Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JACOST**

JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)

Vol. 7 No. 1 (2026) 1 – 8 | ISSN: 2723-1453 (Media Online)

Generative Chatbot Berbahasa Indonesia Dengan Menggunakan Arsitektur Transformer

Winarto Saputro[1], Edi Winarko[2]

[1,2]Departemen Ilmu Komputer dan Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada

[1]winartosaputro@ugm.ac.id*, [2]ewinarko@ugm.ac.id

Abstract

A chatbot is a computer program designed to interact with humans through text or voice messages. One widely studied approach is the generative chatbot, which produces responses dynamically based on conversational data, in contrast to retrieval-based or rule-based approaches that rely on fixed templates or knowledge bases. This study specifically aims to develop a sequence-to-sequence Transformer-based model for Indonesian-language conversations and to perform an empirical comparison with a GRU architecture enhanced with an Attention mechanism. The dataset consists of Indonesian question–answer pairs collected from previous studies and expanded through synonym-based data augmentation to increase variation and diversity in the training data. The model is evaluated using the BLEU score to measure the quality of responses generated, as well as computational efficiency indicators during training and inference. The experimental results show that the Transformer architecture performs better in maintaining context over long sentence sequences, as reflected by higher BLEU scores compared to GRU+Attention across the evaluated datasets. In addition, the parallel processing nature of the Transformer contributes to more efficient training time than the sequential GRU+Attention model. This study indicates the potential of the Transformer as an effective foundation for the development of Indonesian generative chatbots.

Keywords: *BLEU score, chatbot, self-attention, sequence-to-sequence, transformer*

Abstrak

Chatbot merupakan program komputer yang dirancang untuk berinteraksi dengan manusia melalui pesan teks maupun suara. Salah satu pendekatan yang banyak dikaji adalah *Generative Chatbot*, yang menghasilkan respons secara dinamis berdasarkan data percakapan, berbeda dengan pendekatan *Retrieval* maupun *Rule-based* yang bergantung pada templat atau basis pengetahuan tetap. Penelitian ini secara khusus bertujuan untuk mengembangkan model *sequence-to-sequence* berbasis *Transformer* untuk percakapan berbahasa Indonesia serta melakukan pembandingan empiris dengan arsitektur *GRU* yang diperkaya mekanisme *Attention*. Dataset yang digunakan berupa pasangan tanya–jawab berbahasa Indonesia yang diambil dari penelitian terdahulu dan diperluas melalui teknik augmentasi berbasis sinonim guna meningkatkan variasi dan keberagaman data pelatihan. Model dievaluasi menggunakan metrik *BLEU-Score* untuk mengukur kualitas respons yang dihasilkan serta indikator efisiensi komputasi selama pelatihan dan inferensi. Hasil eksperimen menunjukkan bahwa arsitektur *Transformer* menunjukkan kinerja yang lebih baik dalam mempertahankan konteks pada urutan kalimat yang panjang, yang tercermin pada peningkatan nilai *BLEU-Score* dibandingkan *GRU+Attention* pada data setiap dataset yang diuji. Selain itu, sifat pemrosesan paralel pada *Transformer* berkontribusi pada efisiensi waktu pelatihan yang lebih baik dibandingkan model berbasis *GRU+Attention* yang bersifat sequential. Penelitian ini menunjukkan potensi *Transformer* sebagai fondasi yang efektif untuk pengembangan *generative chatbot* berbahasa Indonesia.

Kata kunci: BLEU score, chatbot, self-attention, sequence-to-sequence, transformer

1. Pendahuluan

Chatbot merupakan program komputer yang dirancang untuk berkomunikasi dengan manusia menggunakan teks atau suara sebagai sarana interaksi. Banyak *chatbot* yang sudah ada dibangun sesuai dengan topik dan permasalahan yang ingin dipecahkan oleh seseorang, baik untuk keperluan pribadi ataupun keperluan bisnis yang dapat menggantikan manusia untuk berinteraksi dengan pelanggan atau *customer* [1]. Penerapan chatbot dalam suatu bisnis terbukti dapat meningkatkan *engagement* antara *user* dengan pemangku bisnis karena *chatbot* dapat menyediakan informasi secara lengkap dan mudah dikarenan dapat diakses kapan saja dan dimana saja [2]. Oleh karena itu penelitian tentang pengembangan *chatbot* menjadi topik yang populer dilakukan pada beberapa tahun ini.

Beberapa pendekatan yang digunakan oleh peneliti dalam mengembangkan *chatbot* diantaranya *Rule Based Chatbot*, *Retrieval Chatbot*, dan *Generative Chatbot* [3]. Pada pengembangan *Rule Based Chatbot*, peneliti

menggunakan pendekatan *Natural Language Processing* (NLP) dan *Pattern Matching* untuk menganalisa pertanyaan dari user lalu menghasilkan output yang telah tersedia dalam *database* dengan memanfaatkan fitur *TFIDF* dan *Cosine Similarity* untuk mengukur similaritas antara inputan user dengan aturan yang tersedia dalam *database* [4]. Pada *Retrieval chatbot* pendekatan yang digunakan berupa *machine learning classifier* dan *NLP* [5] dalam mengklasifikasi *intent* pertanyaan dari *user* lalu sistem akan memberikan jawaban sesuai dengan intent yang telah terdeteksi oleh sistem. Sedangkan pada *Generative Chatbot* peneliti banyak memanfaatkan *Deep Learning* dengan teknik *sequence-to-sequence* yang terdiri dari *encoder* dan *decoder* dalam membangun model percakapan. Pada penelitian [6-8] memanfaatkan *RNN* dan varianya seperti *LSTM* dan *GRU* dengan memanfaatkan *word embedding* seperti *Fasttext* sebagai fitur kemudian model *chatbot* tersebut dievaluasi performanya. Pada pengembanganya beberapa peneliti mencoba meningkatkan performa model *Generative Chatbot* dengan menggunakan *attention mechanism* yang digabungkan dengan *LSTM* maupun *GRU* seperti yang dilakukan oleh [9-11] Hal tersebut terbukti dapat meningkatkan performa *chatbot* yang ditunjukan dengan meningkatnya nilai *Bleu Score*.

