Analisis Perbandingan Arsitektur RNN dengan Attention dan Transformer untuk Penerjemahan Mesin Saraf Inggris-Indonesia

6th Gusti Ahmad Muttahid Bilhaq *Teknik Informatika / A1*
Universitas Darussalam Gontor 442023611097

***Abstrak*—Penerjemahan Mesin Saraf (NMT) telah menjadi pendekatan dominan untuk tugas penerjemahan otomatis, dengan arsitektur berbasis Recurrent Neural Network (RNN) dan Transformer sebagai fondasi utamanya. Penelitian ini bertujuan untuk mengimplementasikan dan membandingkan secara empiris kinerja model baseline RNN dengan mekanisme Attention dengan model Transformer untuk tugas penerjemahan dari bahasa Inggris ke bahasa Indonesia (EN-ID). Kedua model dilatih pada dataset paralel dari koleksi Anki, menggunakan tokenisasi subword Byte-Pair Encoding (BPE) dengan 8000 kosakata dan dilatih selama 10 epoch menggunakan framework PyTorch. Evaluasi dilakukan menggunakan metrik SacreBLEU pada test set yang terpisah. Hasil eksperimen menunjukkan temuan yang signifikan: model baseline RNN dengan Attention mencapai skor BLEU sebesar 21.50, secara substansial mengungguli model Transformer yang hanya mencapai skor 5.96. Hasil ini mengindikasikan bahwa dalam kondisi dataset yang terbatas dan durasi pelatihan yang singkat, arsitektur yang lebih sederhana seperti RNN dapat menggeneralisasi lebih baik. Kinerja sub-optimal Transformer menyoroti sensitivitasnya terhadap hyperparameter dan potensi kebutuhan akan data yang lebih besar serta jadwal pelatihan yang lebih canggih untuk mencapai potensi penuhnya.**

Kata Kunci—Transfer Learning, Klasifikasi Citra, ResNet50, Convolutional Neural Network, Visi Komputer

I. PENDAHULUAN

Komunikasi lintas bahasa merupakan fondasi utama dalam era globalisasi digital. Seiring meningkatnya interaksi global, kebutuhan akan sistem penerjemahan otomatis yang cepat dan akurat menjadi semakin krusial. Dalam beberapa dekade terakhir, bidang ini telah mengalami pergeseran paradigma dari metode statistik (SMT) ke Penerjemahan Mesin Saraf (NMT), yang memanfaatkan model deep learning untuk menghasilkan terjemahan yang lebih fasih dan kontekstual. Pendekatan NMT telah terbukti secara signifikan unggul dan menjadi standar industri saat ini.

Arsitektur NMT yang paling awal dan berpengaruh adalah model Encoder-Decoder berbasis Recurrent Neural Network (RNN). Model ini memproses kalimat sumber secara sekuensial dan mengompresnya menjadi sebuah vektor konteks, yang kemudian digunakan oleh decoder untuk menghasilkan kalimat target. Terobosan besar datang dengan diperkenalkannya mekanisme Attention, yang memungkinkan decoder untuk secara dinamis fokus pada bagian-bagian relevan dari kalimat sumber pada setiap langkah penerjemahan, sehingga secara dramatis meningkatkan kinerja pada kalimat panjang. Namun, sifat sekuensial dari RNN membatasi potensi paralelisasi dan terkadang mengalami kesulitan dalam menangkap dependensi jarak jauh.

Sebagai respons terhadap keterbatasan tersebut, arsitektur Transformer diperkenalkan dan dengan cepat merevolusi bidang NLP. Transformer sepenuhnya meninggalkan konsep rekurensi dan sebagai gantinya mengandalkan mekanisme self-attention untuk menimbang pentingnya setiap kata dalam kalimat secara bersamaan. Kemampuannya untuk diproses secara paralel membuat waktu pelatihan menjadi jauh lebih efisien pada perangkat keras modern dan terbukti sangat efektif dalam memodelkan hubungan kompleks antar kata, menjadikannya arsitektur pilihan untuk berbagai tugas NLP berskala besar.

