

Penerapan Speech to Text Berbasis Deep Learning pada Aplikasi Paperless Hospital Menggunakan Metode Gate Recurrent Unit (GRU)

Salahuddin¹, M.Khadafi², Zaharatul Dahlia³

^{1,2,3} Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹*salahuddintik@pnl.ac.id

Abstrak— Sistem informasi rekam medis memiliki peranan yang sangat penting dalam penyediaan informasi dan pelayanan kesehatan pasien di rumah sakit. Selama ini penggunaan aplikasi rekam medis elektronik pada Rumah Sakit Umum Daerah (RSUD) masih mengandalkan inputan/entri data secara manual terutama dalam pencatatan rekam medis pasien kedalam aplikasi. Urgensi penelitian adalah mengubah suara (voice) tenaga medis kedalam bentuk teks/tulisan sebagai inputan pencatatan rekam medis otomatis pada aplikasi rekam medis elektronik RSUD. Tujuan akhir penelitian untuk menghasilkan produk hilirisasi dalam penerapan model inputan pencatatan data rekam medis otomatis pada aplikasi rekam medis elektronik. Metodologi penelitian terdiri dari pengumpulan data, analisa kebutuhan sistem, desain sistem, koding sistem, pengujian sistem/uji coba model, implementasi dan pengoperasian sistem pada mitra. Sedangkan luaran yang ditargetkan yaitu : enerapan modul aplikasi Automatic Speech Recognition (ASR) Aplikasi Rekam Medis Elektronik Rumah Sakit Berbasis Deep Learning Menggunakan Metode Gate Recurrent Unit (GRU). Hasil penelitian menunjukkan bahwa model GRU-MK1 mencapai akurasi 83%, presisi 83%, dan recall 82%, dengan waktu pelatihan 5 jam dan 18 menit. Model GRU-MK2 dan GRU-MK3 menunjukkan kinerja yang serupa, dengan akurasi 92%, presisi 93%, dan recall 92%. Untuk waktu pelatihan, GRU-MK1 membutuhkan waktu lebih sedikit dibandingkan model lainnya, yaitu 5 jam 13 menit, sementara model GRU-MK2 membutuhkan waktu 6 jam 40 menit GRU-MK3 membutuhkan waktu 5 jam 52 menit.

Kata kunci— Deep Learning, Speech to text, GRU, Rekam Medis Elektronik.

Abstract— The medical record information system plays a crucial role in providing patient information and health services in hospitals. The use of electronic medical record applications at Regional General Hospitals (RSUD) currently relies on manual data input/entry, particularly in recording patient medical records into the application. The research urgency is to convert medical personnel's voice into text/writing as input for automatic medical record recording in the RSUD electronic medical record application. The ultimate goal of the research is to produce downstream products in the application of an automatic medical record data input model in the electronic medical record application. The research methodology consists of data collection, system requirements analysis, system design, system coding, system testing/model trials, implementation and operation of the system at partners. While the targeted outputs are: the implementation of the Automatic Speech Recognition (ASR) application module for the Hospital Electronic Medical Record Application Based on Deep Learning Using the Gate Recurrent Unit (GRU) Method. The results show that the GRU-MK1 model achieves 83% accuracy, 83% precision, and 82% recall, with a training time of 5 hours and 18 minutes. The GRU-MK2 and GRU-MK3 models showed similar performance, with 92% accuracy, 93% precision, and 92% recall. For training time, GRU-MK1 took less time than the other models, at 5 hours and 13 minutes, while the GRU-MK2 model took 6 hours and 40 minutes, and the GRU-MK3 model took 5 hours and 52 minutes.

Keywords— Deep Learning, Speech to text, GRU, Electronic Medical Records.

I. PENDAHULUAN

Teknologi saat ini berkembang pesat, dan manusia semakin memanfaatkan teknologi, terutama dalam bidang Deep Learning. Deep Learning adalah salah satu cabang dari Machine Learning yang mengadopsi prinsip ilmu komputer dan statistik untuk mengenali pola data kompleks [1]. Dalam konteks analisis data, salah satu teknik yang penting adalah klasifikasi. Berbagai metode klasifikasi memiliki kelebihan dan kelemahan yang perlu dipertimbangkan. Evaluasi kinerja metode klasifikasi dapat dilakukan dengan menggunakan berbagai kriteria atau metrik, seperti akurasi, presisi, recall, dan F1-score, yang membantu dalam memahami sejauh mana metode tersebut efektif dalam memprediksi kategori atau label data [2].