Seperti yang dilakukan pada penelitian [9] yang mengkombinasikan *LSTM* dengan *Bahdanau Attention Mechanism* yang dilatih dengan menggunakan dialog dataset berbahasa ingris dengan representasi *Bag of Words* mampu menunjukan performa *BLEU-Score* sebesar 0.85. Selain menggunakan corpus berbahasa ingris, penelitian [10] dan [11] menggunakan bahasa indonesia dalam membangun chatbot dengan arsitektur *sequence-to-sequence*. Pada penelitian [10] *chatbot* digunakan dengan mengunakan dataset dari pasangan pertanyaan dan jawaban yang berkaitan dengan *faculty administration* di sebuah universitas. Data diperoleh dengan wawancara dan panduan dari dokumen administrasi yang ada di fakultas sehingga dataset akhir terkumpul sejumlah 1102 pasangan pertanyaan dan jawaban. *Bi-LSTM* dipilih sebagai metode yang digunakan karena sifatnya yang 2 arah sehingga dapat memberikan performa yang lebih dari pada *LSTM*. *Chatbot* tersebut lalu dievaluasi dengan *BLEU-Score* dan *Chatbot Usability Questionnare* (*CUQ*). Hasil penelitian tersebut menunjukan *BLEU-Score* sebesar 0.7724 dan *CUQ* sebesar 62.2.

Pemanfaatan *Attention Mechanism* lalu dimanfaatkan pada penelitian [11] untuk *Chatbot* pada domain layanan kesehatan. Dengan kombinasi *Bi-LSTM* dan *Attention Mechanism* pada *encoder* serta *LSTM* pada *decoder*, model tersebut mampu menghasilkan nilai *BLEU-Score* sebesar 0.85 dengan 6000 *epoch* pada proses pelatihan.

Dengan memasukkan *attention mechanism*, model tidak lagi hanya bergantung pada representasi tersembunyi terakhir, tetapi dapat secara dinamis memfokuskan perhatian pada *token-token* input yang paling relevan untuk setiap tahapan generasi respon *chatbot* serta model dengan *attention mechanism* memiliki kemampuan lebih baik dalam mempertahankan koherensi respons, menangani input panjang, serta meminimalkan kesalahan yang sebelumnya sering muncul akibat *information bottleneck* pada *RNN* tanpa *attention mechanism.*

Meskipun *RNN* dan varian seperti *LSTM* atau *GRU* yang ditambahkan dengan *Attention Mechanism* memberikan performa yang baik untuk beberapa kasus, namun metode tersebut tetap memiliki keterbatasan. Salah satunya adalah *vanishing gradient* [12], yang terjadi ketika proses pelatihan model berulang kali memperbarui parameter, namun gradien yang diterima sangat kecil sehingga pembelajaran model menjadi sangat lambat atau bahkan berhenti. Selain itu, sifat *sequensial processing* di *RNN*, di mana data diproses satu langkah demi satu langkah, membuatnya tidak efisien untuk percakapan panjang karena model harus melalui seluruh urutan input secara berurutan, yang akan berpengaruh pada kecepatan pelatihan [12], [13]. Arsitektur *Transformer* menawarkan solusi untuk kekurangan tersebut. *Transformer* menggunakan mekanisme *self-attention* daripada menggunakan *RNN* dan varianya. Dengan adanya arsitektur ini, maka sebuah data dalam sebuah *sequence* dapat memberikan perhatian lebih kedalam semua data yang berada dalam *sequence* tersebut serta dapat di proses secara parallel sehingga dapat mempercepat waktu pelatihan. Dalam penerapanya, arsitektur *Transformer* sering digunakan untuk beberapa tugas *NLP* seperti *Text Classification* [14] dengan memanfaatkan *encoder* dari *Transformer* [15], [16], mesin penerjemah otomatis [17], dan *question-answering system* [18].

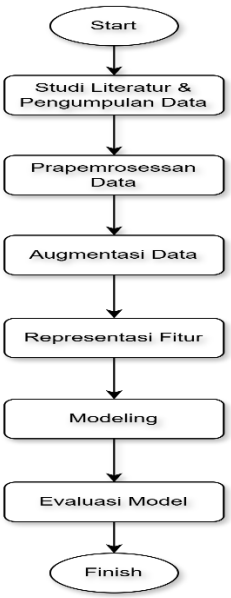
Berdasarkan studi-studi terdahulu, pengembangan *generative chatbot* berbahasa Indonesia masih didominasi oleh penggunaan arsitektur *RNN* beserta variannya, sehingga masih terdapat kesenjangan baik dari sisi performa maupun efisiensi waktu pelatihan. Penelitian ini bertujuan untuk mengembangkan *generative chatbot* berbasis arsitektur *Transformer* serta melakukan perbandingan kinerja menggunakan metrik *BLEU-Score* dengan model *GRU* yang dilengkapi mekanisme *Attention*. Melalui evaluasi komparatif terhadap kedua pendekatan tersebut, penelitian ini mengkaji metode yang memberikan performa paling optimal dari segi nilai *BLEU-Score* dan kecepatan pelatihan model dengan menggukan dataset yang tersedia, sekaligus model yang dikembangkan dapat diterapkan ke dalam sistem yang dapat di *deploy* dan dimanfaatkan secara langsung oleh pengguna.

