Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2024117984
Vol. 11, No. 5, Oktober 2024, hlm. 1009-1016 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

ARSITEKTUR SISTEM PERCAKAPAN OTOMATIS BERBAHASA INDONESIA DENGAN NORMALISASI BAHASA INFORMAL MENJADI BAKU

Muhammad Fathur Rahman Khairul[1], Rizal Setya Perdana[2]

[1,2]Universitas Brawijaya, Malang
Email: [1]fathurrk@student.ub.ac.id, [2]rizalespe@ub.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 10 November 2023, diterima untuk diterbitkan: 30 Oktober 2024)

Abstrak

Komunikasi merupakan hal yang paling penting dalam kehidupan sehari-hari. Setiap orang berkomunikasi dengan cara mereka berdasarkan latar belakang serta kedekatan antar pembicara. Oleh karena itu, perkembangan bahasa informal terjadi sangat cepat dan tidak jarang menciptakan kata-kata baru sebagai pengganti bahasa formal. Hal ini menjadi masalah jika dilihat dari perspektif pemrosesan bahasa alami (NLP). NLP umumnya hanya dapat dilakukan dengan bahasa yang formal dan tidak mampu menginterpretasikan makna dari kalimat informal. Maka dari itu, penulis mengusulkan pendekatan untuk memungkinkan mesin memahami bahasa informal dengan melakukan normalisasi bahasa infomal menjadi baku dengan memanfaatkan NLP. Pendekatan yang dilakukan akan melatih model *pre-trained* GPT-2 berbahasa Indonesia dengan data *parallel corpus* untuk memahami makna dari bahasa informal dan mampu menerjemahkannya ke dalam bentuk baku. Melalui eksperimen yang dilakukan, pendekatan ini mencapai tingkat akurasi 91% dan dapat menerjemahkan bahasa informal dengan baik. Performa ini dapat diraih dengan konfigurasi hiperparameter yaitu Adam *optimizer* dengan *learning rate* 1e-4, *batch size* sebesar 16 dan *dropout rate* sebesar 0,5.

Kata kunci: *NLP, fine-tuning, GPT-2, normalization, machine translation, deep learning*

ARCHITECTURE OF AUTOMATIC CONVERSATION SYSTEM IN INDONESIAN LANGUAGE WITH NORMALIZATION OF INFORMAL LANGUAGE INTO STANDARD LANGUAGE

Abstract

Communication is the most essential thing in daily life. Everyone communicates in their own way based on their background and the closeness between speakers. Thus, the development of informal language occurs quickly and it is often to create new words as a substitute for formal language. This is an issue from a natural language processing (NLP) perspective. NLP generally only works with formal language and is unable to interpret the meaning of informal sentences. Therefore, the authors propose an approach to enable machines to understand informal language by normalizing the informal language to standard by utilizing NLP. The approach will train a pre-trained GPT-2 model in Indonesian with parallel corpus data to understand the meaning of informal language and be able to translate it into standardized form. Through experiments, the method achieved 91% accuracy and can translate informal language well. This performance can be achieved with a hyperparameter configuration, namely Adam optimizer with a learning rate of 1e-4, batch size of 16 and dropout rate of 0.5.

***Key*words**: *NLP, fine-tuning, GPT-2, normalization, machine translation, deep learning*

1. PENDAHULUAN

Perkembangan bahasa informal di Indonesia terjadi sangat cepat. Hal ini terjadi karena sebagian besar orang menulis tanpa memerhatikan struktur kalimat serta kesalahan pengetikan (Wibowo dkk., 2020). Tujuan dari penggunan bahasa informal ialah untuk menciptakan komunikasi yang efektif dan menjadi simbol kedekatan antar pembicara (Arifin dkk., 2022). Pemahaman atas bahasa informal tersebut tentunya menjadi faktor yang menentukan baik tidaknya komunikasi yang terjalin. Interpretasi yang tidak selaras terhadap bahasa informal rentan terjadi diantara individu dengan latar belakang yang berbeda (Andersen, 2005).

Hal ini menjadi masalah ketika dikaji melalui perspektif pemrosesan bahasa alami. Bahasa informal menjadi tantangan dalam mengembangkan model karena umumnya model yang dikembangkan hanya

dapat memahami bahasa formal (Wang dkk., 2019). Selain itu, keterbatasan pustaka serta *dataset* berbahasa Indonesia menjadi tantangan utama dalam pengembangan model pemrosesan bahasa alami berbahasa Indonesia (Wibowo dkk., 2020).