Rumah sakit merupakan lembaga penting dalam sistem pelayanan kesehatan yang menyediakan berbagai layanan medis bagi masyarakat. Sebagai pusat perawatan kesehatan, rumah sakit tidak hanya menyediakan layanan pengobatan dan perawatan medis, tetapi juga merupakan tempat di mana pasien mendapatkan diagnosis, perawatan, dan pemantauan kondisi kesehatannya. Di dalamnya, rekam medis memainkan peran utama sebagai fondasi sistem pelayanan kesehatan.

Rekam medis mencatat semua informasi penting tentang kesehatan pasien, termasuk riwayat medis, diagnosis, tindakan medis, dan perkembangan kondisi kesehatan seiring waktu [3].

Proses penulisan rekam medis melibatkan tiga aspek kunci: menjelaskan kondisi pasien, merinci rencana perawatan, dan mengevaluasi respons pasien terhadap terapi, yang diatur dalam empat langkah SOAP (Subjective, Objective, Assessment, Planning) [4]. Meskipun penting, dokter sering mengalami kesulitan dalam memasukkan data ke dalam sistem rekam medis digital karena multitasking dan fokus pada pasien. Informasi SOAP diperlukan dalam klaim BPJS untuk menguatkan justifikasi pelayanan kesehatan pasien. Pengembangan teknologi seperti pengenalan suara ke teks berbasis deep learning menjadi fokus untuk mengatasi kompleksitas kebutuhan kesehatan.

Penelitian ini melakukan pengujian performa metode Gated Recurrent Unit (GRU) pada masalah klasifikasi teks rekam medis yang diperoleh dari inputan suara dengan menggunakan data sebanyak 267.940 data teks dengan evaluasi model menggunakan confusion matrix. Dataset pada penelitian ini diperoleh dari RSUD Cut Meutia Aceh Utara.

Hasil penelitian adalah membuat suatu sistem yang mampu melakukan klasifikasi teks rekam medis yang diperoleh dari

input menggunakan suara dan akan di implementasi ke sistem yang sudah ada Paperless Hospital.

II. METODOLOGI PENELITIAN

Metodologi berisi tahapan teknis yang akan dilakukan pada tahap penelitian.

A. Pengumpulan Data

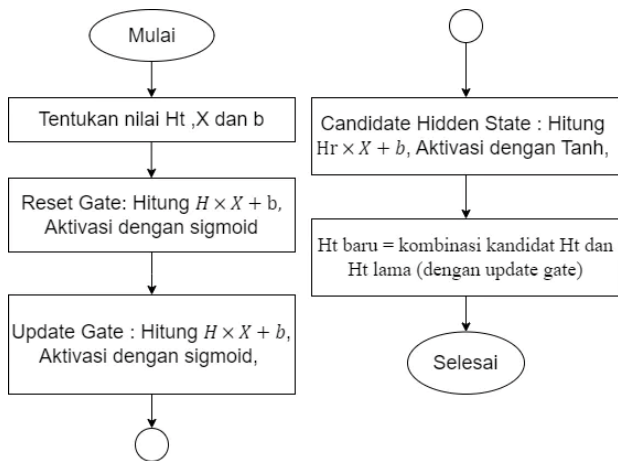
Data merupakan kumpulan fakta yang belum diolah sepenuhnya dan memerlukan pengolahan lebih lanjut. Dalam pengembangan sistem, diperlukan data yang relevan dan dapat dipertanggungjawabkan kebenarannya tanpa manipulasi. Selain dari data, pengumpulan informasi juga menjadi suatu kebutuhan. Data yang digunakan dalam penelitian ini adalah dataset teks rekam medis dari RSUD Cut Meutia yang berjumlah 267.940 teks. Contoh data sampel yang digunakan pada sistem ini dapat dilihat pada tabel 1.

Tabel 1. Contoh data SOAP

No	Data	Label
1	tampak terpasang gips	objek
2	close displaced fraktur ulna medial shaft sinistra dengan narkose	assessment
3	rencana re close reduction, reposition, re long arm cast	Plan
4	riwayat patah tulang tangan kiri akibat terjatuh saat bermain sudah dipasang gips. tidak ada nyeri pada tangan	subjek

B. Arsitektur model Gate Recurrent Unit (GRU)

Arsitektur model GRU ini melewati beberapa tahapan yang akan dilakukan. Berikut adalah arsitektur model pada Gambar 1.