2. Metode Penelitian

Dalam penelitian pembuatan model *generative chatbot* dengan arsitektur *Transformer* dengan Bahasa Indonesia

DOI : https://doi.org/10.52158/v1x82029

ini dilakukan dengan beberapa tahapan Pengembangan yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Pada Gambar 1 dapat dilihat bahwa alur penelitian dimulai dengan mempelajari studi literatur berupa mempelajari penelitian sebelumnya dan pengumpulan data yang akan digunakan dalam penelitian, lalu di lanjutkan dengan prapemrosesan data, kemudian data tersebut diaugmentasi untuk keperluan analisis, lalu dilakukan pembentukan representasi fitur data agar data telah siap di inputkan kedalam model yang akan dibangun. Setelah data telah siap maka akan dibangunkan model untuk dilakukan pelatihan dan pengujian, lalu tahapan penelitian diakhiri dengan melakukan evaluasi performa model yang dibangun.

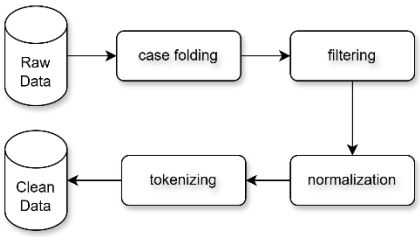
2.1. Studi Literatur dan Pengumpulan Data

Dalam tahapan ini, studi literatur dilakukan dengan mempelajari metode atau algoritma yang digunakan pada penelitian terdahulu dan metode terbaru atau *state-of-the-arts* dengan mempelajari publikasi ilmiah yang diperoleh dari sumber terpercaya.

Pada tahapan ini juga dilakukan pengumpulan data yang digunakan sebagai bahan penelitian. Data yang digunakan pada penelitian ini adalah data question-answering yang digunakan pada penelitian [19]. Data tersebut merupakan data mentah dengan format file csv dengan 3 kolom yaitu *question*, *passage* yang berisi tentang paragraf dimana jawaban tersedia, dan *sequence label* dimana *index* jawaban pada paragraf. Dari dataset tersebut terkumpul data dengan jumlah 2806 pasangan pertanyaan dan jawaban. Dataset tersebut akan dibagi menjadi 80% untuk data latih dan 20% untuk data *testing*.

2.2. Pra-pemrosesan Data

Tujuan dari tahapan ini adalah membersihkan dan mempersiapkan data sesuai dengan format yang diinginkan. Seluruh data pada *dataset* seperti pertanyaan dan jawaban akan di lakukan pra-pemrosesan terlebih dahulu sebelum dilatihkan kedalam *model*. Bererapa tahapan pra-pemrosesan yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Prapemrosesan Data

Tahapan awal yang dilakukan pada pra-pemrosesan data adalah *case folding*, dimana proses ini merubah kata yang menggunakan huruf kapital menjadi huruf kecil. Tahapan selanjutnya adalah *filtering*. Tahap filtering ini bertujuan untuk menghilangkan karakter yang dianggap tidak diperlukan dalam sebuah kalimat seperti karakter ($%/*^#@ dan lain sebagainya. Selanjutnya teks akan di normalisasi. Tahapan normalisasi kata ini bertujuan untuk merubah kata yang dianggap tidak baku menjadi kata yang baku. Penelitian ini menggunakan dataset kamus baku yang dihasilkan oleh [20]. Tahapan terakhir pada pra-pemrosesan data adalah *tokenizing*, dimana tahapan ini merubah kalimat kedalam *token* kecil. Contoh detil tahapan pra-pemrosesan data dapat di lihat pada Tabel 1.

Tabel 1. Tabel Tahapan Prapemrosesan Data

Tahapan	Kalimat
Raw text	brp kapasitas PLTS yg sudah dapat digunakan masyarakat *Yogyakarta sekarang?
Case folding	brp kapasitas plts yg sudah dapat digunakan masyarakat *yogyakarta sekarang?
Filtering	brp kapasitas plts yg sudah dapat digunakan masyarakat yogyakarta sekarang?
Normalization	berapa kapasitas plts yang sudah dapat digunakan masyarakat yogyakarta sekarang?
Tokenizing	['berapa', 'kapasitas', 'plts', 'yang', 'sudah', 'dapat', 'digunakan', 'masyarakat', 'Yogyakarta', 'sekarang', '?']

2.3. Augmentasi Data

Augmentasi data dilakukan pada penelitian ini untuk memperbanyak jumlah data serta untuk mengetahui performa *model* terhadap data yang telah dilatih. Teknik augmentasi yang digunakan pada penelitian ini adalah teknik persamaan kata [21]. Dalam penelitian ini hanya data latih yang berjumlah 80% dari *dataset* original yang didapatkan pada tahap pengumpulan data yang akan diaugmentasikan sejumlah angka tertentu lalu dimasukan kedalam *dataset* baru sehingga terdapat 3

DOI : https://doi.org/10.52158/v1x82029

dataset yang akan diujikan pada penelitian ini yaitu *dataset* original, dataset 7k yang merupakan dari original yang di augmentasi sebanyak 2 kali dari setiap baris data, dan dataset 20k yang merupakan hasil dari dataset original yang di augmentasi sebanyak 8 kali dari dataset original. Pada proses augmentasi akan di pilih 1 kata secara acak pada data *question* lalu kata tersebut di substitusikan dengan kamus sinonimnya dan tahap terakhir adalah penghapusan data apabila terdapat duplikasi data question dalam dataset. Untuk keperluan *validating* dan *testing* digunakan 20% dari tiap *dataset* yang telah dibentuk.

