

# KLASIFIKASI TEKS BERBASIS *LONG SHORT-TERM MEMORY* UNTUK CHATBOT KONSELING GANGGUAN KECEMASAN SOSIAL

Kiki Widiyanto, , Suryo Adhi Wibowo, Unang Sunarya<sup>3</sup>

<sup>1,2,3</sup>Fakultas Teknik Elektro, Universitas Telkom, Bandung

kikiwidiyanto@student.telkomuniversity.ac.id<sup>1</sup>, suryoadhiwibowo@telkomuniversity.co.id<sup>2</sup>,

unangsunarya@telkomuniversity.ac.id<sup>3</sup>

## Abstrak

*Chatbot* adalah teknologi kecerdasan buatan yang dapat melakukan percakapan seperti manusia melalui teks atau suara. Sistem *chatbot* berupa tanya jawab dapat membantu kegiatan manusia, melakukan konsultasi untuk suatu masalah bahkan menawarkan solusi untuk masalah medis. Setiap manusia memiliki tingkat kecemasan. Tingkat kecemasan yang berlebihan dapat menjadi gangguan kecemasan sosial yang mengganggu aktivitas. Kehadiran *chatbot* dapat menjadikan solusi bagi pengguna yang memiliki masalah kecemasan. Oleh karena itu, pada penelitian ini dilakukan klasifikasi teks untuk konseling *chatbot* dan analisa terhadap akurasi *chatbot* gangguan kecemasan sosial dengan menggunakan parameter performasni accuracy, precision, recall, dan f1-score. Klasifikasi teks menggunakan *Long Short-Term Memory* (LSTM) dengan pemodelan teks *pre-processing* menggunakan modifikasi dataset ISEAR serta tanggapan dari beberapa orang. Metode LSTM yang diusulkan yaitu memetakan jawaban pengguna dari *chatbot* berdasarkan kategori label. Sistem dilatih menggunakan dataset berupa teks. Dataset dibuat berupa jawaban dari pengguna diberi label positif dan negatif dengan 70 *training* data. Hasil penelitian menunjukkan model epoch 4 memiliki konfigurasi terbaik yaitu *RMSprop learning rate* 0,001 dengan *test accuracy* 85,71%. Sedangkan pada model epoch 6 memiliki konfigurasi terbaik yaitu *RMSprop learning rate* 0,01 dengan *test accuracy* 89,29%. Selain itu, parameter performansi pada epoch 4 rata-rata *precision* 97%, *recall* 97%, dan *f1-score* 97%. Parameter performansi pada epoch 6 rata-rata *precision* 97%, *recall* 97%, dan *f1-score* 97%.

**Kata Kunci:** *chatbot*, gangguan kecemasan sosial, *long short-term memory*.

## Abstract

*Chatbots are artificial intelligence technologies that can conduct human-like conversations. Chatbot systems in the form of Q&A can help human activities, consulting on a problem and even offer solutions to medical problems. Every human being has a level of anxiety. Therefore, chatbots can be a personal solution that act like problem solver. Based on the above problems, on this research conducted test classification for chatbot counseling. The classification of text uses Long Short-Term Memory (LSTM) and the text modeling utilizes the modified dataset of ISEAR as well as the responses from multiple people. The proposed LSTM method in the text classification was used to map the user's responses of chatbot regarding its labels created in advance. The results of this study, epoch 4 model has the best configuration, RMSprop learning rate 0.001 with a test accuracy of 85.71%. Likewise, the epoch 6 model has the best configuration, RMSprop learning rate 0.01 with a test accuracy of 89.29%. In addition, performance parameters at epoch 4 average 97% precision, recall 97%, and f1-score 97%. Then the performance parameters on epoch 6 average precision 97%, recall 97%, and f1-score 97%.*

**Keywords:** *Chatbot, Long Short-Term Memory, Social Anxiety Disorder.*

## I. PENDAHULUAN

Manusia adalah makhluk yang istimewa dengan memiliki akal, pikiran, perasaan, serta emosi. Pada aktivitas sehari-hari yang penuh dengan tekanan tentu tidak menutup kemungkinan seseorang merasa cemas. Permasalahan yang biasanya dihadapi pada usia remaja akhir salah satunya yaitu gangguan kecemasan sosial atau *Social Anxiety Disorder* (SAD). Gangguan kecemasan sosial merupakan ketakutan yang terus menerus dan tidak realistis terhadap lingkungan sosial dapat menarik perhatian orang asing, atau bahkan hanya mengekspos mereka [1]. Pada penelitian [2] kecemasan sosial adalah masalah kesehatan jiwa (psikologis) yang cukup banyak terjadi di seluruh dunia, termasuk masyarakat Indonesia. Populasi di Indonesia yang

mengalami kecemasan sosial yaitu sekitar 15,8%. Selain itu, aspek keistimewaan manusia salah satunya adalah kecerdasan. Dalam hal ini manusia mampu membuat mesin yang memiliki kecerdasan buatan atau *Artificial Intelligence* (AI). Dalam teknologi AI dikenal sebuah program komputer yaitu *chatbot* yang mana dapat mensimulasikan percakapan seperti manusia. Sistem tanya jawab dalam bentuk *chatbot* dapat membantu dalam keadaan darurat maupun menawarkan solusi untuk masalah medis yang dangkal serta mengurangi biaya perawatan kesehatan dan waktu pengguna, karena tidak memungkinkan sebagai pengguna untuk mengujungi ahli atau dokter saat dibutuhkan.

