TESIS

PEMODELAN PERCAKAPAN BAHASA BANJAR MENGGUNAKAN ARSITEKTUR SEQ2SEQ DENGAN MEKANISME ATENSI

Disusun oleh:

Nama : **Bambang Abdi Setiawan**
NIM : **20.77.1278**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2021

TESIS

PEMODELAN PERCAKAPAN BAHASA BANJAR MENGGUNAKAN ARSITEKTUR SEQ2SEQ DENGAN MEKANISME ATENSI

CONVERSATION MODELING OF BANJAR LANGUAGE USING SEQ2SEQ ARCHITECTURE WITH ATTENTION MECHANISM

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Bambang Abdi Setiawan**
NIM	**: 20.77.1278**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2021

HALAMAN PENGESAHAN

PEMODELAN PERCAKAPAN BAHASA BANJAR MENGGUNAKAN ARSITEKTUR SEQ2SEQ DENGAN MEKANISME ATENSI

CONVERSATION MODELING OF BANJAR LANGUAGE USING SEQ2SEQ ARCHITECTURE WITH ATTENTION MECHANISM

Dipersiapkan dan Disusun oleh

Bambang Abdi Setiawan

20.77.1278

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 3 Februari 2022

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Februari 2022
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

PEMODELAN PERCAKAPAN BAHASA BANJAR MENGGUNAKAN ARSITEKTUR SEQ2SEQ DENGAN MEKANISME ATENSI

CONVERSATION MODELING OF BANJAR LANGUAGE USING SEQ2SEQ ARCHITECTURE WITH ATTENTION MECHANISM

Dipersiapkan dan Disusun oleh

Bambang Abdi Setiawan

20.77.1278

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 3 Februari 2022

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Anggota Tim Penguji

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

Pembimbing Pendamping

Alva Hendi M., S.T., M. Eng., Ph.D.
NIK. 190302493

Anggit Dwi Hartanto, M.Kom.
NIK. 190302163

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Februari 2022
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Bambang Abdi Setiawan
NIM : 20.77.1278
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Pemodelan Percakapan Bahasa Banjar Menggunakan Arsitektur Seq2Seq Dengan Mekanisme Atensi

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 Februari 2022
Yang Menyatakan,

Bambang Abdi Setiawan

HALAMAN PERSEMBAHAN

Kupersembahkan untuk istriku tercinta, *Kartika Listiyaningsih* yang telah mendampingi dengan setia dan memberi semangat untuk menyusun setiap rangkaian kata dalam laporan ini. Untuk puteraku tersayang, *Mikhael Jevarya Evangelion* yang selalu memberikan hiburan dalam setiap perjuanganku. Ayahanda *Riyono* dan Ibunda *Wiji Astuti* terkasih yang selalu memanjatkan doa terbaik untuk keberhasilanku dalam meraih setiap cita-cita.

Teman-teman seperjuangan, terima kasih sudah menjadi sahabat terbaik selama menempuh perjuangan meraih gelar magister. Keep on moving!

HALAMAN MOTTO

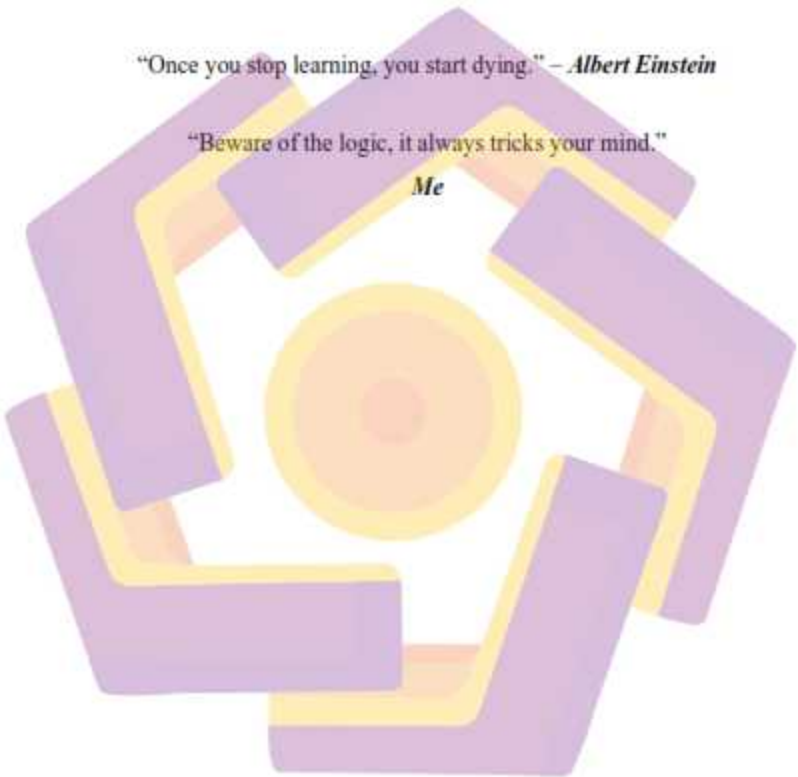
"But as it is written, Eye hath not seen, nor ear heard, neither have entered into the heart of man, the things which God hath prepared for them that love him."

1 Corinthians 2:9

"Once you stop learning, you start dying." – ***Albert Einstein***

"Beware of the logic, it always tricks your mind."

Me



KATA PENGANTAR

Puji dan syukur tak terkira kepada Tuhan Yang Maha Kuasa, Yeshua Hamashiakh atas semua anugerah dan rahmat-Nya sehingga laporan penelitian ini dapat diselesaikan. Terima kasih yang sebesar-besarnya juga penulis ucapkan kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M. selaku Rektor Universitas AMIKOM Yogyakarta.
2. Ibu Prof. Dr. Kusrini, M.Kom. selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. selaku Wakil Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta sekaligus Dosen Pembimbing Utama.
4. Bapak Anggit Dwi Hartanto, M.Kom. selaku Dosen Pembimbing Pendamping.
5. Para dosen selaku Dewan Penguji yang telah memberikan arahan dan perbaikan dari seminar proposal, seminar hasil penelitian sampai ujian tesis.

Kritik dan saran yang membangun sangat diharapkan agar penelitian yang masih belum sempurna ini dapat lebih bermanfaat bagi banyak orang.

Balangan, 3 Februari 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR ISTILAH

Dataset: Kumpulan data yang dapat diproses oleh komputer untuk tujuan analitik dan prediksi. Ini berarti bahwa data yang dikumpulkan harus dibuat seragam dan dapat dimengerti oleh mesin yang tidak melihat data dengan cara yang sama seperti yang dilakukan oleh manusia.

Encoder-Decoder: Arsitektur yang masing-masing terdiri dari sekuensial *neural network* dimana input dibaca secara berurutan dan di-*encode* menjadi representasi internal yang berukuran tetap. Jaringan *decoder* kemudian menggunakan representasi internal ini untuk memproses output kata sampai akhir token sekuensial telah tercapai.

Epoch: Satu epoch dalam jaringan saraf sama dengan satu iterasi siklus training penuh pada dataset training.

Batch Size: Parameter untuk menentukan jumlah sample training yang digunakan dalam satu iterasi.

Dropout: Teknik regularisasi dimana *neuron* yang dipilih secara acak diabaikan selama dilakukan proses training untuk mengurangi terjadinya *overfitting*.

Overfitting: Peristiwa yang terjadi ketika model statistik sangat cocok dengan data trainingnya. Namun, sayangnya algoritma tidak dapat bekerja secara akurat terhadap data yang tidak terlihat, sehingga menggagalkan tujuannya.

Activation Function: Digunakan dalam jaringan saraf tiruan yang menghasilkan nilai kecil untuk input kecil dan nilai lebih besar jika inputnya melebihi ambang batas (*threshold*). Jika inputnya cukup besar, fungsi aktivasi diaktifkan, jika tidak, tidak melakukan apa-apa. Dengan kata lain, fungsi aktivasi seperti gerbang yang memeriksa bahwa nilai yang masuk lebih besar dari angka kritis.

Metric Scores: Nilai pengukuran yang didapatkan sebagai evaluasi terhadap hasil pemodelan yang telah diselesaikan oleh algoritma.

INTISARI

Perkembangan teknologi *chatbot* secara signifikan dimulai dengan implementasi model Seq2Seq (*Sequence-to-Sequence*) yang berbasis pada dua arsitektur RNN (*Recurrent Neural Network*). Model Seq2Seq terbukti mampu menangani masalah *vanishing* atau *exploding gradient* yang terjadi saat melakukan training jangka panjang pada arsitektur *vanilla* RNN. Penggunaan variasi unit gerbang LSTM (*Long-Short Term Memory*) dan GRU (*Gated Recurrent Unit*) pada lapisan *encoder-decoder* dengan menambahkan mekanisme atensi dan optimasi gradien dapat menangani data dengan urutan yang panjang dan meminimalkan tingkat kesalahan.

Dataset yang digunakan adalah percakapan Bahasa Banjar dengan dialek campuran yang diperoleh dari database sistem *live support* pada kampus XYZ yang terdiri dari 6.100 baris pasang percakapan setelah melalui tahapan *preprocessing*. Pemrosesan dataset diatur dengan komposisi 80% data training dan 20% data testing. Pengujian performansi model diperoleh dengan memperhitungkan nilai *loss* dan *confusion matrix* berdasarkan pengaturan parameter berupa optimasi gradien Adam, *learning rate* 0.001, *epoch* 400, *batch size* 64 dan 128, *dropout* 0.5 dan 0.8. Hasil terbaik diperoleh oleh model LSTM-GRU pada epoch ke-50 menggunakan parameter batch size 128 dan dropout 0.8. Model ini mendapatkan nilai loss sebesar 2.9955, accuracy 61.53%, precision 89.11%, recall 54.61%, F1 67.04% dengan durasi testing selama 15.50 detik.

Penggunaan mekanisme atensi yang diterapkan pada arsitektur Seq2Seq dengan pengaturan nilai dropout dan batch size yang sesuai dapat menurunkan overfitting dan meningkatkan kinerja model.

Kata kunci: *seq2seq*, mekanisme atensi, model gerbang, *chatbot*, bahasa banjar

ABSTRACT

The significant development of chatbot technology began with the implementation of the Seq2Seq (Sequence-to-Sequence) model based on two RNN (Recurrent Neural Network) architectures. The Seq2Seq model is proven to handle vanishing or exploding gradient problems that occur while processing long-term training on the vanilla RNN architecture. The use of variations of LSTM (Long-Short Term Memory) and GRU (Gated Recurrent Unit) gate units in the encoder-decoder layer by adding attentional mechanism and gradient optimization can handle long sequences of data and minimize the error rates.

This research uses datasets of Banjar language conversation with mixed dialects obtained from the live support system database on the XYZ college which consists of 6,100 pairs of conversation lines after going through the preprocessing stage. The datasets were processed with a composition of 80% training data and 20% testing data. The model performance test is obtained by calculating the loss and confusion matrix values based on parameter settings of Adam gradient optimization, learning rate 0.001, epoch 400, batch size 64 and 128, dropout 0.5 and 0.8. The best results were obtained by the LSTM-GRU model in the 50th epoch using batch size 128 and dropout 0.8. This model gets a loss value of 2.9955, accuracy 61.53%, precision 89.11%, recall 54.61%, F1 67.04% with a testing duration of 15.50 seconds.

Using the attention mechanism applied to the Seq2Seq architecture with appropriate settings of dropout and batch size can reduce overfitting and improve model performance.

Keyword: seq2seq, attention mechanism, gate model, chatbot, banjar language

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Bahasa Banjar adalah satu dari bahasa suku yang digunakan di Indonesia dan dipakai oleh suku Banjar yang berdomisili di Kalimantan serta merupakan bahasa yang paling banyak dituturkan sebagai *lingua franca* di Kalimantan Selatan (Humaidi, 2017). Berdasarkan data sensus pada tahun 2010 (Na'im, 2012), tercatat sebanyak 3,5 juta orang menuturkan Bahasa Banjar dan menurut data statistik pada tahun 2011, Bahasa Banjar menjadi bahasa daerah pada urutan ke-7 yang terbanyak digunakan di Indonesia.

Bahasa Banjar bukan saja digunakan di wilayah Kalimantan Selatan, tetapi mencakup area Kalimantan Tengah serta Kalimantan Timur. Karena popularitasnya, suku Dayak yang berdomisili di seluruh wilayah Kalimantan mampu menuturkan Bahasa Banjar. Hal ini membuat Bahasa Banjar menjadi bahasa perantara antara warga di wilayah Kalimantan Selatan, Kalimantan Tengah, serta Kalimantan Timur (Hapip, 1977) dan telah mulai digunakan juga di Kabupaten Indragiri Hilir, Provinsi Riau serta di beberapa wilayah di Provinsi Jambi sehingga menjadikan Bahasa Banjar kian populer di berbagai kalangan masyarakat (Sugono, 2017).

Penelitian tentang pembangunan model chatbot di Indonesia yang dilakukan hingga saat ini masih terbatas pada Bahasa Inggris dan Bahasa Indonesia saja. Sedangkan Indonesia memiliki beragam bahasa daerah yang memiliki keunikannya

masing-masing. Perkembangan teknologi pemrosesan bahasa alami atau yang disebut dengan NLP (Natural Language Processing) semakin mendorong kebutuhan akan pemrosesan bahasa khususnya bahasa daerah yang dapat diterapkan untuk memenuhi kebutuhan pelayanan publik di berbagai wilayah di Indonesia dalam bentuk chatbot. Dengan demikian, baik pemerintah maupun pihak swasta dapat memanfaatkannya untuk memberikan pelayanan otomatis yang dirancang dengan baik agar mempu melayani kebutuhan pelanggan ataupun masyarakat umum. Sistem tersebut mampu memperbarui pengetahuan yang dimilikinya sesuai dengan yang diinginkan oleh pembuatnya. Manusia sebagai pengguna dapat bertanya kepada sistem tersebut selayaknya bertanya kepada sesama manusia dalam masa 1x24 jam, sehingga akan sangat menghemat tenaga, waktu dan biaya.

Penelitian yang dilakukan oleh Vinyals pada tahun 2015 menandai kesuksesan dalam membangun sebuah model percakapan menggunakan dataset besar pada arsitektur RNN (Recurrent Neural Network) dengan model Sequence-to-Sequence (Seq2Seq) yang sukses menanggulangi kesulitan pemetaan antara kueri dengan respon yang kerap dibatasi oleh domain kecil dan masih memerlukan lebih banyak porsi campur tangan manusia ketimbang otomatisasi mesin. Hal inilah yang menjadikan Seq2Seq menjadi model yang gampang diaplikasikan dalam bermacam pekerjaan NLP, khususnya untuk keperluan chatbot.

Penerapan arsitektur Seq2Seq dalam berbagai bahasa tentu tidak sama, sehingga dapat menghasilkan pemodelan bahasa yang baru. Dari hal itu maka diangkatlah penelitian tentang interaksi antara manusia dengan komputer

menggunakan arsitektur Seq2Seq untuk chatbot Bahasa Banjar dengan mekanisme atensi, optimasi gradien dan pengaturan parameter. Penerapan variasi arsitektur RNN pada model Seq2Seq dilakukan dengan menambahkan gerbang Long-Short Term Memory (LSTM) (Sutskever, 2014), gerbang Gated Recurrent Unit (GRU) (Cho, 2014), dan mekanisme atensi (Bahdanau, 2014). Pemanfaatan gradient descent akan memproduksi model yang lebih bagus dengan mencari tingkat kesalahan yang paling rendah. Sedangkan pengaturan parameter dapat mengurangi terjadinya overfitting pada jaringan dan membantu meningkatkan kinerja model.

Penelitian ini berfokus pada implementasi model percakapan Seq2Seq menggunakan variasi unit gerbang LSTM dan GRU pada lapisan encoder dan decoder dengan menambahkan mekanisme atensi. Optimasi yang dilakukan adalah penggunaan gradient descent Adam untuk meminimalkan tingkat kesalahan dan *tuning hyperparameter* berupa batch size dan dropout. Mekanisme atensi bekerja dengan meniru dan memperhatikan ciri-ciri visual khusus pada kata tertentu dalam hubungannya untuk merangkai sebuah kalimat. Pada mekanisme atensi, sebuah nilai dapat ditentukan berdasarkan seberapa kuat hubungan sebuah kata dengan kata yang lain lalu dikontribusikan menjadi sebuah struktur hierarki yang mendukung neural network dalam membuat kesimpulan tingkat kepentingan dari sebuah kata dalam konteks kalimat tertentu (Vinyals, 2015).

Beberapa penelitian sebelumnya tentang pemanfaatan arsitektur Seq2Seq dengan penambahan mekanisme atensi memberikan informasi bahwa pengujian terhadap model yang dilakukan menghasilkan output yang cukup koheren (Ali, 2019) sedangkan nilai BLEU mengalami peningkatan sebesar 1,07 persen

(Chandra, 2019) yang memperkuat bukti bahwa penambahan mekanisme atensi pada arsitektur Seq2Seq mampu memberikan hasil yang lebih baik. Penelitian tentang chatbot Seq2Seq terus dikembangkan, seperti penelitian yang dilakukan oleh Sojasingarayar (2020) yaitu dengan menempatkan gerbang LSTM pada lapisan encoder dan decoder diketahui mampu meningkatkan kinerja model secara optimal. Pekerjaan yang telah dilakukan oleh Ali (2019) menghasilkan model chatbot dengan perplexity test yang optimal sebesar 48,22. Namun, semua penelitian yang telah dilakukan tersebut masih belum melakukan perhitungan nilai loss dan metrik secara komprehensif berdasarkan pengaturan parameter yang diterapkan pada setiap model.

Upaya optimasi yang dilakukan untuk meningkatkan performansi model adalah dengan menyusun layer pada struktur model berdasarkan jenis arsitektur neural network yang sesuai. Selain itu, tuning hyperparamater merupakan aspek yang sangat penting untuk menentukan kualitas pembangunan model, baik dari segi durasi training dan testing maupun tingkat akurasi yang dihasilkan. Model yang memiliki perhitungan nilai metrik terbaik akan digunakan sebagai referensi dalam menentukan arsitektur neural network yang ideal beserta parameter-parameternya untuk membangun model percakapan Bahasa Banjar yang optimal.

1.2. Rumusan Masalah

Rumusan masalah yang akan dikaji pada penelitian ini antara lain:

a. Model apa dengan parameter bagaimana yang memiliki nilai pengujian paling baik berdasarkan hasil pada setiap skenario yang dilakukan?

b. Bagaimana pengaruh penggunaan mekanisme atensi dan pengaturan nilai parameter pada proses training model terhadap hasil pengujian?

1.3. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka penelitian yang dilakukan dibatasi oleh ketentuan berikut ini:

a. Dataset yang digunakan adalah percakapan Bahasa Banjar dengan dialek campuran (Kuala dan Hulu) antara calon mahasiswa dengan pihak admisi yang diperoleh dari database sistem *live support* pada kampus XYZ. Dataset ini berjumlah 6.100 baris pasang percakapan.
b. Tahapan *preprocessing* yang dilakukan adalah: filter karakter alfanumerik dan simbol tanda baca percakapan (koma, titik, tanda tanya, tanda seru, garis mendatar, garis miring, *ampersand*), penghapusan spasi berlebih, *casefolding*, dan pembatasan panjang kalimat maksimal 15 kata. Proses *stemming*, *stopword* dan normalisasi kata yang intensif tidak dilakukan.
c. Komposisi dataset adalah 80% data training dan 20% data testing.
d. Optimasi gradient descent menggunakan Adam.
e. Variasi unit gerbang yang digunakan adalah LSTM dan GRU.
f. Epoch yang digunakan untuk semua model adalah 400.
g. Learning rate yang digunakan untuk semua model adalah 0.001 sesuai dengan nilai default yang ditentukan oleh framework Keras.
h. Variasi unit gerbang yang memiliki hasil terbaik pada Skenario 1 akan digunakan sebagai acuan pada Skenario 2.

i. Parameter yang digunakan pada Skenario 1 untuk semua model adalah batch size 64 dan dropout 0.5 menggunakan mekanisme atensi.

j. Parameter yang digunakan pada Skenario 2 terdiri dari: batch size 64 dan dropout 0.5 tanpa mekanisme atensi diterapkan pada Model 1, batch size 64 dan dropout 0.8 dengan mekanisme atensi diterapkan pada Model 2, batch size 128 dan dropout 0.8 dengan mekanisme atensi diterapkan pada Model 3,

k. Indikator yang dipakai untuk mengukur performansi model adalah loss, accuracy, precision, recall dan F1.

l. Proses training dan testing dilakukan menggunakan unit komputer mandiri.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai di dalam penelitian ini adalah:

a. Mengetahui variasi unit gerbang yang menghasilkan nilai metrik terbaik beserta parameter yang digunakan.

b. Mengetahui faktor-faktor yang mempengaruhi performansi model.

1.5. Manfaat Penelitian

Manfaat yang didapat dari melakukan penelitian ini antara lain:

a. Menjadi referensi dalam penelitian pemodelan chatbot menggunakan arsitektur Seq2Seq dan mekanisme atensi.

b. Memberikan kontribusi pada bidang pemrosesan bahasa alami (NLP) khususnya Bahasa Banjar sebagai bentuk dari pengabdian masyarakat.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Model percakapan semakin berkembang ditandai dengan banyaknya riset tentang jaringan saraf tiruan. Model percakapan menggunakan jaringan saraf tiruan yang populer adalah Sequence-to-Sequence (Seq2Seq). Model ini disusun dari dua jaringan RNN (2-RNN) yang dikenal dengan model *encoder-decoder*. Popularitas model Seq2Seq dimulai pada tahun 2014 saat Sutskever melakukan tugas terjemahan mesin untuk menerjemahkan Bahasa Inggris ke Bahasa Prancis. Ia menggunakan layer berlapis LSTM yang diterapkan pada model Seq2Seq kemudian membalik urutan kata pada kalimat yang diinput. Kemudian, ada penelitian tentang model Seq2Seq yang menggunakan Gated Recurrent Unit (GRU). GRU diketahui memiliki kemampuan untuk mempelajari representasi frasa bahasa dan sintaksis secara semantik pada terjemahan mesin (Cho, 2014).