Normalisasi bahasa perlu dilakukan pada kata tidak baku. Hal ini menjadi penting karena dapat membantu pengurai kata (*language parser*) untuk memahami makna leksikal dengan lebih baik (Aw dkk., 2006). Normalisasi yang dilakukan umumnya melibatkan konversi istilah tidak baku dan singkatan menjadi bentuk baku yang lebih mudah dimengerti (Salsabila dkk., 2019). Berbagai pendekatan dimungkinkan untuk melakukan proses ini, salah satunya adalah pendekatan *deep learning* dalam pemrosesan bahasa alami.

Penelitian yang dilakukan oleh Wibowo dkk. menunjukkan perbandingan antara beberapa pendekatan yang mungkin dilakukan dalam normalisasi bahasa informal antara lain pendekatan *dictionary*, statistik dan *machine learning*. Melalui penelitian tersebut diperoleh bahwa pendekatan *machine learning* seperti GPT-2 memiliki performa terbaik diantara beberapa metode lainnya. Terdapat fakta lain berupa metode statistikal merupakan metode yang menggunakan sumber daya paling minim (Wibowo dkk., 2020).

Penelitian mengenai pengaruh normalisasi bahasa informal menjadi baku dalam bahasa Indonesia telah dilakukan sebelumnya oleh beberapa peneliti. Terdapat penelitian yang membahas tentang pengaruh *colloquial words* atau bahasa sehari-hari terhadap metode deteksi *spam* pada *comment* di *Instagram* (Salsabila dkk., 2019). Penelitian tersebut mengimplementasikan metode *dictionary* dalam normalisasi bahasa informal menjadi baku. Melalui penelitian ini diperoleh bahwa tidak ada pengaruh yang signifikan terhadap normalisasi yang dilakukan pada bahasa informal dalam proses *spam detection*. Namun hal ini tidak menutup kemungkinan adanya pengaruh normalisasi terhadap tugas lain. Disisi lain, normalisasi dengan pendekatan *dictionary* dapat meningkatkan akurasi proses kategorisasi complain pada Twitter (Hanafiah dkk., 2017). Penelitian tersebut membuktikan bahwa pada beberapa bidang tertentu, normalisasi dapat memiliki pengaruh yang signifikan.

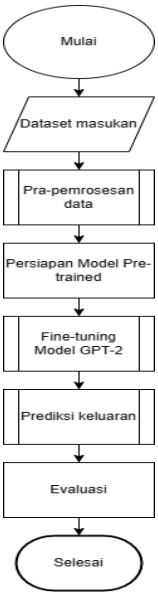
Bidang riset NLP berkembang pesat setelah penemuan arsitektur *transformer* (Vaswani dkk., 2017). Pemanfaatan arsitektur tersebut juga dibuktikan dengan penciptaan model yang bergantung pada *transformer* seperti GPT dan BERT. Pendekatan ini memiliki performa yang lebih baik dari *transformer* dan menjadi *state-of-the-art* untuk tugas NLP (Gillioz dkk., 2020). Selain itu, terdapat implementasi Transformer dalam skema *tagger* dan *generator* (Madaan dkk., 2020) dengan menempatkan *tag* pada kata-kata yang dianggap tidak sopan. Sebagai kekurangannya, pendekatan *tagger generator* ini memiliki kendala yaitu tidak mampu menangani kesalahan penulisan (*typo*). Selain itu, terdapat tantangan yang mungkin dihadapi ketika mengembangkan model NLP yaitu kesalahan semantik dengan adanya homograf ataupun polisemi yaitu satu kata dengan lebih dari satu makna (Banitz, 2020).

Melalui penelitian ini, peneliti mengusulkan metode normalisasi bahasa informal menjadi baku dengan melakukan *fine-tuning* pada model GPT-2 untuk keperluan *machine translation*. Pendekatan ini memiliki kelebihan antara lain adalah menggunakan sumberdaya seperti komputasi dan memori yang sedikit namun memiliki akurasi prediksi yang cukup baik (Wibowo dkk., 2020). Perbedaan penelitian ini dengan penelitian referensi ialah pada model GPT-2 yang digunakan. Penelitian ini menggunakan model GPT-2 yang sudah berbahasa Indonesia, sedangkan penelitian referensi menggunakan model GPT-2 *default* pada *Huggingface*.