Penelitian sebelumnya [3] menggunakan metode *pattern-matching* mengenai pengetahuan dalam *database* yang mana

masukannya akan cocok dengan pola yang telah ada pada pengetahuannya. Penelitian mengenai pembuatan percakapan otomatis [4] mengembangkan *chatting bot* dengan menerapkan *Long Short-Term Memory* (LSTM) memiliki sistem tingkat akurasi menggunakan *neural network* sekitar 30% dari keseluruhan percakapan. Selanjutnya penelitian mengenai *chatbot* pendaftaran mahasiswa baru menggunakan metode *Recurrent Neural Network* (RNN) [5] menghasilkan tingkat akurasi aplikasi sebesar 88%, presisi sebesar 95% dan *recall* sebesar 92%.

Penelitian ini mengusulkan untuk menggunakan metode *Long Short-Term Memory*. Oleh karena itu, untuk mengatasi keterbatasan tingkat akurasi pada penelitian sebelumnya maka menambahkan beberapa parameter performansi *precision*, *recall*, dan *f1-score*. Sistem yang akan dirancang menggunakan *supervised learning* dilakukan klasifikasi teks dengan hasil luaran berupa analisis dan evaluasi kinerja LSTM berdasarkan parameter *accuracy*, *precision*, *recall*, dan *f1-score*.

## II. KAJIAN TEORI

Penelitian menggunakan gangguan Kecemasan Sosial, *Artificial Intelligence*, *Chatbot*, *Long-Short Term Memory* (LSTM), dan Bahasa pemrograman *Python* sebagai konsep dasar penelitian.

### A. Gangguan Kecemasan Sosial

Gangguan kecemasan sosial adalah ketakutan yang terus-menerus dan tidak realistis terhadap situasi sosial yang dapat menarik perhatian orang asing atau bahkan hanya terpapar pada orang asing [1]. Meskipun penyakit ini ditandai sebagai fobia sosial di DSM-IV-TR, istilah gangguan kecemasan sosial diusulkan dalam DSM-5 karena masalah yang ditimbulkannya cenderung lebih luas dan mengganggu aktivitas normal daripada masalah. Orang dengan gangguan kecemasan sosial takut dan menghindari pengawasan orang lain [6].

### B. Artificial Intelligence (AI)

AI merupakan bidang studi yang menguji pemahaman dan peniruan kecerdasan manusia, perilaku cerdas, dan hukum perilaku cerdas. Tugas utamanya adalah membangun teori pemrosesan informasi cerdas, dan kemudian merancang beberapa sistem komputer yang dapat menunjukkan beberapa tindakan serupa dari kecerdasan manusia. Kemampuan kecerdasan buatan adalah perilaku cerdas yang berhubungan dengan kecerdasan manusia, biasanya dilakukan oleh mesin cerdas, seperti penilaian, penalaran, verifikasi, pengenalan, persepsi, pemahaman, komunikasi, desain, pemikiran, pembelajaran dan pemecahan masalah[7].

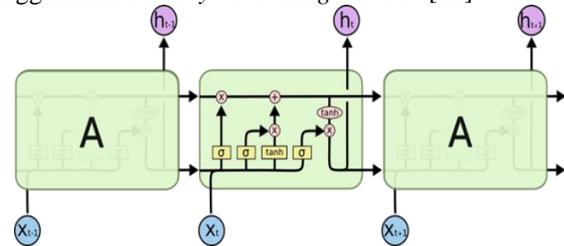
### C. Chatbot

*Chatbot* adalah program komputer yang merespons seperti entitas cerdas ketika berbicara dengan seseorang dengan melakukan percakapan yang dapat dilakukan melalui teks atau suara [8]. Pada dasarnya *chatbot* berinteraksi dengan pengguna dalam subjek tertentu menggunakan

bahasa alami. Banyak *chatbot* yang telah digunakan dengan tujuan pendidikan, layanan pelanggan, panduan, hiburan [9].

### D. Long-Short Term Memory (LSTM)

*Long-Short Term Memory* adalah jaringan *Recurrent Neural Network* (RNN) yang dilengkapi dengan mekanisme gerbang khusus yang mengontrol akses ke *memory cells* [10]. LSTM merupakan unit RNN yang didesain untuk menanggulangi ketergantungan jangka panjang. Dalam standar RNN, modul ini akan memiliki struktur yang sangat sederhana, seperti *layer tanh* tunggal [11]. LSTM dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units* [12].



