cases} 1, & \text{jika } c > r \\ e^{\left(\frac{1-r}{c}\right)}, & \text{jika } c \leq r \end{cases} \tag{27}$$

Langkah pertama dalam menghitung nilai BLEU adalah menentukan *Brevity Penalty* (BP) dengan memperhitungkan rasio panjang terjemahan hasil (c) terhadap panjang referensi (r). Jika panjang terjemahan hasil (c) lebih panjang dari panjang referensi (r), maka tidak mengurangi skor karena terjemahan cukup panjang dan tidak ada penalti untuk hal ini. Sedangkan, jika panjang terjemahan hasil (c) lebih pendek atau sama dengan panjang referensi (r), maka dapat memberikan penalti untuk panjang terjemahan yang terlalu pendek. Kemudian, perhitungan nilai BLEU dilakukan dengan menggunakan Persamaan (28).

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{28}$$

dengan:
N = Jumlah n-gram yang digunakan dalam perhitungan,
w_n = Bobot untuk setiap n-gram,
$\log p_n$ = Keakuratan n-gram ke-n untuk mengurangi bias.

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester ganjil tahun ajaran 2024/2025 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Pengerjaan terbagi menjadi tiga tahap, tahap pertama melakukan pengumpulan studi literatur dan penentuan tema penelitian. Setelah itu, dilakukan pengumpulan data terkait dan dimulainya proses pengolahan data. Tahap pertama memerlukan waktu pengerjaan kurang lebih 20 minggu. Tahap kedua, melakukan penyusunan proposal penelitian dan penyampaian seminar proposal penelitian. Tahap terakhir merupakan tahap penyusunan hasil penelitian dan penyampaian hasil penelitian melalui seminar hasil serta sidang komprehensif yang diharapkan selesai pada tahun 2025.

3.2 Data dan Alat

3.2.1 Data Penelitian

Data yang digunakan dalam penelitian ini terbagi menjadi tiga, yaitu data pelatihan *fine-tuning* sebagai *train, validation* dan data uji sebagai *test*. Data yang digunakan sebagai data pelatihan adalah korpus parallel TED-Talks2020 yang bersumber pada https://opus.nlpl.eu/TED2020/en&id/v1/TED2020 berisi 160.632 pasangan kalimat terjemahan Bahasa Inggris ke Bahasa Indonesia dengan hasil terjemahan formal. Sedangkan, data uji yang digunakan dalam penelitian ini adalah 5.000 pasangan kalimat terjemahan Bahasa Inggris ke Bahasa Indonesia dari kumpulan data NeuLab-TEDTalks yang diperoleh melalui *website* Open Parallel Corpora yang dapat diakses pada https://opus.nlpl.eu/NeuLab-TedTalks/en&id/v1/NeuLab-TedTalks.

Melalui data uji tersebut akan dilakukan evaluasi apakah model mampu menerjemahkan data NeuLab-TEDTalks menjadi terjemahan formal. Sampel data yang digunakan dalam penelitian ini disajikan dalam Tabel 3 sebagai data *train* dan *validation*, serta dalam Tabel 4 sebagai data *test*.

Tabel 3. Sampel Data TED2020

English text	**Indonesian text**
I update the slide show every time I give it.	Saya perbaharui presentasinya setiap kali saya berikan.
Put yourselves in my position!	Cobalah kalian mengerti perasaan saya!
And I say that sincerely	dan saya katakan itu dari dalam hati saya
I was going to talk about information ecology.	Saya ingin berbicara tentang ekologi informasi.
.....	.....

Tabel 4. Sampel Data NeuLab-TEDTalks

English text	**Indonesian text**
And i think as long as i keep grieving	Dan saya pikir selama saya tetap berduka
Well we know that they can roll balls in a straight line	Jadi kita tahu bahwa dia dapat menggulirkan bola pada jalur yang lurus
Our challenge is to find the compassion for others	Tantangan kita adalah mencari kasih sayang terhadap orang lain
When these mutate they change their shape	Ketika mereka bermutasi mereka mengubah bentuk mereka
.....	.....