Penelitian ini akan membahas langkah-langkah dan hasil dalam pelatihan model *deep learning* yang tepat untuk normalisasi bahasa informal menjadi baku. Selain itu, akan dibahas pengaruh dari tiap hiperparameter terhadap performa model yang dibangun. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam perkembangan pemrosesan bahasa alami berbahasa Indonesia.

2. METODE PENELITIAN

Penelitian ini melakukan *fine-tuning* terhadap *pre-trained* model dengan menggunakan *parallel corpus* dengan tujuan memberi pemahaman terhadap model untuk melakukan normalisasi bahasa. Detil mengenai alur penelitian dapat dilihat melalui diagram alir pada Gambar 1.



Gambar 1 Diagram alir penelitian

Dataset akan melewati *pre-processing* untuk menghasilkan *dictionary* berisi *embedding* yang merupakan representasi numerik dari data teks.

Kemudian *embedding* tersebut akan menjadi masukan dan label ke dalam model yang sudah didefinisikan. Dengan begitu, model diharapkan dapat melakukan prediksi sesuai tugas normalisasi yang diberikan.

2.1 Dataset

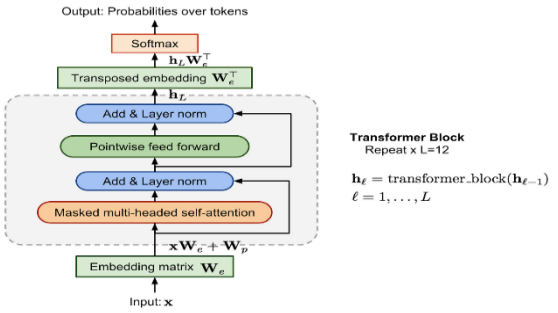
Dataset yang digunakan pada penelitian ini merupakan *parallel corpus* dari penelitian STIF Indonesia (Wibowo dkk., 2020) yang merupakan pasangan kalimat non baku dan baku berbahasa Indonesia yang memiliki dua label yaitu informal dan formal. Data ini terdiri dari 1922 baris data latih dan 214 baris data uji. Data ini diperoleh dari Twitter dengan teknik *scraping* yang kemudian memasuki tahap *pre-processing* yang mana terdapat *case folding*, *masking* dan *text transformation*.

2.2 Pra-Pemrosesan Data

Sebelum memasuki model, data akan melewati proses *pre-processing* untuk menyesuaikan struktur masukan agar dapat dimengerti oleh model. Bahasa informal dikonversi menggunakan *tokenizer* menjadi *embedding input*, sedangkan bahasa formal atau baku dikonversi menjadi *embedding* label. Setiap *embedding* memiliki *attention mask* yang berisi bilangan biner untuk memisahkan data yang harus diolah dan tidak. Seluruh parameter tersebut kemudian dimuat ke dalam *data loader* dengan format *dictionary* dengan *key* antara lain adalah *input*, *attention mask* dari *input*, label, dan *attention mask* dari label.

2.3 Model

Penelitian ini menggunakan *pre-trained* model GPT-2 *Small Indonesian* 522M yang diperoleh melalui *Huggingface*. Model ini sama seperti GPT-2 pada umumnya yaitu dipergunakan untuk keperluan text generation. Namun pada penelitian ini, model dilatih dengan *parallel corpus* sehingga teks yang di *generate* terarah menjadi bahasa formal dari masukan informal. Arsitektur GPT-2 dapat dilihat pada Gambar 2.



Gambar 2 Arsitektur GPT-2

Model ini menggunakan fungsi aktivasi GeLU New yang merupakan varian dari GeLU (Gaussian Error Linear Unit) yang dirancang untuk pemrosesan bahasa alami. Fungsi GeLU New memberikan keluaran yang mirip dengan fungsi GeLU standar, tetapi dengan beberapa perbedaan. Fungsi ini menambahkan pengali skala (0.5) sehingga nilai output lebih mendekati rentang (0, 1) dibandingkan dengan GeLU standar, yang dapat membantu dalam beberapa kasus pemrosesan bahasa alami.