GAMBAR 1  
ARSITEKTUR LSTM

Pada *forget gates* informasi tentang setiap data masukan diproses dan data mana yang akan disimpan atau dibuang di unit penyimpanan dipilih [13]. Data  $x_t$  yaitu data *input* (vektor *input*  $x$  dalam *timestamp*  $t$ ) serta  $h_{t-1}$  yaitu vektor *hidden state* pada *timestamp* sebelumnya  $t-1$  [14]. Hasil *outputnya* yaitu antara 0 dan 1. Jika *outputnya* 1 semua data akan disimpan, sebaliknya jika *outputnya* 0 semua data akan dibuang. Maka pada persamaan 1 menunjukkan perhitungan nilai *forget gate* adalah sebagai berikut.

$$f_t = s(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Pada *input gates* informasi diolah menggunakan perhitungan pada persamaan 2 *Input gates* ada dua gerbang yang akan diimplementasikan, pertama akan menentukan informasi nilai tertentu yang diperbarui ke bagian *cell state* dengan menggunakan fungsi aktivasi *sigmoid* [13]. Kemudian pada langkah ini membentuk kandidat vektor baru menggunakan fungsi aktivasi *tanh* yang akan disimpan pada *memory cell* menggunakan perhitungan pada persamaan 3 [13].

$$i_t = s(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Pada *cell gates* akan memperbarui nilai *cell state* lama  $c_{t-1}$  dengan nilai *cell state* baru  $c_t$ . Dimana nilai ini diperoleh dengan menggabungkan nilai-nilai yang terdapat pada *forget gate* dan *input gate* melalui perhitungan pada persamaan 4 [13].

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (4)$$

Pada *output gates* terdapat dua gerbang yang akan diimplementasikan, pertama memproses fungsi *sigmoid* untuk menghasilkan *output* pada *hidden state* serta menempatkan *cell state* pada *tanh*. Setelah menghasilkan nilai *output sigmoid* dan nilai *output tanh* maka kedua *gates* tersebut dikalikan untuk menghasilkan nilai yang akan

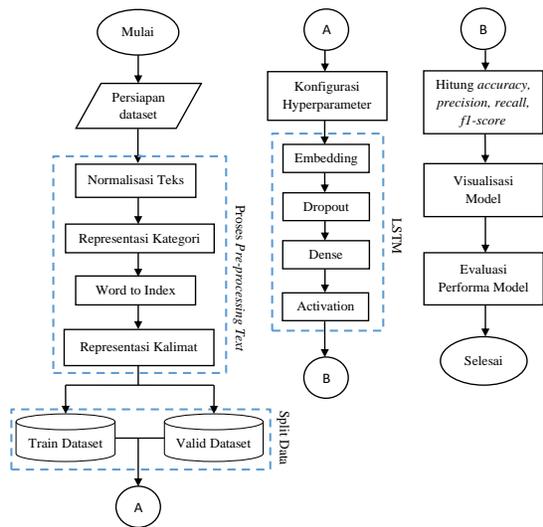
dikeluarkan. Perhitungan pada *output* dilakukan melalui persamaan 5 dan 6 [13].

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

### III. METODE

Penelitian ini melakukan perancangan dan pengujian sistem yang mengklasifikasi teks untuk menganalisis jawaban pengguna pada *chatbot* konseling gangguan kecemasan sosial dengan menggunakan metode LSTM. Alur kerja sistem pada penelitian dapat dilihat pada Gambar 2.



GAMBAR 2. DIAGRAM ALIR SISTEM KLASIFIKASI TEKS LSTM

.....

#### A. Datasets

Pengambilan *dataset* dilakukan dengan tautan *spreadsheet* untuk mengisi permasalahan mengenai gangguan kecemasan sosial yang dibagi menjadi dua jenis yaitu positif (1) dan negatif (0). *Dataset* yang digunakan berbentuk teks/kalimat, dimana teks tersebut merupakan kumpulan tanggapan yang diperoleh dari 70 responden yang terbagi menjadi dua label dengan total 140 data sebagaimana ditunjukkan pada Tabel 1. Untuk sistem ini *dataset* dibagi menjadi 80% data latih dan 20% data uji.

TABEL 1. SKEMA DATASET GANGGUAN KECEMASAN SOSIAL

| No    | Kategori | Jumlah |
|-------|----------|--------|
| 1     | Positif  | 70     |
| 2     | Negatif  | 70     |
| Total |          | 140    |

#### B. Pre-processing

Pada proses *pre-processing* merupakan tahapan dimana aplikasi melakukan penyeleksian data yang akan diproses sebelum masuk ke dalam metode LSTM *Pre-processing* ini memiliki tahapan-tahapan sebagai berikut.

#### 1. Normalisasi Teks

Normalisasi teks meliputi perubahan singkatan, akronim, angka, tanggal, waktu, karakter khusus, dan simbol dengan bentuk huruf alphabet sehingga tidak terjadi ambiguitas. Berikut contoh hasil normalisasi teks pada Tabel 2.

TABEL 2. PROSES NORMALISASI TEKS

| Teks                           | Normalisasi                  |
|--------------------------------|------------------------------|
| SaYa meRAsA GugUP              | saya merasa gugup            |
| JantuNG Sayaa teRaSA berdEbarr | jantung saya terasa berdebar |

#### 2. Representasi Kategori

Proses *labelling* dilakukan dengan mengubah semua jenis kalimat menjadi array menggunakan `LabelEncoder()`. Label encoder untuk mengubah variabel target atau setiap nilai dalam kolom menjadi angka yang berurutan. Detail contoh proses labelling dapat dilihat pada Tabel 3.