Penelitian tentang chatbot menggunakan arsitektur Seq2Seq dan mekanisme atensi telah dilakukan oleh Sojasingarayar (2020) dengan memanfaatkan model encoder-decoder yang tersusun dari rangkaian gerbang LSTM. Penelitian ini mengolah dataset berupa korpus subtitle film versi Cornell yang terdiri dari 220.579 baris percakapan yang diekstrak dari 617 film. Penelitian lain yang menggunakan dataset serupa dilakukan oleh Ali (2019) yang masih memanfaatkan gerbang LSTM dalam struktur vanilla RNN dengan 50 *epochs* dan 32 *batch size*. Masih dengan sumber dataset yang sama, Abdullahi (2019)

menggunakan arsitektur Seq2Seq dan gerbang GRU dengan mekanisme atensi ditambah dengan Adam optimizer dengan *learning rate* sebesar 0.5 dan 1500 epochs. Dataset diproses dengan pembagian 80% training dan 20% testing. Hasil pengujian dari model ini didapatkan nilai *loss* sebesar 0.23 setelah lebih dari 1000 iterasi sebelum terjadi *overfitting*. Dua sukarelawan disajikan 100 respon dari chatbot. Mereka menilai hasil dengan tiga variabel linguistik, yaitu: OK, Tidak OK, dan Tidak Meyakinkan, dengan komposisi 80% OK, 9% Tidak OK dan 11% Tidak Meyakinkan.

Aalipour (2018) merancang chatbot menggunakan arsitektur *bi-directional* LSTM untuk keperluan otomatisasi jawaban atas pertanyaan-pertanyaan teknis dari pelanggan dan membuat ringkasannya untuk menghasilkan respon terkait, menujukkan hasil pengujian ringkasan teks 16% lebih baik dari model dasar dalam melakukan peringkasan percakapan dan respon yang diberikan kepada pelanggan.

Pengimplementasian chatbot menggunakan model Seq2Seq dan mekanisme atensi juga telah menarik perhatian Raj (2021) untuk melakukan penelitian lebih lanjut dengan menambahkan model gerbang LSTM pada pada lapisan encoder dan menambahkan mekanisme atensi pada lapisan decoder dengan 3 (tiga) pengujian menggunakan *hyperparamater* yang berbeda, yaitu: 500 epochs, 100 epochs dan 50 epochs. Setelah dilakukan pengujian percakapan, model dengan 100 epochs menujukkan hasil terbaiknya.

Penggunaan gerbang LSTM pada arsitektur Seq2Seq juga dilakukan oleh Sonawane (2019), menggunakan dataset yang diambil dari percakapan mahasiswa pada sebuah perguruan tinggi dengan pembagian 90% untuk training dengan 150

epochs dan 10% untuk testing. Selama proses training model, loss pada epoch ke-150 menghasilkan nilai 0.012 dengan tingkat akurasi sebesar 92,89%.

Beberapa penelitian yang menggunakan dataset selain Bahasa Inggris juga telah dilakukan. Nguyen (2018) menggunakan Bahasa Vietnam sebagai dataset yang terdiri dari 1331 pasang pertanyaan dan jawaban dengan tahapan *preprocessing* berupa: membaca file teks, memisah teks ke dalam baris, memisah baris ke dalam pasangan kalimat, normalisasi teks, dan filterisasi teks. Sedangkan Palasundram (2019) menggunakan gerbang GRU pada lapisan encoder-decoder untuk memproses dataset berupa 100 pasang percakapan Bahasa Melayu dengan domain *computer science and applications*. Sedangkan untuk Bahasa Indonesia telah dilakukan penelitian oleh Chandra (2019) dengan mengolah dataset percakapan tentang penerimaan mahasiswa baru pada Telkom University yang diambil dari percakapan WhatsApp dengan komposisi 2.506 baris training dan 397 baris testing. Hasil evaluasi menunjukkan bahwa model yang dihasilkan menggunakan mekanisme atensi memiliki respon yang lebih baik dibandingkan dengan model tanpa menggunakan mekanisme atensi.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks Literatur Review dan Posisi Penelitian
Pemodelan Percakapan Bahasa Banjar Menggunakan Arsitektur Seq2seq Dengan Mekanisme Atensi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	A Neural Network based Vietnamese Chatbot	Trang Nguyen dan Maxim Shcherbakov, International Conference on System Modeling & Advancement in Research Trends (SMART), 2018	Membangun chatbot Bahasa Vietnam berdasarkan model Seq2Seq dengan kombinasi mekanisme atensi.	Model yang dibangun menghasilkan percakapan dasar dan sederhana yang mampu mengekstrak pengetahuan dari dataset umum.	Ukuran data training masih terbatas dan respon yang dihasilkan masih terlalu umum. Perlu penambahan dataset dan GPU yang lebih besar untuk percepatan waktu training.	Penelitian ini menggunakan dataset yang relatif sedikit dan masih menggunakan arsitektur vanilla RNN. Penelitian yang akan dilakukan menggunakan dataset yang lebih besar dengan implementasi variasi model gerbang pada arsitektur 2-RNN.
2	Open Domain Chatbot Based on Attentive End-to-End Seq2Seq Mechanism	Saheed Salahudeen Abdullahim, Sun Yiming, Abdulkadir Abdullahi, Umar Aliyu, International Conference on Algorithms, Computing and Artificial Intelligence (ACAI), 2019	Melatih *open-domain chatbot* dengan melakukan *mapping* pada tag input sekuensial untuk men-*generate* tag output sekuensial yang optimal.	Model yang dibangun berhasil mengatasi kompleksitas pemahaman bahasa, ekstraksi fitur, pengenalan domain, deteksi makna, pengisian slot semantik, dan generasi respon.	Metode pengukuran terhadap hasil evaluasi masih berupa klasifikasi manual oleh 100 responden dengan klasifikasi variabel linguistik berupa: OK, Not OK dan *Inconclusive* (tidak meyakinkan).	Penelitian ini belum menggunakan confusion matrix sebagai alat pengukuran hasil evaluasi. Penelitian yang akan dilakukan menggunakan confusion matrix sehingga dapat diketahui pengukurannya secara komprehensif.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Applications of Sequence to Sequence Models for Technical Support Automation	Ghodrat Aalipour, Pranav Kumar, Santosh Aditham, Trung Nguyen, Aditya Sood. IEEE International Conference on Big Data, 2018	Merancang arsitektur 2-RNN untuk keperluan otomatisasi dalam menjawab pertanyaan-pertanyaan teknis dari pelanggan dan membuat ringkasannya untuk menghasilkan respon terkait dengan keperluan *support tickets*.	Dengan menggunakan bi-directional LSTM, hasil pengujian menunjukkan skor BLEU 0,21 untuk ringkasan teks yang 16% lebih baik dari model dasar dalam melakukan peringkasan percakapan dan respon yang dihasilkan kepada pelanggan.	Hasil evaluasi hanya dilakukan dengan cara membuat perbandingan antara *human generated summary* dan *machine generated summary*.	Penelitian ini hanya menggunakan skor BLEU dan ROUGE sebagai alat evaluasi tanpa adanya confusion matrix sehingga belum bisa diketahui hasil evaluasinya secara komprehensif. Penelitian yang akan dilakukan selain menggunakan loss sebagai alat evaluasi, juga menggunakan confusion matrix untuk mengukur nilai evaluasi yang dihasilkan oleh tiap model sesuai dengan skenario.
4	Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model	Yogi Wisesa Chandra, Suyanto Suyanto, International Conference on Computer Science and Computational Intelligence (ICCSCI), 2019	Mengembangkan chatbot berdasarkan model sequence-to-sequence untuk dilatih menggunakan data percakapan dari penerimaan mahasiswa baru pada sebuah universitas.	Menunjukkan bahwa model menghasilkan skor BLEU yang cukup tinggi yaitu 41,04. Teknik mekanisme atensi menggunakan kalimat terbalik meningkatkan model untuk menghasilkan BLEU yang lebih tinggi hingga 44,68.	Menggunakan dataset yang berasal dari WhatsApp dimana sebagian besar masih menggunakan kosakata yang tidak baku dan tidak dilakukan tahapan normalisasi yang intensif terhadap dataset.	Penelitian ini hanya menggunakan skenario dengan dan tanpa mekanisme atensi pada sel LSTM dan BLEU sebagai alat evaluasi model. Penelitian yang akan dilakukan menggunakan variasi model gerbang LSTM dan GRU dengan mekanisme atensi dan confusion matrix sebagai alat evaluasi model.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Sequence to Sequence Model Performance for Education Chatbot	Kulothunkan Palasundram, Nurfadhlina Mohd Sharef, Nurul Amelina Nasharuddin, Khairul Azhar Kasmiran, Azreen Azman, International Journal of Emerging Technologies in Learning (iJET), 2019	Memberikan evaluasi empiris pada literatur Seq2Seq dan memberikan wawasan tentang bagaimana dataset yang dikurasi dapat dikembangkan serta bagaimana pertanyaan yang dirancang dapat dilatih dan diuji kinerja modelnya untuk menjawab pertanyaan.	Penyematan kata dilakukan secara konsisten lebih baik daripada penyematan karakter. *Fine tuning* menunjukkan peningkatan besar pada kinerja model terutama dalam mengurangi overfitting dan menghasilkan jawaban yang benar tanpa perlu menambahkan kerumitan pada model.	Tidak menyajikan hasil training dan evaluasi model dalam bentuk grafik serta hanya berfokus pada analisis perbandingan antara *word embedding* dan *char embedding* dengan skenario tuning paramater.	Penelitian ini hanya menggunakan skor BLEU sebagai alat evaluasi. Penelitian yang akan dilakukan selain menggunakan loss juga menggunakan confusion matrix.
6	ChatBot for College Website	Shubham Sonawane, Shanmughasundaram R, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019	Menerapkan layanan chat pada website perguruan tinggi yang akan secara otomatis menjawab pertanyaan pengguna tanpa interaksi manusia dengan melakukan implementasi model Seq-to-Seq dalam *deep neural network*.	Hasil pengujian antara pertanyaan dan jawaban yang tepat mencapai tingkat akurasi model sebesar 92,89%.	Belum memberikan informasi yang jelas tentang metode yang digunakan untuk melakukan evaluasi hasil pengujian untuk mendapatkan nilai akurasinya.	Penelitian ini masih belum banyak melakukan optimasi model selain menggunakan sel LSTM pada lapaisan RNN. Penelitian yang akan dilakukan menggunakan beberapa skenario optimasi seperti mekanisme atensi, kombinasi gerbang, optimizer dan tuning hyperparameter.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	Conversational AI Chatbot Based on Encoder-Decoder Architectures with Attention Mechanism	Amir Ali, Muhammad Zain Amin, Artificial Intelligence Festival 2.0, NED University of Engineering and Technology, 2019	Mengembangkan chatbot dengan mengimplementasikan arsitektur mekanisme atensi pada lapisan encoder-decoder dengan sel LSTM.	Hasil training dataset perlu dilakukan perbaikan dengan spekulasi pada parameter training dengan eksperimen dataset lain. Selain itu, menambahkan lebih banyak data yang berkualitas akan semakin meningkatkan kinerja dari model.	Pengujian terhadap hasil evaluasi masih dilakukan secara manual dengan perspektif manusia terhadap 7 (tujuh) hasil prediksi.	Penelitian ini hanya menggunakan BLEU dan perplexity sebagai alat evaluasi model. Penelitian yang akan dilakukan menggunakan confusion matrix dan loss sehingga dapat diketahui hasil evaluasinya secara komprehensif.

2.3. Landasan Teori

2.3.1. Bahasa Banjar

Suku Banjar banyak tinggal di wilayah Provinsi Kalimantan Selatan. Hal inilah yang menjadikan Bahasa Banjar identik dengan provinsi tersebut. Istilah "Banjar" juga identik dengan Banjarmasin (ibukota Kalimantan Selatan), Kabupaten Banjar dengan ibukota Martapura, dan juga kota di sebelahnya, yakni Banjarbaru yang berstatus sebagai kotamadya.

Bahasa Banjar memiliki 2 (dua) dialek, yaitu Bahasa Banjar Hulu yang dipakai oleh masyarakat di kawasan yang disebut sebagai Banua Anam (Benua Enam), yang terdiri dari kabupaten: Tapin (Rantau), Hulu Sungai Selatan (Kandangan), Hulu Sungai Tengah (Barabai), Hulu Sungai Utara (Amuntai), Balangan (Paringin) dan Tabalong (Tanjung). Sedangkan Bahasa Banjar Kuala digunakan oleh masyarakat yang berada di wilayah Kabupaten Banjar (Martapura), Kota Banjarbaru, Kota Banjarmasin, Kabupaten Barito Kuala (Marabahan), Tanah Laut (Pelaihari), Tanah Bumbu (Sungai Danau) dan Kotabaru. Kedua dialek tersebut memiliki perbedaan pada kosakata, pelafalan, dan bunyi vokal.

Dialek Banjar Hulu menggunakan bunyi vokal "A", "I", dan "U" saja, dan dialek Banjar Kuala memakai semua bunyi vokal "A", "I", "U", "E" dan "O". Mayarakat Banjar Hulu menyebut bunyi vokal "O" dengan "U bulat" dan bunyi vokal "U" sebagai "U pecah". Sehingga, dialek Banjar Kuala secara umum hampir sama dengan Bahasa Indonesia. Secara umum, pembentukan struktur kalimat Bahasa Banjar tidak jauh berbeda dengan Bahasa Indonesia, yaitu memakai pola subjek diikuti dengan predikat dan objek (SPO) serta tambahan keterangan.

Tabel berikut menyajikan perbandingan dialek dari Bahasa Banjar Hulu dan Bahasa Banjar Kuala.

Tabel 2.2. Perbandingan Dialek Kuala dan Hulu

No.	Kosakata		
	Dialek Kuala	Dialek Hulu	Arti
1	ko-reng	ku-ring	borok
2	lo-ngor	lu-ngur	botak
3	ge-met	gi-mit	pelan
4	o-rang	u-rang	orang
5	se-dang	sa-dang	cukup

Afiks formator derivasional Bahasa Banjar, pembentuk kata kerja menjadi kata benda terdiri dari 2 (dua) awalan dan 1 (satu) akhiran. Pembentukan kata kerja menjadi kata benda berbentuk awalan, yakni /pa-/ dan /ka-/, sedangkan akhiran, yakni /-an/. Sedangkan afiks majemuk derivasional, yaitu konfiks berperan dalam pembentukan kata benda. Terdapat 2 (dua) konfiks yang dapat membentuk kata kerja menjadi kata benda, yaitu /ka-andan/ dan /sa-an/ (Humaidi, 2018).

a) Prefiks (Awalan)

Awalan adalah morfem yang terletak di depan kata yang mengikatnya, misalnya: /di-/ dalam kata *dipacul*. Contoh selengkapnya ditampilkan pada Tabel 2.3.

Tabel 2.3. Contoh Prefiks

Prefiks	Morfem	Arti
/di-/	dipacul	dilepas
/pa-/	pangulir	pemalas
/ba-/	batakun	bertanya
/ta-/	takurihing	tersenyum

b) Infiks (Sisipan)

Sisipan adalah morfem yang terletak di tengah kata yang mengikatnya, misalnya: /el-/ dalam kata *telungkup*. Contoh selengkapnya ditampilkan pada Tabel 2.4

Tabel 2.4. Contoh Infiks

Infiks	Morfem	Arti
/-el-/	telungkup	tiarap
/-em-/	gemirang	gembira
/-in-/	sinambung	sambung
/-ah-/	bahagian	bagian

c) Sufiks (Akhiran)

Akhiran adalah morfem yang terletak di belakang kata yang mengikatnya, misalnya: /an-/ dalam kata *lakian*. Contoh selengkapnya ditampilkan pada Tabel 2.5

Tabel 2.5. Contoh Sufiks

Sufiks	Morfem	Arti
/-an/	lakian	pria
/-kan/	bariakan	berikan
/-i/	ancapi	dipercepat

d) Konfiks (Gabungan Awalan dan Akhiran)

Konfiks merupakan morfem yang ditempatkan di depan dan akhir kata yang mengikatnya, misalnya: /ma-i/ dalam kata *mangalihi*. Contoh selengkapnya ditampilkan pada Tabel 2.6

Tabel 2.6. Contoh Konfiks

Konfiks	Morfem	Arti
/ma-i/	mangalihi	menyulitkan
/ba-an/	baramian	beramai-ramai
/ta-i/	talihati	nampak
/di-i/	dimamai	dimarahi

e) Simulfiks

Simulfiks adalah morfem yang terletak secara simultan ke dalam suku kata pertama dari kata yang mengikatnya, misalnya: /ny-/ dalam kata *nyarik*. Contoh selengkapnya ditampilkan pada Tabel 2.7

Tabel 2.7. Contoh Simulfiks

Simulfiks	Morfem	Arti
/ny-/	nyurui	menyisir
/ng-/	ngintu	itu

Partikel kata dalam Bahasa Banjar meliputi: gin, pang, lah, kah, ja, ha, lih dan am, ditulis setelah kata yang mendahuluinya. Contoh penulisan dalam kalimat antara lain:

a. *Adingku gin badiam di Banjarbaru jua*

   Adikku pun tinggal di Banjarbaru juga

b. *Beapa bulik sungsung, kalo kena pina ditawaakan lawang*

   Kenapa pulang cepat, kalau nanti ditertawakan pintu

c. *Sudah mencariakan, inya jua lah nang disuruh manggawiakan?*

   Setelah mencarikan, dia juga yang disuruh mengerjakan?

d. *Mama tulak sorangan haja ka rumah nini*

   Ibu pergi sendiri saja ke rumah nenek

e. *Inya ja nang tulak madam, kakawalannya tatap bahuma di kampung*

   Dia saja yang pergi merantau, teman-temannya tetap bertani di kampung

f. *Sudah saharian maunjun, si Udin dapat iwak saikung ha*

   Sudah seharian memancing, si Udin dapat seekor ikan saja

g. *Uma ai talalunya, sidin lih nang bahuma sorangan?*

   Aduhai terlalu, beliau yang bertani sendiri?

h. *Ikam tu pang nang pambungasnya di sia*

   Kamulah yang paling cantik di sini

i. *Manggawi tugas sorangan kah, baduakah, nang penting tuntung*

   Mengerjakan tugas sendiri atau berdua, yang penting selesai

2.3.2. Chatbot

Pada tahun 1950, seorang ilmuwan bernama Alan Turing menciptakan Turing Test yang berguna untuk menguji kemampuan komputer dalam mengidentifikasi secara benar sebuah percakapan dari seseorang apakah sedang berkomunikasi dengan komputer atau manusia (Raj, 2019).

Chatbot pertama yang diberi nama Eliza dibangun di MIT Artifial Intelligence Laboratory oleh Joseph Weizenbaum pada tahun 1966 (Khan, 2018). Eliza dirancang untuk menjadi seorang terapis dan sistemnya didasarkan pada simulasi percakapan dengan melakukan pencocokan pola dan metodologi substitusi sehingga dapat memberikan gambaran pemahaman kepada pengguna. Setelah Eliza, ada chatbot bernama Parry yang dikembangkan oleh Kenneth Colby, seorang ilmuwan di Universitas Standford. Parry disimulasikan sebagai seorang pasien yang memiliki sifat paranoid dan berkepribadian ganda ketika dia berkomunikasi dengan terapisnya (Raj, 2019).

Beberapa sistem chatbot kemudian muncul selain di bidang kedokteran. Jabberwocky, diciptakan oleh Rollo Carpenter seorang programmer Inggris. Chatbot ini dimaksudkan untuk mensimulasikan percakapan manusia secara alami tentang hobi, hiburan, dan hal-hal yang menyenangkan atau humor. Lalu ada Tommy, chatbot nirkabel yang dirancang untuk mengatakan ulang setiap pesan yang direkam di dalamnya (Raj, 2019).

Perkembangan teknologi chatbot sampai pada tahun 1995 dimana Richard Wallace membangun A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) dan mendapat Nobel Prize (Raj, 2019). Sayangnya chatbot ini gagal dalam ujicoba

Turing Test. Namun demikian, A.L.I.C.E. mampu memenangkan kompetisi kecerdasan buatan sebanyak 3 (tiga) kali dan mendapatkan Loebner Prize (Khan, 2018). Selanjutnya sistem chatbot semakin menunjukkan perkembangannya melalui *instant messenger* yang diberi nama SmarterChild yang dibangun oleh ActiveBuddy untuk melayani kebutuhan pelanggan, seperti: akses berita, informasi cuaca, informasi saham, jadwal film, dan penampil *yellow pages*. SmarterChild digunakan sebagai asisten pribadi yang dijadikan dasar dari sistem *virtual assistant* sekarang ini (Raj, 2019).

Timeline beberapa sistem chatbot yang telah populer dapat disimak pada Gambar 2.1 berikut ini:

Gambar 2.1. Timeline Aplikasi Chatbot

2.3.3. Natural Language Processing

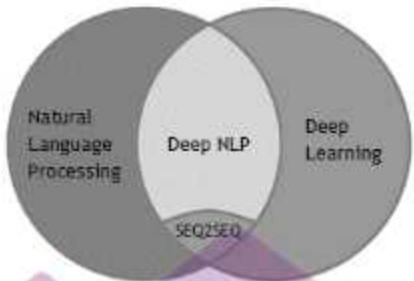
NLP merupakan teknologi yang merepresentasikan kemampuan dari komputer atau sistem yang dirancang untuk menirukan cara berpikir manusia dalam memahami dan memproses bahasa manusia secara alami (Goyal, 2018). NLP merupakan kategori kecerdasan buatan yang dibentuk dari metode komputasi analisis otomatis serta penyajian ulang bahasa manusia yang bertujuan untuk mempermudah jalinan komunikasi dengan mesin. Sekarang ini, riset tentang NLP terus meningkat serta memfokuskan aplikasi teknik-teknik baru pada *deep learning*

yang semakin populer dan telah banyak digunakan, seperti *supervised learning* atau pembelajaran dengan pengawasan. NLP sudah menciptakan beraneka ragam aplikasi yang bermanfaat bagi semua orang, yaitu terjemahan bahasa oleh mesin, klasifikasi kalimat, pendeteksian spam, penentuan konteks kalimat, meringkas naskah, mesin obrolan, serta aplikasi kedokteran (Khurana, 2017).