2.4. Representasi Fitur

Pada tahapan ini dilakukan pembuatan representasi fitur kalimat setelah dilakukan pra-pemprosesan kedalam format yang telah ditentukan oleh *model* yang nantinya akan digunakan pada *encoder* dan *decoder* dari arsitektur *Transformer*. Adapun beberapa tahapan yang dilakukan dalam tahapan ini yaitu penambahan *token* spesial seperti *token start of sentence* <*sos*> yang pada penelitian ini dijadikan awalan dari sebuah kalimat atau *sequence* dan *token end of sentence* <*eos*> yang akan dijadikan akhiran dari sebuah kalimat atau *sequence*, token *unknown* <*unk*> dan penambahan *token padding* <*pad*>untuk menyeragamkan panjang input dari data yang telah ditentukan. Tahapan terakhir dari proses ini adalah memetakan *token* tersebut kedalam indeks kata atau *vocabulary* yang telah di bentuk. Sebagai contoh, jika teks yang telah melewati tahapan pra-pemrosesan data adalah *"200 wattjam"*, maka teks tersebut akan dirubah kedalam represntasi fitur dengan panjang yang telah ditentukan misalnya 10 maka contoh detil tahapan pembentukan representasi fitur dapat di lihat pada Tabel 2.

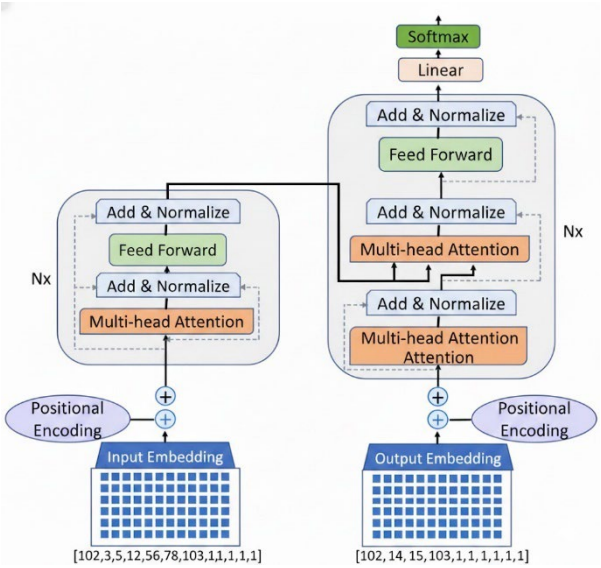
Tabel 2. Tabel Tahapan Pembuatan Repsesntasi Fitur

Tahapan	*Text*
Preprocesing	['200', ' wattjam ']
penambahan <*sos*> dan <*eos*>	['<sos>', '200', ' wattjam', '<eos>']
Penambahan <*pad*>	['<sos>','200','wattjam', '<eos>', '<pad>', '<pad>','<pad>','<pad>','<pad>','<pad>']
Konversi *token* kedalam *index vocabulary*	[102, 14, 15, 103, 1, 1, 1, 1, 1, 1]

2.4. *Modeling*

Pada penelitian ini menggunakan arsitektur *sequence-to-sequence* berbasis *Transformer* yang pertama kali dikenalkan oleh [12] dan *GRU+Attention Mechanism* sebagai metode pembandingnya. Model arsitektur *Transformer* pada penelitian ini dapat dilihat pada Gambar 3.

Seluruh *dataset* dalam penelitian ini akan dilatih dan diujikan kedalam model. Data pertanyaan yang telah melawati tahapan pembentukan representasi fitur seperti pada Tabel 2 akan dimasukan *Embedding Layer* serta yang akan ditambahkan dengan *Positional Embedding* kemudian akan diteruskan kedalam blok *Encoder* yang didalamnya terdapat *self-attention Layer* dan *Feed Forward Layer* dimana seluruh nilai dari parameter tersebut akan di definisikan kedalam pengujian *hyperparameter*. Sedangkan pada blok *Decoder*, data jawaban akan dimasukan kedalam *Embedding Layer* yang akan ditambahkan *Positional Embedding* lalu diteruskan kedalam blok *Decoder* yang terdiri dari 2 *self-attention Layer* dan 1 *Feed Forward Layer*. *Self-attention* pertama digunakan untuk data jawaban dan *Self-attention* kedua digunakan membantu *Decoder* untuk fokus kedalam konteks yang diberikan oleh blok *Encoder* lalu diteruskan kedalam *Feed Forward Layer*. *Output* terakhir akan diteruskan kedalam *Linear Layer* yang akan memetakan dimensi model kedalam jumlah *vocabulary* yang telah dibuat sebelumnya dan *Softmax Layer* sehingga menghasilkan nilai probabilitas output berupa *index* dari *vocabulary* yang telah dibentuk.