TABEL 3. PROSES LABELLING

| Teks                                      | Label   | LabelEncoder |
|---|---------|--------------|
| saya takut berada di tempat umum          | Negatif | 0            |
| gembira saat kontrak kerja ditandatangani | Positif | 1            |

#### 3. Stopwords dan Stemmer

*Stopwords* merupakan sebuah kata penghubung berfungsi untuk menghapus kata-kata yang tidak spesifik sehingga kata-kata tertentu yang paling sering muncul dapat dideteksi. Sedangkan *stemming* adalah proses menghilangkan suffix dan prefix pada token/kata (*reduce inflected*) seperti di-, me-, dan lainnya untuk mengembalikan kata depan ke bentuk dasar, contoh diberikan pada Tabel 4.

TABEL 4. PROSES STOPWORDS DAN STEMMER

| Teks  | Label   | Stopwords                       | Stemmer                   |
|---|---------|---------------------------------|---------------------------|
| saya merasa gugup                                 | Negatif | merasa gugup                    | rasa gugup                |
| gembira ketika berhasil dalam membantu orang lain | Positif | gembira berhasil membantu orang | gembira hasil bantu orang |

#### 4. Representasi Kalimat

Proses membagi serangkaian kata dalam kalimat atau paragraf menjadi token atau bagian kata tunggal yang berdiri sendiri. Menggunakan `pad_sequences` untuk mengubah masing-masing kata ke dalam matriks berdimensi 1x10. Nilai 10 menunjukkan nilai maksimal kata yang digunakan. Contoh diberikan pada Tabel 5.

TABEL 5  
PROSES TOKENISASI

| Teks                      | Representasi kalimat       |
|---------------------------|----------------------------|
| rasa cemas keringat       | [3, 6, 23, ..., 0, 0, 0]   |
| gembira hasil bantu orang | [11, 223, 92, ... 0, 0, 0] |

5. *Word to Index*

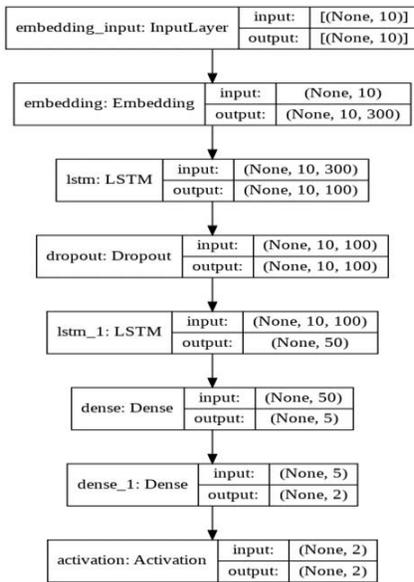
Pada tahapan ini berfungsi untuk mengubah setiap kata ke dalam bentuk indeks. *Word-to-index* merupakan suatu bagian dari proses tokenisasi. Detail contoh diberikan pada Tabel 6.

TABEL 6  
PROSES WORD TO INDEX

| Teks                         | Index  |
|------------------------------|--|
| cemas harus temu orang asing | [6: 'cemas', 16: 'harus', 35: 'temu', 1: 'orang', 22: 'asing'] |
| keringat bicara depan tonton | [23: 'keringat', 5: 'bicara', 8: 'depan', 81: 'tonton']        |

C. Proses Long Short-Term Memory (LSTM)

Pada proses LSTM menggunakan *library Keras* sebagai pembuatan model. Arsitektur model yang digunakan dilihat pada Gambar 3.



GAMBAR 3  
ARSITEKTUR MODEL

Model sistem terdiri dari *input layer*, *embedding layer*, *LSTM layer*, *dropout layer*, *dense layer*, *activation layer* dan *output layer*.

*Hyperparameter* yang digunakan pada model dapat dilihat pada Tabel 7.

TABEL 7  
HYPERPARAMETER MODEL

| Hyperparameter | Value         |
|----------------|---------------|
| Dropout        | 0.2           |
| Activation     | Sigmoid       |
| Epoch          | 4 ; 6         |
| Batch Size     | 8             |
| Optimizer      | Adam, RMSprop |

|               |                       |
|---------------|-----------------------|
| Learning Rate | 0,01 ; 0,001 ; 0,0001 |
| Loss Function | Binary Cross Entropy  |
| Max Length    | 10                    |

D. Analisis Parameter Kinerja

Parameter kinerja yang diuji dalam penelitian ini yaitu *accuracy*, *precision*, *recall*, dan *f1-score* dengan menggunakan acuan *confusion matrix* yang merepresentasikan prediksi dan kondisi sebenarnya (aktual) dari data yang dihasilkan.