2.3.4. Deep Learning

Merupakan kategori pembelajaran mesin dimana prosesnya dilakukan di dalam banyak layer proses untuk mengekstrasi dan mengubah bentuk serta melakukan pembelajaran representasi terstuktur pada data statistik dengan model non-linier. Teknologi deep learning sudah menciptakan teknologi canggih di bermacam aspek tugas, yaitu klasifikasi atau *supervised* serta analisa pola atau *unsupervised* khususnya dalam mengolah data citra, suara, serta teks (Goyal, 2018).

Hal yang menjadikan deep learning semakin populer sekarang ini, yaitu: ketepatan, kemampuan, serta fleksibilitas dengan dukungan peningkatan spesifikasi perangkat keras untuk melakukan komputasi atau perhitungan yang besar. Dalam deep learning banyak terdapat teknik-teknik untuk memproses data yang besar dengan sedikit upaya dari manusia sebab dilakukan dengan cara otomatis yang dilakukan oleh mesin dalam melaksanakan ekstraksi fitur dengan rinci. Hal tersebut memerlukan waktu dengan durasi yang lama serta usaha secara intensif (Chollet, 2018).



Gambar 2.2. Ruang Lingkup Seq2Seq dalam Deep NLP

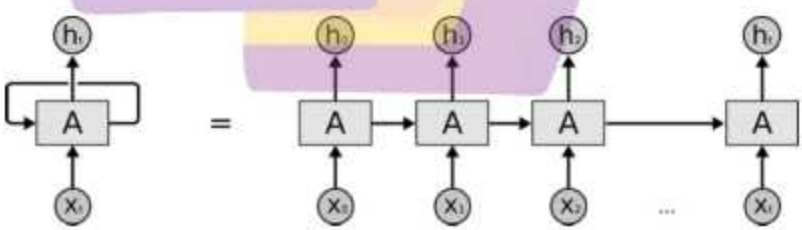
Deep NLP berada pada titik potong antara Natural Language Processing dengan Deep Learning. Sedangkan model Seq2Seq yang digunakan pada penelitian ini berada di dalam area Deep NLP.

2.3.5. Recurrent Neural Network

RNN masuk dalam kategori *supervised deep neural network*. Training RNN untuk level supervised memerlukan dataset training dari sepasang input dan target untuk meminimalisir perbandingan angka loss dari pasangan tersebut dengan cara melakukan optimalisasi terhadap bobot dari jaringan yang tersedia (Salehinejad, 2017). Arsitektur RNN dibangun atas *neuron-neuron* dengan *feedback* yang diulang-ulang. Di tiap tahapan waktu (epoch), neuron menyambut data, melaksanakan perhitungan, lalu menghasilkan output. RNN mampu melakukan perhitungan-perhitungan yang dinamis dalam kondisi tersembunyi atau *hidden state* secara jangka panjang, alhasil dapat membuat model ekspresif serta sangat kokoh untuk melakukan tugas-tugas sekuensial. Misalnya: identifikasi suara, pengucapan sintesis, computer vision, pengolahan keterangan pada video, serta

susunan teks. Selain itu, RNN juga dapat diterapkan dalam ranah dua dimensi di seluruh data spasial seperti gambar, dan bahkan ketika diterapkan pada data yang melibatkan waktu, jaringan dimungkinkan memiliki koneksi yang mundur ke masa lalu, asalkan seluruh urutan diamati sebelum diumpankan ke jaringan (Goodfellow. 2016).

RNN terdiri dari tiga layer, yaitu *input layer*, *recurrent hidden layer*, serta *output layer* (Salehinejad, 2017). Input layer mempunyai bagian input dan terhubung secara penuh dengan bagian tersembunyi yang terdapat pada hidden layer. Bagian tersembunyi tersebut saling tersambung secara berulang-ulang. Hidden layer dapat didefinisikan dengan ingatan atau ruang kondisi yang berukuran besar dan bersifat non-linier agar bisa mengingat serta mengerjakan informasi di masa lampau. *Hidden state* merangkum seluruh informasi yang bersifat unik yang dibutuhkan selaku kondisi terakhir dari jaringan lewat serangkaian tahapan waktu. Informasi tersebut kemudian diintegrasikan sehingga bisa memastikan perilaku dari jaringan untuk masa akan datang serta melaksanakan perkiraan yang tepat pada output layer (Goodfellow, 2016). Arsitektur RNN dapat diamati pada Gambar 2.3.



Gambar 2.3. Arsitektur RNN Sederhana (Colah, 2015)

Sekumpulan jaringan syaraf (A) menerima input (X_t) dan menghasilkan output (H_t). Perulangan memungkinkan informasi untuk diteruskan dari satu langkah jaringan ke langkah berikutnya. Perulangan ini membuat jaringan saraf berulang tampak misterius. Namun, sebenarnya tidak begitu berbeda dari jaringan syaraf biasa. RNN dapat dianggap sebagai beberapa salinan dari jaringan yang sama, masing-masing meneruskan pesan ke penerusnya. Sifat seperti rantai ini mengungkapkan bahwa RNN sangat terkait dengan data sekuensial yang merupakan arsitektur alami dari jaringan saraf (Colah, 2015).

2.3.6. Mekanisme Gerbang

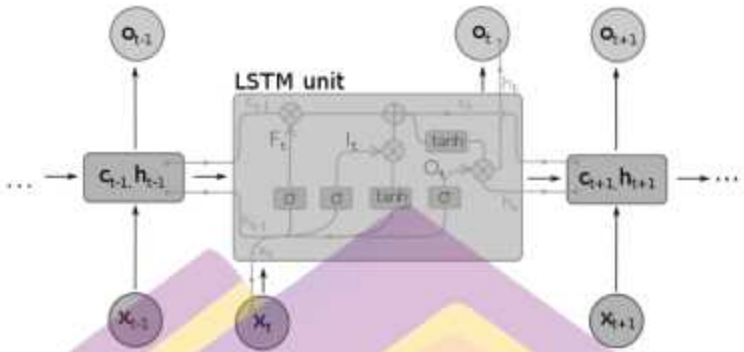
Di balik keunggulannya, RNN memiliki kekurangan dimana saat dilakukan training dalam waktu yang panjang dengan *gradient descent* dapat menciptakan permasalahan gradien yang hilang atau meledak. Ketika training, saat angka gradien dipropagasi ulang, angka itu lalu dikalikan bobot yang nilainya kurang dari satu. Hal ini menyebabkan angka gradien yang diperoleh ketika perulangan lama kelamaan akan berkurang dan menghadapi kehilangan angka. Angka gradien dengan cara eksponensial dapat membengkak, sebab angka gradien dikalikan bobot bernilai lebih dari satu. Dengan demikian, angka gradien perlu dicegah agar tidak memiliki nilai kurang dari atau lebih dari satu (Salehinejad, 2017).

Untuk mengatasi gradien yang hilang atau meledak yaitu dengan cara melakukan modifikasi terhadap arsitektur dari model dengan menambahkan mekanisme gerbang yang didesain agar dapat menyimpan informasi dalam rentang

waktu yang panjang. Mekanisme gerbang yang sangat populer yaitu LSTM (Long-Short Term Memory) serta GRU (Gated Recurrent Unit).

Mekanisme gerbang LSTM dan GRU memang telah sukses menanggulangi permasalahan gradien yang menghilang ataupun meledak sepanjang training jangka panjang. Namun, diperlukan beberapa fakta empiris untuk membuktikan unit gerbang manakah yang paling bagus untuk permasalahan optimasi model percakapan. Chung (2014) mendapatkan hal baru bahwa GRU memberikan hasil model yang sebanding dengan LSTM, namun GRU lebih berdaya guna atau efisien untuk komputasinya. Hal ini dikarenakan GRU disusun dari 2 gerbang: *reset* dan *update*; sedangkan LSTM disusun dari 3 gerbang, yaitu: *input*, *output*, serta *forget* dan ada unit ingatan yang dikontrol oleh gerbang itu (Chung, 2014). GRU memiliki sedikit parameter sehingga waktu training serta konvergensinya lebih singkat dibandingkan dengan LSTM. Namun, LSTM dapat tetap menjaga agar pemahaman informasi pada tiap tahap lebih lama sehingga untuk training jangka panjang dengan data yang banyak, LSTM lebih bagus dari GRU secara teori. Pada prinsipnya, LSTM bisa memakai bagian memorinya untuk menyimpan ingatan informasi yang jauh jaraknya serta dapat melacak bermacam ciri teks yang diproses saat itu (Karpathy, 2016). Di sisi lain, dari penelitian tentang model Seq2Seq menggunakan Gated Recurrent Unit (GRU) diketahui memiliki kemampuan untuk mempelajari representasi frasa bahasa dan sintaksis secara semantik pada terjemahan mesin (Cho, 2014).

Gambar 2.4 di bawah ini menunjukkan diagram untuk satu unit Long Short-Term Memory (LSTM).

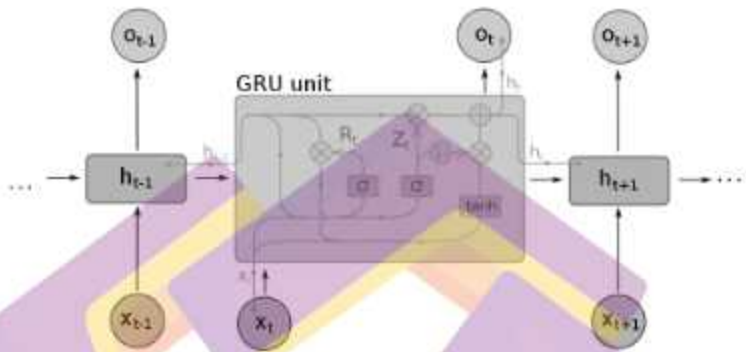


Gambar 2.4. Unit LSTM (Wikimedia Commons, 2017)

LSTM memiliki persamaan dengan variabel X_t yang merupakan vektor masukan ke unit LSTM, Variabel F_t merupakan vektor aktivasi untuk *forget gate*, Variabel I_t merupakan vektor aktivasi untuk *input gate*, Variabel O_t merupakan vektor aktivasi untuk *output gate*, Variabel H_t merupakan vektor untuk *hidden state* dimana vektor ini populer dengan vektor output dari LSTM, Variabel C_t merupakan vektor untuk *cell state*, Sedangkan W-U-b merupakan matrik berbobot serta sebagai acuan untuk vektor bias yang harus dipahami saat dilakukan proses training (Chung, 2014).

$$
\begin{aligned}
F_t &= \sigma_g\,(W_f x_t + U_f h_{t-1} + b_f) \\
I_t &= \sigma_g\,(W_i x_t + U_i h_{t-1} + b_i) \\
O_t &= \sigma_g\,(W_o x_t + U_o h_{t-1} + b_o) \\
C_t &= f_t \circ c_{t-1} + i_t \circ \sigma_g\,(W_c x_t + U_c h_{t-1} + b_c) \\
H_t &= O_t \circ \sigma_h\,(c_t)
\end{aligned}
\qquad (1)
$$

Gambar 2.5 di bawah ini menunjukkan diagram untuk satu unit Gated Recurrent Unit (GRU).



Gambar 2.5. Unit GRU (Wikimedia Commons, 2017)

GRU memiliki persamaan dengan variabel X_t yang merupakan vektor masukan ke unit GRU, Variabel Z_t merupakan vektor untuk *update gate*, Variabel R_t merupakan vektor untuk *reset gate*, Variabel O_t merupakan vektor aktivasi untuk *output gate*, Variabel Ht merupakan vektor untuk *hidden state* dimana vektor ini populer dengan sebutan vektor output dari GRU, W-U-b merupakan matrik berbobot serta sebagai acuan untuk vektor bias yang harus dipahami saat dilakukan proses training (Chung, 2014).

$$\begin{aligned} F_t &= \sigma_g (W_z x_t + U_z h_{t-1} + b_z) \\ R_t &= \sigma_g (W_r x_t + U_r h_{t-1} + b_r) \\ C_t &= Z_t \circ h_{t-1} + (1-Z_t) \circ \sigma_g (W_h x_t + U_h (r_t \circ h_{t-1}) + b_h) \end{aligned} \quad (2)$$

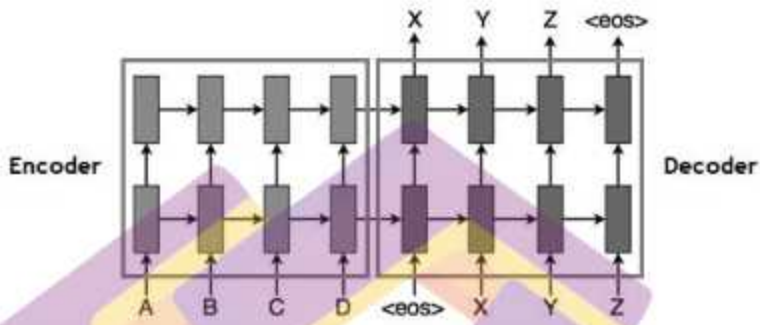
2.3.7. Sequence to Sequence

RNN terbukti bermanfaat untuk membuat model informasi secara sekuensial yang dapat diterapkan pada pemrosesan bahasa alami. Model percakapan menggunakan jaringan saraf tiruan yang populer adalah Sequence-to-Sequence (Seq2Seq). Popularitas model Seq2Seq dimulai pada tahun 2014 saat Sutskever melakukan tugas terjemahan mesin untuk menerjemahkan Bahasa Inggris ke Bahasa Prancis. Model ini disusun dari dua jaringan RNN (2-RNN) yang dikenal dengan model encoder-decoder. Encoder pada jaringan RNN melaksanakan proses encoding secara sekuensial terhadap masukan menuju vektor konteks dalam dimensi yang tidak berubah kemudian decoder menggunakannya menjadi benih agar dapat dibentuk sasaran yang bersifat sekuensial. Hal inilah yang menjadikan Seq2Seq kerap pula dikenal dengan model encoder-decoder (Goodfellow, 2016).

Model Seq2Seq sanggup memperhitungkan kata selanjutnya berdasarkan kata atau konteks sebelumnya dari sebuah obrolan sehingga akan membentuk model generatif yang memiliki kemampuan untuk menghasilkan rangkaian kata dan menghasilkan kalimat baru. Sedangkan bentuk *retrieval* mengutip perkataan secara utuh pada pra-konstruksi repositori (Goodfellow, 2016).

Riset tentang Seq2Seq memberikan peningkatan kemampuan untuk membentuk model dengan ketepatan yang sangat baik. Ada 2 (dua) metode dalam Seq2Seq, yaitu dengan menggunakan variasi dua gerbang yang berlainan (LSTM dengan GRU) pada model encoder-decoder dan dengan menggunakan mekanisme gerbang multi-layer, seperti dua layer GRU pada model encoder serta dua layer

GRU pada model decoder (Bay, 2017). Model Seq2Seq bisa dilihat pada ilustrasi Gambar 2.6.



Gambar 2.6. Model Seq2Seq (Prasanna, 2020)

A, B, C, D adalah input di mana W adalah vektor konteks yang menyimpan informasi vektor pemetaan dari semua masukan menggunakan urutan output vektor konteks X, Y, Z yang dipetakan dan diprediksi (Prasanna, 2020).

2.3.8. Mekanisme Atensi

Arsitektur Seq2Seq mempunyai keterbatasan sebab membutuhkan semua input yang dikodekan ke dalam *single vector* dengan ukuran panjang yang tetap, yang mengakibatkan model tidak bisa menggeneralisasi input lebih lama dari kapasitas tetapnya. Salah satu metode untuk menanggulangi permasalahan ini yaitu dengan menggunakan mekanisme atensi (Bahdanau, 2014).

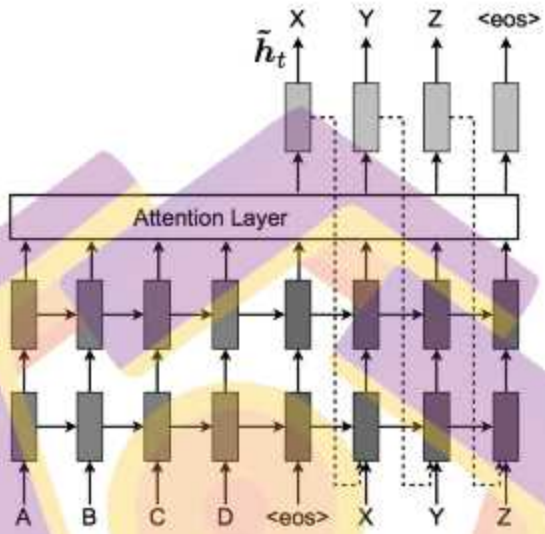
Mekanisme atensi sudah memperoleh ketenaran dalam training neural network dengan menirukan mekanisme atensi oleh manusia secara visual. Atensi

visual oleh manusia dilakukan dengan memperhatikan secara khusus area tertentu dari obyek gambar yang memiliki resolusi tinggi sembari memperhatikan obyek gambar di sekelilingnya yang memiliki resolusi rendah, lalu menentukan kesesuaian titik fokus dalam tahapan-tahapan waktu. Mekanisme atensi mengalami kemajuan serta menjadikan model bisa mempelajari keberpihakan di antara modalitas yang berlainan, seperti visualisasi gambar serta penjelasan gambar dalam proses *captioning* pada sebuah citra (Luong, 2015).

Pada pemrosesan bahasa alami, input dikonversi menjadi kode dengan dimensi tetap agar rangkaian kata yang panjang tidak bisa digeneralisasi lebih panjang dari ukuran yang telah ditetapkan. Mekanisme atensi diaplikasikan agar dapat menyeleraskan kalimat pada pemrosesan teks mesin dan menanggulangi kemampuan yang kurang baik pada pemrosesan teks yang panjang. Selain itu juga dapat meluaskan vektor dimensi tetap dengan membuat model agar otomatis melakukan pencarian bagian dari sumber teks yang sesuai untuk melakukan prediksi kata sasaran (Bahdanau, 2014).

Metode atensi membuat jaringan dapat menyimpan memori tertentu yang telah diterima dari input secara baik dan dipakai untuk melakukan generasi kata di dalam decoder. Mekanisme atensi melakukan perhitungan bobot setiap kata yang diinput untuk menentukan banyaknya atensi yang ada pada input kata tersebut. Hasil penjumlahan bobot paling besar diberikan nilai satu dan setelah itu dipakai untuk melakukan perhitungan nilai rata-rata dari hidden layer yang terakhir sesudah melakukan proses pada tiap input kata yang disebut dengan konteks. Setelah itu

dimasukkan ke lapisan *softmax* dengan hidden layer terakhir dari tahapan selama decoding (Lopyrev, 2015). Layer atensi Seq2Seq dapat diamati pada Gambar 2.7.



Gambar 2.7. Layer Atensi dalam Seq2Seq (Luong, 2015)

Vektor atensi $\tilde{h}_t$ diumpankan sebagai input ke tahapan berikutnya untuk menginformasikan model tentang keputusan penyelarasan data di masa lalu.

2.3.9. Gradient Descent

Gradient Descent merupakan teknik optimasi sederhana dan telah terkenal penggunaannya di lingkup deep learning. Model deep learning umumnya mempunyai acuan bobot, bias dan *cost function* yang disebut dengan nilai error untuk melakukan evaluasi sejauh mana idealitas dari set acuan tersebut. Ide gradient descent didasari dari penyesuaian bobot dari model dengan cara melakukan

pencarian turunan cost function yang terkait dengan setiap anggota matriks pembobotan pada model. Gradient descent dimulai dari angka awal yang dijadikan acuan lalu secara berulang mengarah ke angka acuan yang melakukan minimalisasi nilai error tersebut (Kingma, 2014).

Teknik optimasi gradient descent memiliki ekstensi yang bisa dijadikan alternatif, seperti SGD (Stochastic Gradient Descent), AdaGrad (Adaptive Gradient Algorithm), RMSProp (Root Mean Square Propagation), serta Adam (Adaptive Moment Estimation). SGD menjaga tingkatan *single training* bagi seluruh update bobot serta tingkatan training yang berganti sepanjang proses training. AdaGrad menjaga tingkatan training pada setiap acuan agar mengoptimalkan kemampuan pada permasalahan gradien yang menyebar. Demikian pula dengan RMSProp yang menjaga laju training pada setiap acuan yang dicocokkan sesuai dengan besar nilai rata-rata dari bobot gradien terakhir sehingga menjadikan RMSProp bagus dipakai untuk permasalahan update bobot dengan cara sekuensial. Sedangkan Adam mengkombinasikan kelebihan dua ekstensi tersebut dalam melakukan perhitungan laju training yang mampu beradaptasi untuk acuan yang berbeda dari prediksi momen awal dan momen berikutnya (Kingma, 2014). Alasan inilah yang menjadikan Adam dianjurkan untuk dipakai sebagai algoritma standar sebab memiliki kinerja lebih bagus dibandingkan RMSProp dan AdaGrad (Zhu, 2020).