Gambar 3. Arsitektur *Transformer*

2.5. Evaluasi Model

Pada tahapan evaluasi, semua model akan dievaluasi dengan *Bleu* (*Bilingual Evaluation Understudy*) [22] yang mana merupakan metric yang biasanya digunakan untuk mengevaluasi kualitas mesin penerjemah otomatis. *Bleu Score* mengukur skor presisi dari *n-gram* termodifikasi (*modified* n-*gram precision score*) atau *pn* yang ditunjukan pada Persamaan 1 antara terjemahan otomatis dengan terjemahan rujukan dan menggunakan konstanta yang dinamakan *brevity penalty* (*BP*) seperti yang ditunjukan pada Persamaan 2.

$$P_n = \frac{\Sigma_{c\in\{candidate\}}\Sigma_{n-gram\in c}Count(n-gram)}{\Sigma_{c'\in\{candidate\}}\Sigma_{n-gram\in c'}Count(n-gram')} \quad (1)$$

$$BP = \begin{cases} 1 \; if \; c > r \\ e^{(1-r/c)} \; if \; c \leq r \end{cases} \quad (2)$$

Pada persamaan 1 dan 2 *c* merupakan jumlah kata dari hasil terjemahan otomatis dan *r* merupakan jumlah kata

DOI : https://doi.org/10.52158/v1x82029

dari rujukan. Untuk menghitung nilai *Bleu Score* dapat dilihat pada Persamaan 3.

$$BLEU = BP.\exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (3)$$

Pada Persamaan 3 wn = 1/N bobot seragam dengan nilai umum (strandar nilai N adalah 4). Pada penelitian ini *Bleu Score* akan digunakan untuk mengevaluasi respon sebenarnya dengan respon yang dihasilkan oleh *model*.

3. Hasil dan Pembahasan

Setelah menjalani serangkaian proses pelatihan dan evaluasi model, diperoleh beberapa hasil yang menunjukkan performa dari model *sequence-to-sequence* berbasis *Transformer* dan *sequence-to-sequence* berbasis *GRU* yang telah ditambahkan *Attention Mechanism*.

3.1. Augmentasi *Dataset*

Augmentasi data bertujuan untuk memperbanyak data yang digunakan untuk membangun model. Teknik augmentasi yang digunakan adalah dengan menggunakan sinonim atau persamaan kata dengan cara mengambil 1 kata secara acak dalam kalimat, lalu menggantinya dengan kata yang terdapat pada kamus sinonim. Adapun hasil augmentasi dataset dapat dilihat pada Tabel 3.

Tabel 3. Tabel Hasil Augmentasi Dataset

Dataset	Jumlah pertanyaan yang Diaugmentasi	Jumlah Total Data
Original	0	2806
7k	2	7775
20k	8	20109

Pada hasil augmentasi, proses ini hanya dilakukan untuk memperkaya data sehingga model mampu dievaluasi performanya terhadap kuantitas data yang diberikan. Meskipun metode yang digunakan adalah substitusi kata dengan sinonim yang tidak sepenuhnya mempertimbangkan validitas semantic dan berpotensi bias, Teknik ini mampu meningkatkan ukuran sampel sehingga model tidak mudah menghafal pola spesifik dari data latih yang terbatas.

3.2. *Modeling*

Hasil *training*, *evaluation*, dan *testing* pada pemodelan *Generative Chatbot* ini dipengaruhi oleh beberapa *hyperparameter* pada model. Pengujian *hyperparameter* dilakukan untuk mendapatkan *hyperparameter* yang paling optimal pada setiap model. Detil nilai *hyperparameter* yang diujikan untuk model yang dibangun pada penelitian ini dapat dilihat pada Tabel 4.

Selain Tabel 4, terdapat juga *hyperparameter* yang bernilai tetap yang digunakan untuk kedua model tersebut yaitu nilai *dropout* sebesar 0.2 dan jumlah *epoch* sebesar 50.

Tabel 4. Tabel Nilai *Hyperparamer* yang Diujikan

Model	*Hyperparameter*	*Value*
Transformer	*Batch Size (BS)*	64, 128, 256
	Learning Rate (LR)	1e-3, 3e-5, 5e-5
	Hidden Size (HS)	256, 512
	Encoder & Decoder (N-stack)	3, 6, 8
	Attention Head (AH)	4, 8, 16
	FFN Layer (FFN)	512, 1024, 2048
GRU	*Batch Size (BS)*	64, 128, 256
	Learning Rate (LR)	1e-3, 1e-4, 1e-5
	Hidden Size (HS)	512, 1024
	Embedding Size (ES)	512, 1024

Setelah dilakukan serangkaian eksperimen berdasarkan data dari Tabel 4 untuk setiap model yang dibangun, maka semua hasil dievaluasi berdasarkan nilai *BLEU-Score* dan lama proses pelatihan. Hasil *hyperparameter* optimal pada model *Transformer* untuk *dataset* original, 7k, dan 20k pada penelitian ini dapat dilihat pada Tabel 5 - 7.

Tabel 5. Hasil Pengujian *Transformer* Terbaik Untuk Dataset Original

Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.0001	4.163	1 Menit
Batch Size	64		15 Detik
Hidden Dimension	512		
Attention Head	4		
N-Stack	3		
FFN	1024		

Tabel 6. Hasil Pengujian *Transformer* Terbaik Untuk Dataset 7k

Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.0001	66.145	5 Menit
Batch Size	64		4 Detik
Hidden Dimension	512		
Attention Head	16		
N-Stack	3		
FFN	2048		

Tabel 7. Hasil Pengujian *Transformer* Terbaik Untuk Dataset 20k

Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.0001	63.08	4 Menit
Batch Size	64		56 Detik
Hidden Dimension	256		
Attention Head	8		
N-Stack	3		
FFN	1024		

Sedangkan hasil *hyperparameter* optimal pada model *GRU* dengan *Attention Mechanism* untuk dataset original, 7k, dan 20k pada penelitian ini dapat dilihat pada Tabel 8 - 10.