Meskipun Transformer dianggap sebagai state-of-the-art, perbandingan kinerjanya dengan arsitektur RNN+Attention dalam kondisi sumber daya yang spesifik (misalnya, dataset yang tidak terlalu besar) masih menjadi studi kasus yang menarik. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan, melatih, dan mengevaluasi secara empiris dua model NMT untuk tugas penerjemahan dari bahasa Inggris ke bahasa Indonesia (EN-ID). Model pertama adalah baseline RNN dengan mekanisme Attention, dan model kedua adalah arsitektur Transformer. Kinerja kedua model akan dievaluasi secara kuantitatif menggunakan metrik SacreBLEU untuk memberikan wawasan tentang kelebihan dan kekurangan masing-masing pendekatan dalam skenario yang terkontrol.

II. METODOLOGI

A. Dataset

Penelitian ini menggunakan dataset pasangan kalimat paralel bahasa Inggris-Indonesia yang bersumber dari koleksi Anki yang tersedia di situs ManyThings.org. Dataset mentah ini berisi sekitar 89,000 pasangan kalimat. Setelah melalui tahap pembersihan awal, dataset kemudian diacak dan dibagi menjadi tiga bagian dengan rasio 80:10:10, yaitu:

- Data Latih (Training Set): 71,200 pasangan kalimat, digunakan untuk melatih model.
- Data Validasi (Validation Set): 8,900 pasangan kalimat, digunakan untuk memantau kinerja model selama pelatihan dan mencegah overfitting.
- Data Uji (Test Set): 8,900 pasangan kalimat, digunakan untuk evaluasi akhir model yang sudah dilatih.

B. Pra-pemrosesan Data

Sebelum dimasukkan ke dalam model, data mentah melewati serangkaian langkah pra-pemrosesan. Pertama, semua teks diubah menjadi huruf kecil (lowercase) dan spasi berlebih di awal atau akhir kalimat dihapus.

Langkah krusial berikutnya adalah tokenisasi. Untuk mengatasi masalah kata yang tidak ada dalam kosakata (Out-

of-Vocabulary atau OOV) dan menangani morfologi bahasa Indonesia yang kaya, penelitian ini menggunakan pendekatan tokenisasi subword dengan algoritma Byte-Pair Encoding (BPE). Proses ini diimplementasikan menggunakan library SentencePiece. Sebuah model BPE dengan ukuran kosakata 8,000 token dilatih pada gabungan seluruh kalimat bahasa Inggris dan Indonesia dari dataset. Token-token spesial seperti <sos> (awal kalimat) dan <eos> (akhir kalimat) juga ditambahkan pada setiap urutan kalimat sebelum diproses oleh model.

C. *Arsitektur Model*

Dua arsitektur Neural Machine Translation (NMT) diimplementasikan dan dibandingkan dalam penelitian ini.

D. *Model Baseline: RNN dengan Attention*

Model pertama adalah arsitektur Encoder-Decoder sekuensial berbasis Recurrent Neural Network (RNN).

- Encoder: Terdiri dari sebuah lapisan embedding yang diikuti oleh satu lapisan Gated Recurrent Unit (GRU). Encoder memproses urutan token input dan menghasilkan serangkaian output state serta sebuah hidden state final (vektor konteks). Dimensi embedding dan hidden state diatur sebesar 256.
- Decoder: Juga menggunakan satu lapisan GRU. Pada setiap langkah waktu, decoder menerima embedding dari token target sebelumnya, hidden state sebelumnya, dan sebuah vektor konteks yang dihitung oleh mekanisme Attention.
- Attention Mechanism: Mekanisme attention aditif (gaya Bahdanau) diimplementasikan untuk memungkinkan decoder secara dinamis memberikan bobot relevansi pada output state encoder. Hal ini membantu model fokus pada bagian-bagian paling relevan dari kalimat sumber saat menghasilkan setiap kata terjemahan.

E. *Model Transformer*

Model kedua didasarkan pada arsitektur Transformer, yang sepenuhnya mengandalkan mekanisme self-attention dan tidak menggunakan rekurensi. Implementasi ini memanfaatkan modul nn.Transformer bawaan dari PyTorch.

