Implementasi Long Short-Term Memory dalam Mendeteksi Kesalahan Pronunciation Bahasa Inggris Berbasis Audio

Roro Ayu Fasha Dewatri[1], Ricky Eka Putra[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]roro.20059@mhs.unesa.ac.id
[2]rickyeka@unesa.ac.id

***Abstrak*— Perkembangan kecerdasan buatan membuka peluang dalam mendukung pembelajaran bahasa, khususnya dalam mendeteksi dan memperbaiki kesalahan pengucapan (*pronunciation*). Bahasa Inggris, sebagai bahasa internasional, sering kali menimbulkan tantangan dalam pengucapan yang dapat mengubah makna pesan jika ada kesalahan. Penelitian ini memanfaatkan model Long Short-Term Memory (LSTM) untuk mendeteksi kesalahan pengucapan bahasa Inggris berbasis audio. Data yang digunakan terdiri dari dataset TIMIT, yang mewakili penutur asli Amerika, dan Common Voice untuk penutur non-Amerika. Data diproses dengan resampling, padding zero, trimming berbasis energi, dan normalisasi untuk mengekstraksi fitur yang lebih fokus pada bagian audio yang signifikan. Pembagian data dilakukan menggunakan K-Fold Cross-Validation (k=10) dengan proporsi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Eksperimen dilakukan dengan fokus pada optimizer Adam dengan berbagai kombinasi hyperparameter, seperti batch size (16, 32, 64), epoch (50, 75, 100), dan learning rate (0.001, 0.0001), dengan evaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Kombinasi hyperparameter yang optimal ditemukan pada akurasi 94% dan F1-score 95% pada kombinasi batch size 32, epoch 100, dan learning rate 10^{-4}. Penelitian ini mengidentifikasi kombinasi hyperparameter yang optimal untuk mencapai stabilitas model yang baik dan membuka peluang untuk pengembangan sistem yang dapat memberikan umpan balik korektif otomatis bagi pengguna.**

***Kata Kunci*— Long Short-Term Memory (LSTM), Pengucapan Bahasa Inggris, Pemrosesan Audio, TIMIT, Common Voice, Hyperparameter Tuning.**

I. PENDAHULUAN

Kemajuan dalam pengembangan sistem kecerdasan buatan (AI) telah membawa dampak besar pada berbagai bidang, termasuk pengenalan suara dan pengajaran bahasa. AI, atau Artificial Intelligence, memungkinkan mesin untuk memahami dan menanggapi data dengan cara yang menyerupai manusia, menjadikannya alat yang sangat berguna dalam menghadapi berbagai tantangan, terutama di bidang pendidikan dan komunikasi. Dalam hal ini, salah satu kemampuan yang menarik dari AI adalah mendeteksi serta memperbaiki kesalahan pengucapan dalam berbagai bahasa, termasuk Bahasa Inggris.

Bahasa Inggris merupakan bahasa internasional yang digunakan sebagai media untuk berbagi informasi, perdagangan, pendidikan, dan banyak hal lainnya di seluruh dunia [1]. Banyak negara termasuk Cina, Denmark, Swedia, Iran, Jepang, Korea, Indonesia, dan masih banyak lagi yang memperlakukan Bahasa Inggris sebagai bahasa asing, English as Foreign Language (EFL) [2]. Meskipun dianggap sebagai bahasa asing, tidak dipungkiri bahwa bahasa secara umum digunakan sebagai alat komunikasi sehingga dapat menyampaikan pesan dalam bentuk suara atau tulisan. Pengucapan atau pronunciation yang baik merupakan hal yang krusial karena pengucapan yang salah akan membuat penerima salah menginterpretasikan pesan. Pengucapan bunyi huruf dalam kata, serta penekanan suku kata pada bagian kata, sering kali secara drastis akan mengubah arti dan konteks kata, sehingga mengubah makna kalimat yang sedang dikomunikasikan [3]. Kesalahan dalam penggunaan bahasa dapat terjadi karena pengucapan yang tidak tepat (*mispronunciation*) sehingga menimbulkan pergeseran makna. Dalam hal ini, ketepatan pelafalan menjadi penting agar maksud dari pembicaraan dapat dimengerti dengan baik oleh pendengar. Pengaruh bahasa daerah dapat mengubah bentuk intonasi dan pengucapan manusia. Seperti yang diketahui bersama, ada banyak jenis bahasa daerah di suatu negeri yang menciptakan aksen dan pengucapan yang beragam di antara orang-orang [4].

Dalam upaya untuk mengatasi masalah tersebut, teknologi deep learning, khususnya Long Short-Term Memory (LSTM) digunakan untuk membuat sistem deteksi kesalahan pronunciation berbasis audio. Deep learning merupakan cabang dari machine learning yang menggunakan jaringan neural dengan beberapa lapisan (layer) untuk menginterpretasikan data yang kompleks. Salah satu model deep learning yang paling dikenal adalah LSTM, Recurrent Neural Network (RNN) yang dirancang khusus untuk memahami dan memproses urutan data, seperti teks atau suara, dengan ketergantungan jangka panjang [5]. Bagian kunci yang meningkatkan kemampuan LSTM untuk memodelkan jangka Panjang adalah komponen yang disebut blok memori. Dengan kemampuannya untuk mempertahankan informasi dari waktu ke waktu, LSTM menjadi pilihan yang ideal dalam konteks mendeteksi dan memperbaiki kesalahan pronunciation dalam Bahasa Inggris berbasis audio.