3.2.2 Alat Penelitian

Adapun alat yang digunakan dalam penelitian ini, sebagai berikut:

1. **Perangkat Keras (*Hardware*)**
   Perangkat keras yang digunakan pada penelitian ini untuk melaksanakan penelitian, yaitu:
   a. Processor AMD Ryzen 7 8845HS w/ Radeon 780M Graphics, 3801 Mhz, 8 Core(s), 16 Logical Processor(s)

b. Memory DDR5 RAM 16 GB

2. **Perangkat Lunak (*Software*)**

   Perangkat lunak yang digunakan pada penelitian ini untuk melaksanakan penelitian, yaitu:

   a. Operating System Windows 11 Home Single Language 64-bit
   b. *Python* 3.12.3

3. ***Library***

   Library yang digunakan pada penelitian ini untuk melaksanakan penelitian, yaitu:

   a. *Transformers 4.40.2*

      Transformers merupakan *library* utama dari Hugging Face yang menyediakan model *pre-trained* dan dapat digunakan untuk berbagai macam tugas salah satunya adalah penerjemahan bahasa (Ibrahim, *et al.*, 2025).

   b. *Numpy 1.26.4*

      Numpy merupakan *library* dasar dalam komputasi numerik di Python yang menyediakan *array* multidimensi dan berbagai fungsi matematika yang efisien (Hermanto, *et al.*, 2022).

   c. *Pandas 2.2.2*

      Pandas merupakan *library* yang membantu dalam membaca, mengolah, membersihkan, dan menganalisis data sebelum digunakan dalam model berbasis tabel (*DataFrame*) (Hermanto, *et al.*, 2022).

   d. *Sentencepiece 0.2.0*

      Sentencepiece merupakan tokenizer berbasis *subword* yang dikembangkan oleh Google. *Library* ini sering digunakan dalam model *multilingual* dan *machine translation* karena lebih efektif mengatasi kata-kata yang jarang muncul. *SentencePiece* memecah teks menjadi unit-unit kecil (*subwords*) yang lebih fleksibel dibanding tokenisasi berbasis kata (Haris, *et al.*, 2024).

   e. *PyTorch 2.3.0 + cu121*

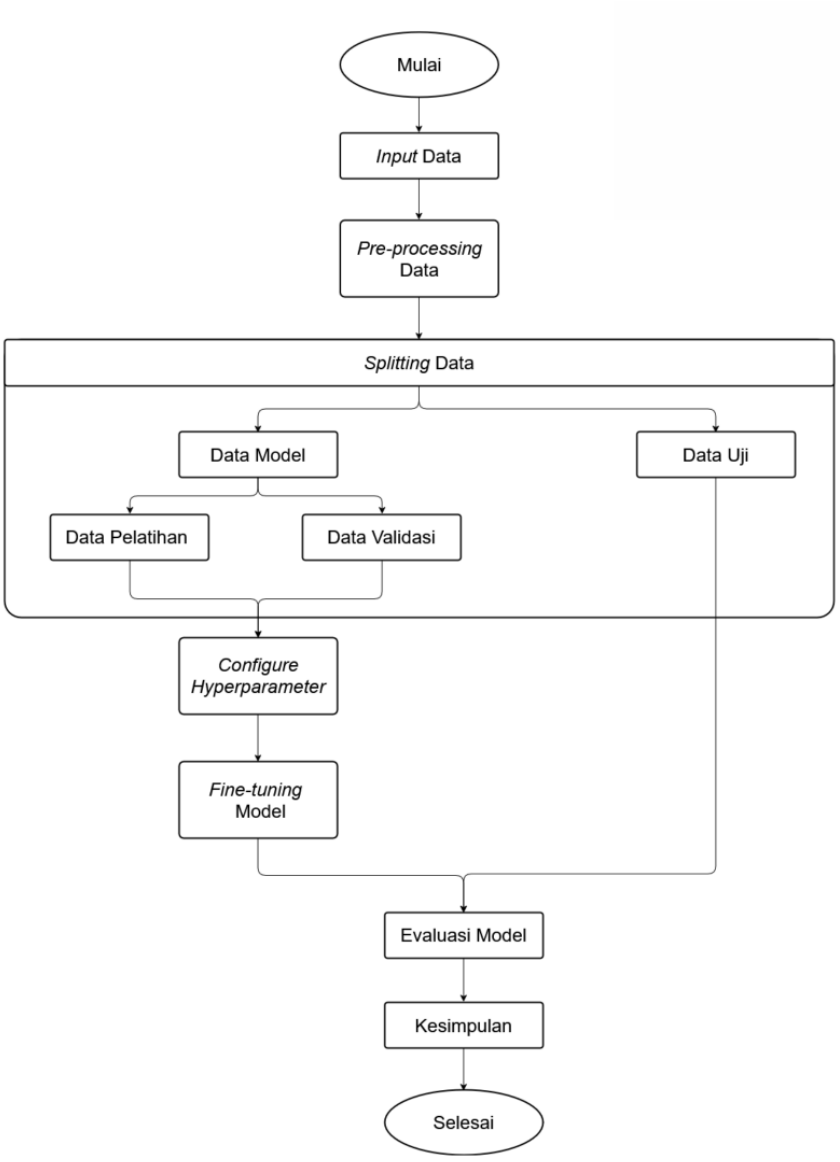
      PyTorch merupakan salah satu *library* dalam komputasi *deep learning* yang digunakan untuk melatih model, menangani *loss function*, dan menjalankan model di GPU (Dzil, *et al.*, 2024)