- Komponen Utama: Arsitektur ini terdiri dari tumpukan lapisan encoder dan decoder. Setiap lapisan encoder memiliki sub-lapisan Multi-Head Self-Attention dan Feed-Forward Network. Lapisan decoder memiliki tiga sub-lapisan: Masked Multi-Head Self-Attention, Multi-Head Attention (antara encoder-decoder), dan Feed-Forward Network.
- Positional Encoding: Karena model ini tidak memproses data secara sekuensial, informasi posisi token disuntikkan ke dalam input embedding menggunakan Positional Encoding berbasis fungsi sinus dan kosinus.
- Hyperparameter: Arsitektur Transformer dikonfigurasi dengan dimensi model (d_model) 256, 8 attention heads (nhead), 3 lapisan encoder, 3 lapisan decoder, dan dimensi feed-forward sebesar 512.

F. *Pengaturan Pelatihan*

Seluruh model diimplementasikan menggunakan framework PyTorch dan dilatih pada lingkungan Google Colab dengan akselerasi GPU. Untuk kedua model, Adam optimizer digunakan. Learning rate diatur ke 0.001 untuk model RNN dan 0.0001 untuk model Transformer. Fungsi kerugian (loss function) yang digunakan adalah Cross-Entropy Loss, dengan mengabaikan indeks token padding saat kalkulasi. Proses pelatihan dijalankan selama 10 epoch dengan ukuran batch sebesar 64. Pada model RNN, teknik Teacher Forcing dengan rasio 0.5 diterapkan untuk menstabilkan pelatihan.

G. *Metrik Evaluasi*

Kinerja akhir dari kedua model dievaluasi pada test set menggunakan metrik SacreBLEU. BLEU (Bilingual Evaluation Understudy) adalah metrik standar dalam evaluasi NMT yang mengukur kesamaan antara terjemahan yang dihasilkan mesin dan terjemahan referensi oleh manusia berdasarkan presisi n-gram. Skor yang lebih tinggi mengindikasikan kualitas terjemahan yang lebih baik.

III. EKSPERIMEN

A. *Skenario Eksperimen*

Eksperimen dilakukan dengan melatih kedua model dari awal (from scratch) pada dataset latih yang sama. Proses pelatihan dijalankan untuk jumlah epoch yang sama, yaitu 10 epoch, untuk memastikan bahwa perbandingan dilakukan dalam batasan sumber daya komputasi dan waktu yang setara. Kinerja model pada setiap epoch dipantau menggunakan data validasi, dan evaluasi final dilakukan pada data uji yang belum pernah dilihat oleh kedua model.

B. *Konfigurasi Model dan Pelatihan*

Konfigurasi hyperparameter utama untuk kedua model dirangkum dalam Tabel 1. Pengaturan ini dipilih berdasarkan praktik umum dalam literatur NMT dan disesuaikan dengan skala dataset yang digunakan.

Parameter	**Model RNN + Attention**	**Model Transformer**
Ukuran Kosakata (BPE)	8,000	8,000
Dimensi Embedding	256	256
Dimensi Hidden/Model	256	256
Lapisan Encoder	1	3
Lapisan Decoder	1	3
Jumlah Attention Heads	N/A	8
Dimensi Feed-Forward	N/A	512
Optimizer	Adam	Adam
Learning Rate	0.001	0.0001
Ukuran Batch	64	64

Parameter	Model RNN + Attention	Model Transformer
Jumlah Epoch	10	10
Dropout	0.1	0.1

C. *Lingkungan Komputasi*

Seluruh proses pelatihan dan evaluasi model dijalankan pada platform Google Colaboratory. Eksperimen ini memanfaatkan akselerasi hardware dari Graphics Processing Unit (GPU) yang disediakan oleh platform (umumnya NVIDIA Tesla T4 atau seri serupa) untuk mempercepat waktu komputasi yang dibutuhkan oleh model deep learning.