GeLU *New* juga menambahkan penyesuaian kecil (x + 0.044715 * x^3) sebelum melewati fungsi tanh, yang dapat membantu mengatasi masalah dari fungsi GeLU standar yang lebih lambat dalam beberapa implementasi perangkat keras. Persamaan (1) merupakan formula matematis dari fungsi aktivasi GeLU *New*. x merupakan input dari fungsi serta $tanh$ merupakan tangen hiperbolik yaitu sebuah fungsi aktivasi

$$GeLU(x) = \frac{1}{2}x\,(1 + tanh\sqrt{\frac{2}{\pi}(x + 0{,}0447(x^3))}) \quad (1)$$

2.4 Fine-tuning GPT-2

Fine-tuning dilakukan untuk menggunakan pengetahuan dasar yang telah dimiliki oleh model dan melakukan sedikit modifikasi terhadap prediksi yang akan dilakukan oleh model. Proses ini bertujuan untuk memanfaatkan bobot yang telah dimiliki oleh model dan kemudian melatih model dengan *dataset* tertentu untuk mengarahkan prediksi model. Pada kasus ini dilakukan dengan memberi masukan berupa pasangan kalimat informal dan formal sehingga jika model diberikan masukan kalimat informal akan mengeluarkan bentuk formal dari kalimat tersebut.

Fine-tuning model diawali dengan menginisiasi setiap pustaka yang dibutuhkan antara lain model, *tokenizer*, *optimizer*,dropout, dan *scheduler*. Model akan memasuki perulangan sebanyak jumlah *epoch* dalam proses pelatihan. Setiap *epoch* berisi perulangan untuk seluruh data latih, perulangan untuk seluruh data validasi, dan perhitungan evaluasi metrik untuk penyesuaian nilai *learning rate* dari *optimizer*.

Pada proses pelatihan, model akan melakukan prediksi sesuai *input* yang diberikan. Kemudian hasil prediksi tersebut akan dibandingkan dengan data label untuk menghitung nilai loss dan akurasi yang diperoleh. Setelah itu akan dilakukan *backpropagation* untuk menghitung nilai gradien dari prediksi yang dilakukan. Nilai gradien tersebut akan digunakan sebagai parameter dalam melakukan pembaruan bobot pada model. Setelah itu akan dilakukan dropout untuk menghilangkan pengaruh beberapa persen dari seluruh bobot terhadap prediksi dengan memberi nilai nol pada bobot.

Kemudian model memasuki proses validasi yang mana model akan diberikan masukan berupa data yang belum pernah dipelajari sebelumnya untuk melihat seberapa jauh model dapat melakukan generalisasi. Prediksi tersebut akan dibandingkan dengan data label untuk memperoleh nilai *validation loss* dan *validation accuracy*. Jika model memiliki

nilai *validation accuracy* yang tinggi dan nilai *validation* yang rendah, maka model dapat dikatakan sudah mampu memahami dan melakukan generalisasi terhadap data yang diberikan.

Pada akhir *epoch*, nilai seluruh metrik akan dirata-ratakan untuk memperoleh nilai metrik untuk tiap *epoch*. Nilai metrik tiap *epoch* akan dicetak dan dijadikan parameter dalam proses pembaruan *learning rate*. Pembaruan *learning rate* akan dilakukan oleh *scheduler* dengan kriteria tertentu jika tidak terjadi perubahan yang signifikan terhadap performa model.

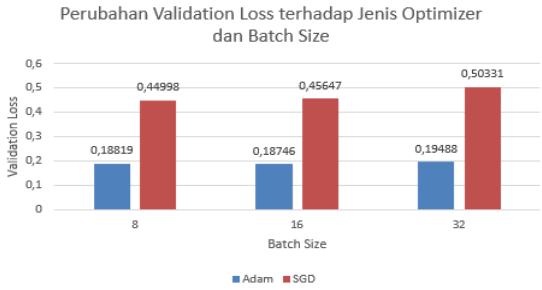
Melalui proses perulangan tersebut, model akan memperoleh pengetahuan tentang normalisasi bahasa informal menjadi baku dalam bahasa Indonesia. Keluaran dari proses *fine-tuning* adalah model dengan bobot yang sudah diperbarui sesuai tugas yang diberikan. Kualitas proses ini sangat bergantung pada hiperparameter yang ditentukan serta variasi pada data masukan. Sehingga banyak faktor yang perlu diperhatikan dalam melakukan *fine-tuning* model.