1. *Accuracy*

Parameter *accuracy* digunakan untuk menghitung rasio prediksi benar (positif dan negatif) dengan keseluruhan data [13]. Perhitungan *accuracy* dapat dihitung dengan menggunakan persamaan 7

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

2. *Precision*

Parameter *precision* digunakan untuk menghitung rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif [13]. Perhitungan pada presisi yang diterapkan menggunakan persamaan (8)

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

3. *Recall*

Parameter *recall* digunakan untuk menghitung rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif [13]. Perhitungan *recall* yang diterapkan menggunakan persamaan 9

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

4. *F1-score*

Parameter *f1-score* digunakan untuk menghitung rata-rata dari *precision* dan *recall* [13]. Perhitungan pada *f1-score* dapat dilihat di persamaan 10

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (10)$$

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian Sistem

Penelitian menggunakan 140 dataset dengan empat skenario pengujian yang bertujuan untuk mendapatkan konfigurasi terbaik dari model yang telah dirancang. Berikut skenario pengujian yang dilakukan:

1. Skenario 1: Optimasi *Learning Rate* dan *Optimizer* pada Epoch 4

Skenario ini bertujuan untuk menentukan model sistem terbaik yang dilakukan dengan mengkombinasikan *learning rate* dan *optimizer* yang diujikan pada nilai *epoch* 4. Pengujian dilakukan dengan tiga *learning rate*, yaitu  $10^{-2}$ ,  $10^{-3}$ , dan  $10^{-4}$  serta *optimizer* Adam dan RMSprop. Konfigurasi terbaik ditentukan oleh nilai *test accuracy* yang dihasilkan dari setiap *learning rate* dan *optimizer*.

2. Skenario 2: Pengujian Parameter *Accuracy*, *Precision*, *Recall*, dan *F1-score* pada *Epoch* 4 dengan Konfigurasi Terbaik

Skenario 2 merupakan konfigurasi dari model terbaik yang dihasilkan dari Skenario 1 melalui *test accuracy* dengan mempertimbangkan pada grafik *training* dan *testing*. *Confusion matrix* dari hasil *testing* model terbaik dijelaskan dan dianalisis.

3. Skenario 3: Optimasi *Learning Rate* dan *Optimizer* pada *Epoch* 6

Skenario pengujian sama dengan Skenario 1, yang berbeda terdapat pada nilai *epoch* yang digunakan. Bertujuan untuk menentukan model sistem terbaik yang dilakukan mengkombinasikan *learning rate* dan *optimizer* pada nilai *epoch* 6. Pengujian dilakukan dengan tiga *learning rate*, yaitu  $10^{-2}$ ,  $10^{-3}$ , dan  $10^{-4}$  serta *optimizer* Adam dan RMSprop. Konfigurasi terbaik ditentukan oleh nilai *test accuracy* yang dihasilkan dari setiap *learning rate* dan *optimizer*.

4. Skenario 4: Pengujian Parameter *Accuracy*, *Precision*, *Recall*, dan *F1-score* pada *Epoch* 6 dengan Konfigurasi Terbaik

Skenario 2 merupakan konfigurasi dari model terbaik yang dihasilkan dari Skenario 3 melalui *test accuracy* dengan mempertimbangkan pada grafik *training* dan *testing*. *Confusion matrix* dari hasil *testing* model terbaik dijelaskan dan dianalisis.

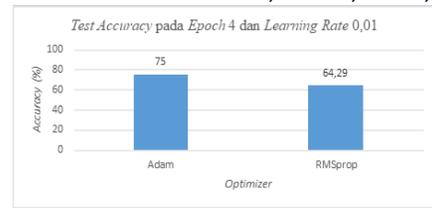
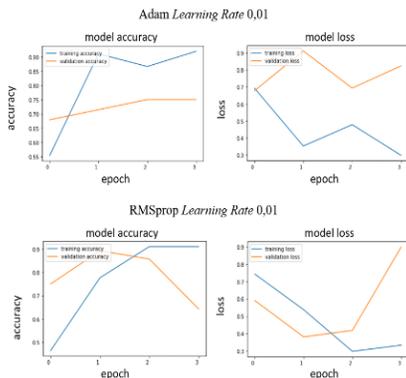
B. Hasil dan Analisis Pengujian Sistem

Hasil dan analisis dari pengujian sistem sebagai berikut.

1. Skenario 1: Optimasi *Learning Rate* dan *Optimizer* pada *Epoch* 4

Pada proses optimasi, terdapat dua *hyperparameter* yang digunakan untuk mencapai kinerja yang maksimal, antara lain *learning rate* dan *optimizer*. *Learning rate* yang digunakan yaitu  $10^{-2}$ ,  $10^{-3}$ , dan  $10^{-4}$  sedangkan untuk *optimizer* menggunakan Adam dan RMSprop.

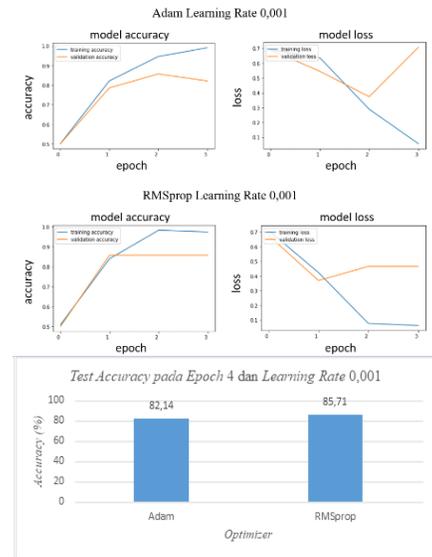
Gambar berikut menunjukkan grafik *training* dan *testing* terhadap nilai *epoch* dengan nilai *learning rate* 0,01.