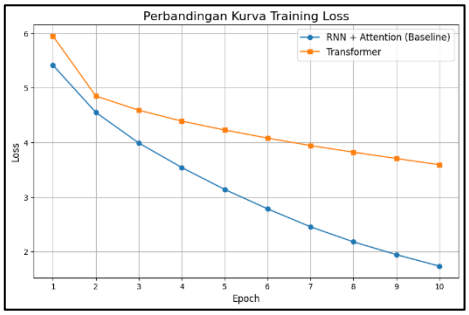
D. *Metrik Evaluasi*

Untuk mengukur kualitas terjemahan secara kuantitatif, metrik SacreBLEU digunakan. Metrik ini dipilih karena merupakan standar de facto dalam riset penerjemahan mesin yang memastikan proses evaluasi yang konsisten dan dapat direproduksi. Skor dihitung pada keseluruhan data uji dengan membandingkan hasil terjemahan model (hipotesis) dengan terjemahan referensi.

IV. HASIL DAN PEMBAHASAN

A. *Kinerja Pelatihan Model*

Proses pembelajaran kedua model dipantau dengan mencatat nilai loss pada data latih di setiap akhir epoch. Gambar 1 menampilkan perbandingan kurva training loss antara model baseline RNN dengan Attention dan model Transformer selama 10 epoch.



Gambar 1. Perbandingan Kurva Training Loss antara Model RNN dan Transformer

Dari kurva tersebut, dapat diamati bahwa kedua model menunjukkan tren penurunan loss yang konsisten, mengindikasikan bahwa keduanya berhasil belajar dari data pelatihan. Namun, terdapat perbedaan yang jelas dalam konvergensi. Model RNN dengan Attention menunjukkan penurunan loss yang lebih curam dan stabil, mencapai nilai loss akhir sebesar 1.735. Sebaliknya, model Transformer, meskipun juga mengalami penurunan loss, memiliki kurva yang lebih landai dan mencapai nilai loss akhir yang lebih tinggi, yaitu 3.592. Hal ini menunjukkan bahwa dalam 10 epoch, model RNN mampu menyesuaikan diri dengan data latih secara lebih efektif.

B. *Hasil Evaluasi Kuantitatif*

Evaluasi final dilakukan pada 8,900 pasangan kalimat dalam test set untuk mengukur kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Kualitas terjemahan diukur menggunakan skor SacreBLEU. Hasil perbandingan kuantitatif disajikan dalam Tabel 2.

Metrik	Model RNN + Attention	Model Transformer
Final Training Loss	1.735	3.592
Skor SacreBLEU	21.50	5.96
Waktu Inferensi (Test Set)	~17 detik	~2 menit 5 detik

Hasil evaluasi kuantitatif menunjukkan perbedaan kinerja yang sangat signifikan antara kedua model. Model baseline RNN dengan Attention secara telak mengungguli model Transformer, dengan mencapai skor BLEU sebesar 21.50 poin. Skor ini menunjukkan kemampuan terjemahan yang cukup baik untuk model yang dilatih dalam waktu singkat. Sebaliknya, model Transformer hanya mampu meraih skor BLEU 5.96, yang mengindikasikan kualitas terjemahan yang masih sangat rendah.

C. *Analisis Kualitatif*

Untuk mendapatkan pemahaman yang lebih dalam tentang perbedaan perilaku kedua model, analisis kualitatif dilakukan pada beberapa contoh terjemahan. Tabel 3 menyajikan salah satu contoh representatif.

Tipe	Kalimat
Sumber (EN)	a group of people are playing football.
Referensi (ID)	sekelompok orang sedang bermain sepak bola.
Hasil RNN	sekelompok orang bermain sepak bola.
Hasil Transformer	orang-orang itu adalah seorang orang yang baik.

Dari contoh di atas, terlihat jelas perbedaan kualitas. Model RNN berhasil menghasilkan terjemahan yang akurat dan relevan dengan kalimat sumber. Sebaliknya, model Transformer menghasilkan kalimat yang secara gramatikal koheren namun secara semantik sama sekali tidak relevan. Fenomena ini sering disebut sebagai "halusinasi", di mana model menghasilkan teks yang fasih tetapi tidak berdasar pada input yang diberikan.