DOI : https://doi.org/10.52158/v1x82029

Tabel 8. Hasil Pengujian *GRU* Terbaik Untuk Dataset Original

Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.001	0.761	3 Menit
Batch Size	256		38 Detik
Embedding Size	1024		
Hidden Size	1024		

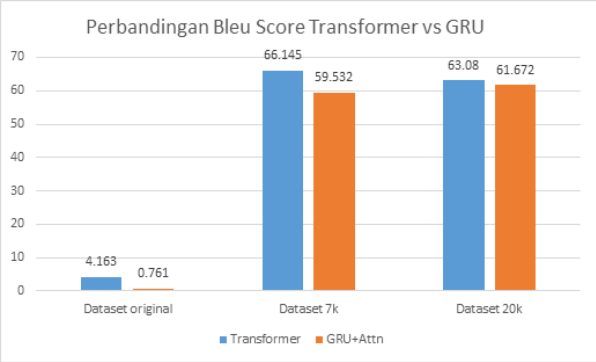
Tabel 9. Hasil Pengujian *GRU* Terbaik Untuk Dataset 7k

Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.001	59.532	9 Menit
Batch Size	256		40 Detik
Embedding Size	1024		
Hidden Size	1024		

Tabel 10. Hasil Pengujian *GRU* Terbaik Untuk Dataset 20k

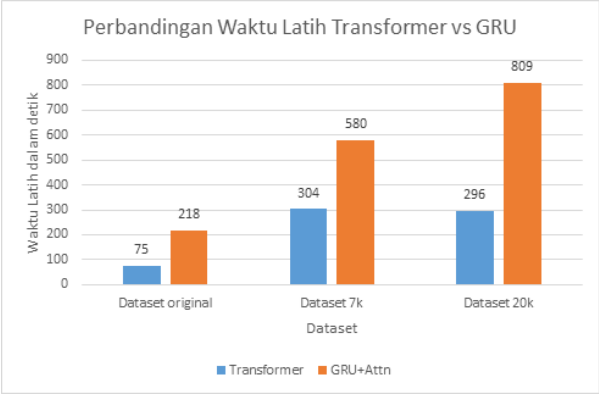
Hyperparameter Terbaik	Nilai	*Bleu Score*	Waktu Pelatihan
Learning Rate	0.001	61.672	13 Menit
Batch Size	128		29 Detik
Embedding Size	512		
Hidden Size	1024		

Untuk mengetahui perbandingan antara performa model baik model *Generative Chatbot* berbasis *Transformer* maupun berbasis *GRU* dan mekanisme *Attention* yang telah dibuat baik dari segi nilai *Bleu Score* maupun waktu pelatihan model dapat dilihat pada Gambar 4 dan 5.



Gambar 4. Perbandingan *BLEU Score* Untuk Setiap *Dataset*

Dari Gambar 4 terlihat bahwa model *Transformer* selalu dapat mengungguli performa model *GRU+Attention*. Hal ini menunjukan bahwa mekanisme *self-attention* pada *Transformer* bekerja lebih efektif daripada metode *Reccurent* seperti *GRU*. Namun terlihat bahwa pada dataset original baik model *Transformer* maupun *GRU+Attention* menunjukan performa *BLEU Score* yang sangat rendah. Hal ini terjadi karena data yang digunakan terlalu sedikit untuk model *sequence to sequence* dimana model tersebut merupakan turunan dari *deep learning* yang terkenal akan *data hungry* sehingga pengembanganya tergantung pada jumlah dan keberagaman data yang relatif banyak agar dapat mencapai performa optimalnya. Serta untuk ukuran dataset original, baik model *Transformer* dan model *GRU+Attention* memiliki kapasitas parameter yang besar sehingga model cenderung *overfitting* untuk data yang terlalu sedikit. Kelemahan ini dapat diperbaiki dengan cara menambah jumlah training dataset sehingga model mampu belajar dengan lebih optimal. Hal ini selaras dengan hasil pengujian pada dataset 7k dan dataset 20k dimana performa model tersebut dapat dikatakan bagus.



Gambar 5. Perbandingan Waktu Pelatihan Model

Pada dataset 7k dan 20k menunjukan bahwa pemanfaatan teknik data augmentasi dan jumlah augmentasi dapat meningkatkan performa model, meskipun pada model *Transformer* nilai *Bleu-Score* tertinggi didapatkan pada dataset 7k. Sedangkan pada Gambar 5 terlihat jelas bahwa model *Transformer* mampu mengungguli kecepatan pelatihan pada dataset original, dataset 7k dan dataset 20k. Hal lain juga terlihat semakin banyak dataset yang digunakan maka waktu pelatihan pun akan bertambah. Namun, pada pada dataset 7k dan dataset 20k model *Transformer* memiliki waktu pelatihan paling cepat meskipun dengan banyaknya parameter pada model tersebut. Pada hal ini terlihat bahwa peran *sequential processing* pada *GRU* sangat mempengaruhi lamanya waktu pelatihan model, berbeda dengan *Transformer* yang di process secara parallel sehingga mampu mempersingkat waktu pelatihan tanpa mengorbankan performa model.

3.3. Hasil Respon

Untuk mengetahui respon yang diberikan oleh *chatbot*, maka setiap model yang telah dibuat dan dilatih dengan dataset *training* menggunakan *hyperparameter* terbaiknya maka kedua model tersebut akan diujikan dengan dataset *testing* untuk mengetahui perbedaan respon yang dihasilkan oleh setiap model. Tabel 11 - 13 menunjukan perbandingan respon *chatbot* dari *Transformer* dan *GRU* yang telah dilatih dengan setiap subset dataset.