Gambar 1. Arsitektur Model Gate Recurrent Unit (GRU)

Keterangan :
 Ht : Hidden State
 X : Nilai Input
 b : Bias

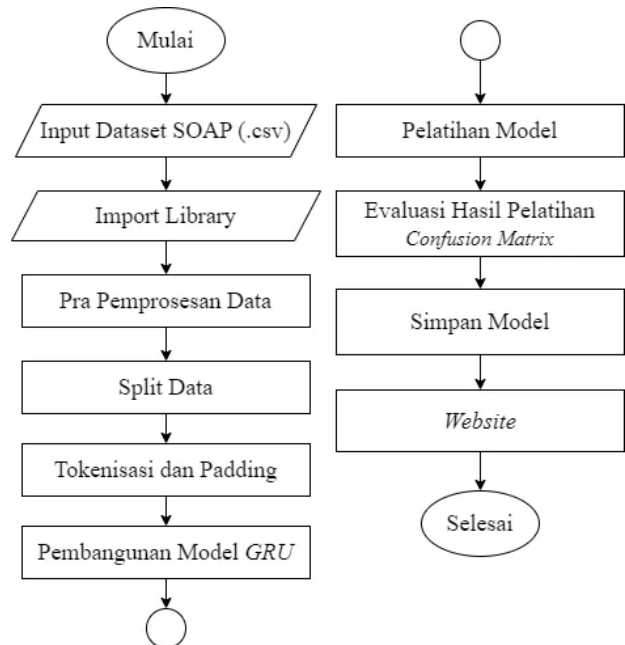
Sigmoid : Fungsi sigmoid adalah fungsi aktivasi yang memetakan nilai input menjadi nilai antara 0 dan 1.
 Tanh : Fungsi Tanh adalah fungsi aktivasi yang memetakan nilai input menjadi nilai antara -1 dan 1. Adapun Langkah-langkah tersebut dijelaskan sebagai berikut [5,6]:

Adapun Langkah-langkah tersebut dijelaskan sebagai berikut :

1. Menentukan Nilai Ht dan X: Menetapkan nilai hidden state sebelumnya (Ht) dan nilai input saat ini (X) sebagai dasar untuk perhitungan berikutnya.
2. Reset Gate: Menghitung nilai yang menentukan seberapa banyak informasi dari hidden state sebelumnya yang akan dilupakan, menggunakan fungsi sigmoid untuk menghasilkan nilai antara 0 dan 1.
3. Update Gate: Mengatur keseimbangan antara informasi baru dan lama dalam hidden state baru dengan menggunakan perhitungan produk dari hidden state sebelumnya dan input saat ini.
4. Candidate Hidden State: Membuat kandidat hidden state baru dengan memperhitungkan informasi yang telah disaring oleh reset gate, dan menerapkannya dalam fungsi Tanh untuk menghasilkan representasi baru data.
5. Output Ht Baru: Menggabungkan kandidat hidden state dengan hidden state sebelumnya berdasarkan hasil update gate untuk menghasilkan hidden state baru yang akan digunakan selanjutnya.

C. Perancangan Gate Recurrent Unit (GRU)

Berikut *flowchart* dari rancangan metode *Gate Recurrent Unit* (GRU) terdapat pada Gambar 2.



Gambar 2. Perancangan Penerapan GRU

Pada gambar 2 menggambarkan proses implementasi model *Speech to Text* berbasis Deep Learning pada aplikasi Paperless Hospital dengan metode *Gate Recurrent Unit* (GRU). Dimulai dengan penginputan dataset SOAP dalam

format .csv, dilanjutkan dengan pengimporan library yang dibutuhkan untuk analisis. Setelah itu, dilakukan pra-pemrosesan data dan pembagian data (split data) menjadi set pelatihan dan pengujian. Langkah berikutnya adalah tokenisasi dan *padding* data, yang mempersiapkan data teks untuk dimasukkan ke dalam model. Setelah model GRU dibangun, dilanjutkan dengan pelatihan model menggunakan data yang telah diproses. Hasil pelatihan dievaluasi menggunakan matriks kebingungan (*Confusion Matrix*). Model yang telah dilatih kemudian disimpan dan digunakan dalam aplikasi berbasis web. Flowchart ini diakhiri dengan simbol "END" yang menunjukkan bahwa proses telah selesai.

D. Evaluasi Model Menggunakan Confussion Matrix

Confussion matrix adalah salah satu tabel yang berisi perhitungan evaluasi performa model klasifikasi berdasarkan jumlah objek penelitian yang diprediksi dengan benar dan salah terdapat pada tabel 2.