   f. *Sacrebleu 2.4.2*

      Sacrebleu merupakan *library* yang digunakan untuk mengevaluasi hasil terjemahan mesin menggunakan BLEU *score* secara konsisten. SacreBLEU dirancang untuk menghindari ketidakcocokan antar evaluator BLEU konvensional dengan menyediakan tokenisasi standar serta mendukung *multi-reference evaluation* (Post, 2018).

3.3 Metode Penelitian

Penelitian ini menggunakan model *pre-trained* MarianMT berbasis *transformer* dengan *fine-tunning* data formal. Penelitian dilakukan dengan perangkat lunak python menggunakan bantuan teks editor Kaggle. Hasil evaluasi dinilai berdasarkan hasil BLEU *score* pada data uji. Alur penelitian diilustrasikan pada Gambar 9.



Gambar 9. *Flowchart* Metode Penelitian.

Penjelasam dari alur penelitian yang akan dilaksanakan dalam penelitian ini adalah sebagai berikut:

1. *Input* data

   Melakukan *Input* data *in-domain* yang akan digunakan untuk *fine-tuning* model dan *Input* data *test* yang akan digunakan dalam evaluasi model.

2. *Pre-processing* data

   Melakukan *pre-processing* pada seluruh data dengan melakukan *tokenisasi* untuk memecah teks menjadi unit kecil yang terstruktur, *case-folding* untuk mengubah semua huruf menjadi kecil, *filtering* untuk menghapus karakter non-alfabet yang tidak relevan, dan *padding* untuk memastikan panjang data seragam. Tahapan ini mempersiapkan data agar siap digunakan oleh model dengan efisiensi dan akurasi yang lebih baik.

3. *Splitting* data

   Proses pembagian data dilakukan dengan cara membagi dataset menjadi dua bagian utama, yaitu data model dan data *testing*. Data model dibagi menjadi dua, yaitu data pelatihan dan data validasi. Terdapat lima skenario proporsi *splitting* data yang akan dilakukan pada data model, yaitu 70:30, 75:25, 80:20, 85:15, dan 90:10. Untuk pengujian model (*testing*), seluruh data *testing* digunakan secara terpisah. Dataset uji tidak terlibat dalam proses pelatihan atau validasi, sehingga hasil pengujian dapat memberikan gambaran yang objektif mengenai kemampuan model dalam menggeneralisasi pada data baru.