II. METODOLOGI PENELITIAN

A. *Pendekatan Penelitian*

Penelitian ini menggunakan metode pendekatan eksperimental yang terstruktur untuk menyelidiki dan mengevaluasi kinerja model Long Short-Term Memory (LSTM) dalam mendeteksi kesalahan pengucapan Bahasa Inggris berbasis audio. Pendekatan ini melibatkan pengendalian dan manipulasi variabel-variabel tertentu dalam

lingkungan terkendali. Proses eksperimental dilakukan melalui serangkaian percobaan dimana hyperparameter pada model LSTM dikombinasikan secara sistematis dan hasilnya dievaluasi untuk menemukan kombinasi yang memberikan kinerja paling optimal. Metode eksperimental dipilih karena memberikan dapat mengontrol variabel-variabel yang relevan dengan penelitian ini. Dalam kerangka metode eksperimental, variabel-variabel yang dikendalikan termasuk, tetapi tidak terbatas pada, pengaturan seperti ukuran batch, epoch, learning rate, optimizer, dan metrik evaluasi.

B. *Dataset*

Penelitian ini menggunakan dua dataset publik utama, yaitu TIMIT dan Common Voice, untuk memberikan variasi dalam jenis penutur bahasa Inggris, baik penutur asli (*native*) dengan aksen Amerika maupun penutur asli (*native*) dengan aksen non-Amerika.

TIMIT Acoustic-Phonetic Continuous Speech Corpus merupakan hasil kerja sama antara Massachusetts Institute of Technology (MIT), SRI International (SRI), dan Texas Instruments, Inc. (TI). Pidato dalam dataset ini direkam di TI, ditranskrip di MIT, serta diverifikasi dan dipersiapkan untuk produksi CD-ROM oleh National Institute of Standards and Technology (NIST) [6].

Common Voice, di sisi lain, adalah dataset publik yang dapat diakses melalui platform Kaggle. Dataset ini disediakan oleh Mozilla dan terdiri dari korpus data ucapan yang direkam oleh pengguna di situs web Common Voice. Teks dalam dataset ini berasal dari berbagai sumber domain publik, seperti posting blog yang dikirimkan pengguna, buku-buku lama, film, dan korpus ucapan publik lainnya [7].

Kedua dataset ini dipilih sebagai data utama dalam deteksi kesalahan pengucapan dan dikelola untuk menyederhanakan struktur data serta mempermudah proses selanjutnya.

1) *Dataset TIMIT*

Dataset TIMIT awalnya terstruktur dalam beberapa folder berdasarkan informasi penutur dan wilayah dialek. Namun, dalam penelitian ini, hanya file audio yang digunakan, sehingga semua file audio dari TIMIT digabungkan ke dalam satu folder "timit_data" tanpa memisahkan berdasarkan penutur atau wilayah dialek. Langkah ini dilakukan mengingat penelitian lebih berfokus pada deteksi kesalahan pengucapan daripada analisis berdasarkan penutur individu. Dataset TIMIT digunakan sebagai acuan untuk pengucapan yang benar karena data ini berisi rekaman audio dari penutur asli (*native speaker*) bahasa Inggris dengan dialek Amerika Serikat. Dataset TIMIT memiliki kualitas audio yang tinggi dan menyediakan transkripsi fonetik yang akurat, sehingga dapat membantu model dalam mempelajari pola pengucapan yang benar (Label 0).

2) *Dataset Common Voice*

Dataset Common Voice yang terdiri dari berbagai folder akan menggunakan dua folder untuk penelitian ini yakni "cv-valid-dev" dan "cv-valid-test". Kedua folder tersebut terdiri dari data penutur Bahasa Inggris beraksen non-Amerika—baik dari *native speaker* negara-negara berbahasa Inggris lainnya seperti Inggris dan Kanada maupun *non-native speaker* dari negara-negara di luar wilayah berbahasa Inggris. Dataset Common Voice ini digunakan untuk menandakan adanya kemungkinan perbedaan atau kesalahan dalam pengucapan (Label 1) dibandingkan dengan standar aksen Amerika.

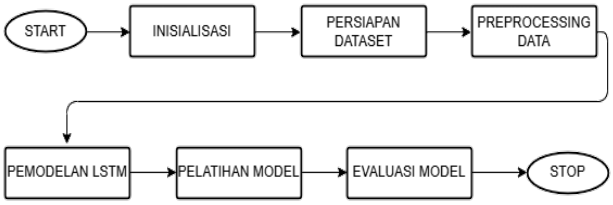
Keseluruhan data yang digunakan sebanyak 14372 file audio dengan rinciannya dapat dilihat pada Tabel 1.

TABEL 1
RINCIAN JUMLAH DATASET

Dataset	Jumlah Data	Jumlah Label 0	Jumlah Label 1
TIMIT	6301	6301	-
Common Voice	8071	-	8071
TOTAL	14372	6301	8071

C. *Alur Sistem*

Alur perancangan sistem deteksi kesalahan pronunciation dengan model LSTM akan dijelaskan secara detail dimana memberikan panduan langkah demi langkah dalam mengimplementasikan sistem. Proses dimulai dari dari inisialisasi hingga evaluasi model, dengan penekanan khusus pada langkah-langkah seperti persiapan dataset, preprocessing data audio, pembangunan model LSTM, pelatihan model, dan evaluasi kinerja. Gbr. 1 memberikan gambaran visual yang mendetail mengenai alur perancangan sistem ini.



Gbr. 1 Alur Perancangan Sistem Deteksi Kesalahan Pronunciation

1) *Inisialisasi*

Pada tahap inisialisasi, langkah awal yang dilakukan adalah mengimpor sejumlah library utama yang dibutuhkan untuk pengolahan data audio dan pengembangan model Long Short-Term Memory (LSTM). Library yang digunakan meliputi Librosa, yang dirancang untuk ekstraksi fitur dan analisis data audio; TensorFlow dan Keras, yang berfungsi sebagai kerangka kerja pembelajaran mesin berbasis deep learning; Numpy yang memungkinkan manipulasi data numerik secara efisien; serta Matplotlib dan Seaborn, yang digunakan untuk membuat visualisasi data guna mempermudah eksplorasi dan analisis pola.