2.3.10. Activation

Activation function memegang peranan yang sangat penting dalam training jaringan syaraf karena berfungsi untuk memetakan input dan output sebagai data

non-linier. Fungsi aktivasi yang umum digunakan adalah: *tanh*, *softmax*, *sigmoid*, *ReLu* (Breier, 2018) dan *Leaky ReLu* (Sharma, 2019). Arsitektur vanilla RNN masih sering menggunakan fungsi aktivasi tanh atau sigmoid, atau bahkan keduanya. Misalnya, LSTM menggunakan aktivasi sigmoid untuk koneksi berulang dan aktivasi tanh untuk output (Ienco, 2017). Tanh memiliki output pada range -1 sampai 1 yang dapat membedakan nilai negatif dan positif. Softmax digunakan untuk mengatasi permasalahan *multi-class classification* yang menghasilkan output pada range 0 sampai 1 dimana semua jumlah output dari softmax adalah 1. Hal ini mendasari softmax digunakan sebagai fungsi aktivasi pada layer decoder di dalam mekanisme atensi (Lopyrev, 2015).

2.3.11. Layer Model

Embedding merupakan teknik untuk mengubah alfanumerik menjadi vektor yang biasa digunakan dalam aplikasi terkait NLP atau tugas lain yang melibatkan jaringan saraf. Beberapa metode yang populer antara lain: Word2Vec, fastText dan GloVe. Namun, framework Keras juga telah menyediakan layer embedding yang praktis untuk digunakan dalam training model. Layer embedding mengubah setiap kata menjadi vektor dengan panjang tetap. Vektor yang dihasilkan adalah vektor padat dengan nilai *real*, bukan hanya 0 dan 1. Panjang vektor kata yang tetap merepresentasikan kata dengan cara yang lebih baik dengan dimensi yang diperkecil. Layer embedding berfungsi seperti tabel pencarian. Kata adalah kunci dalam tabel ini, sedangkan vektor padat (*dense*) adalah nilainya.

Layer *RepeatVector* digunakan untuk mengkonversi input menjadi tensor dalam bentuk *shape* meliputi *batch size*, *sequence length*, dan *input size*. Sebagai contoh, jika shape akan dimasukkan ke layer GRU, maka dapat digunakan layer RepeatVector untuk mengulang input dalam jumlah yang ditetapkan sebanyak *n* kali. Misalnya, jika RepeatVector dengan nilai 15 diterapkan ke layer yang memiliki input shape "*batch_size, 256*", maka bentuk output dari layer tersebut akan menjadi "*batch_size, 15, 256*".

Layer *Batch Normalization* digunakan agar distribusi dari input ke layer tertentu tidak berubah dari waktu ke waktu akibat pembaruan parameter dari setiap batch. Hal ini memungkinkan untuk menggunakan learning rate yang lebih tinggi tanpa risiko divergensi. Batch Normalization melakukan regularisasi pada model dan mengurangi kebutuhan untuk *dropout* (Srivastava, 2014) dan memungkinkan untuk mencegah jaringan terjebak dalam mode saturasi (Ioffe, 2015).

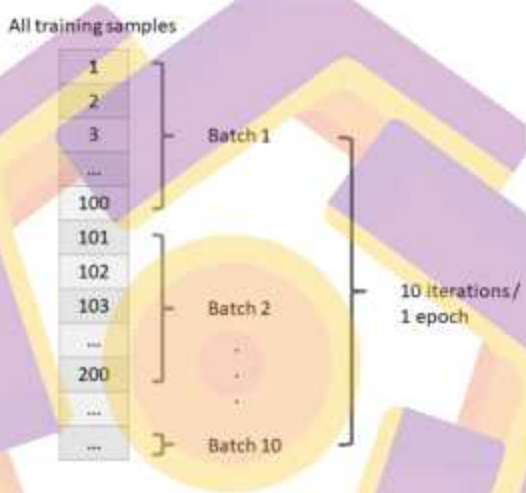
2.3.12. Parameter Model

Parameter model merupakan variabel konfigurasi internal sebagai kunci algoritma pembelajaran mesin yang dibutuhkan oleh model saat membuat prediksi yang dipelajari dari data training historis. Setiap model memiliki parameter penting yang tidak dapat diestimasi secara langsung berdasarkan data. Pengaturan parameter model secara manual disebut sebagai *tuning hyperparameter* karena tidak ada rumus analitik yang tersedia untuk menghitung nilai yang sesuai (Kuhn, 2013). Tuning hyperparameter perlu dilakukan untuk menemukan kombinasi optimal dari parameter yang digunakan untuk meminimalkan cost function.

Batch size atau ukuran batch adalah hyperparameter yang digunakan untuk menentukan jumlah sampel yang harus dipropagasi melalui jaringan. Batch digambarkan sebagai iterasi *for-loop* atas satu atau lebih sampel dalam membuat prediksi. Pada akhir batch, prediksi dibandingkan dengan variabel output yang diharapkan, kemudian dihitung nilai lossnya sebelum dilakukan pembaruan bobot pada model. Nilai batch size dapat ditentukan dari 1 hingga seterusnya, tetapi secara umum berupa 2^n diawali dari 2 hingga 2048 sesuai dengan penelitian yang dilakukan Masters (2018) tentang analisis perbandingan kinerja model berdasarkan ukuran batch. Dari penelitian tersebut didapati bahwa secara umum nilai batch size yang efektif dalam melakukan training yang stabil yaitu pada kisaran 4 hingga 64. Hal ini juga sejalan dengan yang dilakukan oleh Su (2020) yang menggunakan batch size sebesar 64 untuk membuat model chatbot berbahasa Mandarin. Namun, Masters juga mendapati bahwa ukuran batch 128 juga efektif untuk menangani dataset yang lebih besar.

Jumlah epoch adalah hyperparameter yang digunakan untuk menentukan berapa kali algoritma pembelajaran akan bekerja melalui seluruh dataset training. Satu epoch terdiri dari satu atau lebih batch. Epoch digambarkan sebagai iterasi for-loop yang di dalamnya terdapat for-loop bersarang yang memproses perulangan pada setiap batch sampel data, dimana satu batch memiliki jumlah sampel dari ukuran batch yang telah ditentukan. Jumlah epoch yang digunakan dapat diatur dan ditentukan dari 10, 50, 100, 200 sampai 1000 atau lebih besar lagi sesuai dengan kebutuhan penelitian. Penelitian yang dilakukan oleh Faza (2018) dalam upaya meningkatkan kinerja klasifikasi dengan mengolah 4000 baris dataset pada training

deep neural network menggunakan 1000 epoch menunjukkan nilai *crossentropy error* yang mulai stabil pada epoch ke-400. Untuk evaluasi, Faza melakukan pengamatan nilai metrik pada setiap 50 epoch yang dihasilkan selama proses training. Ilustrasi keterkaitan antara batch dan epoch dapat disimak pada Gambar 2.8.

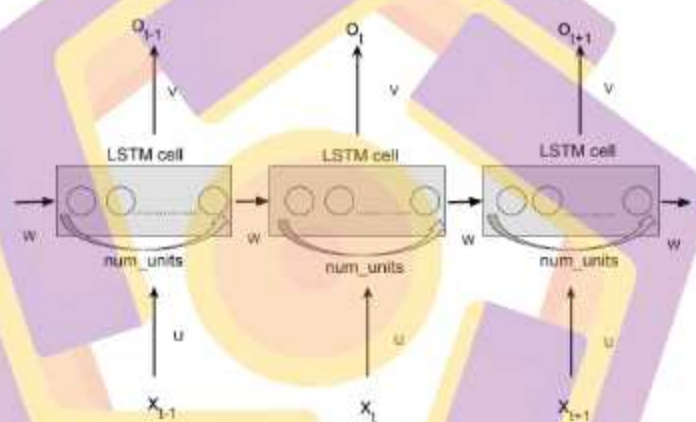


Gambar 2.8. Ilustrasi Batch dan Epoch

Learning rate digunakan untuk melakukan perhitungan update bobot selama proses training (Moolayil, 2019). Idealnya, semakin besar nilai learning rate, maka proses training akan berjalan semakin cepat dan semakin cepat pula nilai bobot mendekati nilai sebenarnya. Learning rate pada neural network umumnya berada pada range 0.0 dan 1.0. Variasi nilai learning rate yang umum digunakan adalah 0.1, 0.01, 0.001 dan 0.0001. Nilai default learning rate yang digunakan oleh Keras

adalah 0.001 dimana nilai ini dianggap paling ideal untuk menangani berbagai macam kasus secara umum.

Number of units atau num_units adalah jumlah unit memori RNN untuk setiap input berupa urutan yang melekat satu sama lain. Masing-masing num_units meneruskan informasi yang disaring ke unit memori berikutnya pada neural network. Gambar 2.9 menunjukkan interpretasi contoh dari num_units yang terdapat di dalam sel LSTM.



Gambar 2.9. Number of Units dalam Sel LSTM (Chhabra, 2017)

Nilai dari num_units yang umum digunakan adalah 64, 128, 256 dan 512 (Martin, 2018). Penyesuaian nilai-nilai tersebut dapat dipilih secara eksperimental untuk melihat pengaruhnya terhadap akurasi training. Semakin banyak jumlah num_units maka semakin baik juga hasil akurasinya terhadap data training, tetapi memungkinkan dapat terjadi overfitting. Jumlah num_units pada hidden layer juga berpengaruh pada durasi training. Semakin banyak jumlah num_unit maka semakin menambah durasi training pada pembangunan model. Nilai num_units yang biasa

digunakan dalam training model adalah 256 seperti yang tertulis pada dokumentasi Keras (Zhu, 2020).

Dropout adalah teknik untuk mencegah overvitting dengan cara memutuskan unit pada layer jaringan syaraf beserta semua koneksi masuk dan keluarnya (Srivastava, 2014). Unit yang akan diputuskan dilakukan secara acak. Petunjuk yang tertulis pada dokumentasi Keras (https://keras.io/ko/getting-started/sequential-model-guide) menggunakan 0.25 dan 0.5 sebagai nilai dropout pada data sekuensial. Srivastava menggunakan variasi nilai 0.5 pada hidden unit dan 0.8 pada input unit sebagai nilai yang mendekati optimal untuk model.

2.3.13. Confusion Matrix dan Loss

Analisis hasil merupakan tahapan untuk melakukan pengujian model yang sudah dihasilkan sampai pada tahap implementasi. Tahapan ini dimaksudkan untuk melakukan pengujian terhadap indikator-indikator yang terkait dalam rangka memberikan informasi model manakah yang bagus dan sesuai dengan rumusan masalah. Tahapan evaluasi yang dilakukan bersumber pada parameter yang sudah ditetapkan, serta kesimpulannya bisa dijadikan sebagai materi rujukan untuk riset berikutnya.

Pengukuran yang dipakai dalam melakukan pengujian berupa nilai *accuracy*, *precision*, *recall*, *F1*, dan *loss* untuk setiap kategori model. Hasil akurasi dari proses generasi rangkaian kalimat dinilai dari seberapa tepat pemetaan konteks kalimat dengan kalimat yang dijadikan target percakapan. Pengujian juga memperhatikan hasil precision dan recall. Nilai precision memperlihatkan

kedekatan dari koleksi konteks sesudah dipetakan ke kalimat yang dijadikan target. Hal ini penting untuk mengetahui ketepatan dari informasi pertanyaan dengan respon jawaban. Recall memperlihatkan kelengkapan sistem dalam menemukan lagi hasil prediksi untuk nilai positif. Accuracy, precision dan recall didefinisikan dengan:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Dimana,

True Positif (TP) = jumlah data dengan kelas positif dan hasil prediksi benar

True Negatif (TN) = jumlah data dengan kelas negatif dan hasil prediksi benar

False Positif (FP) = jumlah data dengan kelas positif dan hasil prediksi salah

False Negatif (FN) = jumlah data dengan kelas negatif dan hasil prediksi salah

F-measure (F1) merupakan metrik evaluasi yang menggabungkan angka precision dan recall sebab kedua nilainya bisa mempunyai bobot yang berbeda. Sehingga, F1 merupakan rata-rata harmonis yang diperoleh dari hasil precision dan recall dengan rentang nilai dari 0 hingga 1. Nilai F1 didefinisikan dengan:

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \quad (6)$$

Nilai metrik digunakan untuk mengevaluasi dan memilih model yang terbaik sedangkan loss digunakan untuk mengevaluasi dan menganalisis optimasi model. Semakin rendah nilai loss yang dihasilkan oleh suatu model maka semakin

baik pula kualitas model tersebut. Teknik optimisasi menggunakan optimizer digunakan untuk meminimalkan nilai loss.

2.3.14. TensorFlow dan Keras

TensorFlow adalah framework pembelajaran mesin yang dibuat dengan lisensi *open source* dengan dukungan dari Google. TensorFlow merupakan *interface* pemrograman untuk melakukan kalkulasi numerik menggunakan grafis alir data. Bermacam node yang terdapat di dalam grafis berupa perhitungan matematika dan tepi grafis berupa data array dengan banyak dimensi yang disebut sebagai tensor yang mengalir di antaranya. TensorFlow bersifat fleksibel serta bisa dipakai untuk keperluan training pada model deep learning menggunakan bermacam-macam algoritma (Abadi, 2016) serta sudah banyak dipakai di banyak bidang riset seperti identifikasi suara, *computer vision*, robotika, pemrosesan teks, NLP, dan masih banyak lagi.

Keras adalah API (Application Programming Interface) dari deep learning yang dibangun dengan Python dan berjalan pada platform pembelajaran mesin TensorFlow. Keras dikembangkan dengan tujuan untuk fokus pada eksperimen instan dan memiliki kemampuan untuk mewujudkan ide menjadi hasil secepat mungkin sebagai kunci dalam melakukan riset yang baik dan efektif.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian dilakukan dengan eksperimen, dimana dilakukan beberapa kali percobaan pembangunan model dengan menerapkan variasi mekanisme gerbang dan variasi hyperparamater tertentu. Penelitian ini merupakan penelitian deskriptif yang menggambarkan kondisi faktual dari obyek yang diteliti berdasarkan karakteristik tertentu dengan hasil yang dianalisis secara cermat. Sedangkan pendekatan yang dilakukan bersifat kuantitatif karena menganalisis hasil evaluasi eksperimen berdasarkan data empiris yang telah didapatkan.

3.2. Metode Pengumpulan Data

Studi Literatur dilakukan dengan mencari sumber rujukan yang berkaitan dengan arsitektur 2-RNN (Seq2Seq). Berikutnya, mencari sumber rujukan mengenai optimasi Seq2Seq yang terdiri dari model gerbang, mekanisme atensi, serta gradient descent. Sumber rujukan yang diambil berasal dari jurnal nasional dan internasional serta diperoleh dari buku atau pustaka elektronik dan literatur lainnya.

Data percakapan Bahasa Banjar diambil dari database aplikasi live support yang memuat pertanyaan dan jawaban antara pendaftar atau calon mahasiswa dengan staf admisi pada kampus XYZ. Topik percakapan yang dilakukan adalah seputar penerimaan mahasiswa baru, perkuliahan dan kegiatan akademik.

3.3. Metode Analisis Data

Dataset percakapan yang telah melewati proses *cleaning* disusun dalam bentuk pasangan kalimat sesuai dengan konteks dan target, sehingga sesuai dengan arsitektur model Seq2Seq menggunakan kerangka kerja deep learning. Model dibangun dengan bahasa pemrograman Python dengan framework TensorFlow dan Keras sebagai alat penelitian serta library Matplotlib untuk visualisasi training.

3.4. Dataset

Dataset percakapan Bahasa Banjar diperoleh dari hasil percakapan antara pendaftar atau calon mahasiswa dengan pihak akademisi tentang hal-hal yang berkaitan dengan penerimaan mahasiswa baru, perkuliahan dan kegiatan akademik. Dataset ini diperoleh dari database aplikasi live support yang tersedia pada sistem kampus XYZ selama 8 bulan dari Maret hingga Oktober 2021.

Gambar 3.1. Tampilan Aplikasi Live Support

Tabel 3.1 berikut ini merupakan sample percakapan Bahasa Banjar yang dilakukan antara pendaftar atau calon mahasiswa sebagai pengguna dengan pihak akademisi sebagai admin.

Tabel 3.1. Sample Percakapan Antara Pengguna dengan Admin

(Q) Pengguna	*Kawa kah ulun mandaftar wayah ini?* (Bisakah saya registrasi sekarang?)
(A) Admin	*Kawa haja ding ai amun pian handak balalakas* (Bisa saja dik kalau anda ingin cepat)
(Q) Pengguna	*Apa haja garang jurusannya? Sebarataan hinggan jurusan kumputer haja kah?* (Apa saja sih jurusannya? Semuanya apakah jurusan komputer saja?)
(A) Admin	*Kada ding, ada ai jurusan nang lain. Pian handak jurusan apa?* (Tidak dik, ada juga jurusan yang lain. Anda mau jurusan apa?)
(Q) Pengguna	*Unda handak jurusan teknik sipil pang ada lah?* (Aku mau registrasi jurusan teknik sipil apakah ada?)
(A) Admin	*Inggih kawa ai pian amun handak mandaftar jurusan nitu.* (Iya, bisa saja kalau anda ingin registrasi jurusan itu.)

Untuk memberikan gambaran tentang proses pengambilan, pembuatan dan pengolahan dataset, disajikan alur pembuatan dataset seperti yang terlihat pada Gambar 3.2 berikut ini:

Gambar 3.2. Alur Pembuatan Dataset

Pengambilan dataset dilakukan dengan menyampaikan permohonan izin untuk mengambil data percakapan kepada pimpinan kampus terlebih dahulu. Setelah permintaan dikabulkan, maka langkah selanjutnya adalah melakukan login pada server MySQL sesuai kredensial yang telah diberikan untuk mengakses

database percakapan. Data yang terdapat pada tabel percakapan terdiri dari 8.394 baris dimana setiap percakapan ditandai dengan 3 (tiga) jenis konteks, yaitu: teks, gambar dan file.

Gambar 3.3. Tabel Percakapan Aplikasi Live Support

Selanjutnya, data percakapan diekspor ke dalam bentuk file CSV menggunakan kueri MySQL dengan kriteria baris yang hanya memiliki jenis konteks berupa teks. Hal ini akan mengeliminasi data percakapan yang memiliki jenis konteks berupa gambar dan file. File CSV yang telah berhasil diekspor terdiri dari 7.648 baris, berupa teks kalimat, emoji, dan berbagai macam karakter non-alfabetik. Dalam hal ini, maka perlu dilakukan tahapan filtering untuk membersihkan teks percakapan sehingga akan didapatkan dataset yang hanya terdiri dari kalimat, angka dan tanda baca saja.

Proses filtering digunakan untuk menghilangkan semua karakter khusus sehingga akan didapatkan teks percakapan yang terdiri dari karakter a-z, angka 0-9 dan tanda baca percakapan, seperti: koma, titik, tanya tanya, tanda seru, garis mendatar, garis miring dan *ampersand*. Semua tanda baca tersebut tetap

dipertahankan dengan tujuan agar antara konteks dan target percakapan dapat memberikan makna yang sesuai dengan maksud pembicaraan.

Proses normalisasi yang dilakukan berupa penghapusan spasi yang berlebihan antar token kata, perbaikan singkatan dan kesalahan pengetikan pada teks percakapan. Misalnya, "*mun kd bisa komputer kw leh kuliah*" menjadi "*mun kada bisa komputer kawa leh kuliah*". Hingga saat ini masih belum tersedia pustaka *word normalizer* untuk Bahasa Banjar sehingga proses ini dilakukan secara manual dan terbatas. Contoh daftar perbaikan kata dapat disimak pada Tabel 3.2.

Tabel 3.2. Contoh Daftar Perbaikan Kata

Kata Asal	Kata Pengganti	Arti
kam	ikam	kamu
lun	ulun	saya
kd, kadak	kada	tidak
gih, geh	inggih	iya
kdd	kadada	tidak ada
kw	kawa	bisa

Proses *casefolding* bertujuan untuk mengubah penulisan semua karakter pada teks percakapan menjadi huruf kecil dengan maksud untuk menyeragamkan teks yang akan diproses. Sebagai contoh, "*Kawa haja Ding ai amun Pian handak balalakas*" akan diubah menjadi "*kawa haja ding ai amun pian handak balalakas*".

Dataset yang diperoleh setelah dilakukan proses filtering dan normalisasi menjadi 6.100 baris percakapan. Setiap percakapan berisi satu atau lebih pertanyaan (sebagai kalimat konteks) dan satu atau lebih jawaban (sebagai kalimat target).

Hasil akhir dari tahapan preprocessing terhadap data percakapan seperti ditunjukkan pada Gambar 3.4.

	user-admin
0	kawa kah ulun mandaftar wayah ini ?
1	kawa haja ding ai amun pian handak balalakas
2	apa haja garang jurusannya ? sebarataan hinggan jurusan kumputer haja kah ?
3	kada ding, ada ai jurusan nang lain. pian handak jurusan apa ?
4	ulun handak jurusan teknik sipil pang ada lah ?
5	inggih kawa ai pian amun handak mandaftar jurusan nitu.

Gambar 3.4. Sample Percakapan

Terakhir, teks yang sudah bersih diubah ke dalam bentuk pasangan konteks dan target (*pairing*). Pairing adalah proses menjadikan teks percakapan ke dalam bentuk pasangan-pasangan yang berurutan dan saling berkaitan antara satu sama lain. Tabel 3.3 memberikan gambaran dari proses tersebut.

Tabel 3.3. Bentuk Pairing Konteks dan Target

Baris	Teks
1	kawa kah ulun mandaftar wayah ini ? <TAB> kawa haja ding ai amun pian handak balalakas
2	kawa haja ding ai amun pian handak balalakas <TAB> apa haja garang jurusannya ? sebarataan hinggan jurusan kumputer haja kah ?
3	apa haja garang jurusannya ? sebarataan hinggan jurusan kumputer haja kah ? <TAB> kada ding , ada ai jurusan nang lain . pian handak jurusan apa ?
4	kada ding , ada ai jurusan nang lain . pian handak jurusan apa ? <TAB> unda handak jurusan teknik sipil pang ada lah ?
5	unda handak jurusan teknik sipil pang ada lah ? <TAB> inggih kawa ai pian amun handak mandaftar jurusan nitu .