4. Inisialisasi dan konfigurasi *hyperparameter* model

   Inisialisasi model MarianMT *pre-trained* merupakan langkah pertama dalam mempersiapkan model untuk diterapkan pada tugas terjemahan bahasa. Setelah inisialisasi, konfigurasi *hyperparameter* dimodifikasi untuk mengoptimalkan proses pelatihan. Parameter yang disesuaikan meliputi *learning rate* yang diatur menjadi 2×10^{-5} untuk pembaruan bobot yang stabil, *batch size* sebesar 64 untuk efisiensi pemrosesan data, *dropout* sebesar 0.3 untuk mencegah *overfitting*, dan jumlah *epoch* sebanyak 5 untuk memastikan model menjalani pelatihan yang cukup tanpa *overtraining*. Melalui konfigurasi tersebut, model diharapkan mampu mencapai performa yang baik dalam tugas terjemahan tanpa kehilangan kemampuan generalisasi.

5. *Fine-tuning* model

   Melakukan *fine-tuning* pada model dengan melibatkan proses pelatihan model menggunakan data latih (*training data*) dengan tujuan meningkatkan kinerja model. Proses ini dilakukan dengan menerapkan teknik *early-stopping,* yaitu metode untuk menghentikan pelatihan secara dini jika kinerja model pada data validasi tidak mengalami peningkatan dalam beberapa epoch berturut-turut. Hal

ini bertujuan untuk mencegah *overfitting*, mengoptimalkan penggunaan sumber daya komputasi, dan menjaga model tetap generalis terhadap data baru.

6. Evaluasi model
   Evaluasi model dilakukan menggunakan data *test* untuk mengukur kemampuan model dalam menggeneralisasi pada data baru yang tidak terlibat dalam pelatihan. Model diuji dengan data uji dan hasilnya dibandingkan dengan nilai sebenarnya menggunakan BLEU *score*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan sebelumnya dalam *fine-tuning* model MarianMT menggunakan data TEDTalks pada terjemahan bahasa Inggris ke bahasa Indonesia dengan konteks terjemahan formal diperoleh beberapa kesimpulan utama dapat diambil sebagai berikut:

1. *Fine-tuning* model MarianMT menggunakan dataset *in-domain* terbukti mampu meningkatkan kualitas terjemahan bahasa Inggris ke bahasa Indonesia. Evaluasi menggunakan metrik BLEU menunjukkan bahwa model yang telah melalui proses *fine-tuning* dengan penghapusan tanda baca pada tahap *pre-processing* mampu menghasilkan terjemahan yang lebih baik dibandingkan model awal dengan skor BLEU tertinggi yang diperoleh sebesar 32,14.
2. Penerapan teknik *dropout* pada lapisan *feedforward, attention*, dan fungsi aktivasi bertujuan untuk mengurangi risiko *overfitting* selama proses pelatihan model. Selain itu, penggunaan *early stopping* dengan metrik BLEU memungkinkan proses pelatihan berhenti lebih awal apabila tidak terjadi peningkatan yang signifikan dalam beberapa *epoch* berturut-turut. Kombinasi kedua teknik ini berkontribusi dalam meningkatkan stabilitas model serta memastikan bahwa model tidak hanya menghafal data latih, tetapi juga mampu melakukan generalisasi dengan baik terhadap data uji.
3. Pada penelitian ini, dilakukan eksperimen dengan lima skenario pembagian data untuk mengeksplorasi dampaknya terhadap performa model. Hasil analisis menunjukkan bahwa skenario pembagian data 90:10 (pelatihan:validasi) menghasilkan model dengan kinerja terbaik berdasarkan skor BLEU serta kestabilan nilai *loss* selama proses pelatihan dan validasi. Model dengan proporsi ini dapat belajar secara optimal dari data latih, sementara validasi yang cukup kecil tetap memberikan indikasi akurat terhadap kinerja model tanpa mengorbankan jumlah data yang digunakan untuk pembelajaran.