2) *Persiapan Dataset*

Setelah mengimpor library, langkah berikutnya adalah memuat dataset audio yang akan digunakan berasal dari dua sumber utama yakni TIMIT dan Common Voice dari Mozilla.

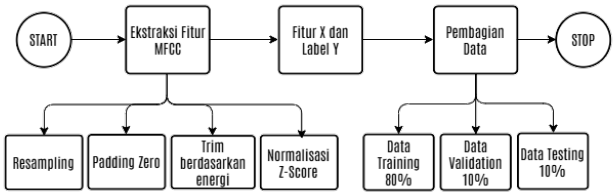
a. TIMIT: dataset ini telah tersedia dalam format WAV dan sudah siap pakai tanpa perlu konversi tambahan. Data pada TIMIT awalnya terstruktur per penutur yang kemudian digabung menjadi satu folder"timit-data". Data ini terdiri dari 6301 file WAV dengan total ukuran 591.76 MB.

b. Common Voice: Data audio dalam format MP3 dimana yang dibutuhkan adalah bentuk WAV. Maka, data akan dikonversi agar kompatibilitas dengan pipeline pemrosesan yang digunakan. Konversi dilakukan dalam dua tahap:
   - Folder cv-valid-dev yang berisi 4076 file MP3 dikonversi sepenuhnya ke dalam format WAV.
   - Folder cv-valid-test berjumlah 3995 file MP3 dikonversi ke dalam format WAV dengan penomoran file dimulai dari 4076 agar tidak terjadi duplikasi nama file dengan cv-valid-dev.

Secara keseluruhan, terdapat 8071 file WAV dari folder "common-voice" dengan total ukuran 3294.64 MB yang siap digunakan.

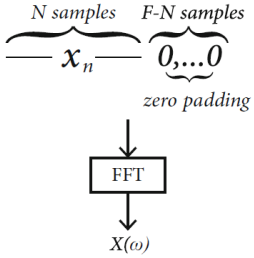
3) *Preprocessing Data*

Tahap preprocessing data berfokus pada persiapan fitur dari data audio sebelum digunakan dalam pelatihan model LSTM. Proses ini dirancang untuk memastikan konsistensi panjang, kualitas fitur, dan kompatibilitas dataset. Gbr. 2 menunjukkan alur preprocessing data.



Gbr. 2 Sub Alur Bagian Preprocessing Data

a. Ekstraksi Fitur MFCC: tahap krusial dalam analisis suara karena fitur-fitur tersebut menyediakan representasi yang lebih kompak dan bermakna dari data audio mentah. Salah satu fitur yang paling umum digunakan dalam analisis suara adalah Mel-Frequency Cepstral Coefficients (MFCC). MFCC berfungsi untuk mewakili spektrum frekuensi dari sinyal audio dalam domain mel-scale, yang lebih mendekati cara telinga manusia merasakan frekuensi [8].
   1. Resampling: Resampling data audio bertujuan agar semua data dapat diproses secara konsisten, mengurangi kemungkinan terjadinya kesalahan atau ketidaksesuaian ketika model menerima input yang beragam. Untuk menyelaraskan kedua dataset yang memiliki sample rate yang berbeda tersebut akan dilakukan resampling menjadi 48 kHz.
   2. Padding Zero: Jika audio memiliki panjang padding kurang dari 262 (panjang maksimum yang ditentukan), maka akan dilakukan padding zero atau penambahan nol di akhir data audio. Ilustrasi dalam Gbr. 3 merupakan proses padding zero pada data $x(n)$ yang memiliki N sampel dengan melakukan augmentasi dengan $F - N$ zero untuk mendapatkan F sampel yang dari $F - point\ Fourier\ transform$ dapat dihitung. Hal ini membuat data dapat ditambahkan sejumlah nol ke penjumlahan dengan tidak mengubah spektrum audio [9, pp. 102–103].



Gbr. 3 Proses Padding Zero

   3. Trim berdasarkan energi: Trimming berbasis energi pada sinyal audio adalah teknik yang bertujuan untuk memangkas bagian-bagian sinyal dengan energi rendah, seperti bagian diam atau noise, untuk mempertahankan hanya bagian-bagian yang relevan secara akustik. Hal tersebut digunakan ketika audio memiliki panjang padding lebih dari 262.
   4. Normalisasi Z-Score: normalisasi MFCC untuk menyelaraskan skala data, menjadikan rata-rata dan variansnya seragam, yang memastikan bahwa fitur memiliki distribusi yang konsisten. Normalisasi yang digunakan adalah normalisasi z-score. Rumus untuk normalisasi z-score dijelaskan pada (1).

$$\boldsymbol{Z} = \frac{(\boldsymbol{X}-\boldsymbol{\mu})}{\boldsymbol{\sigma}} \quad (1)$$

Dimana Z adalah nilai hasil standarisasi, X adalah nilai fitur yang sudah dilakukan manipulasi, μ merupakan rata-rata dari semua nilai fitur, dan σ yakni standar deviasi dari semua nilai fitur.

b. Fitur X dan Label Y: Fitur dari kedua dataset, yaitu Common Voice dan TIMIT, digabungkan dalam satu list dan dikonversi menjadi array numpy bernama X sebagai input model Long Short-Term Memory (LSTM). Proses ini memastikan format data konsisten untuk diolah oleh model. Label y ditentukan berdasarkan asal dataset: label 0 untuk TIMIT (pengucapan benar, native speaker aksen Amerika Serikat) dan label 1 untuk Common Voice (kemungkinan mengandung kesalahan pengucapan, native speaker aksen non-Amerika Serikat). Kedua label digabungkan dan dikonversi menjadi array numpy y sebagai target output. Dimensi akhir X dan y dijelaskan pada Gbr. 4.

```
Shape of X: (14372, 13, 262)
Shape of y: (14372,)
```

Gbr. 4 Bentuk Dimensi Fitur X dan Label y

c. Pembagian Data: Pembagian menjadi tiga bagian, yakni data latih, data validasi, dan data pengujian, sering digunakan dalam kasus di mana evaluasi kinerja model membutuhkan pengujian yang lebih komprehensif [10].