GAMBAR 4 KONFIGURASI OPTIMIZER & TEST ACCURACY PADA LEARNING RATE 0,01

Pada konfigurasi ini grafik hasil *training* dan *testing* terhadap nilai *epoch* dapat dilihat pada Gambar 4. *Test accuracy* terbaik didapat oleh *optimizer* Adam dengan nilai 75,00%. Hal ini dikarenakan model *optimizer* pada Adam menunjukkan *training accuracy* dan *validation accuracy* yang semakin naik. Sedangkan pada *optimizer* RMSprop menunjukkan grafik yang lebih buruk dibandingkan dengan *optimizer* Adam.

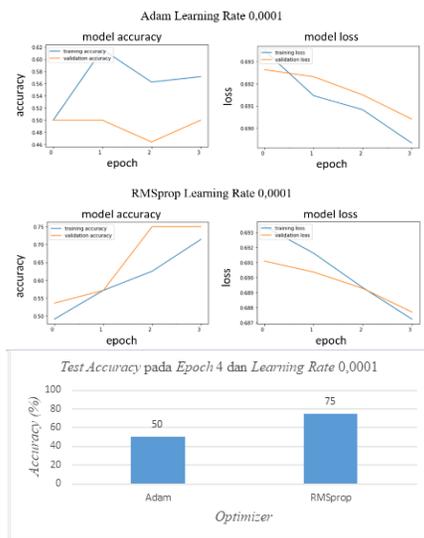
Gambar 5 adalah grafik *training* dan *testing* terhadap nilai *epoch* dengan nilai *learning rate* 0,001. Dapat dilihat grafik diatas bahwa kedua *optimizer* tersebut menunjukkan performa yang kurang baik. Model dengan *optimizer* Adam mengalami *underfitting* dilihat dari kurva *training loss* yang belum menemukan penurunan ke titik yang stabil. RMSprop mendapatkan hasil *test accuracy* tertinggi sebesar 85,71%.



GAMBAR 5 KONFIGURASI OPTIMIZER & TEST ACCURACY PADA LEARNING RATE 0,001

Gambar 6 merupakan grafik *training* dan *testing* terhadap nilai *epoch* dengan nilai *learning rate* 0,0001. Dari grafik dibawah kedua model tersebut, model dengan *optimizer* Adam menunjukkan performa yang buruk. Hal ini terdapat pada *loss* yang terindikasi *overfitting* dan *test accuracy* yang cukup jauh menandakan tidak stabil. Sedangkan pada *optimizer* RMSprop menunjukkan performa yang cukup baik

untuk setiap *epoch*. *Test accuracy* tertinggi dihasilkan oleh model *optimizer* RMSprop dengan nilai 75.00%.



GAMBAR 6 KONFIGURASI OPTIMIZER & TEST ACCURACY PADA LEARNING RATE 0,0001

2. Skenario 2: Pengujian Parameter *Accuracy*, *Precision*, *Recall*, dan *F1-score* pada *Epoch 4* dengan Konfigurasi Terbaik

Pada model dengan konfigurasi terbaik yaitu ditentukan dengan memilih *test accuracy* yang tertinggi. Pada Gambar 5, model *optimizer* RMSprop 0,001 menunjukkan grafik *training* dan *testing* terhadap *epoch 4* yang cukup baik. Oleh karena itu, konfigurasi dengan model terbaik diperoleh *optimizer* RMSprop dengan nilai *learning rate* 0,001 serta *test accuracy* sebesar 85,71%.

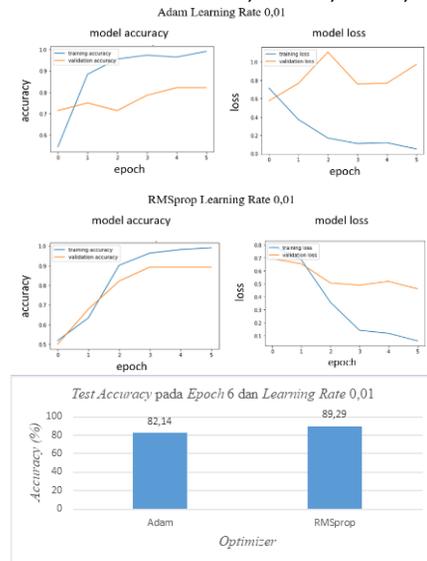
Hasil *confusion matrix* pada *epoch 4* dengan 140 memetakan empat kondisi. 69 data kondisi TP, 67 data kondisi TN, 3 data kondisi FP, dan 1 data kondisi FN. Tabel 8. menunjukkan hasil *precision*, *recall*, dan *f1-score* dengan menggunakan persamaan (7) sampai dengan (10)