Tabel 2. Confusion Matrix

Data Aktua	Data Prediksi	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	False Negative (TN)

Perhitungan *confusion matrix* melibatkan empat kombinasi nilai prediksi dan nilai aktual, yang secara umum dikenal sebagai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). True Positive merujuk pada data positif yang berhasil diprediksi secara tepat, sementara True Negative adalah jumlah data negatif yang berhasil terdeteksi dengan benar. False Positive, juga dikenal sebagai Type-1 Error, terjadi ketika data negatif salah terdeteksi sebagai positif. Sementara itu, False Negative, atau Type-2 Error, mengindikasikan data positif yang keliru terdeteksi sebagai negative [7,8].

III. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan dataset sebanyak 267.940 data. Pembangunan model *deep learning* menggunakan beberapa prosedur. Berikut implementasi dari masing-masing prosedur.

A. Preprocessing

Subbab ini, akan dilakukan pembahasan secara terperinci mengenai teknik-teknik preprocessing yang diterapkan, termasuk proses pembersihan data. Berikut contoh data sebelum dilakukan proses pembersihan data pada tabel 3.

Tabel 3. Data SOAP Sebelum Proses Processing

Subjek	Asesmen	Objek	Plan
demam 2 hari,mual dan muntah.	observasi febris + dyspepsia	conjuctiva inf anemis,perut kembung	pemeriksaan darah rutin,elektroli t,widal tes

Pada tahap ini, data mentah dari file Excel diolah melalui beberapa langkah:

1. Mengimpor Data: Data dari file *datamentah.xlsx* diimpor menggunakan pustaka Pandas.
2. Penanganan Nilai Kosong: Nilai NaN diganti dengan string kosong ("") untuk memudahkan proses berikutnya.
3. Pembuatan UUID: Setiap baris diberi kolom "id" berisi UUID unik untuk identifikasi.
4. Pembuatan Label Data: Label dibuat berdasarkan kolom subjek, objek, asesmen, dan plan, dengan pembersihan karakter yang tidak diperlukan.
5. Penyimpanan Data: Data dilabeli dan disimpan sebagai *data_labeled2.csv* dengan header berisi id, dataset, dan label.

Hasil dari proses preprocessing data dapat dilihat pada tabel 4.

Tabel 4. Hasil proses preprocessing data

Id	Dataset	Label
93533a80-980b-4064-8747-3c3ededa6c26	demam 2 hari,mual dan muntah	subjek
93533a80-980b-4064-8747-3c3ededa6c26	conjuctiva inf anemis,perut kembung	objek
93533a80-980b-4064-8747-3c3ededa6c26	observasi febris + dyspepsia	assessment
93533a80-980b-4064-8747-3c3ededa6c26	pemeriksaan darah rutin,elektrolit,widal tes	plan

B. Split Data

Split data merupakan tahap krusial dalam pembuatan model pembelajaran mesin. Proses ini membagi dataset menjadi dua subset berbeda: data pelatihan (training set) dan data pengujian (testing set). Pada penelitian ini, dataset dibagi dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian.

Pembagian ini memastikan bahwa model memiliki cukup data untuk belajar (training) serta sejumlah data yang terpisah untuk mengukur kinerjanya (testing). Selain itu, dalam pembagian dataset ini, diterapkan pengaturan *random state* yang tetap untuk memastikan konsistensi dan replikasi hasil. Dengan pengaturan ini, data akan selalu terbagi dengan cara yang sama, memudahkan evaluasi dan perbandingan akurasi model.

C. Penerapan Nilai Parameter

Subbab ini, akan dijelaskan mengenai penerapan nilai parameter yang digunakan dalam membangun model pembelajaran mesin.

1. Nilai parameter model GRU pertama

Pada penerapan parameter GRU pertama, nilai-nilai terdapat pada tabel 5.

Tabel 5. Nilai parameter model GRU pertama

Layer	Type	Units	Return Sequences	Activation Function	Dropout Rate
1	GRU	64	Yes	Tanh	0.2
2	GRU	64	Yes	Tanh	0.2
3	GRU	64	Yes	Tanh	0.2
4	GRU	64	Yes	Tanh	0.2
5	DENSE	4	-	Softmax	-

Model GRU yang dibangun terdiri dari empat lapisan GRU dan satu lapisan output. Lapisan pertama GRU memiliki 64 unit dengan pengaturan *return sequences* aktif, yang berarti keluaran dari setiap langkah waktu akan diteruskan ke lapisan berikutnya. Fungsi aktivasi yang digunakan adalah Tanh untuk menangani transformasi non-linear, sedangkan *dropout* dengan tingkat 0,2 diterapkan untuk mencegah *overfitting* dengan menonaktifkan 20% unit secara acak selama pelatihan. Lapisan kedua dan ketiga memiliki konfigurasi yang sama, yaitu menggunakan 64 unit, *return sequences* aktif, dan fungsi aktivasi Tanh, serta *dropout* sebesar 0,2 untuk meningkatkan generalisasi model.