4. Salah satu aspek yang diamati dalam penelitian ini adalah dampak penghapusan tanda tanya pada tahap *pre-processing* terhadap hasil terjemahan yang dihasilkan oleh model. Hasil pengamatan menunjukkan bahwa penghapusan tanda tanya, model cenderung menerjemahkan kalimat tanya seperti kalimat pernyataan sehingga mempengaruhi makna terjemahan yang dihasilkan. Hal ini berdampak pada kalimat pendek dan sedang dimana tanda baca berperan penting dalam menandai intonasi dan maksud pesan yang disampaikan. Sementara itu, pada kalimat panjang struktur kompleks dapat membantu model mempertahankan konteks meskipun tanda tanya dihilangkan.

5.2 Saran

Adapun beberapa saran yang perlu diperhatikan untuk penelitian berikutnya adalah sebagai berikut:

1. Melakukan eksplorasi teknik augmentasi data untuk meningkatkan *robustness* model. Upaya ini bertujuan untuk meningkatkan ketahanan model terhadap berbagai variasi *input*, teknik augmentasi data seperti *back-translation*, *paraphrasing*, atau penambahan variasi kalimat tanya dan kalimat formal dapat diterapkan. Penambahan contoh dari berbagai struktur kalimat memungkinkan model dapat belajar untuk mengenali serta menerjemahkan kalimat dengan lebih fleksibel tanpa kehilangan makna aslinya.
2. Penelitian ini menemukan bahwa tanda tanya memiliki pengaruh terhadap hasil terjemahan dalam beberapa kasus. Oleh karena itu, diperlukan analisis yang lebih mendalam terhadap struktur sintaksis dan semantik dalam proses penerjemahan untuk memahami bagaimana model memproses dan menghasilkan terjemahan yang sesuai. Penelitian lebih lanjut mengenai bagaimana model menangani kalimat ambigu atau kalimat dengan banyak interpretasi juga dapat menjadi area penelitian yang menarik.

DAFTAR PUSTAKA

Aryanti, W. 2023. Penerapan artificial neural network dengan algoritma backpropagation untuk memprediksi harga saham. *Jurnal Riset Statistika*. **3**(2): 107-118.

Aristyanto, M. Y., & Kurniawan, R. 2021. Pengembangan Metode Neural Machine Translation Berdasarkan Hyperparameter Neural Network, hlm. 935-946. In Seminar Nasional Official Statistics, Jakarta.

Awalina, A., Bachtiar, F. A., & Utaminingrum, F. 2022. Perbandingan Pretrained Model Transformer pada Deteksi Ulasan Palsu. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, **9**(3): 597–604.

Baquero, A.P. 2023. Transformer Models for Machine Translation and Streaming Automatic Speech Recognition (Disertasi). Departement De Sistemes Informatics I Computacio Universitat Politecnica De Valencia, Valencia.

Bahdanau, D., Cho, K., & Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, San Diego.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, **19**(2): 263-311.

Do, N. Q., Selamat, A., Krejcar, O., Yokoi, T., & Fujita, H. 2021. Phishing webpage classification via deep learning-based algorithms: an empirical study. *Applied Sciences*, **11**(19): 9210.

Dwiastuti, M. 2019. English-Indonesian neural machine translation for spoken language domains, hlm. 309-314. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence.

Dzil, F., Khumaidi, A., Rahmat, M. B., Endrasmono, J., Syai'in, M., & Riananda, D. P. 2024. Deteksi Objek di Lapangan pada Robot Sepakbola Beroda Menggunakan Metode YOLOV5. *Jurnal Elektronika dan Otomasi Industri*. **11**(2): 604-611.

Feldman, R., & Sanger, J. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, New York.

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

Firmanto, B., & Sesoca, J. 2024. Tinjauan Perkembangan Kecerdasan Buatan Berbasis Arsitektur Transformer. *SMARTICS Journal*, **10**(1), 33-38.

Ghirrid, A.A., Sari, R.T.K., & Aldisa, R.T. 2024. Algoritma Natural Language Processing Untuk Aplikasi Penerjemah (Indonesia – Jawa) Menggunakan Metode Speech Processing*Journal Teknologi Informasi dan Komunikasi*. **8**(3): 746-759.