Dari Tabel 11 - 13 terlihat bahwa model *Transformer* mampu memberikan respon yang mendekati aktual respon terutama pada respon yang mengandung kalimat yang panjang, sedangkan model *GRU+Attention* terlihat

DOI : https://doi.org/10.52158/v1x82029

kesulitan pada respon dengan kalimat yang panjang. Hal ini menunjukan bahwa mekanisme *self-attention* pada arsitektur *Transformer* mampu bekerja lebih baik daripada mekanisme *attention* yang dikombinasikan pada model *GRU*. Namun pada beberapa data terlihat bahwa model *GRU+Attention* lebih baik dalam menghasilkan respon jika respon tersebut terdiri dari kata yang lebih sedikit. Hal ini menunjukan bahwa penambahan data dengan teknik augmentasi dapat meningkatkan performa pada model *GRU+Attention* sehingga dapat mengimbangi performa model *Transformer* dari segi nilai *BLEU Score*. Meskipun demikian, respon yang dihasilkan oleh model *Transformer* pada data dengan kalimat yang pendek pada dasarnya masih mempunyai konteks yang mirip dengan respon sebenarnya.

Tabel 11. Tabel Perbandingan Respon Model Untuk Dataset Original

Kategori	Isi
Pertanyaan	sejak kapan realisasi penanaman modal dalam negeri(pmdn) dan penanaman modal asing(pma) di riau terus menurun?
Aktual Respon	tahun 1999
Transformer Respon	tahun 2005
GRU Respon	tahun /

Tabel 12. Tabel Perbandingan Respon Model Untuk Dataset 7k

Kategori	Isi
Pertanyaan	berapa jarak kota tua kopenick dari kota berlin mitte?
Aktual Respon	sekitar 15 kilometer
Transformer Respon	sekitar 15 kilometer
GRU Respon	seputar 15 kilometer

Tabel 13. Tabel Perbandingan Respon Model Untuk Dataset 20k

Kategori	Isi
Pertanyaan	berapakah ketinggian situs tinggihari dari muka laut?
Aktual Respon	antara 700 - 1.000 meter di atas permukaan laut
Transformer Respon	antara 700 - 1.000 meter di atas permukaan laut
GRU Respon	antara 700 - 1.000 meter di atas permukaan bahar

Pada *dataset* dengan ukuran kecil, model tidak memperoleh cukup variasi data untuk mempelajari distribusi respon secara menyeluruh, sehingga kemampuan generalisasi pada tahap inferensi menjadi terbatas. Kondisi ini menyebabkan model cenderung menghasilkan respons yang bersifat umum atau mengalami perbedaan struktur kalimat dibandingkan *actual respons* dari dataset. Selain itu, *BLEU-Score* yang mengandalkan pencocokan *four-gram* secara eksak menjadi semakin sensitif pada dataset dengan jumlah referensi terbatas, karena variasi kecil dalam pemilihan kata atau urutan token dapat menghasilkan penalti yang signifikan. Akibatnya, meskipun respons yang dihasilkan oleh model masih relevan secara kontekstual, nilai *BLEU-Score* tetap rendah. Temuan ini menunjukkan bahwa rendahnya nilai *BLEU-Score* pada model *Transformer* dan *GRU+Attention* lebih merefleksikan keterbatasan data pelatihan dan karakteristik metrik evaluasi, dibandingkan kelemahan dari arsitektur model yang digunakan.

4. Kesimpulan

Penelitian ini menggunakan arsitektur *Transformer* dalam membangun *generative chatbot* berbahasa indonesia dengan 3 subset dataset yaitu dataset 2k, 7k, dan 20k yang merupakan hasil augmentasi dari dataset penelitian sebelumnya. model *Transformer* mampu secara konsisten mengungguli model *GRU+attention* baik dari segi nilai *BLEU-Score* maupun kecepatan pelatihan. Hal ini diperoleh dari hasil eksperimen yang menunjukan nilai *BLEU-Score* untuk dataset 2k, 7k, dan 20k pada model *Transformer* mampu memperoleh nilai sebesar 4.163, 66.145, dan 63.08 berturut-turut. sedangkan *GRU+Attention* hanya memperoleh nilai sebesar 0.761, 59.532. dan 61.672 untuk ketiga subset dataset yang sama. Dari segi kecepatan pelatihan model *Transformer* menunjukan waktu pelatihan pada dataset 2k, 7k, dan 20k adalah 1 menit 15 detik, 5 menit 4 detik dan 4 menit 56 detik secara berturut turut yang mana menunjukan waktu lebih efektif dari pada model *GRU+Attention* hanya memperoleh waktu 3 menit 38 detik, 9 menit 40 detik dan 13 menit 29 detik. Temuan ini mengindikasikan bahwa ukuran dan kualitas *dataset* memiliki pengaruh yang signifikan terhadap performa model, serta bahwa augmentasi data efektif dalam meningkatkan kualitas respons. Namun demikian, performa *BLEU Score* yang kecil pada dataset 2k menunjukan bahwa baik model *Transformer* dan *GRU+Attention* masih memiliki karakteristik model *deep learning* pada umumnya yaitu membutuhkan data banyak mencapai performa yang optimal. Sebagai arah penelitian selanjutnya, disarankan untuk mengintegrasikan *pretrained language model* serta menerapkan *contextual embedding* seperti *BERT* atau model sejenisnya guna meningkatkan kemampuan generalisasi model, khususnya pada skenario *low-resource*. Selain itu, eksplorasi teknik augmentasi yang lebih semantik serta evaluasi menggunakan metrik tambahan yang berorientasi pada kualitas percakapan juga menjadi peluang penelitian lanjutan untuk meningkatkan keandalan sistem *chatbot generatif*.