Lapisan keempat juga memiliki 64 unit dan fungsi aktivasi Tanh, tetapi tidak mengembalikan urutan (*return sequences* non-aktif). Hal ini menjadikannya langkah pemrosesan GRU terakhir sebelum menuju lapisan output. Pada lapisan ini, tingkat *dropout* tetap dipertahankan sebesar 0,2. Lapisan terakhir adalah lapisan *dense* dengan 4 unit dan menggunakan fungsi aktivasi *softmax*. Lapisan ini berfungsi menghasilkan prediksi akhir dalam bentuk distribusi probabilitas untuk empat kelas. Arsitektur model ini dirancang untuk memproses data berurutan secara efektif, menangkap pola temporal yang kompleks, serta menghindari *overfitting* dengan penerapan *dropout* pada setiap lapisan GRU.

2. Nilai parameter model GRU kedua

Nilai parameter GRU kedua beserta parameter-parameternya terdapat pada tabel 6.

Tabel 6 Nilai parameter model GRU kedua

Layer	Type	Units	Return Sequences	Activation Function	Dropout Rate
1	Embedding	128	-	-	-
2	Bidirectional GRU	128	Yes	Tanh	0.2
3	Bidirectional GRU	128	Yes	Tanh	0.2
4	GlobalAveragePooling 1D	-	-	-	-
5	DENSE	256	-	ReLU	-
6	Dropout	-	-	-	0.1
7	Dense	64	-	ReLU	-
8	Dropout	-	-	-	0.1
9	Dense	4	-	Softmax	-

Model GRU kedua ini memiliki arsitektur yang lebih kompleks dibandingkan model sebelumnya, dengan

kombinasi lapisan *embedding*, lapisan *bidirectional GRU*, dan lapisan *dense*. Lapisan pertama adalah lapisan *embedding* yang memiliki *input_dim* sebesar ukuran kosa kata dan *output_dim* sebesar 128, yang berfungsi untuk memetakan setiap kata dalam dataset menjadi vektor berdimensi 128. Selanjutnya, model ini memiliki dua lapisan *bidirectional GRU*, masing-masing dengan 128 unit dan pengaturan *dropout* sebesar 0,2, serta menggunakan *return sequences* untuk menangkap informasi urutan data secara lebih efektif.

Setelah lapisan GRU, terdapat lapisan *GlobalAveragePooling1D* untuk meratakan keluaran dari lapisan sebelumnya. Kemudian, terdapat dua lapisan *dense* dengan 256 dan 64 unit, yang masing-masing menggunakan fungsi aktivasi ReLU. Pada setiap lapisan *dense* ini, diterapkan *batch normalization* untuk menstabilkan proses pelatihan, serta *dropout* sebesar 0,1 untuk mencegah *overfitting*. Akhirnya, lapisan terakhir adalah lapisan *dense* dengan 4 unit dan fungsi aktivasi *softmax* yang bertugas menghasilkan prediksi dalam bentuk distribusi probabilitas untuk empat kelas yang berbeda. Arsitektur ini dirancang untuk menangkap pola sekuensial data dengan lebih efektif dan akurat.

3. Nilai parameter model GRU ketiga

Berikut nilai pada model GRU ketiga beserta parameter-parameternya terdapat pada tabel 7.

Tabel 7. Nilai parameter model GRU ketiga

Layer	Type	Units	Return Sequences	Activation Function	Dropout Rate
1	Embedding	128	-	-	-
2	Bidirectional GRU	128	Yes	Tanh	0.2
3	Bidirectional GRU	128	Yes	Tanh	0.2
4	GlobalAveragePooling 1D	-	-	-	-
5	DENSE	512	-	ReLU	-
6	Dropout	-	-	-	0.1
7	Dense	128	-	ReLU	-
8	Dropout	-	-	-	0.1
9	Dense	4	-	Softmax	-

Model GRU ketiga ini memiliki arsitektur yang lebih kompleks dengan penambahan beberapa lapisan untuk menangkap pola dalam data sekuensial secara lebih mendalam. Lapisan pertama adalah lapisan *embedding*, yang memetakan setiap kata dalam dataset menjadi vektor berdimensi 128. Ini membantu model dalam memahami hubungan antar kata dalam urutan data.