Hafifah, F., Rahman, S., & Asih, M. S. 2021. Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN). *TIN: Terapan Informatika Nusantara*, **2**(5): 292-301.

Haksoro, E. I., & Setiawan, A. 2021. Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, **5**(2): 81-91.

Haris, M., Suharso, A., & Nurkifli, E. H. 2024. Analisis Sentimen Pada Game Efootball di Google Play Store Menggunakan Algoritma INDOBERT. *JATI (Jurnal Mahasiswa Teknik Informatika)*. **8**(6): 12108-12121.

Hermanto, K., Salim, D., Wu, B., Salim, O. R., Gunadi, R. B. 2022. Penggunaan python untuk menganalisis pola penyebaran covid-19 di masa pandemi. Journal of Student. *Journal of Student Development Information System (JoSDIS)*. **2**(2): 120-133.

Hidayat, D., & Maki, W. F. A. 2022. Deteksi Glaukoma Menggunakan Metode Convolutional Neural Network Dan Grabcut Segmentation. *eProceedings of Engineering*, **9**(3).

Ibrahim, A. K., Mustikasari, M., & Bastian, I. 2025. Analisis Sentimen Pada Ulasan Aplikasi Astro-Groceries in Minutes di Google Play Store Menggunakan Metode Bidirectional Encoder Representation From Transformers (BERT). *Jurnal Informatika dan Teknik Elektro Terapan*. **13**(1).

Jamaluddin, T., Bijaksana, M. A., & Asror, I. 2020. Perbandingan Algoritma Sentencepiece BPE dan Unigram Pada Tokenisasi Artikel Bahasa Indonesia. *eProceedings of Engineering*, **7**(2).

Junczys-Dowmunt, M., Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, F.A., Bogoychev. N., Martins, A.F.T., Birch, A. 2018. Marian: Fast Neural Machine Translation in C++. hlm 116-121. In Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, University of Edinburgh.

Koehn, P., Och, F. J., & Marcu, D. 2003. Statistical phrase-based translation, hlm 48-54. In 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Langauge Technology (HLT-NAACL 2003), Edmonton.

Kudo, T. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959.

Lubis, N., Siambaton, M. Z., & Aulia, R. 2024. Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN). *sudo Jurnal Teknik Informatika*. **3**(3): 113-126.

Maksymenko, D., Turuta, O., Saichyshyna, N., Yerokhin, M., & Turuta, O. 2023. Controllability for English-Ukrainian Machine Translation Based on Specialized Corpora, hlm. 1-9. Proceedings of the 1st International Workshop on Multilingual, Multimodal and Multitask Language Generation, Kharkiv.

Miyagawa, S. 2023. Machine Translation for Highly Low-Resource Language: A Case Study of Ainu, a Critically Endangered Indigenous Language in Northern Japan, hlm. 120-124. Proceedings of the Joint 3rd International Conference on Natural Language Processing for Digital Humanities and 8th International Workshop on Computational Linguistics for Uralic Languages, Tokyo.

Nugraha, K. A., Santoso, A. J., & Suselo, T. 2013. Algoritma backpropagation pada jaringan saraf tiruan untuk pengenalan pola wayang kulit, hlm 8-13. In Seminar Nasional Informatika (SEMNASIF), Yogyakarta.

Och, F. J., & Ney, H. 2022. Discriminative training and maximum entropy models for statistical machine translation, hlm 295-302. In Proceedings of the 40th Annual meeting of the Association for Computational Linguistics, Philadelphia.

Pertiwi, A., Sujaini, H., & Safriadi, N. 2024. Pengaruh Penggunaan Attention pada Kualitas Mesin Penerjemah Berbasis Neural Bahasa Indonesia ke Bahasa Madura Bangkalan. *Jurnal Linguistik Komputasional*. **7**(1): 11-20.

Post, M. 2018. A call for clarity in reporting BLEU scores, hlm 186–191. In Association for Computational Linguistics. Brussels.

Pratiwi, H. A., Cahyanti, M., & Lamsani, M. 2021. Implementasi Deep Learning Flower Scanner Menggunakan Metode Convolutional Neural Network. *Sebatik*. **25**(1): 124-130.