Dalam kondisi sebenarnya, hasil dari proses pairing terhadap dataset final yang dimaksudkan dapat disimak pada Gambar 3.5.

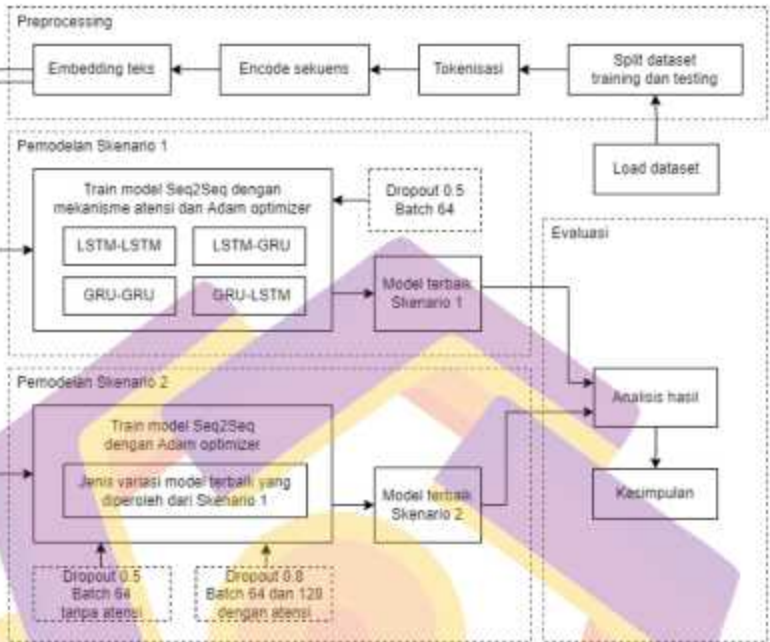
	user	admin
0	kawa kah ulun mandaftar wayah ini ?	kawa haja ding ai amun pian handak balalakas
1	kawa haja ding ai amun pian handak balalakas	apa haja garang jurusannya ? sebarataan hingga...
2	apa haja garang jurusannya ? sebarataan hingga...	kada ding, ada ai jurusan nang lain. pian hand...
3	kada ding, ada ai jurusan nang lain. pian hand...	ulun handak jurusan teknik sipil pang ada lah ?
4	ulun handak jurusan teknik sipil pang ada lah ?	inggih kawa ai pian amun handak mandaftar juru...

Gambar 3.5. Hasil Pairing Konteks dan Target

Dataset final tersebut disimpan dengan format *tab separated values* (TSV). File ini memisahkan elemen kalimat konteks dan target dengan karakter '\t' (tab) sebagai delimiter. Di dalam fungsi tersebut terjadi pembacaan data TSV sesuai alamat direktori *default* dari tempat penyimpanan dataset. Setelah itu, setiap baris pada dataset tersebut disimpan ke dalam *array* penampung. Proses ini berakhir ketika seluruh sample data berhasil dimuat dan array tersebut dibalikkan untuk dipakai pada langkah selanjutnya.

3.5. Alur Penelitian

Alur penelitian yang dilakukan dapat disimak pada Gambar 3.6 berikut ini:



Gambar 3.6. Alur Penelitian

Dataset yang telah melewati tahapan preprocessing berjumlah 6.100 baris pasang percakapan, untuk kemudian dipisah dengan komposisi 80:20 yang terdiri dari 4.879 data training dan 1.220 data testing. Selanjutnya, tahapan tokenisasi digunakan untuk membentuk token kata pada baris kalimat. Proses tokenisasi kata ini nantinya berguna untuk mendefinisikan kamus kata (vocabulary) beserta ukuran panjangnya. Keduanya dipakai untuk proses training dan generasi respon. Proses ini juga melakukan pemangkasan jumlah kata (*trimming*) jika suatu baris teks didapati memiliki lebih dari 15 kata dengan maksud agar sekuens pada kalimat tidak terlalu panjang. Setelah token kata terbentuk berikutnya dilakukan proses *encode sequences* yang digunakan untuk menghasilkan daftar urutan integer yang

menyandikan kata-kata sesuai kalimat yang diberikan ke dalam bentuk sekuensial dan memastikan bahwa semua sekuens dalam daftar memiliki panjang yang sama sesuai dengan panjang maksimal yang telah ditentukan. Terakhir, dilakukan proses embedding untuk diubah menjadi vektor padat yang berukuran tetap dengan menggunakan framework Keras.

Dalam penelitian ini, dua tahapan skenario dilakukan untuk mendapatkan model terbaik berdasarkan hasil evaluasi dari masing-masing skenario. Percobaan dilakukan dengan melatih model Seq2Seq ditambah mekanisme atensi dan gradient descent Adam menggunakan variasi mekanisme gerbang LSTM dan GRU pada lapisan encoder dan decoder. Hyperparameter yang diterapkan pada Skenario 1 di setiap model adalah dropout sebesar 0.5 (Srivastava, 2014) dan batch size sebesar 64 (Su, 2020). Untuk membuktikan dampak dari penggunaan mekanisme atensi terhadap kualitas model Seq2Seq, maka dilakukan kembali proses training dengan hyperparameter yang sama tetapi tanpa menggunakan mekanisme atensi.

Variasi mekanisme gerbang dari model terbaik yang didapatkan berdasarkan hasil evaluasi pada Skenario 1 selanjutnya akan digunakan sebagai acuan model pada Skenario 2 yang dilatih dengan hyperparameter berupa dropout sebesar 0.5 dan 0.8 (Srivastava, 2014) serta batch size sebesar 64 dan 128 (Masters, 2018).

Variasi gerbang LSTM dan GRU pada lapisan encoder dan decoder ini diterapkan sebagai *baseline* untuk pembangunan model berdasarkan skenario yang dilakukan sebagaimana terlihat pada Tabel 3.4.

Tabel 3.4. Skenario Pembangunan Model

Model Skenario 1	Parameter Skenario 1	Model Skenario 2	Parameter Skenario 2
LSTM-LSTM GRU-GRU LSTM-GRU GRU-LSTM	Dropout 0.5 Batch Size 64 GD Adam Attention	Model terbaik dari Skenario 1	Dropout 0.5/0.8 Batch Size 64/128 GD Adam Attention/No Attention

Model LSTM-LSTM mempunyai 6 lapisan pada neural network seperti yang ditunjukkan pada Tabel 3.5. Lapisan embedding digunakan untuk mengubah ukuran jumlah kata (*vocabulary*) menjadi vektor padat (*dense*) dengan ukuran tetap. Variasi gerbang model ini adalah 1 lapisan LSTM pada encoder dan 1 lapisan LSTM pada decoder serta menerapkan fungsi dropout untuk menghindari overfitting. Keduanya lapisan tersebut dihubungkan oleh lapisan Repeat Vector agar bisa mengulangi input dari encoder ke decoder. Lapisan Batch Normalization digunakan pada lapisan berikutnya agar distribusi dari layer input ke layer berikutnya tidak berubah selama periode training akibat pembaruan parameter dari setiap batch. Lapisan Attention dengan fungsi aktivasi *softmax*, *time distributed*, dan *dense* dipasang di akhir berdekatan dengan lapisan decoder (Bahdanau, 2014). Ukuran unit dari lapisan embedding adalah sebesar *n* vocabulary sedangkan lapisan yang lainnya masing-masing berjumlah 256 unit.

Tabel 3.5. Struktur Lapisan Model LSTM-LSTM

Lapisan	Ukuran Unit	Jumlah Lapisan
Embedding	1 x *n* vocabulary	1
Encoder LSTM	1 x 256	1
Repeat Vector	1 x 256	1
Decoder LSTM	1 x 256	1
Batch Normalization	1 x 256	1
Attention	1 x 256	1
Total Lapisan		6

Model GRU-GRU mempunyai 6 lapisan pada neural network seperti yang ditunjukkan pada Tabel 3.6. Lapisan embedding digunakan untuk mengubah ukuran jumlah kata (*vocabulary*) menjadi vektor padat (*dense*) dengan ukuran tetap. Variasi gerbang model ini adalah 1 lapisan GRU pada encoder dan 1 lapisan GRU pada decoder serta menerapkan fungsi dropout untuk menghindari overfitting. Keduanya lapisan tersebut dihubungkan oleh lapisan Repeat Vector agar bisa mengulangi input dari encoder ke decoder. Lapisan Batch Normalization digunakan pada lapisan berikutnya agar distribusi dari layer input ke layer berikutnya tidak berubah selama periode training akibat pembaruan parameter dari setiap batch. Lapisan Attention dengan fungsi aktivasi *softmax*, *time distributed*, dan *dense* dipasang di akhir berdekatan dengan lapisan decoder (Bahdanau, 2014). Ukuran unit dari lapisan embedding adalah sebesar *n* vocabulary sedangkan lapisan yang lainnya masing-masing berjumlah 256 unit.

Tabel 3.6. Struktur Lapisan Model GRU-GRU

Lapisan	Ukuran Unit	Jumlah Lapisan
Embedding	1 x *n* vocabulary	1
Encoder GRU	1 x 256	1
Repeat Vector	1 x 256	1
Decoder GRU	1 x 256	1
Batch Normalization	1 x 256	1
Attention	1 x 256	1
Total Lapisan		6

Model LSTM-GRU mempunyai 6 lapisan pada neural network seperti yang ditunjukkan pada Tabel 3.7. Lapisan embedding digunakan untuk mengubah ukuran jumlah kata (*vocabulary*) menjadi vektor padat (*dense*) dengan ukuran tetap. Variasi gerbang model ini adalah 1 lapisan LSTM pada encoder dan 1 lapisan GRU

pada decoder serta menerapkan fungsi dropout untuk menghindari overfitting. Keduanya lapisan tersebut dihubungkan oleh lapisan Repeat Vector agar bisa mengulangi input dari encoder ke decoder. Lapisan Batch Normalization digunakan pada lapisan berikutnya agar distribusi dari layer input ke layer berikutnya tidak berubah selama periode training akibat pembaruan parameter dari setiap batch. Lapisan Attention dengan fungsi aktivasi *softmax*, *time distributed*, dan *dense* dipasang di akhir berdekatan dengan lapisan decoder (Bahdanau, 2014). Ukuran unit dari lapisan embedding adalah sebesar *n* vocabulary sedangkan lapisan yang lainnya masing-masing berjumlah 256 unit.

Tabel 3.7. Struktur Lapisan Model LSTM-GRU

Lapisan	Ukuran Unit	Jumlah Lapisan
Embedding	1 x *n* vocabulary	1
Encoder LSTM	1 x 256	1
Repeat Vector	1 x 256	1
Decoder GRU	1 x 256	1
Batch Normalization	1 x 256	1
Attention	1 x 256	1
Total Lapisan		6

Model GRU-LSTM mempunyai 6 lapisan pada neural network seperti yang ditunjukkan pada Tabel 3.8. Lapisan embedding digunakan untuk mengubah ukuran jumlah kata (*vocabulary*) menjadi vektor padat (*dense*) dengan ukuran tetap. Variasi gerbang model ini adalah 1 lapisan GRU pada encoder dan 1 lapisan LSTM pada decoder serta menerapkan fungsi dropout untuk menghindari overfitting. Keduanya lapisan tersebut dihubungkan oleh lapisan Repeat Vector agar bisa mengulangi input dari encoder ke decoder. Lapisan Batch Normalization digunakan pada lapisan berikutnya agar distribusi dari layer input ke layer berikutnya tidak

berubah selama periode training akibat pembaruan parameter dari setiap batch. Lapisan Attention dengan fungsi aktivasi *softmax*, *time distributed*, dan *dense* dipasang di akhir berdekatan dengan lapisan decoder (Bahdanau, 2014). Ukuran unit dari lapisan embedding adalah sebesar *n* vocabulary sedangkan lapisan yang lainnya masing-masing berjumlah 256 unit.

Tabel 3.8. Struktur Lapisan Model GRU-LSTM

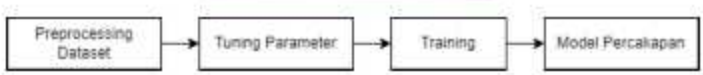
Lapisan	Ukuran Unit	Jumlah Lapisan
Embedding	1 x *n* vocabulary	1
Encoder GRU	1 x 256	1
Repeat Vector	1 x 256	1
Decoder LSTM	1 x 256	1
Batch Normalization	1 x 256	1
Attention	1 x 256	1
Total Lapisan		6

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Model Percakapan

Arsitektur Seq2Seq merupakan salah satu model deep learning yang digunakan dalam bidang NLP dan telah banyak digunakan untuk kebutuhan terjemahan mesin, rangkuman teks, dan percakapan. Pada setiap proses training terhadap dataset teks menggunakan arsitektur Seq2Seq dengan pengaturan parameter tertentu akan menghasilkan sebuah model yang dapat diuji dan dievaluasi performansinya. Model yang dihasilkan pada proses training tersebut berisi bobot (*weight*) berupa array multidimensi dalam format *Hierarchical Data Format* (HDF) dari hasil perhitungan vektor yang dilakukan selama proses training. Dalam konteks percakapan otomatis (chatbot), model ini disebut dengan model percakapan (chatbot model). Penelitian ini dimaksudkan untuk menguji beberapa model percakapan yang dihasilkan berdasarkan variasi tuning hyperparameter untuk mengetahui model manakah yang memiliki performansi yang lebih baik di antara model yang lain. Untuk memberikan gambaran tentang generasi model percakapan yang dimaksud dapat disimak pada Gambar 4.1.



Gambar 4.1. Proses Generasi Model Percakapan

4.2. Mekanisme Training

Alat penelitian yang digunakan untuk melakukan proses training adalah sebuah unit komputer dengan spesifikasi *hardware* berupa CPU Intel Core i5-4460 3.20 GHz, RAM DDR3 12 GB, dan GPU Vurrion GeForce GT-730 128 bits 4GB. Sedangkan spesifikasi untuk *software* terdiri dari environment Python 3.6.13 (Anaconda), TensorFlow 1.12.0, Keras 2.2.5, Numpy, Pandas, CSV, dan Matplotlib untuk keperluan visualisasi grafis.

Pada dasarnya, kecepatan proses training di kerangka kerja deep learning bergantung pada 3 (tiga) faktor, yaitu: CPU (Central Processing Unit), GPU (Graphic Processing Unit) dan TPU (Tensor Processing Unit). Fitur akselerasi training menggunakan GPU dan TPU dapat dilakukan menggunakan Google Colab versi Pro. Sedangkan untuk proses training menggunakan unit komputer secara mandiri dapat menggunakan CPU dan GPU. Dikarenakan GPU yang digunakan dalam penelitian ini belum mendukung teknologi akselerasi hardware, maka proses training dilakukan dengan mengandalkan kinerja CPU saja sehingga menyebabkan durasi training akan berjalan lebih lama.

Parameter yang digunakan untuk melakukan proses training berupa epoch, batch size, num_units, learning rate dan dropout. Jumlah epoch yang digunakan adalah 400. Hal ini didasarkan pada penelitian yang telah dilakukan oleh Faza (2018) yang mendapati bahwa upaya peningkatan kinerja klasifikasi dengan mengolah 4000 baris dataset pada training deep neural network menggunakan 1000 epoch menunjukkan nilai crossentropy error yang mulai stabil pada epoch ke-400. Nilai batch size yang digunakan adalah 64 dan 128 sesuai dengan penelitian yang

dilakukan oleh Masters (2018) tentang analisis perbandingan kinerja model berdasarkan ukuran batch yang mendapati bahwa secara umum nilai batch size yang efektif dalam melakukan training yang stabil yaitu pada kisaran 4 hingga 64 dan 128 untuk dataset yang lebih besar. Batch size 64 juga digunakan oleh Su (2020) untuk membuat model chatbot berbahasa Mandarin. Jumlah unit pada setiap sel neural network atau num_units yang digunakan adalah 256 (Martin, 2018) yang umum digunakan sesuai dengan dokumentasi Keras. Untuk nilai learning rate digunakan sesuai standar Keras yaitu 0.001 dimana nilai ini dianggap paling ideal untuk menangani berbagai macam kasus secara umum. Teknik regularisasi yang digunakan untuk mengurangi overfitting adalah dropout dengan nilai 0.5 dan 0.8 yang sering digunakan pada proses training data sekuensial. Srivastava (2014) menggunakan variasi nilai 0.5 pada hidden unit dan 0.8 pada input unit sebagai nilai yang mendekati optimal.

Training terhadap model yang diteliti diatur dalam 2 (dua) tahapan skenario. Pada skenario yang pertama, proses training dilakukan dengan memasang unit gerbang LSTM dan GRU secara bergantian pada lapisan encoder dan decoder dalam arsitektur Seq2Seq dengan ditambah mekanisme atensi (Bahdanau, 2014) dan gradient descent Adam. Setelah proses training selesai maka akan didapatkan 4 (empat) hasil variasi model, yaitu; LSTM-LSTM, GRU-GRU, LSTM-GRU, dan GRU-LSTM. Selanjutnya dilakukan perbandingan hasil evaluasi terhadap keempat model tersebut berdasarkan nilai metric dan loss. Jenis variasi unit gerbang yang diterapkan pada model terbaik akan dijadikan acuan untuk digunakan pada skenario

berikutnya. Parameter yang digunakan pada skenario ini antara lain: 400 epoch, 64 batch size, 256 num_unit, 0.001 learning rate dan 0.5 dropout.

Jenis variasi unit gerbang pada model terbaik yang diperoleh pada skenario 1 selanjutnya ditraining kembali pada skenario 2 dengan melakukan perubahan pada beberapa parameter. Nilai epoch yang digunakan masih sama dengan skenario sebelumnya yaitu 400. Demikian pula untuk num_unit dan learning rate juga masih sama, yaitu 256 dan 0.001. Variasi nilai batch size yang diterapkan adalah 64 dan 128 dengan menaikkan nilai dropout sebesar 0.8. Pada penelitian yang telah dilakukan oleh Smith (2018) didapatkan temuan bahwa penggunaan ukuran batch yang lebih besar secara signifikan dapat mengurangi jumlah pembaruan parameter yang diperlukan untuk melatih model dan berpotensi mengurangi waktu training model secara dramatis dibandingkan dengan menaikkan ukuran learning rate. Menaikkan nilai dropout diketahui dapat lebih mengeneralisasi data yang digunakan pada neural network sehingga dapat menurunkan tingkat kompleksitas pada model dan dapat mengurangi potensi terjadinya overfitting, dengan demikian dapat meningkatkan tingkat akurasi (Valentina, 2020).

Tahapan pembangunan model yang dilakukan pada setiap skenario terdiri dari pemuatan dataset, pembagian komposisi sample training dan testing, tokenisasi sample data, konversi token kata ke dalam bentuk sekuens, pendefinisian model neural network, lalu proses training dan testing. Proses generasi respon dimulai dari pemuatan sample data, tokenisasi sample data, pemuatan bobot jaringan, input kalimat, penyandian input secara sekuens, proses translasi, dan menampilkan respon sesuai dengan hasil generasi respon. Setelah training selesai dilakukan

selanjutnya dapat diketahui evaluasinya berupa nilai metrik beserta loss, durasi proses training, dan ukuran file yang berisi bobot hasil training.