D. *Pembahasan*

Hasil eksperimen yang menunjukkan keunggulan signifikan model RNN atas Transformer dalam skenario ini adalah temuan yang menarik dan memerlukan pembahasan lebih lanjut. Meskipun arsitektur Transformer secara umum dianggap state-of-the-art, kinerjanya yang rendah dalam penelitian ini dapat diatribusikan pada beberapa faktor potensial:

1. Ukuran Dataset: Transformer dikenal sebagai arsitektur yang "haus data" (data-hungry). Pada dataset dengan ukuran relatif kecil (sekitar 70,000 kalimat latih), model yang lebih sederhana seperti RNN dengan bias induktif sekuensialnya mungkin dapat belajar dan menggeneralisasi pola secara lebih efisien.
2. Sensitivitas Hyperparameter: Arsitektur Transformer sangat sensitif terhadap pengaturan hyperparameter, terutama learning rate. Penggunaan learning rate yang konstan dan tanpa jadwal pemanasan (warm-up) mungkin tidak optimal dan menghambat konvergensi model.
3. Durasi Pelatihan: Sepuluh epoch mungkin cukup bagi model RNN untuk mencapai konvergensi yang baik, namun belum cukup bagi model Transformer yang lebih kompleks untuk mempelajari representasi data yang kaya dan matang.

Dengan demikian, hasil ini tidak serta-merta menyimpulkan bahwa arsitektur RNN lebih superior dari Transformer secara umum. Namun, hasil ini secara kuat menunjukkan bahwa dalam kondisi sumber daya yang terbatas (dataset kecil, waktu pelatihan singkat, dan tanpa tuning hyperparameter yang ekstensif), model yang lebih sederhana seperti RNN dengan Attention dapat menjadi pilihan yang lebih praktis dan efektif.

KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan dan membandingkan kinerja dua arsitektur Neural Machine Translation (NMT) yang fundamental—model baseline Recurrent Neural Network (RNN) dengan mekanisme Attention dan model Transformer—untuk tugas penerjemahan dari bahasa Inggris ke bahasa Indonesia. Eksperimen dilakukan dalam lingkungan yang terkontrol, menggunakan dataset, durasi pelatihan, dan metrik evaluasi yang sama untuk kedua model.

Temuan utama dari penelitian ini adalah bahwa model baseline RNN dengan Attention, meskipun memiliki arsitektur yang lebih sederhana, menunjukkan kinerja yang secara signifikan lebih unggul dibandingkan model Transformer. Hal ini dibuktikan secara kuantitatif melalui skor SacreBLEU, di mana model RNN mencapai 21.50, sementara model Transformer hanya mencapai 5.96. Hasil ini menyoroti bahwa dalam skenario dengan dataset yang terbatas dan tanpa optimisasi hyperparameter yang ekstensif, arsitektur yang lebih sederhana dapat menghasilkan generalisasi yang lebih baik dan lebih efektif.

Keterbatasan utama dari penelitian ini terletak pada durasi pelatihan yang hanya 10 epoch dan penggunaan konfigurasi hyperparameter dasar untuk model Transformer. Kinerja sub-optimal dari Transformer kemungkinan besar disebabkan oleh faktor-faktor tersebut, karena arsitektur ini dikenal memerlukan data dalam skala besar dan penyesuaian hyperparameter yang cermat untuk mencapai potensi maksimalnya.

Untuk penelitian di masa mendatang, beberapa arah pengembangan dapat dieksplorasi. Pertama, melakukan penyetelan hyperparameter yang lebih ekstensif untuk model Transformer, termasuk implementasi learning rate scheduler dengan fase warm-up dan menambah durasi pelatihan secara signifikan. Kedua, melakukan eksperimen pada dataset yang jauh lebih besar untuk memvalidasi apakah keunggulan Transformer dapat terwujud seiring dengan bertambahnya skala data. Terakhir, melakukan studi ablasi yang lebih mendalam, seperti menganalisis dampak jumlah lapisan encoder/decoder atau jumlah attention heads terhadap kinerja model.

Sebagai penutup, penelitian ini menunjukkan bahwa pemilihan arsitektur model tidak dapat dilepaskan dari konteks masalah, termasuk skala data dan sumber daya komputasi yang tersedia. Model yang lebih sederhana seperti RNN dengan Attention masih menjadi alternatif yang sangat relevan dan berkinerja tinggi untuk tugas-tugas dengan sumber daya yang terbatas.

.