Selanjutnya, model memiliki dua lapisan *Bidirectional GRU*, masing-masing terdiri dari 256 unit. Kedua lapisan ini memiliki pengaturan *return sequences* aktif, yang berarti keluaran dari setiap langkah waktu akan diteruskan ke lapisan berikutnya. Penggunaan fungsi aktivasi Tanh dan pengaturan *dropout* sebesar 0,2 pada setiap lapisan GRU ini membantu model menangkap pola non-linear dan mengurangi risiko *overfitting*. Selain itu, parameter *reset_after* disetel ke *true*, memungkinkan unit GRU untuk mereset setelah setiap batch,

yang dapat meningkatkan kinerja model dalam menangani urutan data.

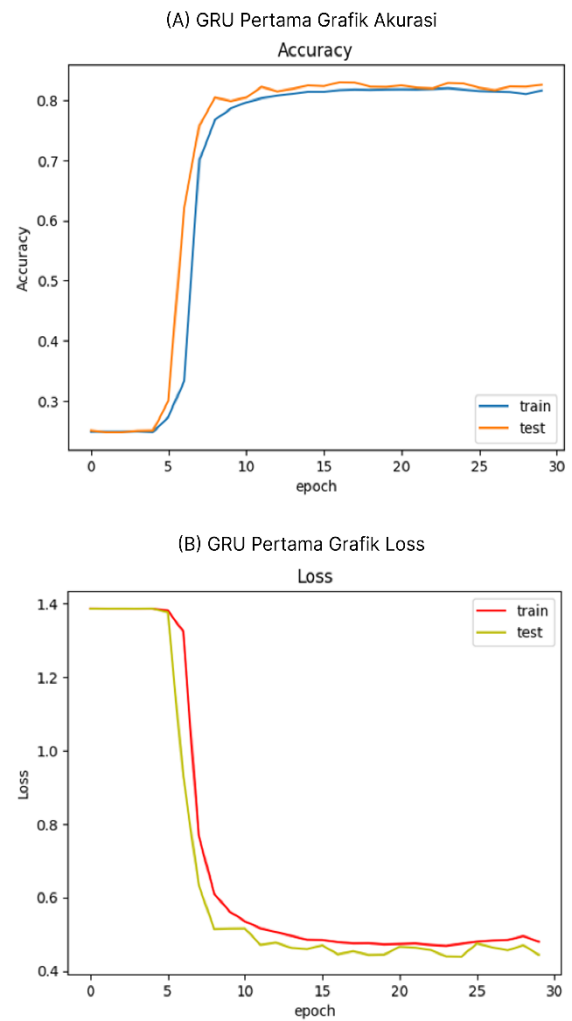
Setelah lapisan GRU, lapisan *GlobalAveragePooling1D* digunakan untuk meratakan keluaran, diikuti oleh dua lapisan *dense* dengan 512 dan 128 unit yang menggunakan fungsi aktivasi ReLU. Pada kedua lapisan ini, *batch normalization* diterapkan untuk menstabilkan proses pelatihan dan mempercepat konvergensi. *Dropout* dengan tingkat 0,1 juga diterapkan pada setiap lapisan *dense* untuk lebih meningkatkan generalisasi model. Akhirnya, lapisan terakhir adalah lapisan *dense* dengan 4 unit dan fungsi aktivasi *softmax*, yang menghasilkan prediksi dalam bentuk distribusi probabilitas untuk empat kelas. Arsitektur ini dirancang untuk memproses data sekuensial secara efektif, menangkap pola yang lebih kompleks, dan menjaga stabilitas selama pelatihan.

D. Analisis Hasil Pelatihan Model

Pada bagian ini, akan dipaparkan hasil pengujian dan evaluasi kinerja model klasifikasi SOAP yang telah dikembangkan. Pengujian dilakukan untuk menilai seberapa baik model dalam mengklasifikasikan data teks SOAP [9].

1. Model GRU pertama

Hasil pelatihan model dan akurasi nilai dan loss terdapat pada Gambar 3.



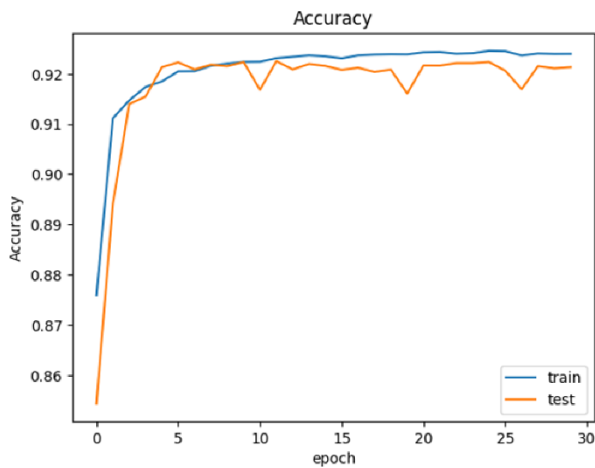
Gambar 3. GRU Pertama Grafik Akurasi dan Loss