Primandhika, R. B., Munawar, M. N., & Saifullah, A. R. 2021. Experiment on a Transformer Model Indonesian-to-Sundanese Neural Machine Translation with Sundanese Speech Level Evaluation, hlm 452-459. In Thirteenth Conference on Applied Linguistics (CONAPLIN 2020), Bandung.

Purwitasari, N. A., & Soleh, M. 2022. Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing. *Jurnal Ilmu Pengetahuan dan Teknologi (IPTEK)*, **6**(1): 81-91: 14-21.

Rehm, G., Sasaki, F., Stein, D., & Witt, A. 2018. *Language Technologies for a Multilingual Europe: TC3 III*. 5^{th} Edition. Language Science Press, Berlin.

Pertiwi, A., Sujaini, H., & Safriadi, N. 2024. Pengaruh Penggunaan Attention pada Kualitas Mesin Penerjemah Berbasis Neural Bahasa Indonesia ke Bahasa Madura Bangkalan. *Jurnal Linguistik Komputasional*. **7**(1): 11-20.

Romanelli, F., Cain, J., & McNamara, P. J. 2014. Should TED talks be teaching us something?. *American journal of pharmaceutical education)*, **78**(6): 113.

Sennrich, R., Haddow, B., & Birch, A. 2015. Neural machine translation of rare words with subword units, hlm 1715–1725. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics , Berlin.

Sharma, S., Sharma, S., Athaiya, A. 2020. Activation functions in neural networks. *Towards Data Sci*. **6**(12): 310-316.

Sinuhaji, F., Tarigan, H., & Tarigan, D. E. 2023. Memperkirakan Kuantitas Masyarakat Di Kabupaten Karo Dengan Pendekatan Jaringan Saraf Tiruan. *Jurnal Curere*, **7**(1): 38-52.

Siregar, U.D. 2023. Bahasa Inggris sebagai Bahasa Komunikasi Bisnis di Era Globalisasi: Persepsi Pebisnis dan Karyawan. *Jurnal Bahasa dan Sastra Indonesia*. **3**(1): 129-135.

Soliman, S.S., Mayada, M.H., & Shaheen, I.S. 2022. MarianCG: A Code Generation Transformer Model Inspired by Machine Translation. *Journal of Engineering and Applied Science*. **69**(1): 104.

Srivastava, S., Shukla, A., & Tiwari, R. 2018. Machine translation: from statistical to modern deep-learning practices. arXiv preprint arXiv:1812.04238.

Surjandari, I., Megawati, C., Dhini, A., & Hardaya, I. B. N. S. 2016. Application of text mining for classification of textual reports: a study of Indonesia's national complaint handling system, hlm 8-10. In 6th International Conference on Industrial Engineering and Operations Management (IEOM 2016), Kuala Lumpur.

Sutskever, I., Vinyals, O., & Le, Q. V. 2014. Sequence to sequence learning with neural networks, hlm 27. Advances in neural information processing systems, Canada.

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

Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. 2023. Fine-tuning large neural language models for biomedical natural language processing. *Patterns*, **4**(4).

Utami, N. W., Purnama, I. N., & Prajna, I. P. R. 2023. Klasifikasi Tanaman Upakara Adat Hindu Di Kebun Raya Eka Karya Bali Menggunakan Algoritma Convolutional Neural Network. *Jurnal Informatika Teknologi dan Sains (Jinteks)*, **5**(4): 671-678.

Vaswani, A. 2017. Attention is all you need, hlm. 5998–6008. Advances in Neural Information Processing Systems Proceedings of NeurIPS, California.

Wairagala, E.P., Mukiibi, J., Tusubira, J.F., Babirye, C., Nabende, N.J., Katumba, A., & Ssenkungu, I. 2022. Gender bias evaluation in Luganda-English machine translation, hlm. 274-286. Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas, Kampala.

Zein, A. 2018. Peran Text Processing Dalam Aplikasi Penerjemah Multi Bahasa Menggunakan Ajax Api Google. *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, **28**(1).