4.3. Pemrosesan Dataset

Dataset final yang akan diproses terdiri dari 6.100 baris pasang percakapan dengan komposisi 80:20 yang terdiri dari 4.879 data training dan 1.220 data testing. Sample data yang telah dimuat diubah dimensinya (*reshape*) ke dalam bentuk 2 dimensi dengan pembagian: sample data 2 elemen untuk keperluan komposisi dan sample data 1 elemen untuk keperluan tokenisasi. *Pseudocode* yang digunakan untuk melakukan pemisahan sample data dapat disimak pada Gambar 4.2.

```
# load dataset
SET dataset TO load_clean_sample_data(param_folder, param_file)
SET dataset1 TO np.reshape(dataset, (-1, 2))
SET dataset2 TO dataset1.reshape(-1, 1)

# total dataset lines
SET total_sentences TO int(len(dataset1))

# proportion of the sentences will be used for the test set
SET test_proportion TO 0.2
SET train_test_threshold TO int((1-test_proportion) * total_sentences)

# split into train and test
SET train, test TO dataset1[:train_test_threshold], dataset1[train_test_threshold:]
```

Gambar 4.2. Pseudocode Pemisahan Sample Data

Cuplikan hasil pemisahan sample data untuk kedua jenis reshape tersebut dapat disimak pada Gambar 4.3.

```
dataset1

array([['kawa kah ulun mandaftar wayah ini ?',
        'kawa haja ding ai amun pian handak balalakas'],
       ['kawa haja ding ai amun pian handak balalakas',
        'apa haja garang jurusannya ? sebarataan hinggan jurusan kumputer haja kah ?'],
       ['apa haja garang jurusannya ? sebarataan hinggan jurusan kumputer haja kah ?',
        'kada ding , ada ai jurusan nang lain . pian handak jurusan apa ?'],
       ...,
       ['sudahkah tersedia wifi di stkom tanjung',
        'iya di stkom tanjung sudah terfasilitasi dengan wifi'],
       ['iya di stkom tanjung sudah terfasilitasi dengan wifi',
        'pendaftarannya waya apa dibuka ?'],
       ['pendaftarannya waya apa dibuka ?',
        'mun yang ulun tahu gelombang 1 dari tanggal 1 februari sampai tanggal 28 mei 2021']],
      dtype='<U131')

dataset2

array([['kawa kah ulun mandaftar wayah ini ?'],
       ['kawa haja ding ai amun pian handak balalakas'],
       ['kawa haja ding ai amun pian handak balalakas'],
       ...,
       ['pendaftarannya waya apa dibuka ?'],
       ['pendaftarannya waya apa dibuka ?'],
       ['mun yang ulun tahu gelombang 1 dari tanggal 1 februari sampai tanggal 28 mei 2021']],
      dtype='<U131')
```

Gambar 4.3. Cuplikan Pemisahan Sample Data

Tahapan berikutnya adalah melakukan proses tokenisasi yang berfungsi untuk membentuk token kata pada baris kalimat dimana nantinya berguna untuk mendefinisikan kamus kata (vocabulary) beserta ukuran panjangnya.

```
# prepare tokenizer

SET all_tokenizer TO create_tokenizer(dataset2[:, 0])
SET all_vocab_size TO len(all_tokenizer.word_index) + 1
SET all_length TO max_length(dataset2[:, 0])

OUTPUT(all_tokenizer)
OUTPUT("All vocabulary size: %d" % (all_vocab_size))
OUTPUT("All maximum question length: %d" % (all_length))
```

Gambar 4.4. Pseudocode Proses Tokenisasi

Berikutnya dilakukan tahapan *encode sequences* yang digunakan untuk menghasilkan daftar urutan integer yang menyandikan kata-kata sesuai kalimat yang diberikan ke dalam bentuk sekuensial dan memastikan bahwa semua sekuens dalam daftar memiliki panjang yang sama sesuai dengan panjang maksimal yang

telah ditentukan. Sedangkan *encode output* digunakan untuk mengubah vektor kelas (integer) dari sekuens target ke matriks kelas biner. Pseudocode untuk proses encode sequence ditunjukkan pada Gambar 4.5.

```
# prepare training data
SET trainX TO encode_sequences(all_tokenizer, all_length, train[:, 0])
SET trainY TO encode_sequences(all_tokenizer, all_length, train[:, 1])
SET trainY TO encode_output(trainY, all_vocab_size)

# prepare testing data
SET testX TO encode_sequences(all_tokenizer, all_length, test[:, 0])
SET testY TO encode_sequences(all_tokenizer, all_length, test[:, 1])
SET testY TO encode_output(testY, all_vocab_size)
```

Gambar 4.5. Pseudocode Encode Sequences

Cuplikan hasil proses encode sequences yang telah dilakukan pada sample dataset dapat disimak pada Gambar 4.6.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	629	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	43	6	211	1268	69	0	0	0	0	0	0	0	0	0
2	5	14	1	84	70	77	10	630	0	0	0	0	0	0	0
3	63	14	152	431	631	632	91	633	14	43	0	0	0	0	0
4	20	1	13	84	91	341	153	77	10	97	83	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	113	35	131	33	0	0	0	0	0	0	0	0	0	0	0
796	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0
797	1139	10	23	9	172	121	155	2	429	171	325	811	0	0	0
798	3	48	97	2	15	53	50	2	429	248	0	0	0	0	0
799	113	40	507	1140	185	14	39	25	123	812	618	469	614	0	0

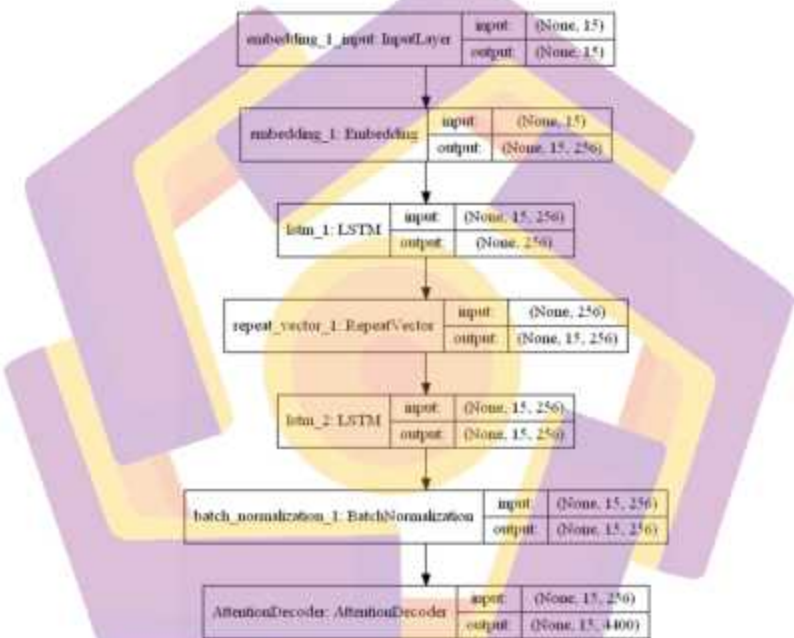
Gambar 4.6. Cuplikan Encode Sequences

Setelah itu dilakukan proses embedding yang berfungsi untuk mengubah setiap kata menjadi vektor dengan ukuran panjang tetap. Tahapan ini dilakukan saat model dikompilasi sesuai dengan kriteria dan pengaturan parameter yang telah didefinisikan sebelumnya. Embedding ditempatkan di input layer pada lapisan pertama dari struktur model yang akan ditraining.

4.4. Visualisasi dan Struktur Model

4.4.1. Model Layer LSTM-LSTM

Fungsi pada model gerbang LSTM-LSTM ini dibuat sesuai dengan struktur layer model jaringan yang telah didefinisikan pada Tabel 3.5. Visualisasi dari struktur layer model tersebut seperti yang tampak pada Gambar 4.7.



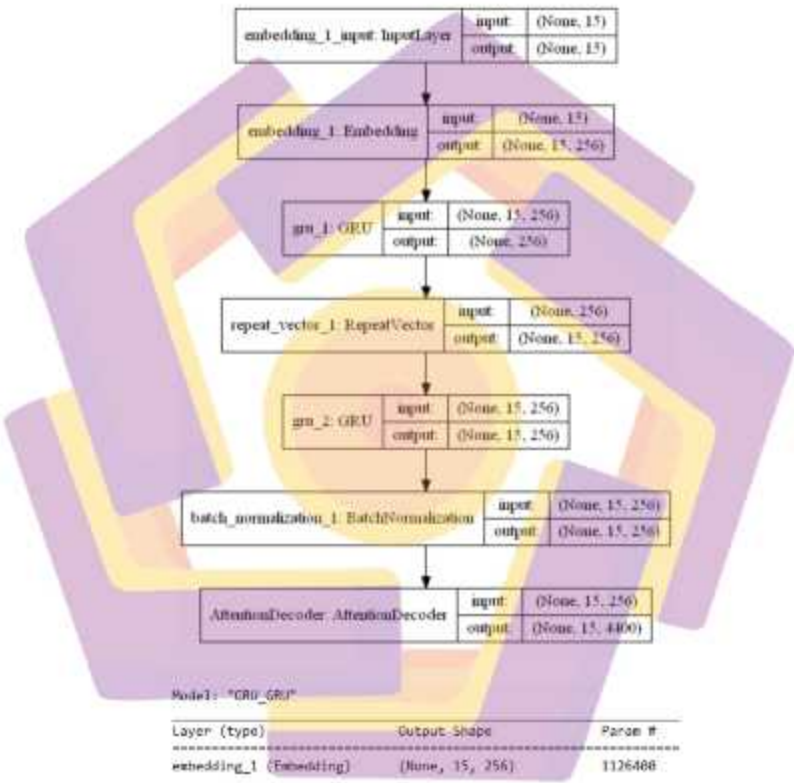
Model: "LSTM_LSTM"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 15, 256)	1126400
lstm_1 (LSTM)	(None, 256)	525312
repeat_vector_1 (RepeatVecto	(None, 15, 256)	0
lstm_2 (LSTM)	(None, 15, 256)	525312
batch_normalization_1 (Batch	(None, 15, 256)	1024
AttentionDecoder (AttentionD	(None, 15, 4400)	25587504

Gambar 4.7. Visualisasi dan Struktur Model LSTM-LSTM

4.4.2. Model Layer GRU-GRU

Fungsi pada model gerbang GRU-GRU ini dibuat sesuai dengan struktur layer model jaringan yang telah didefinisikan pada Tabel 3.6. Visualisasi dari struktur layer model tersebut seperti yang tampak pada Gambar 4.8.



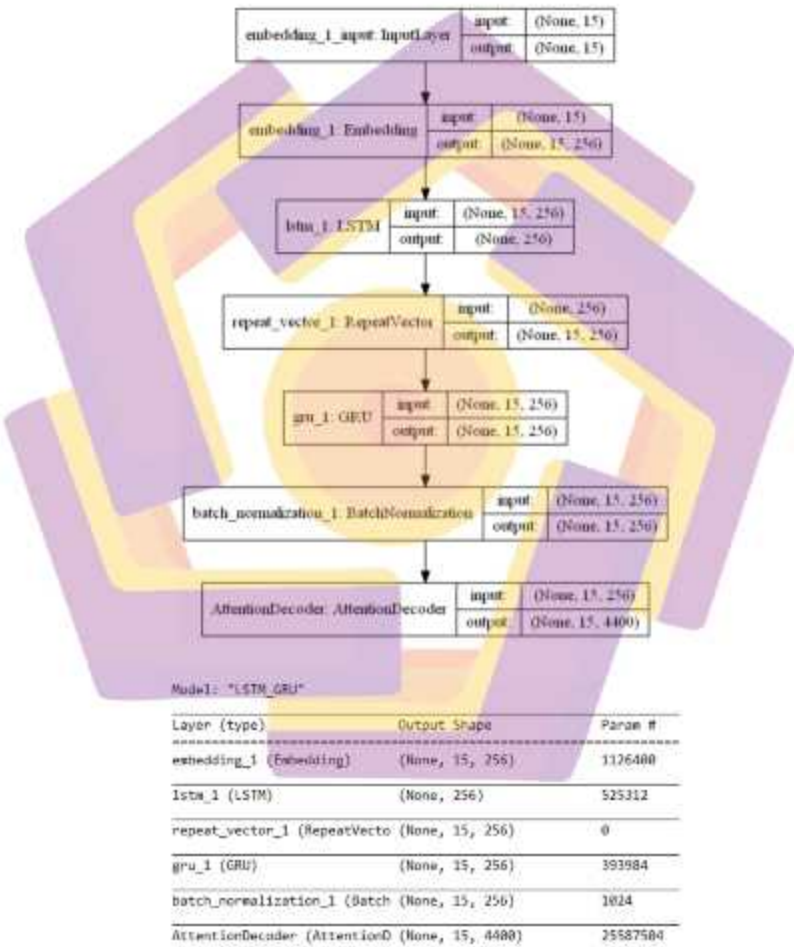
Model: "GRU_GRU"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 15, 256)	1126400
gru_1 (GRU)	(None, 256)	393984
repeat_vector_1 (RepeatVecto	(None, 15, 256)	0
gru_2 (GRU)	(None, 15, 256)	393984
batch_normalization_1 (Batch	(None, 15, 256)	1024
AttentionDecoder (AttentionD	(None, 15, 4400)	25587504

Gambar 4.8. Visualisasi dan Struktur Model GRU-GRU

4.4.3. Model Layer LSTM-GRU

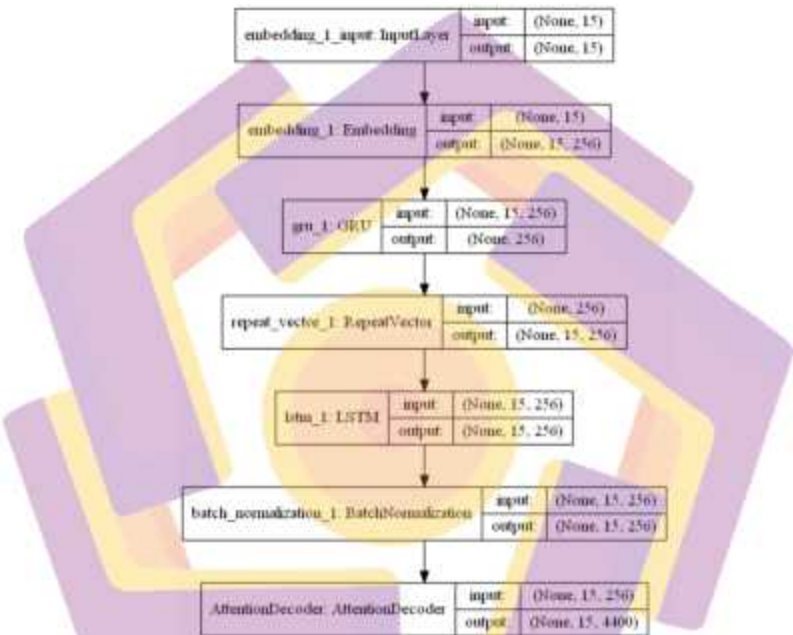
Fungsi pada model gerbang LSTM-GRU ini dibuat sesuai dengan struktur layer model jaringan yang telah didefinisikan pada Tabel 3.7. Visualisasi dari struktur layer model tersebut seperti yang tampak pada Gambar 4.9.



Gambar 4.9. Visualisasi dan Struktur Model LSTM-GRU

4.4.4. Model Layer GRU-LSTM

Fungsi pada model gerbang GRU-LSTM ini dibuat sesuai dengan struktur layer model jaringan yang telah didefinisikan pada Tabel 3.8. Visualisasi dari struktur layer model tersebut seperti yang tampak pada Gambar 4.10.



Model: "GRU_LSTM"

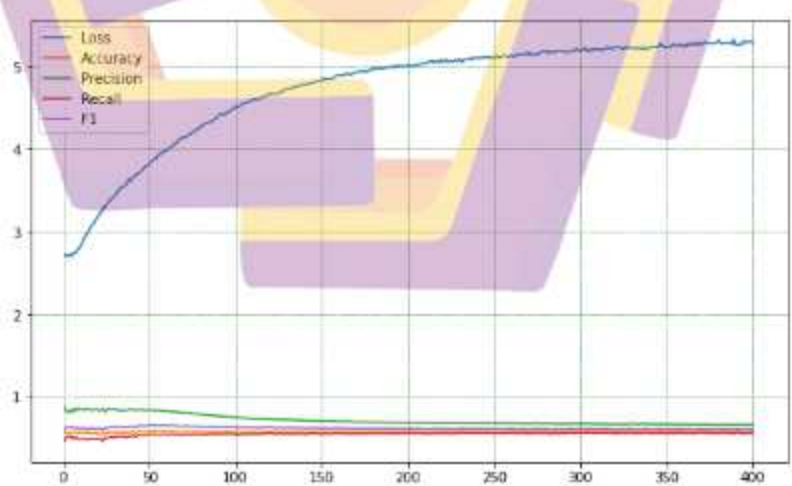
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 15, 256)	1126400
gru_1 (GRU)	(None, 256)	393984
repeat_vector_1 (RepeatVecto	(None, 15, 256)	0
lstm_1 (LSTM)	(None, 15, 256)	525312
batch_normalization_1 (Batch	(None, 15, 256)	1024
AttentionDecoder (AttentionD	(None, 15, 4400)	25587504

Gambar 4.10. Visualisasi dan Struktur Model GRU-LSTM

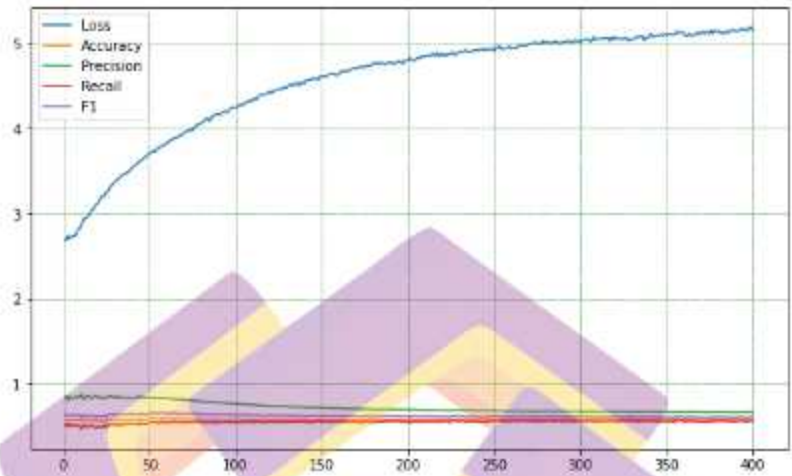
4.5. Penentuan Evaluasi Epoch

Nilai metrik yang ditentukan dalam training dan testing adalah accuracy, precision, recall, dan F1. Nilai metrik yang dihasilkan selama proses training biasanya cenderung naik (Rizki, 2021), artinya proses training berlangsung dengan baik. Loss menunjukkan nilai kesalahan dari hasil proses training dan testing berdasarkan sample data yang tersedia. Idealnya, nilai loss yang dihasilkan selama proses training cenderung menurun, menunjukkan bahwa proses training berlangsung dengan baik (Abdullahi, 2019).

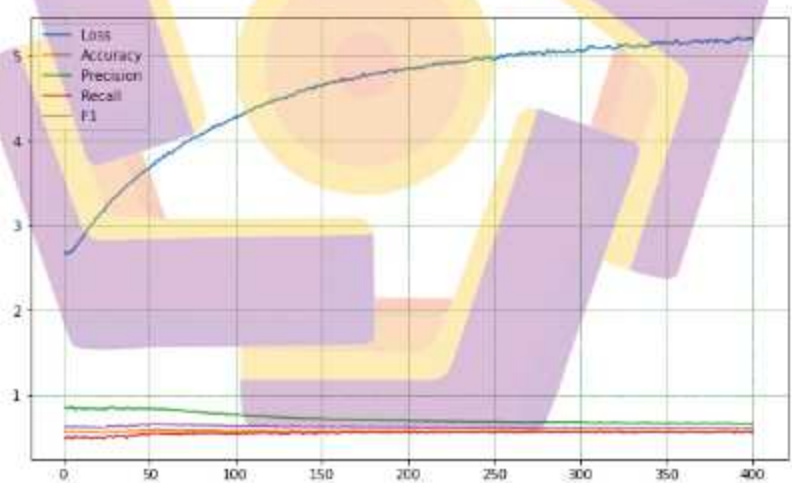
Evaluasi model dilakukan berdasarkan hasil pengamatan terhadap kejadian overfitting saat proses training dan testing yang mengindikasikan nilai metrik yang belum stabil. Visualisasi setiap model pada Skenario 1 yang ditampilkan berupa grafik menunjukkan bahwa nilai metrik terbaik yang dihasilkan mulai stabil pada epoch ke-50 yang kemudian menjadi linier hingga epoch ke-400.



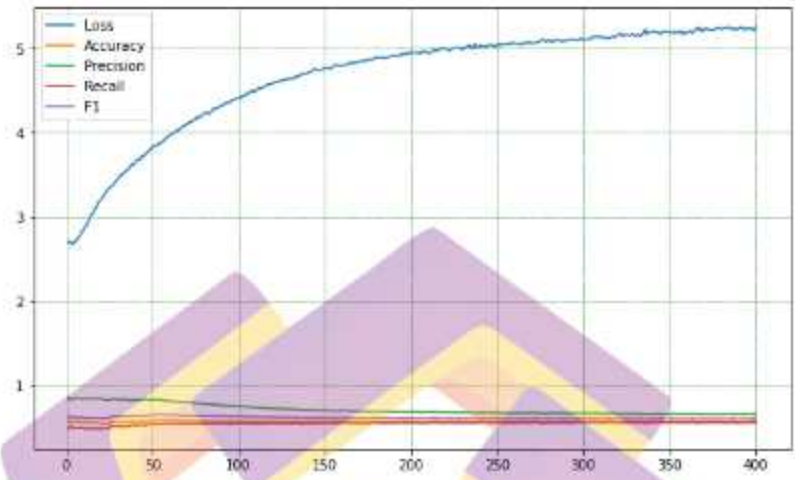
Gambar 4.11. Visualisasi Nilai Metrik Model LSTM-LSTM



Gambar 4.12. Visualisasi Nilai Metrik Model GRU-GRU



Gambar 4.13. Visualisasi Nilai Metrik Model LSTM-GRU



Gambar 4.14. Visualisasi Nilai Metrik Model GRU-LSTM

4.6. Training dan Testing Model Skenario 1

4.6.1. Nilai Metrik Training dan Testing

Pada kondisi nilai metrik dan loss yang tidak stabil, proses training tetap dijalankan sesuai dengan jumlah epoch yang telah ditentukan sebelumnya. Hal ini dilakukan untuk melihat perkembangan nilai tersebut agar menemukan pada epoch berapa nilai-nilai tersebut terlihat baik dan optimal sesuai. Pengamatan nilai metrik dan loss yang dihasilkan oleh masing-masing variasi unit gerbang dilakukan pada setiap 50 epoch berdasarkan hasil perhitungan metrik dan loss terbaik yang diperoleh selama proses training sesuai dengan penelitian Faza (2018).