TABEL 8. EVALUASI PERFORMANSI PADA EPOCH 4 DENGAN OPTIMIZER RMSPROP DAN LEARNING RATE 0,001

| Classes       | Precision | Recall | F1-Score |
|---------------|-----------|--------|----------|
| 0 (Negatif)   | 0,99      | 0,96   | 0,97     |
| 1 (Positif)   | 0,96      | 0,99   | 0,97     |
| Macro Average | 0,97      | 0,97   | 0,97     |

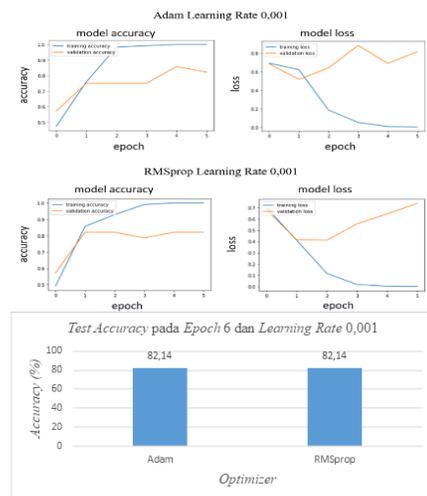
3. Skenario 3: Optimasi *Learning Rate* dan *Optimizer* pada Epoch 6

Skenario pengujian ini bertujuan untuk menentukan model sistem terbaik yang dilakukan dengan mengkombinasikan *learning rate* dan *optimizer* yang diujikan pada nilai *epoch 6*. Pada proses optimasi, terdapat dua *hyperparameter* yang digunakan untuk mencapai kinerja yang maksimal, antara lain *learning rate* dan *optimizer*.



GAMBAR 7 TRAINING DAN TESTING & TEST ACCURACY PADA EPOCH, LEARNING RATE 0,01

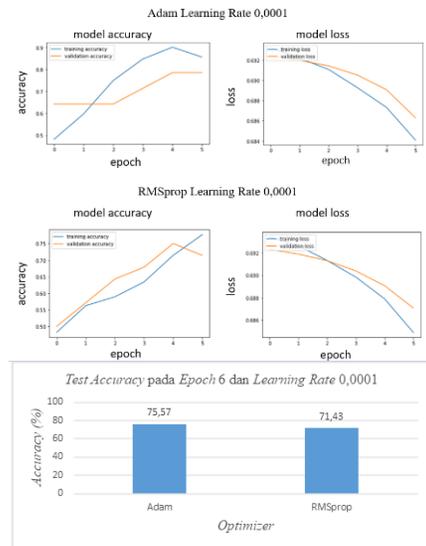
Gambar 6 menunjukkan grafik hasil *training* dan *testing* terhadap nilai *epoch*. Model *optimizer* Adam menunjukkan indikasi *overfitting* dikarenakan *validation loss* lebih tinggi dari *training loss* serta tidak menunjukkan penurunan kurva. Sedangkan RMSprop menunjukkan model yang lebih baik. Dalam konfigurasi *optimizer* ini, nilai *test accuracy* tertinggi didapat oleh RMSprop sebesar 89,29%.



GAMBAR 8 TRAINING DAN TESTING & TEST ACCURACY PADA EPOCH, LEARNING RATE 0,001

Pada Gambar 8. menunjukkan grafik *training* dan *testing* terhadap nilai *epoch* dengan nilai *learning rate* 0,001. Model dengan *optimizer* Adam dan RMSprop mengalami *overfitting* dilihat dari kurva nilai *validation loss* yang semakin besar dan kurva nilai *training loss* yang semakin kecil. *Test accuracy* yang dihasilkan dari kedua model *optimizer* memiliki nilai yang sama dan cukup tinggi yaitu sebesar 82,14%.

|                      |      |      |      |
|----------------------|------|------|------|
| <b>Macro Average</b> | 0,97 | 0,97 | 0,97 |
|----------------------|------|------|------|



GAMBAR 9  
TRAINING DAN TESTING & TEST ACCURACY PADA EPOCH,  
LEARNING RATE 0,0001

Pada Gambar 9. menunjukkan grafik *training* dan *testing* terhadap nilai *epoch* dengan nilai *learning rate* 0,0001. Pada model dengan *optimizer* Adam menunjukkan grafik *training* dan *validation loss* semakin kecil, begitu pula pada *optimizer* RMSprop. Namun, grafik yang dihasilkan RMSprop tidak lebih baik dibandingkan dengan Adam dan juga dapat dibuktikan dengan melihat nilai *validation loss* dan *training loss* yang semakin dekat. Dalam konfigurasi ini nilai *test accuracy* yang cukup tinggi yaitu *optimizer* Adam dengan nilai 78.57%.

4. Skenario 4: Pengujian Parameter *Accuracy*, *Precision*, *Recall*, dan *F1-score* pada *Epoch* 6 dengan Konfigurasi Terbaik

Pada model dengan konfigurasi terbaik yaitu ditentukan dengan memilih *test accuracy* yang tertinggi. Selain dari itu, grafik pada *training* dan *testing* terhadap *epoch* juga dipertimbangkan. Dapat dilihat pada Gambar 7, bahwa model dengan *test accuracy* yang tertinggi yaitu dengan *optimizer* RMSprop pada nilai *learning rate* 0,01. Meskipun demikian perbedaan antara kedua model *optimizer* tidak terlalu jauh apabila dilihat dari *test accuracy*, model dengan RMSprop tetap menunjukkan performa yang lebih baik.