3. PENGUJIAN DAN ANALISIS

Pada bagian ini akan dijelaskan hasil pengujian untuk tiap konfigurasi hiperparameter yang dilakukan. Hiperparameter yang diuji antara lain jenis *optimizer*, *learning rate*, *dropout rate* dan juga *batch size*. Proses pelatihan akan menggunakan data latih dan validasi yang sama dan dilakukan sebanyak 10 *epoch*. Hal ini dikarenakan performa model GPT-2 saat *fine-tuning* konvergen pada epoch ke-3 sampai ke-5 (Wibowo dkk., 2020). Nilai yang dicantumkan pada tabel selanjutnya merupakan nilai yang diperoleh pada *epoch* terakhir untuk tiap pengujian. Parameter yang digunakan pada tabel antara lain ialah ‘optim’ untuk *optimizer*, ‘lr’ untuk *learning rate*, ‘dr’ untuk *dropout rate* dan ‘bs’ untuk *batch size*. Contoh hasil pengujian dapat dilihat pada tabel. Model dapat melakukan normalisasi pada bahasa Indonesia informal yang umum digunakan seperti “ngga”, “nomer” dan “min”. Disisi lain, model tidak dapat melakukan normalisasi kata “hp” ke dalam bentuk bakunya yaitu “ponsel”.

Tabel 3.1 Beberapa sampel normalisasi yang dilakukan

Case	Input	Ground truth	Output
1	*min* tolong respon kelanjutan keluhan saya	Admin, tolong respons kelanjutan keluhan saya.	admin , tolong respons kelanjutan keluhan saya .
2	saya beli pulsa *ko* belum masuk *yaa* ?	Saya beli pulsa kok belum masuk ya?	mengapa saya beli pulsa belum masuk ?
3	pembelian paket internet malam kok *ga* bisa ya *min* ?	Pembelian paket internet malam kok tidak bisa ya, admin?	mengapa pembelian paket internet malam tidak bisa ya , admin ?
4	hari minggu *ngga* ada pengiriman ya *min* ?	Hari Minggu tidak ada pengiriman ya, admin?	hari minggu tidak ada pengiriman admin ?
5	di *hp* lain *nomer* saya *jg* aneh sinyalnya	Di ponsel lain nomor saya juga aneh sinyalnya.	di *hp* lain nomor saya juga aneh sinyalnya .



Gambar 3 Perubahan *Validation loss* terhadap Jenis *Optimizer*

3.1 Pengujian terhadap Jenis Optimizer

Pengujian terhadap jenis *optimizer* dilakukan untuk memperoleh jenis *optimizer* yang paling sesuai dengan tugas yang dilakukan. Pengujian ini melibatkan *optimizer* yaitu Adam dan SGD. Pengujian ini menggunakan nilai *learning rate* yaitu 1e-5 dan *dropout rate* sebesar 0,5. Sementara *batch size* yang digunakan yaitu 8,16 dan 32. Pada pengujian ini diterapkan adaptive *learning rate* dengan memanfaatkan *scheduler* untuk memperoleh prediksi yang lebih akurat. Gambar 3 menunjukkan hasil pengujian terhadap jenis *optimizer*.

Melalui percobaan dengan konfigurasi hiperparameter yang sama, hasilnya menunjukkan bahwa Adam *optimizer* lebih sesuai untuk digunakan dalam tugas normalisasi bahasa. Terdapat beberapa alasan yang mendukung kesimpulan ini:

1. Generalisasi yang lebih baik: Hasil pengujian menunjukkan bahwa Adam *optimizer* menghasilkan nilai *validation loss* yang lebih rendah dibandingkan dengan SGD pada setiap percobaan. *Validation loss* yang rendah menunjukkan bahwa model yang menggunakan Adam *optimizer* mampu melakukan generalisasi dengan lebih baik. Hal ini berarti model tersebut mampu lebih akurat dalam memprediksi data baru yang tidak terlihat selama proses pelatihan.
2. Konvergensi yang lebih cepat: Dalam beberapa percobaan, Adam *optimizer* cenderung mencapai konvergensi lebih cepat dibandingkan dengan SGD. Hal ini terlihat dari train loss dan *validation loss* yang stabil pada nilai yang lebih rendah dalam jumlah iterasi yang lebih sedikit. Konvergensi yang lebih cepat menghemat waktu pelatihan dan memungkinkan model untuk mencapai performa yang baik dengan lebih efisien.
3. Perbedaan Tingkat Akurasi: Meskipun tidak ada perbedaan signifikan dalam tingkat akurasi prediksi antara Adam *optimizer* dan SGD dalam percobaan ini, selisih tingkat akurasi cenderung meningkat seiring bertambahnya nilai *batch size*. Ini menunjukkan bahwa Adam *optimizer* dapat menghitung gradien dengan lebih baik