Tabel 4.1. Nilai Metrik Training Model LSTM-LSTM

Epoch	Loss	Accuracy	Precision	Recall	F1
50	1.129850	0.714012	0.916680	0.606969	0.730070
100	0.437311	0.878090	0.946594	0.811614	0.873820
150	0.226218	0.936831	0.963308	0.909531	0.935620
200	0.143343	0.959923	0.974382	0.947612	0.960796
250	0.110047	0.968040	0.977879	0.960101	0.968898
300	0.085176	0.973465	0.981255	0.968163	0.974657
350	0.081536	0.973874	0.981118	0.969160	0.975095
400	0.088498	0.971401	0.979156	0.966714	0.972887

Tabel 4.2. Nilai Metrik Training Model GRU-GRU

Epoch	Loss	Accuracy	Precision	Recall	F1
50	1.203948	0.703300	0.907507	0.599713	0.722036
100	0.592225	0.838546	0.932563	0.759281	0.836961
150	0.321773	0.910774	0.952939	0.869577	0.909316
200	0.218559	0.939728	0.963525	0.915488	0.938870
250	0.165619	0.953720	0.970486	0.938403	0.954153
300	0.150536	0.956617	0.971103	0.944825	0.957771
350	0.134116	0.961194	0.974066	0.951192	0.962481
400	0.131322	0.961850	0.973930	0.951780	0.962718

Tabel 4.3. Nilai Metrik Training Model LSTM-GRU

Epoch	Loss	Accuracy	Precision	Recall	F1
50	1.230914	0.701988	0.909448	0.596037	0.719956
100	0.580119	0.844271	0.937041	0.764364	0.841832
150	0.293563	0.919259	0.958458	0.881752	0.918469
200	0.194015	0.946082	0.967082	0.926501	0.946335
250	0.153667	0.955660	0.970831	0.942734	0.956560
300	0.138185	0.959637	0.971591	0.949129	0.960212
350	0.121535	0.963995	0.974837	0.955674	0.965145
400	0.111250	0.966318	0.975598	0.959049	0.967242

Tabel 4.4. Nilai Metrik Training Model GRU-LSTM

Epoch	Loss	Accuracy	Precision	Recall	F1
50	1.108026	0.718481	0.911940	0.610931	0.731504
100	0.470734	0.869686	0.942058	0.799590	0.864927
150	0.240110	0.934085	0.962078	0.903792	0.931996
200	0.156751	0.956412	0.972015	0.941231	0.956359
250	0.130991	0.961987	0.973464	0.952026	0.962613
300	0.111361	0.966865	0.976199	0.959418	0.967728
350	0.099876	0.969953	0.978723	0.963585	0.971083
400	0.097906	0.969639	0.978807	0.963845	0.971259

Pengamatan nilai metrik dan loss yang dihasilkan oleh masing-masing variasi unit gerbang dilakukan pada setiap 50 epoch berdasarkan hasil perhitungan metrik dan loss terbaik yang diperoleh selama proses testing (Faza, 2018).

Tabel 4.5. Nilai Metrik Testing Model LSTM-LSTM

Epoch	Loss	Accuracy	Precision	Recall	F1
50	3.831696	0.588415	0.844142	0.544918	0.656656
100	4.519780	0.564918	0.750381	0.539399	0.624291
150	4.838371	0.568361	0.707442	0.550710	0.617409
200	5.018774	0.573005	0.684791	0.560273	0.615027
250	5.103915	0.571421	0.680039	0.560601	0.613412
300	5.191747	0.580000	0.675407	0.570219	0.617410
350	5.242780	0.573770	0.669708	0.564098	0.611443
400	5.287285	0.576284	0.663880	0.567814	0.611309

Tabel 4.6. Nilai Metrik Testing Model GRU-GRU

Epoch	Loss	Accuracy	Precision	Recall	F1
50	3.716670	0.585191	0.839331	0.532350	0.645347
100	4.242997	0.576120	0.760446	0.544317	0.631066
150	4.609390	0.560383	0.708529	0.536831	0.608398
200	4.807797	0.562240	0.694117	0.545628	0.609177
250	4.911642	0.570219	0.680525	0.554536	0.609773
300	5.018897	0.565792	0.668951	0.552623	0.604062
350	5.095069	0.566776	0.666406	0.554372	0.604153
400	5.157089	0.571038	0.662063	0.558579	0.605010

Tabel 4.7. Nilai Metrik Testing Model LSTM-GRU

Epoch	Loss	Accuracy	Precision	Recall	F1
50	3.662454	0.589563	0.845704	0.542350	0.655056
100	4.281678	0.572732	0.771046	0.539836	0.631143
150	4.655153	0.575738	0.716019	0.555191	0.623421
200	4.841725	0.571093	0.698280	0.554317	0.616345
250	4.956077	0.571311	0.688552	0.556831	0.614283
300	5.063366	0.572350	0.677699	0.559617	0.611855
350	5.142062	0.568907	0.669191	0.558251	0.607657
400	5.188345	0.571202	0.669532	0.559891	0.608760

Tabel 4.8. Nilai Metrik Testing Model GRU-LSTM

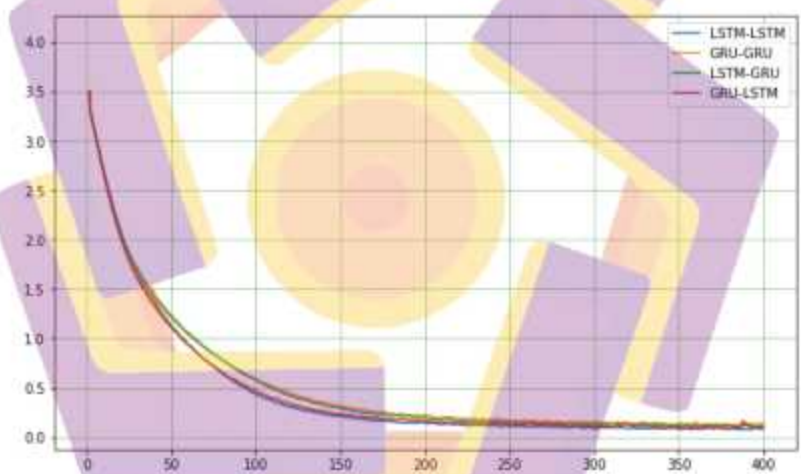
Epoch	Loss	Accuracy	Precision	Recall	F1
50	3.835192	0.581257	0.844276	0.532514	0.646826
100	4.414528	0.571530	0.753167	0.547541	0.630987
150	4.772520	0.572186	0.703843	0.555574	0.619241
200	4.960740	0.572623	0.685406	0.558798	0.614350
250	5.040048	0.574208	0.677025	0.560929	0.612441
300	5.099893	0.574536	0.673905	0.563060	0.612484
350	5.180916	0.572022	0.665744	0.561421	0.608222
400	5.254084	0.566339	0.658445	0.554426	0.601046

4.6.2. Grafik Nilai Loss Training dan Testing

Grafik nilai loss training memberikan visualisasi nilai kesalahan dari hasil proses training berdasarkan sample data yang tersedia. Nilai loss yang dihasilkan selama proses training cenderung menurun (Abdullahi, 2019), menunjukkan bahwa proses training berlangsung dengan baik.

Nilai loss semua model mengalami penurunan yang tajam dari epoch ke-1 sampai ke-100 lalu terjadi penurunan yang tidak banyak dari epoch ke-100 sampai ke-150. Di atas epoch ke-150, terjadi penurunan yang sangat sedikit dan cenderung melambat sampai epoch ke-400. Dari sini terlihat titik jenuh training karena penurunan nilai loss cenderung melambat bahkan stagnan. Model LSTM-LSTM

memiliki pola yang sama dengan model GRU-LSTM. Pada kisaran epoch ke-45 sampai ke-200, kedua model ini memiliki nilai loss yang lebih rendah dari kedua model yang lain. Hal ini tampaknya dikarenakan unit gerbang LSTM yang dipasang pada lapisan dekat dengan teknik batch normalization dapat membantu mengurangi nilai loss (Ioffe, 2015). Pada epoch ke-45 sampai ke-200 model GRU-GRU dan LSTM-GRU memiliki pola yang sama dimana memiliki nilai loss lebih besar dari kedua model yang lain. Visualisasi training model Skenario 1 dapat disimak pada Gambar 4.15.

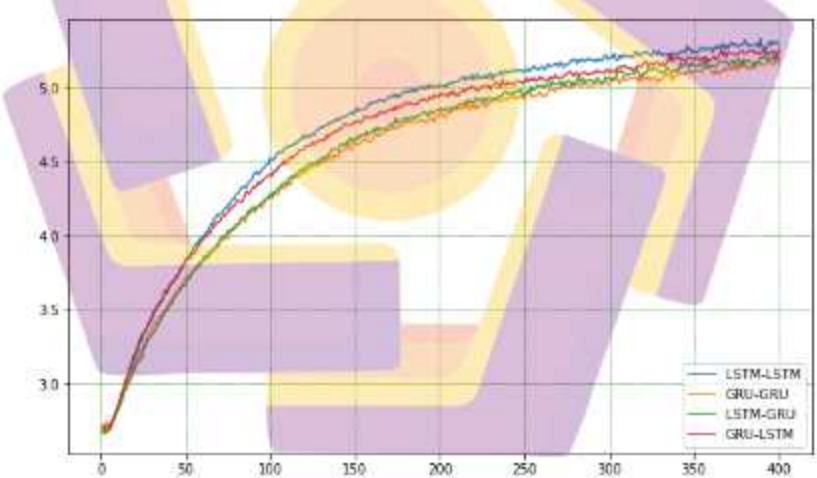


Gambar 4.15. Grafik Nilai Loss Training Skenario 1

Grafik nilai loss selanjutnya memberikan visualisasi nilai kesalahan dari hasil proses testing berdasarkan sample data yang tersedia. Nilai loss testing menunjukkan nilai kesalahan dari hasil proses pengujian berdasarkan sample data testing. Semakin rendah nilai loss menunjukkan tingkat kesalahan model yang

rendah (Abdullahi, 2019). Akan tetapi, hasil pengujian menunjukkan nilai loss yang semakin bertambah pada setiap epoch.

Pada proses testing, semua model menunjukkan kenaikan nilai loss yang sudah terjadi sejak epoch ke-1 sampai epoch ke-50. Setelah itu, nilai loss mengalami kenaikan secara terus menerus dan mengalami perlambatan dimulai dari epoch ke-200 sampai ke-400. Model GRU-GRU memiliki pola yang hampir sama dengan model LSTM-GRU sebagaimana halnya saat proses training. Model LSTM-LSTM dan GRU-LSTM juga memiliki pola yang hampir sama, tetapi model LSTM-LSTM memiliki nilai loss yang paling tinggi di antara model yang lain. Untuk visualisasi testing model Skenario 1 dapat disimak pada Gambar 4.16.



Gambar 4.16. Grafik Nilai Loss Testing Skenario 1

Evaluasi performansi model hanya bagus pada data training daripada data testing. Terdapat overfitting pada semua model meskipun tidak besar. Penyebab kondisi overfitting pada model akan dibahas pada analisis hasil evaluasi.

4.6.3. Durasi Proses Training dan Testing

Proses training mengambil data sebanyak 80% dari ukuran sample data. Setelah training selesai dilakukan, didapatkan informasi durasi yang dibutuhkan untuk melatih model LSTM-LSTM, GRU-GRU, LSTM-GRU, dan GRU-LSTM sesuai dengan paramater yang telah ditentukan. Durasi training model pada Skenario 1 ditunjukkan pada Tabel 4.9 yang diurutkan berdasarkan durasi terlama.

Tabel 4.9. Durasi Training Model Skenario 1

Urutan	Nama Model	Durasi
1	GRU-LSTM	24 jam 5 menit 00 detik
2	LSTM-GRU	24 jam 4 menit 53 detik
3	LSTM-LSTM	24 jam 4 menit 30 detik
4	GRU-GRU	24 jam 4 menit 12 detik

Dari tabel tersebut dapat diketahui bahwa durasi training yang paling cepat dimiliki oleh model GRU-GRU. GRU hanya memiliki 2 (dua) gerbang saja (reset dan update) sehingga dapat melakukan komputasi lebih cepat dari LSTM (Chung, 2014). Sedangkan durasi training yang paling lama dihasilkan oleh model GRU-LSTM. Unit gerbang LSTM yang dipasang pada lapisan decoder dengan penambahan mekanisme atensi tampaknya melakukan komputasi lebih lama saat memproses data sekuensial yang diterima dari unit gerbang GRU. Namun, semua model terlihat membutuhkan waktu training yang tidak jauh berbeda.

Proses testing mengambil data sebanyak 20% dari ukuran sample data. Durasi testing yang dibutuhkan dapat disimak pada Tabel 4.10 sesuai dengan urutan durasi terlama.

Tabel 4.10. Durasi Testing Model Skenario 1

Urutan	Nama Model	Durasi
1	LSTM-GRU	17.81 detik
2	GRU-GRU	16.98 detik
3	LSTM-LSTM	16.96 detik
4	GRU-LSTM	16.40 detik

Durasi testing yang paling cepat dimiliki oleh model GRU-LSTM. Sedangkan yang paling lama dihasilkan oleh model LSTM-GRU. Semua model memiliki selisih durasi testing yang sangat tipis, yaitu dalam hitungan milidetik saja. Sehingga tampaknya perbedaan durasi training tidak berdampak signifikan terhadap durasi testing.

4.6.4. Analisis Hasil Training dan Testing

Durasi training yang paling cepat dimiliki oleh model GRU-GRU, yaitu 24 jam, 4 menit 12 detik. Diketahui GRU hanya memiliki 2 (dua) gerbang, yaitu reset dan update sehingga dapat melakukan komputasi lebih cepat dari LSTM (Chung, 2014) yang memiliki 3 (tiga) gerbang, yaitu input, forget dan output. Nilai metrik yang paling tinggi dan nilai loss yang paling rendah pada saat training dihasilkan oleh model LSTM-LSTM pada epoch ke-350 dengan nilai loss 0.0815, accuracy 97.38%, precision 98.11, recall 96.91, dan F1 97.50. Hal yang menarik adalah untuk model dengan variasi unit gerbang yang berbeda, GRU lebih baik digunakan pada encoder dan LSTM lebih baik digunakan pada decoder. Hal ini diperlihatkan dari hasil nilai metrik dan loss training dari model GRU-LSTM memiliki nilai metrik dan loss yang lebih baik daripada model gerbang GRU-GRU dan LSTM-GRU.

Durasi testing yang paling cepat dimiliki oleh model GRU-LSTM yaitu selama 16.40 detik, terpaut beberapa milidetik dibandingkan dengan model LSTM-LSTM, GRU-GRU dan LSTM-GRU. Model LSTM-GRU memiliki waktu testing yang paling lama yaitu 17.81 detik. Hal ini tampaknya terkait dengan perhitungan nilai metrik dan loss yang dimiliki oleh model LSTM-GRU ternyata paling baik di antara model lain pada epoch ke-50 dengan nilai loss sebesar 3.6624, accuracy 58.95%, precision 84.57%, recall 54.23% dan F1 65.50%.

4.7. Training dan Testing Model Skenario 2

Struktur layer model dengan unit gerbang LSTM-GRU memiliki nilai metrik dan loss terbaik pada Skenario 1 sehingga akan dijadikan acuan untuk proses training pada Skenario 2. Tuning hyperparameter yang digunakan pada skenario ini adalah batch size dan dropout sesuai yang telah ditentukan pada rencana sebelumnya. Ada 3 (tiga) model yang ditraining pada skenario ini, antara lain: Model 1 dengan batch size 64 dan dropout 0.5, sama dengan yang digunakan pada Skenario 1 tetapi tanpa menggunakan mekanisme atensi, Model 2 dengan batch size 64 dan dropout 0.8, dan Model 3 dengan batch size 128 dan dropout 0.8.

4.7.1. Nilai Metrik Training dan Testing

Nilai metrik yang ditentukan dalam training adalah accuracy, precision, recall, dan F1. Nilai metrik yang dihasilkan selama proses training biasanya cenderung naik (Rizki, 2021), artinya proses training berlangsung dengan baik. Loss menunjukkan nilai kesalahan dari hasil proses training berdasarkan sample

data yang tersedia. Idealnya, nilai loss yang dihasilkan selama proses training cenderung menurun (Abdullahi, 2019), menunjukkan bahwa proses training berlangsung dengan baik.

Pada kondisi nilai metrik dan loss yang tidak stabil, proses training tetap dijalankan sesuai dengan jumlah epoch yang telah ditentukan sebelumnya. Hal ini dilakukan untuk melihat perkembangan nilai tersebut agar ditemukan pada epoch berapa nilai-nilai tersebut terlihat baik dan optimal. Pengamatan nilai metrik dan loss yang dihasilkan oleh masing-masing variasi unit gerbang dilakukan pada setiap 50 epoch berdasarkan hasil perhitungan metrik dan loss terbaik yang diperoleh selama proses training sesuai dengan penelitian Faza (2018).

Tabel 4.11. Nilai Metrik Training Model 1

Epoch	Loss	Accuracy	Precision	Recall	F1
50	2.036225	0.495983	0.816831	0.415215	0.559589
100	1.427469	0.572173	0.809351	0.480419	0.608424
150	1.183906	0.613152	0.800310	0.529227	0.640608
200	1.056712	0.634700	0.801563	0.559958	0.661967
250	0.977692	0.649061	0.798638	0.578076	0.672832
300	0.904208	0.664132	0.801324	0.595293	0.684938
350	0.869565	0.671442	0.800278	0.605664	0.691087
400	0.836982	0.676676	0.802845	0.614149	0.697378

Tabel 4.12. Nilai Metrik Training Model 2

Epoch	Loss	Accuracy	Precision	Recall	F1
50	2.357169	0.564405	0.889450	0.496058	0.636623
100	1.895846	0.589219	0.892121	0.507454	0.646747
150	1.619157	0.621166	0.886423	0.525326	0.659481
200	1.434550	0.649341	0.883706	0.550823	0.678397
250	1.308652	0.670329	0.878729	0.576771	0.696249
300	1.266970	0.678705	0.871429	0.587716	0.701842
350	1.264324	0.679156	0.864949	0.593879	0.704072
400	1.312619	0.672187	0.861068	0.588878	0.699246

Tabel 4.13. Nilai Metrik Training Model 3

Epoch	Loss	Accuracy	Precision	Recall	F1
50	2.518169	0.561017	0.890898	0.493653	0.635164
100	2.047218	0.583699	0.896195	0.502767	0.644043
150	1.751260	0.612407	0.897075	0.518057	0.656624
200	1.531717	0.640869	0.892098	0.541750	0.674064
250	1.339448	0.672911	0.891442	0.569762	0.695091
300	1.203237	0.696823	0.891472	0.597896	0.715688
350	1.064386	0.725886	0.898400	0.630553	0.740960
400	0.964971	0.747175	0.898357	0.659329	0.760446

Proses testing memakai sample yang sudah dialokasikan untuk pengujian pada saat proses training, yaitu sebesar 20% dari sample data. Nilai metrik yang digunakan dalam pengujian adalah accuracy, precision, dan F1. Loss menunjukkan nilai kesalahan dari hasil proses testing berdasarkan sample data testing. Semakin tinggi nilai metrik maka menunjukkan hasil model yang semakin bagus (Rizki, 2021), sementara semakin rendah nilai loss maka menunjukkan tingkat kesalahan model yang rendah (Abdullahi, 2019). Pengamatan nilai metrik dan loss yang dihasilkan oleh masing-masing variasi unit gerbang dilakukan pada setiap 50 epoch berdasarkan hasil perhitungan metrik dan loss terbaik yang diperoleh selama proses testing (Faza, 2018).

Tabel 4.14. Nilai Metrik Testing Model 1

Epoch	Loss	Accuracy	Precision	Recall	F1
50	4.455663	0.502896	0.772324	0.447923	0.566828
100	4.960566	0.489071	0.731619	0.444754	0.552973
150	5.233792	0.484317	0.702948	0.446940	0.546284
200	5.383816	0.485246	0.677101	0.453934	0.543339
250	5.498920	0.479454	0.665880	0.446448	0.534349
300	5.575956	0.481311	0.654336	0.449836	0.532984
350	5.641145	0.482404	0.644362	0.455956	0.533850
400	5.684404	0.485683	0.637888	0.459344	0.533930

Tabel 4.15. Nilai Metrik Testing Model 2

Epoch	Loss	Accuracy	Precision	Recall	F1
50	3.100362	0.611967	0.900119	0.539891	0.667517
100	3.470789	0.592404	0.891331	0.531694	0.658445
150	3.723116	0.584481	0.859683	0.535738	0.653629
200	3.911038	0.576503	0.828521	0.532732	0.642813
250	4.061956	0.571530	0.802182	0.537377	0.638898
300	4.148766	0.570328	0.776631	0.538525	0.632059
350	4.210496	0.559891	0.758937	0.525792	0.617195
400	4.198429	0.564809	0.765287	0.534153	0.625323

Tabel 4.16. Nilai Metrik Testing Model 3

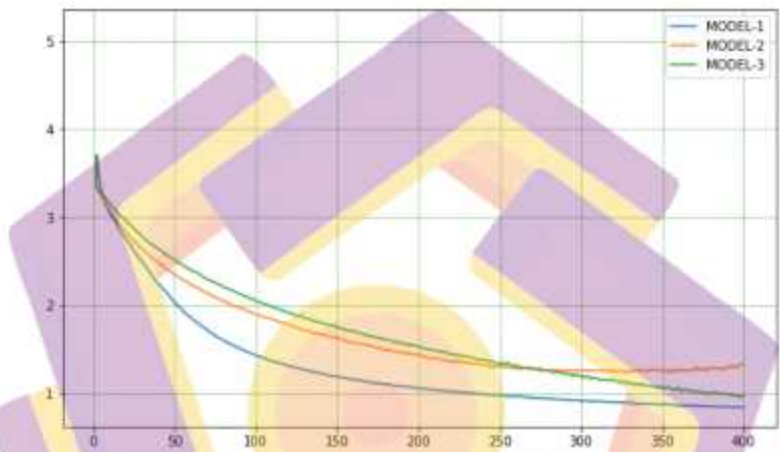
Epoch	Loss	Accuracy	Precision	Recall	F1
50	2.995556	0.615301	0.891182	0.546175	0.670405
100	3.302300	0.599235	0.890684	0.537213	0.663155
150	3.531073	0.595683	0.866672	0.548361	0.665786
200	3.718354	0.584590	0.848691	0.535246	0.650445
250	3.875514	0.571913	0.832850	0.525902	0.638645
300	3.982331	0.578907	0.808760	0.540383	0.643204
350	4.128984	0.566393	0.779502	0.528852	0.625780
400	4.236445	0.560437	0.760283	0.528525	0.619701

4.7.2. Grafik Nilai Loss Training dan Testing

Grafik nilai loss training memberikan visualisasi nilai kesalahan dari hasil proses training berdasarkan sample data yang tersedia. Nilai loss yang dihasilkan selama proses training cenderung menurun (Abdullahi, 2019), menunjukkan bahwa proses training berlangsung dengan baik.