Pada gambar 3 A grafik akurasi dari pengujian model GRU pertama selama 30 epoch, akurasi pelatihan mulai dari 0.2492 pada epoch pertama dan meningkat signifikan, terutama mulai epoch ke-6, dengan akurasi validasi mencapai 0.6204 pada epoch ke-7. Akurasi stabil di kisaran 0.8 pada akhir epoch ke-30, dengan akurasi pelatihan 0.8154 dan validasi 0.8252. Pada gambar 1 B grafik nilai loss menunjukkan penurunan signifikan dari sekitar 1,4 di awal hingga stabil di kisaran 0,4-0,5 setelah epoch keenam. Total waktu pelatihan adalah 5 jam, 13 menit, dan 42 detik.

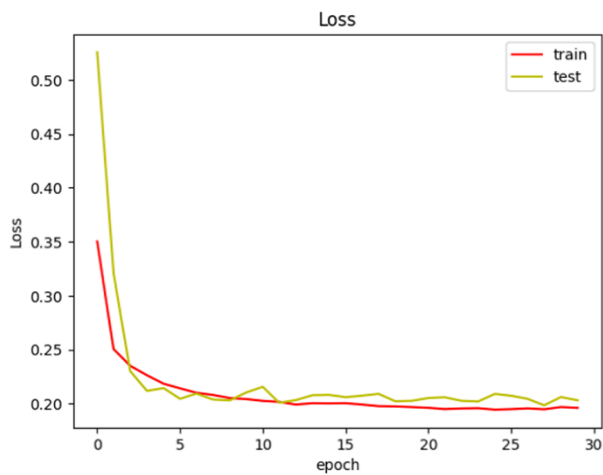
2. Model GRU kedua

Nilai akurasi dan loss dari model GRU kedua terdapat pada Gambar 4.

(A) GRU Kedua Grafik Akurasi



(B) GRU Kedua Grafik Loss



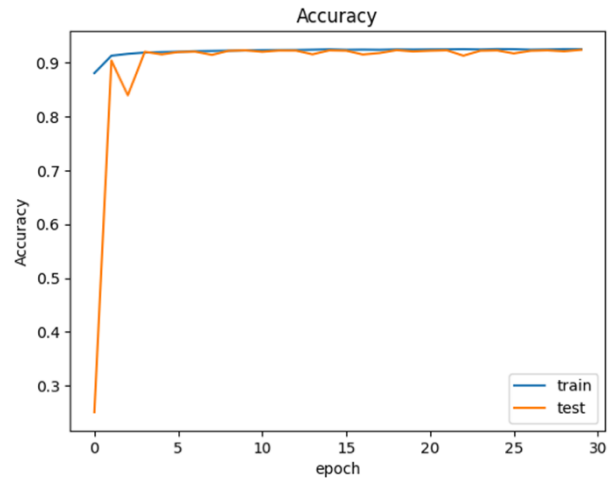
Gambar 4. GRU Kedua Grafik Akurasi dan Loss

Grafik pada gambar 4 A menunjukkan peningkatan akurasi model GRU kedua selama 30 epoch pelatihan, dengan akurasi pelatihan dimulai dari 0.8759 dan akurasi validasi dari 0.8543 pada epoch pertama, yang terus meningkat hingga mencapai 0.9240 (pelatihan) dan 0.9213 (validasi) pada akhir pelatihan. Sementara itu, grafik loss gambar 4 B menggambarkan penurunan signifikan nilai loss, dimulai dari 0.3503 (pelatihan) dan 0.5255 (validasi) pada epoch pertama hingga mencapai stabilitas pada sekitar 0.1963 (pelatihan) dan 0.2032 (validasi) pada akhir pelatihan, menunjukkan konvergensi model dalam waktu 6 jam, 40 menit, dan 1 detik.

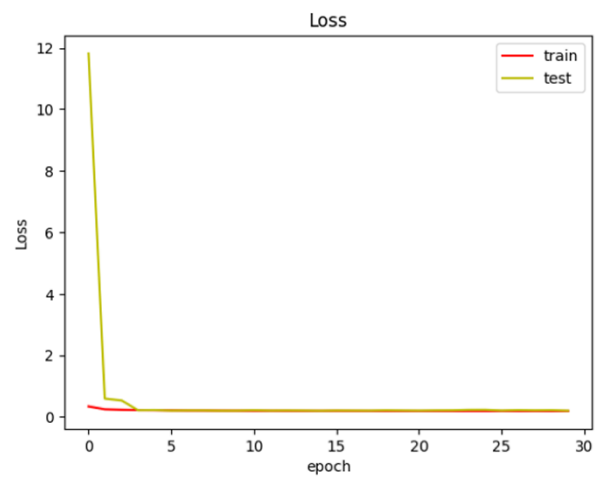
3. Model GRU ketiga

Nilai akurasi dan loss dari model GRU ketiga terdapat pada Gambar 5.