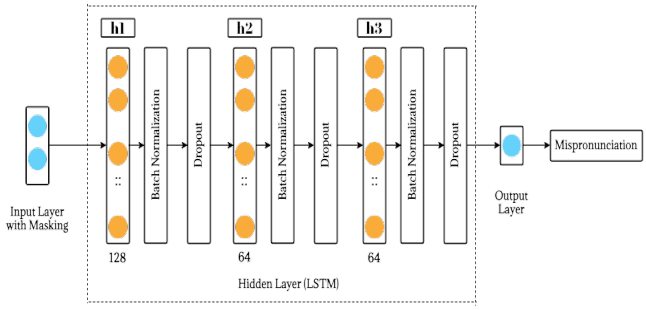
Rasio pembagian yang digunakan adalah 80% untuk data latih, 10% untuk data validasi, dan 10% untuk data pengujian. Pembagian data akan dilakukan menggunakan Teknik K-Fold *Cross-Validation*. Teknik ini membagi dataset menjadi K bagian atau "fold" yang sama atau hampir sama ukurannya. Tujuannya adalah untuk memastikan bahwa model dilatih dan divalidasi pada berbagai subset data yang berbeda, sehingga hasil evaluasi menjadi lebih robust dan generalizable [11]. Jumlah fold (k) ditetapkan 10, yang mana berdasarkan beberapa penelitian yang relevan relevan [12]–[14]. Dikarenakan dalam K-Fold, data dibagi menjadi K bagian yang sama, lalu setiap fold menggunakan bagian yang berbeda untuk validasi, sementara sisa fold digunakan untuk pelatihan dan test. Jadi, di setiap iterasi fold, subset data validasi dan data test bisa berbeda. Dengan K-Fold cross-validation, dataset dapat dibagi secara sistematis dan hasil evaluasi model menjadi lebih akurat dan dapat diandalkan. Tabel 2 menunjukkan hasil pembagian dataset dengan K-Fold.

TABEL 2
HASIL PEMBAGIAN DATASET

Training set	Validation set	Test Set	Total
10.348	1437	2587	14.372

4) *Pemodelan Long Short-Term Memory*

Long Short-Term Memory (LSTM) adalah jenis arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk menangani masalah vanishing gradient dan long-term dependencies dalam data urutan (*sequential data*). Dalam tugas akhir ini, LSTM digunakan untuk mendeteksi kesalahan pengucapan berbasis audio dengan memproses fitur *Mel-Frequency Cepstral Coefficients* (MFCC) yang telah diekstraksi. Gbr. 5 merupakan representasi model LSTM yang akan digunakan untuk mendeteksi kesalahan *pronunciation*. Hal tersebut menggambarkan arsitektur jaringan LSTM secara detail, termasuk layer input dengan masking, hidden layers dengan unit-unit tertentu, batch normalization, dropout, hingga layer output dan deteksi kesalahan pronunciation.



Gbr. 5 Representasi Model LSTM pada Sistem Deteksi Kesalahan Pronunciation

Setelah merancang arsitektur model, selanjutnya adalah mengkompilasi model (*compile model*) dengan menggunakan fungsi kerugian (*loss function*), optimizer, dan metrik yang akan digunakan untuk evaluasi.

1) *Pelatihan Model*

Pelatihan model merupakan proses dimana model akan diperbarui secara berulang menggunakan data pelatihan untuk meminimalkan kerugian (*loss*) dan meningkatkan kinerja dalam memprediksi label yang benar. Parameter yang digunakan adalah 50 epoch dengan batch size 32.

5) *Evaluasi Model*

Evaluasi model adalah tahap penting dalam proses pengembangan model. Ini membantu dalam memahami kinerja model pada data yang belum pernah dilihat selama pelatihan, yaitu data uji. Dengan evaluasi, dapat dinilai seberapa baik model melakukan tugas yang diharapkan, dalam hal ini mendeteksi kesalahan pengucapan berbasis audio. Data uji (test set) adalah subset dari dataset yang tidak digunakan selama pelatihan model. Data ini digunakan secara eksklusif untuk mengevaluasi performa model setelah pelatihan selesai. Dalam konteks ini, data uji memberikan gambaran tentang bagaimana model akan berkinerja dalam situasi dunia nyata. Metode evaluasi menggunakan fungsi 'evaluate' dari Keras untuk menghitung nilai loss dan akurasi model pada data uji. Fungsi ini mengembalikan dua nilai: *loss* (kerugian) dan akurasi model.

D. *Hyperparameter Tuning*

Pemilihan hyperparameter dalam eksperimen ini dilakukan dengan merujuk pada referensi dari [15, pp. 59–66] serta praktik umum dalam deep learning. Buku tersebut menyebutkan beberapa hyperparameter default, seperti batch size 32, learning rate 1e-3, dan optimizer Adam. Berdasarkan referenai ini, eksperimen dilakukan dengan memperluas variasi hyperparameter, meliputi batch size 16 (di bawah default) dan 64 (di atas default), learning rate 1e-4 (lebih kecil dari default), serta jumlah epoch 50, 75, dan 100 untuk memastikan pelatihan model berlangsung cukup lama agar dapat mencapai konvergensi.