ketika ukuran *batch* semakin besar. Dalam tugas normalisasi bahasa, ketepatan perhitungan gradien dapat mempengaruhi kemampuan model untuk memahami dan menyesuaikan pola dalam data teks dengan lebih baik.

3.2 Pengujian terhadap Batch size

Pengujian terhadap *batch size* dilakukan untuk memperoleh pembagian jumlah *batch* terbaik untuk kelas data. Pembagian *batch* berpengaruh pada nilai gradien yang diperoleh. Pada penelitian in nilai *batch* yang digunakan antara lain 8, 16 dan 32. Jenis *optimizer* yang digunakan adalah Adam dengan *learning rate* yaitu 1e-5 dan *dropout rate* sebesar 0.5. Tabel 3.2 menunjukkan hasil pengujian terhadap *batch size*.

Tabel 3.2 Hasil pengujian terhadap *batch size*

case	optim	lr	dr	bs	v_acc	v_loss
1	*Adam*	1e-5	0.5	8	0.9676	**0.1890**
2	*Adam*	1e-5	0.5	16	0.9663	0.1933
3	*Adam*	1e-5	0.5	32	**0.9698**	0.2002

Melalui percobaan yang dilakukan, tidak terdapat perubahan akurasi dan loss yang signifikan terhadap perubahan nilai *batch size*. Namun, *fine-tuned* model dapat melakukan prediksi lebih baik ketika ukuran *batch* lebih besar dikarenakan Adam *optimizer* melibatkan lebih banyak sampel data pada setiap *epoch*nya. Sehingga perkiraan nilai gradien dan fungsi objektif dapat lebih akurat.

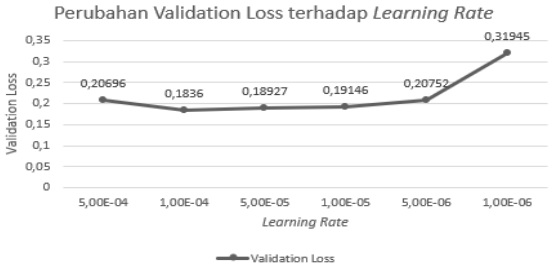
Ukuran *batch* yang lebih besar juga mengakibatkan variansi dari perkiraan gradien dapat berkurang dan cenderung lebih stabil. Hal ini membantu *optimizer* dalam mengarah ke konvergensi dengan lebih cepat. Namun nilai *batch size* yang terlalu besar juga dapat memiliki dampak negatif yaitu mengurangi tingkat keanekaragaman informasi yang diperoleh oleh model. Selain itu, nilai *batch size* yang terlalu besar juga menyebabkan pelatihan menjadi lebih lambat dan penggunaan sumberdaya yang lebih besar. Pemilihan nilai *batch size* yang optimal harus mempertimbangkan keuntungan dari estimasi gradien dan kerugian dari meningkatnya komputasi dan berkurangnya keanekaragaman informasi.

3.3 Pengujian terhadap Learning rate

Pengujian terhadap *learning rate* dilakukan untuk memperoleh nilai *learning rate* yang paling sesuai. Pada pengujian ini, *optimizer* yang digunakan adalah *optimizer* yang paling optimal sesuai pada pengujian 6.1.1. Untuk *learning rate* yang digunakan diantara 5e-4 hingga 1e-6. Tabel 3.3 dan Gambar 4 menunjukkan hasil pengujian terhadap *learning rate*.