(A) GRU Ketiga Grafik Akurasi



(B) GRU Ketiga Grafik Loss



Gambar 5. GRU Ketiga Grafik Akurasi dan Loss

Grafik 5 A menunjukkan bahwa model GRU ketiga mengalami peningkatan akurasi yang signifikan selama 30 epoch pelatihan. Pada awal pelatihan, akurasi model untuk data pelatihan dimulai dari 0,8807, sementara akurasi validasi berada pada 0,2507. Nilai ini terus meningkat secara konsisten seiring bertambahnya jumlah epoch, menunjukkan bahwa model berhasil belajar dari data yang diberikan. Pada akhir pelatihan, model mencapai akurasi sebesar 0,9245 untuk data pelatihan dan 0,9233 untuk data validasi. Peningkatan yang hampir seimbang antara akurasi pelatihan dan validasi ini menandakan bahwa model mampu generalisasi dengan baik terhadap data baru, mengindikasikan kinerja yang optimal dan minim *overfitting*.

Grafik 5 B memperlihatkan penurunan nilai *loss* yang cukup drastis sepanjang proses pelatihan. Pada epoch pertama, nilai *loss* untuk data pelatihan berada di angka 0,3397, sementara untuk data validasi mencapai 0,5867. Seiring bertambahnya epoch, nilai *loss* terus menurun dan akhirnya stabil di sekitar 0,1916 untuk pelatihan dan 0,2028 untuk validasi. Penurunan ini menunjukkan bahwa model berhasil

meminimalkan kesalahan prediksi dan konvergen dengan baik, menyelesaikan proses pelatihan dalam waktu total 5 jam 52 menit 34 detik. Stabilitas nilai *loss* di akhir pelatihan menandakan bahwa model telah mencapai titik optimal, di mana penambahan epoch lebih lanjut kemungkinan tidak akan memberikan peningkatan yang signifikan terhadap kinerja model.

IV. KESIMPULAN

Implementasi speech to text berbasis deep learning dengan metode Gated Recurrent Unit (GRU) pada aplikasi Paperless Hospital berhasil menunjukkan efektivitas dalam klasifikasi data SOAP. Model GRU Ketiga memberikan hasil terbaik dengan akurasi sebesar 92.45%, precision 93%, dan recall 92.33%, serta waktu pelatihan yang lebih efisien dibandingkan dua model lainnya, yaitu 5 jam 46 menit 1 detik. Struktur model yang digunakan, termasuk lapisan embedding, Bidirectional GRU, GlobalAveragePooling1D dan lapisan dense dengan batch normalization dan dropout terbukti mampu meningkatkan performa dalam menangani data medis yang kompleks.

REFERENSI

- [1] J. Schmidhuber, "Deep Learning in neural networks: An overview," 2015. doi: 10.1016/j.neunet.2014.09.003.
- [2] J. Brownlee, "Machine learning mastery," 2020.
- [3] S. Flora et al., "SISTEM PENGELOLAAN REKAM MEDIS PUSKESMAS HEALTH MEDICAL RECORD MANAGEMENT SYSTEM," *Jambura Health and Sport Journal*, vol. 4, no. 2, 2022.
- [4] I. Sudirahayu, A. Harjoko, H. A. Moeloek, and P. Lampung, "Analisis Kesiapan Penerapan Rekam Medis Elektronik Menggunakan DOQ-IT di RSUD Dr. H. Abdul Moeloek Lampung," 2016.
- [5] B. Schloss and S. Konam, "Towards an Automated SOAP Note: Classifying Utterances from Medical Conversations," in *Proceedings of Machine Learning Research*, 2020.
- [6] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in *Lecture Notes in Networks and Systems*, 2023. doi: 10.1007/978-3-031-35314-7_2.
- [7] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing*, Proceedings of the Conference, 2014. doi: 10.3115/v1/d14-1179.
- [8] M. Muntean and F. D. Militaru, "Metrics for Evaluating Classification Algorithms," in *Smart Innovation, Systems and Technologies*, 2023. doi: 10.1007/978-981-19-6755-9_24.
- [9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, [Online]. Available: <http://arxiv.org/abs/1412.3555>