Evaluasi performa model dilakukan menggunakan metrik akurasi, presisi, recall dan F1-score untuk memberikan gambaran menyeluruh tentang kemampuan model dalam mendeteksi kesalahan pengucapan. Variasi dan pilihan hyperparameter ini dirancang untuk mengoptimalkan performa pada dataset TIMIT dan Common Voice. Berikut adalah penjelasan masing-masing hyperparameter yang digunakan:

1) *Optimizer*

Optimizer merupakan algoritam yang digunakan untuk mengoptimalkan fungsi kerugian (*loss function*) dengan memperbarui parameter model selama proses pelatihan. Optimizer yang digunakan adalah Adam, salah satu optimizer yang paling populer dan sering digunakan. Adam menggunakan kombinasi dari metode momentum dan perubahan kecepatan belajar adaptif (*adaptive learning rate*) untuk mengoptimalkan parameter model.

2) *Batch Size*

Ukuran batch size menenutkan jumlah sampel data yang akan diproses oleh model pada setiap iterasi pelatihan. Default batch size 32 diambil dari referensi, sementara 16 dan 64 dipilih untuk mengeksplorasi dampaknya terhadap performa model.

3) *Jumlah epoch*

Jumlah epoch adalah hyperparameter yang menentukan berapa kali keseluruhan dataset akan dilewati oleh model LSTM selama proses pelatihan. Meskipun referensi menyebutkan 4 epoch sebagai default dalam konteks R, nilai ini terlalu kecil untuk model LSTM yang membutuhkan pelatihan lebih panjang agar dapat menangkap pola temporal dalam data. Oleh karena itu, jumlah epoch 50, 75, dan 100 dipilih untuk mencari keseimbangan antara waktu pelatihan dan performa.

4) *Learning Rate*

Learning rate mengontrol seberapa besar perubahan yang dilakukan pada parameter model setiap kali pembaharuan dilakukan selama proses pelatihan. Default 0.001 atau 10^{-3} $(1e-3)$ adalah nilai yang umum digunakan untuk optimizer seperti Adam. Nilai 0.0001 atau 10^{-4} $(1e-4)$ dieksplorasi untuk menghindari overshooting dan memberikan konvergensi yang lebih stabil, terutama pada model LSTM yang cenderung lebih kompleks.

5) *Metrik Evaluasi*

Evaluasi model LSTM akan melibatkan sejumlah metrik evaluasi yang meliputi akurasi (accuracy), presisi (precision), recall, dan F1-score. Confusion Matrix akan digunakan untuk menggambarkan kinerja LSTM dengan mempertimbangkan empat kelas: True Positive, False Positive, True Negative, dan False Negative. Keempat kelas klasifikasi ini akan menjadi dasar perhitungan untuk setiap metrik kinerja yang disebutkan sebelumnya. Detail tentang penggunaan metrik-metrik tersebut akan dijelaskan lebih lanjut dalam Tabel 3.

TABEL 3
CONFUSION MATRIX

Predicted Values	Actual Values: Positive	Actual Values: Negative
Positive	TP (True Positive)	FP (False Positive)
Negative	FN (False Negative)	TN (True Negative)

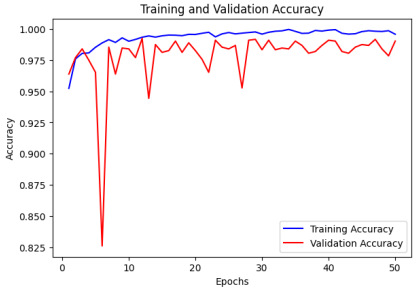
III. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil dari berbagai eksperimen yang dilakukan untuk mengevaluasi performa model Long Short-Term Memory (LSTM) dalam mendeteksi kesalahan *pronunciation* berbasis audio. Penelitian diawali dengan pelatihan model menggunakan parameter *default* untuk menentukan baseline performa model. Selanjutnya, dilakukan eksperimen lanjutan dengan optimizer Adam yang memvariasikan batch size, jumlah epoch, dan learning rate. Setiap variasi hyperparameter tersebut akan diukur dengan empat metrik evaluasi yakni akurasi, presisi, recall, dan F1-score untuk memperoleh hasil yang lebih optimal.

A. *Model Awal dengan Parameter Default*

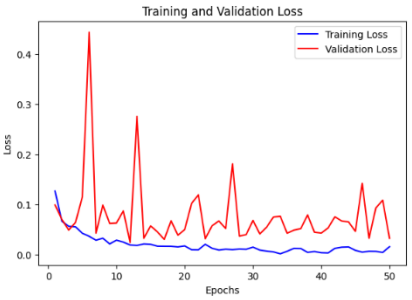
Sebagai baseline, model LSTM dilatih menggunakan parameter default, yaitu batch size 32, jumlah epoch 50, learning rate 1e-3, optimizer Adam, dan metrik evaluasi akurasi. Evaluasi ini meliputi proses preprocessing data, pembagian dataset, pembangunan model, serta evaluasi terhadap performa model menggunakan data test.

Berdasarkan grafik akurasi selama proses pelatihan serta validasi pada Gbr. 6 Dapat dilihat bahwa model menunjukkan tren akurasi pelatihan yang cepat meningkat dan stabil pada tingkat yang tinggi, mendekati nilai 1.00. Akurasi validasi juga cukup baik, meskipun terdapat fluktuasi yang signifikan di beberapa epoch awal sebelum akhirnya stabil mendekati akurasi pelatihan. Fluktuasi ini menunjukkan bahwa model memerlukan waktu untuk menyesuaikan diri pada validasi yang mungkin memiliki variasi yang lebih besar dibandingkan data latih.



Gbr. 6 Grafik Akurasi pada Model Awal

Pada grafik loss pada Gbr. 7, terlihat bahwa loss pelatihan berangsur menurun dan mencapai tingkat yang stabil, sementara loss validasi mengalami fluktuasi yang lebih besar, terutama di epoch awal. Hal ini menandakan bahwa model mungkin mengalami overfitting, dimana model berhasil mempelajari pola pada data latih dengan sangat baik namun belum mampu mempertahankan performa yang konsisten pada data validasi.