Daftar Rujukan

[1] E. Adamopoulou and L. Moussiades, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, no. October, p. 100006, 2020, doi: 10.1016/j.mlwa.2020.100006.

[2] G. Wijayanto, Y. Rivai, A. Alvionita, S. W. Wildah, and J. Jushermi, “Evaluation of the Effect of Chatbot in Improving Customer Interaction and Satisfaction in Online Marketing in Indonesia,” *West Science Business and Management*, vol. 1, no. 04, pp. 304–310, Sep. 2023, doi: 10.58812/wsbm.v1i04.248.

DOI : https://doi.org/10.52158/v1x82029

[3] G. Caldarini, S. Jaf, and K. McGarry, "A Literature Survey of Recent Advances in Chatbots," *Information*, vol. 13, no. 1, p. 41, Jan. 2022, doi: 10.3390/info13010041.

[4] A. Elcholiqi and A. Musdholifah, "Chatbot in Bahasa Indonesia using NLP to Provide Banking Information," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 1, p. 91, Jan. 2020, doi: 10.22146/ijccs.41289.

[5] Zein Hanni Pradana, Hanin Nafi'ah, and Raditya Artha Rochmanto, "in Chatbot-based Information Service using RASA Open-SourceFrameworkin Prambanan Temple Tourism Object," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 656–662, Aug. 2022, doi: 10.29207/resti.v6i4.3913.

[6] Y. S. H. Langgeng, "Long short-term memory-based chatbot for vocational registration information services," *Journal of Applied Data Sciences*, vol. 4, no. 4, pp. 414–430, Dec. 2023, doi: 10.47738/jads.v4i4.128.

[7] M. Ilyas Tri Khaqiqi, N. H. Harani, and C. Prianto, "Performance Analysis and Development of QnA Chatbot Model Using LSTM in Answering Questions," *The Indonesian Journal of Computer Science*, vol. 12, no. 3, Jun. 2023, doi: 10.33022/ijcs.v12i3.3249.

[8] Fahmi Yusron Fiddin, A. Komarudin, and M. Melina, "Chatbot Informasi Penerimaan Mahasiswa Baru Menggunakan Metode FastText dan LSTM," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 33–39, Feb. 2024, doi: 10.52158/jacost.v5i1.648.

[9] P. Choudhary and S. Chauhan, "An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism," *Decision Analytics Journal*, vol. 9, p. 100359, Dec. 2023, doi: 10.1016/j.dajour.2023.100359.

[10] Suryani and Mustakim, "An Intelligent Chatbot for Faculty Administration Using Bidirectional LSTM and Seq2Seq Architecture," in *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, IEEE, Jun. 2024, pp. 226–231. doi: 10.1109/SIML61815.2024.10578161.

[11] M. A. Khadija, W. Nurharjadmo, and Widyawan, "Deep Learning Generative Indonesian Response Model Chatbot for JKN-KIS," in *2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, IEEE, Aug. 2022, pp. 70–74. doi: 10.1109/APICS56469.2022.9918686.

[12] A. Vaswani *et al.*, "Attention Is All You Need," *Adv Neural Inf Process Syst*, vol. arXiv:1706, Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[13] S. M. Jain, "Introduction to Transformers," in *Introduction to Transformers for NLP*, Berkeley, CA: Apress, 2022, pp. 19–36. doi: 10.1007/978-1-4842-8844-3_2.

[14] L. H. Suadaa, I. Santoso, and A. T. B. Panjaitan, "Transfer Learning of Pre-trained Transformers for Covid-19 Hoax Detection in Indonesian Language," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 317, Jul. 2021, doi: 10.22146/ijccs.66205.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, pp. 4171–4186, 2018, doi: https://doi.org/10.48550/arXiv.1810.04805

[16] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," Online. [Online]. Available: https://huggingface.co/

[17] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, "Automatic Correction of Indonesian Grammatical Errors Based on Transformer," *Applied Sciences*, vol. 12, no. 20, p. 10380, Oct. 2022, doi: 10.3390/app122010380.

[18] K. Nassiri and M. Akhloufi, "Transformer models used for text-based question answering systems," *Applied Intelligence*, vol. 53, no. 9, pp. 10602–10635, May 2023, doi: 10.1007/s10489-022-04052-8.

[19] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," *ArXiv*, pp. 843–857, Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.05387

[20] A. S. Nikmatun, A. Y. Winatmoko, A. A. Septiandri, and A. Jamal, "Colloquial Indonesian Lexicon," in *2018 International Conference on Asian Language Processing (IALP)*, Bandung: IEEE, Nov. 2018, pp. 226–229. doi: 10.1109/IALP.2018.8629151.

[21] Abdurrahman and A. Purwarianti, "Effective Use of Augmentation Degree and Language Model for Synonym-based Text Augmentation on Indonesian Text Classification," in *2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)*, Bali: IEEE, Oct. 2019, pp. 217–222. doi: 10.1109/ICACSIS47736.2019.8979733.

[22] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Morristown, NJ, USA: Association for Computational Linguistics, 2002, p. 311. doi: 10.3115/1073083.1073135.