Nilai loss pada Model 1 mengalami penurunan yang tajam dari epoch ke-1 sampai ke-200 lalu terjadi penurunan yang tidak banyak mulai epoch ke-200 sampai ke-250 dan terus mengalami penurunan hingga epoch ke-400. Model 2 dan Model 3 mengalami pola yang sama, dimana nilai loss mengalami penurunan yang tajam mulai epoch ke-1 hingga ke-250. Namun, nilai loss Model 3 lebih tinggi dari Model

2, tetapi mulai epoch ke-250 Model 3 mengalami penurunan nilai loss terus-menerus hingga epoch ke-400. Sedangkan Model 2 mengalami saturasi dan cenderung mengalami kenaikan setelah epoch ke-350 hingga 400. Visualisasi training model Skenario 2 dapat disimak pada Gambar 4.17.

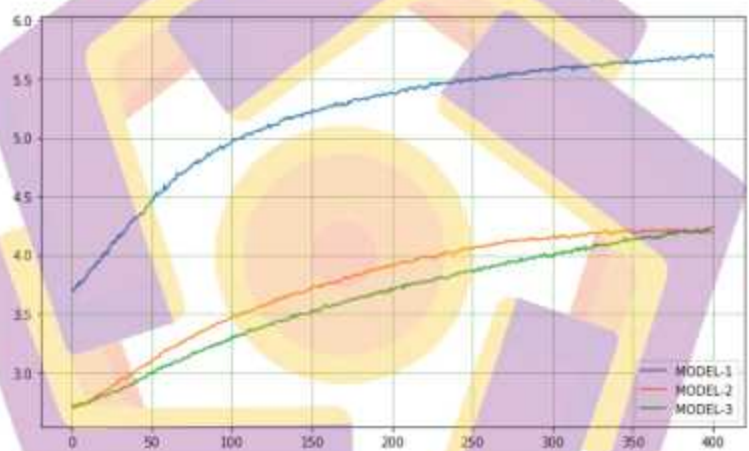


Gambar 4.17. Grafik Nilai Loss Training Skenario 2

Grafik nilai loss selanjutnya memberikan visualisasi nilai kesalahan dari hasil proses testing berdasarkan sample data yang tersedia. Nilai loss testing menunjukkan nilai kesalahan dari hasil proses pengujian berdasarkan sample data testing. Semakin rendah nilai loss maka menunjukkan tingkat kesalahan model yang rendah (Abdullahi, 2019). Akan tetapi, hasil testing menunjukkan nilai loss yang semakin bertambah setiap epoch.

Pada proses testing, semua model menunjukkan kenaikan nilai loss yang sudah terjadi sejak epoch ke-1 sampai epoch ke-400. Model 1 dan Model 2 mengalami tren kenaikan nilai loss dengan pola yang sama, tetapi dengan

perbedaan nilai loss yang signifikan. Nilai loss yang dimiliki oleh Model 1 berada jauh di atas Model 2 dan Model 3 dikarenakan adanya perbedaan arsitektur di antara ketiganya, dimana Model 1 dilatih tanpa menggunakan mekanis atensi. Model 3 yang memiliki nilai batch size lebih tinggi dari Model 2 memiliki nilai loss yang paling rendah di antara kedua model yang lain, mulai pada kisaran epoch ke-20 hingga kisaran epoch ke-380. Untuk visualisasi testing model Skenario 2 dapat disimak pada Gambar 4.18.



Gambar 4.18. Grafik Nilai Loss Testing Skenario 2

Evaluasi performansi model hanya bagus pada data training daripada data testing. Terdapat overfitting pada semua model meskipun tidak besar. Penyebab kondisi overfitting pada model akan dibahas pada analisis hasil evaluasi.

4.7.3. Durasi Proses Training dan Testing

Proses training mengambil data sebanyak 80% dari ukuran sample data. Setelah training selesai dilakukan, didapatkan informasi durasi untuk melatih ketiga

model sesuai dengan paramater yang telah ditentukan. Durasi training model pada Skenario 1 ditunjukkan pada Tabel 4.17 yang diurutkan berdasarkan durasi terlama

Tabel 4.17. Durasi Training Model Skenario 2

Urutan	Nama Model	Durasi
1	MODEL 2	24 jam 04 menit 06 detik
2	MODEL 3	22 jam 50 menit 34 detik
3	MODEL 1	03 jam 47 menit 08 detik

Dari hasil tersebut dapat diketahui bahwa durasi training tercepat dimiliki oleh Model 1 yang memiliki perbedaan sangat dignifikan dari model lainnya. Hal ini dikarenakan model ini dilatih tanpa menggunakan mekanisme atensi sehingga proses komputasi yang dilakukan tidak kompleks dibandingkan dengan kedua model yang lain. Durasi training Model 2 memiliki selisih sekitar 74 menit lebih lama dari Model 3 yang memiliki nilai batch size lebih besar.

Proses testing mengambil data sebanyak 20% dari ukuran sample data. Durasi testing yang dibutuhkan dapat disimak pada Tabel 4.18 sesuai dengan urutan durasi terlama.

Tabel 4.18. Durasi Testing Model Skenario 2

Urutan	Nama Model	Durasi
1	MODEL 2	17.11 detik
2	MODEL 3	15.50 detik
3	MODEL 1	03.02 detik

Durasi testing tercepat dimiliki oleh Model 1 yang dilatih tanpa menggunakan mekanime atensi. Sedangkan Model 2 memiliki selisih durasi hanya 2 detik lebih lama dari Model 3. Sehingga diketahui bahwa model yang dilatih tanpa mekanisme atensi memiliki perbedaan durasi training dan testing yang sangat

signifikan lebih cepat jika dibandingkan dengan model yang menerapkan mekanisme atensi.

4.7.4. Analisis Hasil Training dan Testing

Durasi training yang paling cepat dihasilkan oleh Model 1 (LSTM-GRU), yaitu 3 jam 47 menit 8 detik. Model ini ditraining dengan batch size 64 dan dropout 0.5 tanpa menggunakan mekanisme atensi. Mekanisme atensi melakukan perhitungan bobot setiap kata yang diinput untuk menentukan banyaknya atensi yang ada pada input kata tersebut (Lopyrev, 2015). Proses training tanpa mekanisme atensi akan melewatkan tahapan komputasi kompleks tersebut sehingga durasi training dan testing dapat dilakukan dengan sangat cepat. Nilai metrik yang paling tinggi dan nilai loss yang paling rendah pada saat training dihasilkan oleh Model 3 (LSTM-GRU) pada epoch ke-400 dengan nilai loss 0.9649, accuracy 74.71%, precision 89.83%, recall 65.93%, dan F1 76.04%. Model 3 ditraining menggunakan batch size 128. Waktu yang dibutuhkan untuk melakukan training pada model ini adalah 22 jam 50 menit 34 detik, dimana terdapat selisih yang cukup signifikan jika dibandingkan dengan Model 2 yang ditraining menggunakan batch size 64 membutuhkan waktu sekitar 74 menit lebih lama. Sehingga dari hal ini diketahui bahwa penggunaan batch yang lebih besar membantu mempersingkat waktu training dikarenakan dalam satu iterasi dapat memproses lebih banyak unit neural netwok.

Durasi testing yang paling cepat masih dihasilkan oleh Model 1 (LSTM-GRU), yaitu 3.02 detik. Hal ini dikarenakan struktur model ini tidak dipasang

lapisan mekanisme atensi dengan komputasi bobot yang kompleks, sehingga memiliki durasi training dan testing yang lebih cepat secara signifikan dari model yang lain. Sama halnya pada proses training, Model 3 (LSTM-GRU) memiliki nilai loss yang paling rendah di antara model lain, yaitu 2.9955 dengan nilai metrik yang didapatkan berupa accuracy 61.53%, precision 89.11%, recall 54.61%, F1 67.04%. Model ini dilatih menggunakan batch size 128 dan dropout 0.8. Hal ini sejalan dengan penelitian (Smith, 2018) dimana menaikkan nilai batch size dapat meningkatkan nilai akurasi dari model. Demikian pula penggunaan variasi dropout 0.5 dan 0.8 merupakan nilai yang mendekati optimal (Srivastava, 2014).

4.8. Evaluasi Pemodelan

Berdasarkan perhitungan nilai metrik dan loss testing, didapati bahwa epoch ke-50 untuk semua model memiliki nilai metrik dan loss terbaik, kecuali nilai recall. Selisih hasil nilai metrik antar epoch terlihat mengalami kecenderungan penurunan dimana nilai metrik yang didapatkan lebih tinggi pada epoch ke-50 dibandingkan epoch selanjutnya. Hal ini juga terlihat dari nilai loss untuk setiap model, dimana angkanya terus meningkat pada setiap epoch. Dari hal tersebut dapat diketahui bahwa jumlah iterasi maksimal berada pada epoch ke-50 dan mulai memperlihatkan titik jenuh ketika jumlah epoch tersebut makin bertambah.

Nilai metrik yang dihasilkan setelah epoch ke-50 mulai terlihat beberapa overfitting yang terjadi di semua hasil testing meskipun tidak besar. Semakin bertambahnya epoch dapat membuat nilai recall menjadi tidak stabil dan dapat menurunkan nilai metrik accuracy, precision, dan F1 serta menyebabkan nilai loss

menjadi semakin tinggi. Dengan demikian, jumlah epoch yang besar justru dapat memicu terjadinya overfitting karena nilai metrik yang didapatkan malah menjadi tidak baik khususnya pada evaluasi data di luar dari dataset training. Fungsi dropout dengan pengaturan nilai yang sesuai dapat membantu mengurangi overfitting (Rizki, 2021). Penggunaan nilai dropout 0.8 pada Skenario 2 diketahui dapat mengurangi overfitting secara signifikan mulai epoch ke-50 dan seterusnya jika dibandingkan dengan training model pada Skenario 1 yang menggunakan nilai dropout lebih kecil yaitu 0.5. Visualisasi ini tampak pada Gambar 4.13.

Beberapa penyebab yang membuat nilai metrik dan loss pada proses training lebih baik dibandingkan dengan testing. Salah satunya adalah kualitas dari dataset yang perlu ditinjau lebih lanjut sebab banyak ditemukan kesalahan pengetikan, kesalahan ejaan, singkatan, serta bentuk kata dalam beragam dialek yang pada akhirnya menyebabkan duplikasi pada kamus kata (*vocabulary*). Susunan kata dalam kalimat percakapan juga terlihat bebas, tidak struktural atau tidak mengikuti tatanan subjek, predikat, objek, dan keterangan (SPOK). Selain itu, terdapat ketidaklogisan pemetaan antara konteks dengan target, sehingga percakapan yang terjadi tidak sesuai dengan topiknya. Oleh karena itu, perlu penelitian lebih lanjut untuk penggunaan dataset ini.

4.9. Hasil Percakapan

Pengujian model percakapan dalam bentuk generasi respon dilakukan menggunakan antarmuka pada Jupyter Notebook. Hasil generasi respon ditampilkan dalam bentuk teks menggunakan bahasa pemrograman Python.

Tampilan antarmuka program beserta respon yang dihasilkan oleh model seperti yang ditunjukkan pada Gambar 4.19.

Gambar 4.19. Contoh Hasil Percakapan

BAB V

PENUTUP

5.1. Kesimpulan

Sesuai dengan hasil yang telah dicapai dalam penelitian ini yang dilakukan dengan menerapkan variasi unit gerbang LSTM dan GRU pada arsitektur Seq2Seq dengan mekanisme atensi, maka kesimpulan yang telah didapatkan antara lain:

1. Pada Skenario 1, kinerja terbaik diperoleh model LSTM-GRU yang ditraining dengan parameter batch size 64 dan dropout 0.5, mendapatkan nilai loss sebesar 3.6624, accuracy 58.95%, precision 84.57%, recall 54.23% dan F1 65.50%. Selanjutnya, unit gerbang LSTM-GRU dijadikan acuan untuk proses training pada Skenario 2 dengan parameter batch size 64 dan 128 serta dropout 0.5 (tanpa atensi) dan 0.8 (dengan atensi). Kinerja model terbaik diperoleh pada parameter batch size 128 dan dropout 0.8, dengan nilai loss sebesar 2.9955, accuracy 61.53%, precision 89.11%, recall 54.61%, F1 67.04%. Semua model pada kedua skenario tersebut ditraining menggunakan gradient descent Adam dengan learning rate 0.001 sesuai nilai default pada framework Keras.
2. Penggunaan mekanisme atensi yang diterapkan pada arsitektur Seq2Seq dengan pengaturan nilai dropout dan batch size yang sesuai dapat menurunkan overfitting dan meningkatkan kinerja model.

5.2. Saran

Beberapa saran yang perlu untuk dipertimbangkan dalam pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Perlu penambahan dataset percakapan Bahasa Banjar dengan jumlah yang lebih besar untuk meningkatkan kinerja model yang dilatih menggunakan pendekatan deep learning.
2. Normalisasi kata pada dataset ini masih terbatas. Perlu menjadi perhatian khusus pada penelitian berikutnya untuk meningkatkan kualitas dataset dengan lebih fokus kepada permasalahan ini.
3. Penggunaan variasi metode embedding atau vektorisasi kata untuk memperkaya vocabulary, seperti GloVe, Word2Vec, FastText dan ELMo.
4. Penggunaan unit gerbang atau arsitektur yang lebih bervariasi, seperti BiLSTM, Bi-GRU dan BERT sebagai upaya optimasi model.
5. Percobaan lebih lanjut terhadap penentuan jumlah epoch secara dinamis menggunakan teknik *early stopping* dan tuning hyperparameter yang lebih bervariasi.

DAFTAR PUSTAKA

BUKU

Chollet, F. (2018). Deep Learning with Python. New York: Manning Publications Co.

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. An MIT Press Book.

Goyal, P., Pandey, S., Jain, K. (2018). Deep Learning for Natural Language Processing: Creating Neural Networks with Python. India: Apress Media LLC.

Hapip, A. D. (1977). Kamus Banjar-Indonesia. Pusat Pembinaan dan Pengembangan Bahasa. Departemen Pendidikan dan Kebudayaan.

Khan, R., Das, A. (2018). Build Better Chatbots: A Complete Guide to Getting Started with Chatbots. India: Apress Media LLC.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. doi:10.1007/978-1-4614-6849-3

Moolayil, J., (2019). Learn Keras for Deep Neural Networks – A Fast-Track Approach to Modern Deep learning with Python, Apress, Canada.

Na'im, A., Syaputra, H. (2012). "Kewarganegaraan, Suku Bangsa, Agama, dan Bahasa Sehari-Hari Penduduk Indonesia", Sensus Penduduk 2010, Badan Pusat Statistik, Jakarta.

Raj, S. (2019). Building Chatbots with Python: Using Natural Language Processing and Machine Learning. India: Apress Media LLC.

Sugono, D. (2017). Bahasa dan Peta Bahasa di Indonesia. Kementrian Pendidikan dan Kebudayaan.

JURNAL ILMIAH

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A. (2016). TensorFlow: A system for large-scale machine learning. ArXiv.

Bahdanau, D., Cho, K., Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv.

Bay, A., Sengupta, B. (2017). StackSeq2Seq: Dual Encoder Seq2Seq Recurrent Networks. ArXiv.

Breier, J., Hou, X., Jap, D., Ma, L., Bhasin, S., Liu, Y. (2018). DeepLaser: Practical Fault Attack on Deep Neural Networks. arXiv.

Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. ArXiv.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv.

Humaidi, A., HB, A. (2018). Afiks Pembentuk Nomina Dalam Bahasa Banjar. Stilistika: Jurnal Bahasa, Sastra, dan Pengajarannya, ISSN 2527-4104, Vol. 3 No. 1, Issue 1 April 2018.

Humaidi, A., Kamariah, Harpriyanti, H. (2017). Infleksi dalam Bahasa Banjar. Jurnal Bahasa, Sastra, dan Pengajarannya.

Ienco, D., Gaetano, R., Dupaquier, C., Maurel, P. (2017). Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks. ArXiv.

Ioffe, S., Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv.

Karpathy, A., Johnson, J., Fei-fei, L. (2016). Visualizing and Understanding Recurrent Networks. ArXiv.

Khurana, D., Koli, A., Khatter, K., Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges. ArXiv.

Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. ArXiv.

Lopyrev, K. (2015). Generating News Headlines with Recurrent Neural Networks. ArXiv.

Luong, M.-T., Pham, H., Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. ArXiv.

Masters, D., Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. ArXiv.

Palasundram, K., Sharef, NM., Nasharuddin, NA., Kasmiran, KA., Azman, A. (2019). Sequence to Sequence Model Performance for Education Chatbot. International Journal of Emerging Technologies in Learning (iJET).

Prassanna, J., Nawas, K.K., Jackson C.J., Prabakaran, R., Ramanath, S. (2020). Towards Building A Neural Conversation Chatbot Through Seq2Seq Model. International Journal of Scientific & Technology Reaearch Volume 9.

Raj, VS. (2021). Seq2Seq learning Chatbot with Attention Mechanism. University of Calgary.

Rizki, A., Sibaroni, Y. (2021). Analisis Sentimen Untuk Pengukuran Tingkat Depresi Pengguna Twitter Menggunakan Deep Learning. eProceedings of Engineering. Vol 8, No 5.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S. (2017). Recent Advances in Recurrent Neural Networks, ArXiv.

Sharma, O. (2019). A New Activation Function for Deep Neural Network. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 84-86, doi: 10.1109/COMITCon.2019.8862253.

Smith, S.L., Kindermans, P., Ying, C., Le, Q.V. (2018). Don't Decay the Learning Rate, Increase the Batch Size. ArXiv.

Sojasingarayar, A. (2020). Seq2Seq AI Chatbot with Attention Mechanism. Artificial Intelligence, IA School/University.

Sonawane, S., Shanmughasundaram, R. (2019). ChatBot for College Website. International Journal of Innovative Technology and Exploring Engineering (IJITEE). Vol. 8 Issue-10, August 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (January 2014), 1929–1958.

Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. ArXiv.

Valentina, R., Rostianingsih, S., Tjondrowiguno, A.N. (2020). Pengenalan Gambar Botol Plastik dan Kaleng Minuman Menggunakan Metode Convolutional Neural Network. Jurnal Infra. Vol 8, No 1.

Vinyals, O., Le, Q. (2015). A Neural Conversational Model. ArXiv.

PROSIDING

Abdullahi, SS., Yiming, S., Abdullahi, A., Aliyu, U. (2019). Open Domain Chatbot Based on Attentive End-to-End Seq2Seq Mechanism. International Conference on Algorithms, Computing and Artificial Intelligence (ACAI).

Aalipour, G., Kumar, P., Aditham, S., Nguyen, T., Sood, A. (2018). Applications of Sequence-to-Sequence Models for Technical Support Automation. IEEE International Conference on Big Data (Big Data).

Ali, A., Amin, M.Z. (2019). Conversational AI Chatbot Based on Encoder-Decoder Architectures with Attention Mechanism. Artificial Intelligence Festival 2.0. NED University of Enginering and Technology.

Chandra, Y.W., Suyanto. (2019). Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model. Procedia Computer Science.

Martin C.P., Torresen J. (2018). RoboJam: A Musical Mixture Density Network for Collaborative Touchscreen Interaction. In: Liapis A., Romero Cardalda J., Ekárt A. (eds) Computational Intelligence in Music, Sound, Art and Design. EvoMUSART 2018. Lecture Notes in Computer Science, vol 10783. Springer, Cham. https://doi.org/10.1007/978-3-319-77583-8_11

Nguyen, T., Shcherbakov, M. (2018). A Neural Network based Vietnamese Chatbot. International Conference on System Modeling & Advancement in Research, College of Computing Sciences & Information Technology.

Su, H., Shen, X., Xiao, Z., Zhang, Z., Chang, E., Zhang, C., Niu, C., Zhou, J. (2020). MovieChats: Chat like Humans in a Closed Domain. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2020.emnlp-main.535.

LAPORAN PENELITIAN

Faza, S. (2018). Peningkatan Kinerja dalam Pengklasifikasian Menggunakan Deep Learning, Tesis. Universitas Sumatera Utara.

LINK

Chhabra, J.S. (2017). Understanding LSTM in Tensorflow (MNIST dataset). https://jasdeep06.github.io/posts/Understanding-LSTM-in-Tensorflow-MNIST

Colah's Blog. (2015). Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Wikimedia Commons. (2017). Gated Recurrent Unit. https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit.svg

Wikimedia Commons. (2017). Long Short-Term Memory. https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg

Zhu, S. (2020). Francois Chollet Working with RNNs. https://keras.io/guides/working_with_rnns/

LAMPIRAN 1. PSEUDOCODE FOR FUNCTION

Load Paired Dataset

```
DEFINE FUNCTION load_clean_sample_data(folder, file):
    SET lines_filepath TO (folder, file)
    with open(lines_filepath) as read:
        SET reader TO read(delimiter="tab")
        SET dataset TO []
        FOR row IN reader:
            dataset.append(row)
    read.close()
    RETURN dataset
```

Fit A Tokenizer

```
DEFINE FUNCTION create_tokenizer(lines):
    SET tokenizer TO Tokenizer()
    tokenizer.fit_on_texts(lines)
    RETURN tokenizer
```

Max Sentence Length

```
DEFINE FUNCTION max_length(lines):
    RETURN max(len(line.split()) FOR line IN lines)
```

Encode and Pad Sequences

```
DEFINE FUNCTION encode_sequences(tokenizer, length, lines):
    # encode sequences
    SET X TO tokenizer.texts_to_sequences(lines)
    # pad sequences with 0 values
    SET X TO pad_sequences(X, maxlen=length, padding="post")
    RETURN X
```

One Hot Encoding Target Sequence

```
DEFINE FUNCTION encode_output(sequences, vocab_size):
    SET ylist TO list()
    FOR sequence IN sequences:
        SET encoded TO to_categorical(sequence,
            num_classes=vocab_size)
        ylist.append(encoded)
    SET y TO array(ylist)
    SET y TO y.reshape(sequences.shape[0], sequences.shape[1],
        vocab_size)
    RETURN y
```