Gbr. 7 Grafik Loss pada Model Awal

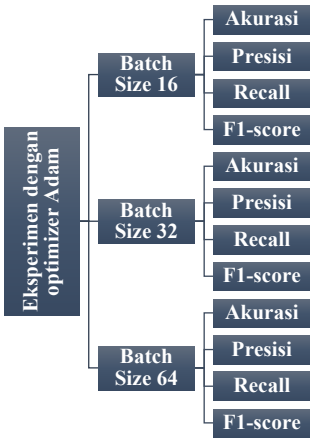
Hasil pelatihan menunjukkan bahwa model mencapai akurasi validasi sebesar 0,987 atau 98,7%. Pencapaian ini menunjukkan bahwa model memiliki performa yang baik dalam mendeteksi kesalahan pengucapan, mencerminkan kemampuan model dalam mempelajari pola-pola penting dari data latih. Namun, fluktuasi yang terjadi pada loss validasi, terutama di epoch awal, memberikan indikasi bahwa model masih memerlukan tuning hyperparameter lebih lanjut untuk meningkatkan stabilitas dan kemampuan generalisasinya pada data yang belum pernah dilihat sebelumnya.

Tahap selanjutnya akan difokuskan pada eksperimen menggunakan optimizer Adam, di mana variasi batch size,

jumlah epoch, dan learning rate akan diatur secara sistematis. Pendekatan ini bertujuan untuk menemukan kombinasi pengaturan terbaik yang dapat mengurangi fluktuasi, meningkatkan akurasi validasi, serta memastikan performa model tetap konsisten pada berbagai kondisi data.

B. *Pendahuluan Eksperimen*

Eksperimen melibatkan berbagai kombinasi batch size dan metrik evaluasi dengan masing-masing kombinasi akan diuji dengan jumlah epoch dan learning rate. Struktur eksperimen dapat dilihat pada Gbr. 8, yang memberikan gambaran alur pengujian untuk setiap konfigurasi.



Gbr. 8 Skenario Eksperimen Hyperparameter Tuning

Alur pengujian pada Gbr. 8 diuraikan lebih lanjut dalam penjelasan berikut.

a. Eksperimen dilakukan dengan tiga ukuran bacth size yaitu 16, 32, dan 64.
b. Pada setiap batch, digunakan empat metrik evaluasi untuk analisis performa model yaitu akurasi, presisi, recall, dan F1-score.
c. Setiap kombinasi batch size dan metrik evaluasi diuji dengan jumlah epoch 50, 75, dan 100 serta learning rate sebesar 10^{-3} dan 10^{-4}.

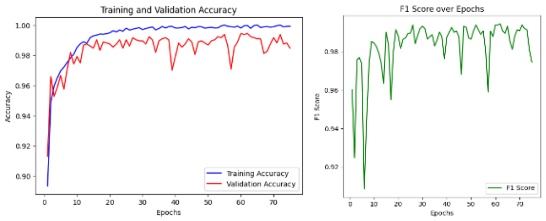
Sebagai ilustrasi, pada batch size 16, eksperimen dilakukan dengan empat metrik evaluasi, tiga jumlah epoch, dan dua learning rate, yang menghasilkan 24 kombinasi pengujian (4 metrik * 3 epoch * 2 learning rate). Pendekatan serupa diterapkan untuk ukuran batch size 32 dan 64, yang masing-masing juga menghasilkan 24 kombinasi. Dengan demikian, total eksperimen keseluruhan adalah 72 kombinasi pengujian.

C. *Hasil Eksperimen dan Analisis*

1) *Hasil Analisis berdasarkan Batch Size*

Berdasarkan hasil eksperimen yang dilakukan dengan tiga variasi batch size (16, 32, 64), dapat dilihat perbandingan kinerja model pada masing-masing skenario. Setiap scenario diuji dengan epoch 50, 75, dan 100 serta dua learning rate (1e-3 dan 1e-4). Berdasarkan analisis, berikut adalah hasil dengan skenario terbaik dan terburuk untuk masing-masing skenario:

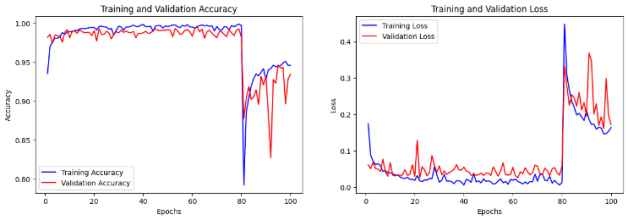
a. Batch size 16
   - Skenario terbaik, akurasi yang tinggi dengan hasil terbaik pada learning rate 10^{-4}, yaitu mencapai 99,4% pada epoch 75. Model ini juga mampu mencapai F1-score yang sangat baik (99,8%) dengan fluktuasi yang relatif kecil. Grafik yang menunjukkan hasil di setiap perkembangan epochnya ketika pelatihan data dapat dilihat pada Gbr. 9.



(a) Akurasi (b) F1-score

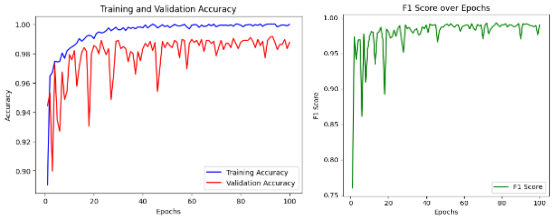
Gbr. 9 Grafik Akurasi dan F1-score di skenario batch 16 dengan epoch 75 dan lr 1e-4

   - Skenario terburuk, memiliki fluktuasi yang cukup signifikan pada akurasi dengan epoch 100 dan learning rate 10^{-3}. Terjadi penurunan yang signifikan pada akurasi pelatihan dan validasi, mengindikasikan potensi overfitting yang lebih jelas dengan hasil akhirnya jika diuji dengan data uji sebesar 91,0%. Untuk visualisasi grafik dapat dilihat pada Gbr. 10.