Tabel 3.3 Hasil pengujian model terhadap *learning rate*

case	optim	lr	dr	bs	v_acc	v_loss
1	*Adam*	5e-4	0.5	16	0.9703	0.2069
2	*Adam*	1e-4	0.5	16	**0.9715**	**0.1836**
3	*Adam*	5e-5	0.5	16	0.9708	0.1892
4	*Adam*	1e-5	0.5	16	0.9672	0.1914
5	*Adam*	5e-6	0.5	16	0.9680	0.2075
6	*Adam*	1e-6	0.5	16	0.9585	0.3194



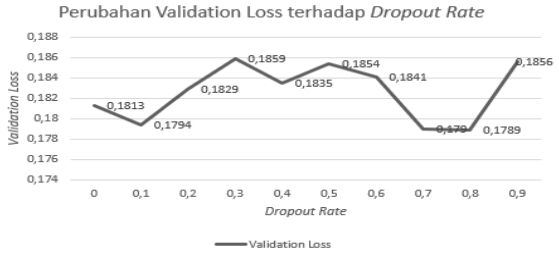
Gambar 4 Perubahan *Validation loss* terhadap *Learning rate*

Tingkat akurasi pada tahap pelatihan dan validasi relatif stabil dan tidak terdapat perbedaan yang signifikan ketika *learning rate* berada dalam rentang 5e-4 hingga 5e-6. Namun, ketika *learning rate* sangat rendah (1e-6), akurasi cenderung menurun. Hal ini menunjukkan bahwa *learning rate* yang terlalu rendah dapat menghambat model untuk melakukan pembelajaran. Sedangkan jika ditinjau dari nilai *validation loss*, model mencapai titik minimum pada nilai *learning rate* 1e-4. Ketika *learning rate* bernilai lebih atau kurang dari 1e-4, maka tren nilai *validation loss* cenderung meningkat.

Berdasarkan pengujian ini diperoleh bahwa *learning rate* yang terlalu tinggi atau terlalu rendah dapat mempengaruhi performa model. *Learning rate* yang terlalu tinggi dapat menyebabkan model sulit untuk mencapai titik konvergen dikarenakan melompati nilai optimum lokal. Sementara jika *learning rate* terlalu rendah, konvergensi akan semakin lambat dan menyebabkan model terjebak dalam nilai optimum lokal.

3.4 Pengujian terhadap Dropout rate

Pengujian terhadap jenis *optimizer* dilakukan untuk memperoleh *dropout rate* yang tepat sehingga model dapat melakukan generalisasi dengan baik. Dropout yang digunakan pada pengujian ini mulai dari 0,1 hingga 0,9. *Optimizer* yang digunakan merupakan Adam dengan *learning rate* terbaik yang diperoleh dari pengujian sebelumnya. Hasil pengujian dapat dilihat pada Gambar 5.



Gambar 5 Perubahan *Validation loss* terhadap *Dropout rate*

Dalam pengujian ini, diperoleh bahwa tingkat akurasi pada pelatihan maupun validasi cenderung stabil atau tidak terdapat perbedaan yang signifikan

ketika terjadi perubahan *dropout rate*. Terdapat perubahan-perubahan minim yang terjadi pada nilai akurasi dan loss. Namun perubahan tersebut tidak memiliki tren yang jelas dan cenderung acak.

Berdasarkan informasi tersebut, dapat disimpulkan bahwa perubahan *dropout rate* dalam kisaran 0 hingga 0.9 tidak memberikan pengaruh yang signifikan terhadap performa model. Sehingga dapat dikatakan bahwa *dropout rate* tidak menjadi faktor kritis dalam konfigurasi model ini. Namun, hal ini juga sangat bergantung pada *dataset* yang digunakan, terdapat kemungkinan bahwa diperlukan variasi *dataset* dan *dropout rate* yang lebih luas untuk dapat melihat pengaruh *dropout rate* pada performa model.

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa arsitektur normalisasi bahasa informal menjadi baku dapat dibangun dengan memanfaatkan *pre-trained* model yaitu GPT-2 Small Indonesian 522M yang kemudian dilakukan *fine-tuning* dengan *parallel corpus* bahasa informal dan baku berbahasa Indonesia. Berdasarkan pengujian yang dilakukan, performa model dapat mencapai tingkat akurasi 0,9716 dan loss 0,1789. Konfigurasi hiperparameter yang digunakan untuk mencapai nilai tersebut antara lain penggunaan *optimizer* Adam dengan *learning rate* sebesar 1e-4, *batch size* sebesar 16 dan *dropout rate* sebesar 0,8. Berdasarkan analisis hasil pengujian yang dilakukan, jenis *optimizer* dan *learning rate* adalah hiperparameter yang berpengaruh secara signifikan terhadap performa model normalisasi ini. Sedangkan *batch size* dan *dropout rate* tidak berdampak signifikan terhadap performa model. Namun, hal ini dapat disebabkan beberapa hal antara lain *dataset* yang digunakan dan tingkat eksplorasi eksperimen yang dilakukan. Tidak menutup kemungkinan terdapat pengaruh signifikan dari hiperparameter tersebut pada kasus lain.