Hasil *confusion matrix* pada *epoch* 6 dengan 140 data memetakan empat kondisi antara lain 69 data kondisi TP, 67 data kondisi TN, 3 data kondisi FP, dan 1 data kondisi FN. Tabel 9. menunjukkan hasil *accuracy*, *precision*, *recall*, dan *f1-score* dengan menggunakan persamaan (7) sampai dengan (10)

TABEL 9  
EVALUASI PERFORMANSI PADA EPOCH 6 DENGAN OPTIMIZER  
RMSPROP DAN LEARNING RATE 0,01

| Classes     | Precision | Recall | F1-Score |
|-------------|-----------|--------|----------|
| 0 (Negatif) | 0,99      | 0,96   | 0,97     |
| 1 (Positif) | 0,96      | 0,99   | 0,97     |

V. KESIMPULAN

Penerapan klasifikasi teks untuk konseling gangguan kecemasan sosial menggunakan *chatbot* pada metode LSTM telah berhasil diterapkan dengan dua jumlah *epoch* yang dilakukan percobaan *trial* dan *error*. Model dengan jumlah *epoch* 4 memiliki konfigurasi *optimizer* dan *learning rate* terbaik yaitu RMSprop pada *learning rate* 0,001 dengan *test accuracy* 85,71%. Demikian juga pada model dengan jumlah *epoch* 6 memiliki konfigurasi *optimizer* dan *learning rate* terbaik yaitu RMSprop pada *learning rate* 0,01 dengan *test accuracy* 89,29%. Berdasarkan hasil pengujian beberapa konfigurasi menghasilkan *test accuracy* yang cukup tinggi namun pada proses *training* dan *testing* yang kurang baik. Hal tersebut memungkinkan ketidakcocokkan pada beberapa faktor yang dapat memengaruhi performa model yang dibuat seperti model arsitektur maupun *hyperparameter*.

Penelitian selanjutnya dapat menggunakan multi-class classification text dalam pengujian model untuk membandingkan dan menganalisa performansi pada sistem chatbot.

REFERENSI

- [1] A. M. Kring, S. L. Johnson, G. C. Davison, and J. M. Neale, *Abnormal Psychology, 12th Edition*, 12th ed. Wiley, 2012.
- [2] E. P. N. Hasibuan, W. Srisayekti, and M. F. Moeliono, "GAMBARAN KECEMASAN SOSIAL BERDASARKAN LIEBOWITZ SOCIAL ANXIETY SCALE (LSAS) PADA REMAJA AKHIR DI BANDUNG," Universitas Padjadjaran, 2015.
- [3] B. Setiaji and F. W. Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling," *Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS*, vol. 0, pp. 72–77, 2016, doi: 10.1109/ISMS.2016.53.
- [4] T. Fujita, W. Bai, and C. Quan, "Long short-term memory networks for automatic generation of conversations," *Proc. - 18th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2017*, pp. 483–487, 2017, doi: 10.1109/SNPD.2017.8022766.
- [5] T. A. Zuraiyah et al., "Mahasiswa Baru Menggunakan Recurrent Neural Network," pp. 91–101.
- [6] M. B. Stein and D. J. Stein, "Social anxiety disorder," *Lancet*, vol. 371, no. 9618, pp. 1115–1125, 2008, doi: 10.1016/S0140-6736(08)60488-2.
- [7] X. Liu, "Artificial intelligence and modern sports education technology," *Proc. - 2010 Int. Conf. Artif. Intell. Educ. ICAIE 2010*, pp. 772–776, 2010, doi: 10.1109/ICAIE.2010.5641441.

- [8] A. Khanna, B. Pandey, K. Vashishta, K. Kalia, B. Pradeepkumar, and T. Das, "A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence," *Int. J. u- e-Service, Sci. Technol.*, vol. 8, no. 7, pp. 277–284, 2015, doi: 10.14257/ijunesst.2015.8.7.28.
- [9] S. V. Doshi, S. B. Pawar, A. G. Shelar, and S. S. Kulkarni, "Artificial Intelligence Chatbot in Android System using Open Source Program-O," *Ijarcce*, vol. 6, no. 4, pp. 816–821, 2017, doi: 10.17148/ijarcce.2017.64151.
- [10] A. M. Rahman, A. Al Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2017, pp. 75–78, doi: 10.1109/R10-HTC.2017.8288910.
- [11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [12] C. Olah, "Understanding LSTM Networks," 2015. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed Oct. 28, 2020).
- [13] M. Wildan, P. Aldi, and A. Aditsania, "Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin," *e-Proceeding Eng.*, vol. 5, no. 2, pp. 3548–3555, 2018.
- [14] R. F. Rahmadzani, "Cara Kerja Long Short-Term Memory (LSTM)," Feb. 21, 2021. <https://rifqifai.com/cara-kerja-long-short-term-memory-lstm/> (accessed Oct. 04, 2021).