Gbr. 10 Grafik akurasi dan loss di skenario batch 16 dengan epoch 100 dan lr 1e-3

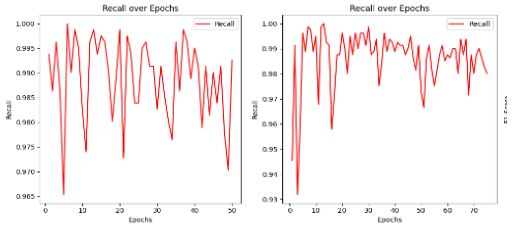
b. Batch size 32
   - Skenario terbaik, pada epoch 100 dengan learning rate 10^{-4} dengan akurasi yang tinggi (98,3%) dan F1-score tertinggi (99,4%). Performa yang ditunjukkan pada Gbr. 11 baik dengan sedikit fluktuasi di awal pelatihan menunjukkan bahwa model memerlukan waktu untuk menyesuaikan diri dengan variasi data validasi.



(a) Akurasi (b) F1-score

Gbr. 11 Grafik Akurasi dan F1-score di skenario batch 32 dengan epoch 100 dan lr 1e-4

   - Skenario terburuk, menunjukkan fluktuasi yang cukup besar pada recall di epoch 50 dan 75 dengan learning rate 10^{-3} yang dapat dilihat pada Gbr. 12.

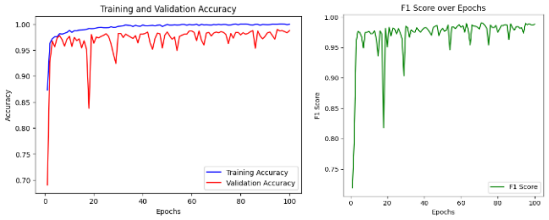


(a) Epoch 50 (b) Epoch 75

Gbr. 12 Grafik Recall di skenario batch 32 dengan lr 1e-3

c. Batch size 64

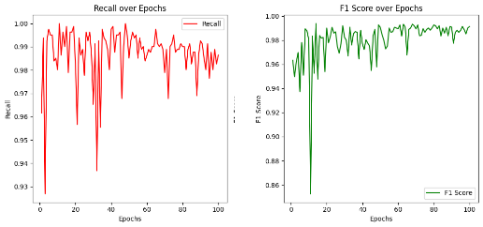
- Skenario terbaik, Performa terbaik ditemukan pada epoch 100 dan learning rate 10^{-4} dengan akurasi dan F1-score tertinggi, di mana stabilitas lebih penting daripada kecepatan konvergensi. Hasil evaluasi dengan data uji mencapai 99,7% dan 98,2% untuk masing-masing akurasi dan F1-score dan grafik dapat dilihat pada Gbr. 13.



(a) Akurasi (b) F1-score

Gbr. 13 Grafik Akurasi dan F1-score di skenario batch 64 dengan epoch 100 dan lr 1e-4

- Skenario terburuk, Performa model dengan learning rate 10^{-3} menunjukkan fluktuasi yang lebih besar pada recall dan F1-score. Recall pada Gbr. 14 (a) menunjukkan fluktuasi yang cukup besar, namun ada peningkatan cepat yang diikuti oleh penurunan tajam pada beberapa epoch. Fluktuasi signifikan terlihat pada F1-score terlihat pada Gbr. 14 (b), mencerminkan ketidakstabilan keseimbangan antara presisi dan recall.



(a) Recall (b) F1-score

Gbr. 14 Grafik Recall dan F1-score di skenario batch 64 dengan lr 1e-3

2) *Perbandingan rata-rata hasil evaluasi model pada data testing*

Dari seluruh eksperimen yang dilakukan, Tabel 4 merupakan hasil rata-rata evaluasi model pada data testing untuk masing-masing metrik evaluasi berdasarkan batch size yang digunakan.

TABEL 4
RATA-RATA HASIL EVALUASI MODEL PADA DATA TESTING

Batch Size	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
16	96,7	100,0	97,0	99,3
32	98,5	100,0	98,9	99,4
64	97,6	100,0	98,0	99,0

Dari Tabel 4, terlihat bahwa batch size 32 memberikan hasil terbaik secara keseluruhan dengan akurasi 98.5%, recall 98.9%, dan F1-score 99.4%. Sedangkan batch size 16 menunjukkan hasil yang sangat baik dengan akurat 96.7% dan performa tinggi pada recall serta F1-score, namun sedikit lebih rendah dibandingkan dengan batch size 32.

D. *Pembahasan*

Presisi memang selalu mencapai 100% di evaluasi model pada data testing di semua skenario eksperimen, sehingga tidak dapat dijadikan acuan untuk menentukan skenario terbaik atau terburuk. Ini menunjukkan bahwa model sangat akurat dalam mengidentifikasi kesalahan pengucapan, tanpa menghasilkan false positives. Namun, karena presisi yang konsisten ini tidak memberikan informasi lebih lanjut mengenai kinerja model, fokus analisis harus lebih pada akurasi, recall, dan F1-score, yang menunjukkan perbedaan lebih besar di antara skenario-skenario eksperimen.

Dari hasil eksperimen yang telah dilakukan pada batch size 16, 32, dan 64, serta menggunakan tiga jumlah epoch (50, 75, dan 100) dan dua nilai learning rate (1e-3 dan 1e-4), hasil keseluruhan yang dapat diambil mengenai kombinasi terbaik dan terburuk berdasarkan hasil metrik evaluasi.