Berdasarkan penelitian yang sudah dilakukan, dapat dijabarkan saran perbaikan dari pengembangan arsitektur normalisasi bahasa informal menjadi baku. Berkaitan dengan *dataset*, disarankan untuk menambah variasi data mengingat banyaknya bahasa informal dalam bahasa Indonesia yang terus bertambah seiring perkembangan zaman. Sehingga diharapkan model normalisasi bahasa informal masih relevan kedepannya. Berkaitan dengan metode, masih dimungkinkan untuk melakukan eksplorasi pendekatan *fine-tuning pre-trained* model lainnya seperti T5, GPT-3, dan BERT yang masing-masing memiliki karakteristik yang berbeda. Berkaitan dengan performa model, tidak menutup kemungkinan bahwa masih terdapat konfigurasi ataupun penyesuaian yang dapat dilakukan seperti jenis *optimizer*, *learning rate*, metode regularisasi, metode pra-pemrosesan serta pelatihan terhadap model yang dibangun guna meningkatkan performa model.

DAFTAR PUSTAKA

ALIYAH SALSABILA, N., ARDHITO WINATMOKO, Y., AKBAR SEPTIANDRI, A., & JAMAL, A. 2019. Colloquial Indonesian Lexicon. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*. https://doi.org/10.1109/IALP.2018.8629151.

ANDERSEN, G. 2005. Pragmatic markers and sociolinguistic variation. *Journal of Pragmatics*, *37*(4).

ARIFIN, M. B., HEFNI, A., & PURWANTI, P. 2022. Slang dalam Bahasa Indonesia: Kajian Morfosemantik. *Diglosia: Jurnal Kajian Bahasa, Sastra, dan Pengajarannya*, *5*(1s). https://doi.org/10.30872/diglosia.v5i1s.402.

AW, A. T., ZHANG, M., Xiao, J., & Su, J. 2006. A phrase-based statistical model for SMS text normalization. *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Main Conference Poster Sessions*. https://doi.org/10.3115/1273073.1273078.

BANITZ, B. 2020. Machine translation: A critical look at the performance of rule-based and statistical machine translation. *Cadernos de Traducao*, *40*(1). https://doi.org/10.5007/2175-7968.2020v40n1p54.

GILLIOZ, A., CASAS, J., MUGELLINI, E., & KHALED, O. A. 2020. Overview of the Transformer-based Models for NLP Tasks. *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, FedCSIS 2020*. https://doi.org/10.15439/2020F20

HANAFIAH, N., KEVIN, A., SUTANTO, C., FIONA, ARIFIN, Y., & HARTANTO, J. 2017. Text Normalization Algorithm on Twitter in Complaint Category. *International Conference on Computer Science and Computational Intelligence*.

MADAAN, A., SETLUR, A., PAREKH, T., PPOCZOS, B., NEUBIG, G., YANG, Y., SALAKHUTDINOV, R., BLACK, A. W., & PRABHUMOYE, S. 2020. Politeness transfer: A tag and generate approach. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.169.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., & POLOSUKHIN, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-December*..

WANG, Y., WU, Y., MOU, L., LI, Z., & CHAO, W. 2019 Harnessing pre-trained neural networks with rules for formality style transfer. *EMNLP-*

IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference. https://doi.org/10.18653/v1/d19-1365.

WIBOWO, H. A., PRAWIRO, T. A., IHSAN M., AJI, A. F., PRASOJO, R. E., MAHENDRA, R., & FITRIANY, S. 2020. Semi-Supervised Low-Resource Style Transfer of Indonesian Informal to Formal Language with Iterative Forward-Translation. *2020 International Conference on Asian Language Processing (IALP)*, 310–315. https://doi.org/10.1109/IALP51396.2020.9310459

Halaman ini sengaja dikosongakn.