Kombinasi terbaik ditemukan pada batch size 32 dengan learning rate 10^{-4} pada epoch 100 dengan metrik akurasi dan metrik F1-score. Pada kombinasi ini, model menunjukkan performa sangat baik dengan hasil akurasi mencapai 98,3% dan F1-score 99,4%. Hasil ini menunjukkan keseimbangan yang sangat baik dan stabil.

Kombinasi yang paling tidak optimal terjadi pada batch size 16 dengan learning rate 10^{-3}, dimana akurasi dan loss mengalami fluktuasi yang lebih besar, meskipun akurasi masih terbilang cukup tinggi (91.0%) jika model dievaluasi dengan data uji. Fluktuasi yang signifikan pada metrik akurasi dan loss menunjukkan tantangan model sehingga kombinasi ini kurang optimal meskipun akurasi tetap tinggi.

IV. KESIMPULAN

Model Long Short-Term Memory (LSTM) telah dikembangkan untuk mendeteksi kesalahan *pronunciation* Bahasa Inggris berbasis audio, dengan fokus pada penggunaan optimizer Adam. Pada model awal, yang dilatih dengan parameter default (batch sizer 32, epoch 50, learning rate 10^{-3}), model mencapai akurasi validasi sebesar 98,7%. Meskipun hasil ini menunjukkan performa yang baik, fluktuasi pada loss validasi mengindikasikan perlunya tuning hyperparameter untuk meningkatkan stabilitas dan kemampuan generalisasi model.

Eksperimen dilakukan dengan tiga ukuran batch size (16, 32, dan 64) dan empat metrik evaluasi (akurasi, presisi, recall, dan F1-score). Setiap kombinasi batch size dan metrik diuji dengan tiga jumlah epoch (50, 75, 100) dan dua nilai learning rate (10^{-3} dan 10^{-4}), menghasilkan total 72 kombinasi eksperimen.

Hasil evaluasi model pada data testing menunjukkan presisi yang selalu mencapai 100% pada semua skenario. Hal tersebut menunjukkan meskipun selalu di angka sempurna dan grafik masih menunjukkan naik turun, model tidak menghasilkan *false positives.* Hal ini tidak cukup untuk menentukan skenario terbaik atau terburuk sehingga analisis lebih difokuskan pada metrik akurasi, recall, dan F1-score, yang memberikan gambaran lebih jelas mengenai performa model di berbagai skenario eksperimen.

Evaluasi kinerja model menunjukkan skenario dengan hasil terbaik dan terburuk sebagai berikut:

- Skenario terbaik: batch size 32, learning rate 10^{-4}, epoch 100 yang menghasilkan akurasi 94% dan F1-score 95%.
- Skenario terburuk: batch size 16, learning rate 10^{-3}, epoch 100, dimana meskipun akurasi 91%, fluktuasi lebih besar.

Secara keseluruhan, model LSTM dengan optimizer Adam efektif dalam mendeteksi kesalahan pronunciation Bahasa Inggris dengan hasil terbaik menggunakan kombinasi hyperparameter batch size 32, epoch 100, dan learning rate 10^{-4} dengan hasil sebesar 94%. Hasil ini memberikan gambaran stabilitas dan keseimbangan yang optimal dalam performa model.

V. Saran

Berdasarkan hasil penelitian, beberapa saran dapat dijadikan acuan untuk pengembangan lebih lanjut dalam bidang deteksi kesalahan pronunciation berbasis audio. Pertama, penggunaan dataset yang lebih beragam dengan label yang ditentukan oleh ahli dapat meningkatkan keakuratan pelabelan kesalahan, sehingga model lebih memahami nuansa pengucapan yang sulit terdeteksi secara otomatis. Kolaborasi dengan ahli linguistik atau pengajar bahasa Inggris disarankan untuk memperkaya dataset. Kedua, meskipun LSTM terbukti efektif, eksplorasi metode lain yang lebih efisien dalam menangani hubungan jangka panjang pada data audio, seperti model deep learning terbaru, dapat memberikan hasil lebih optimal. Ketiga, pengembangan sistem dengan fitur umpan balik berupa saran korektif otomatis, seperti transkrip pengucapan yang benar atau panduan suara, dapat meningkatkan manfaat aplikasi ini dalam pembelajaran bahasa berbasis teknologi. Dengan saran-saran ini, penelitian mendatang diharapkan mampu meningkatkan performa dan kegunaan model secara lebih luas.

Ucapan Terima Kasih

Puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan hidayah-Nya yang telah mengiringi proses penelitian ini hingga selesai. Penulis juga menyampaikan rasa terima kasih yang mendalam kepada orang tua atas doa dan dukungan moral maupun material, kepada dosen pembimbing yang telah memberikan arahan dan masukan, serta kepada rekan-rekan dan semua pihak yang telah memberikan kontribusi dalam penelitian ini.

Referensi

[1] T. Plailek and A. M. Essien, “Pronunciation Problems and Factors Affecting English Pronunciation of EFL Students,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 12, pp. 2026–2033, 2021.

[2] R. Nordquist, “Expanding Circle: Definition and Examples.” Accessed: Mar. 20, 2024. [Online]. Available: https://www.thoughtco.com/expanding-circle-english-language-1690619

[3] N. R. Kobilova, “Importance of Pronunciation in English Language Communication,” *Acad. Res. Educ. Sci.*, vol. 3, no. 6, p. 1, 2022, [Online]. Available: https://t.me/ares_uz

[4] D. R. Pratiwi and L. M. Indrayani, “Pronunciation Error on English Diphthongs Made by EFL Students,” *Teknosastik*, vol. 19, no. 1, p. 24, 2021, doi: 10.33365/ts.v19i1.486.

[5] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, “Deep Learning with Long Short-Term Memory for Time Series Prediction,” *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, 2019, doi: 10.1109/MCOM.2019.1800155.

[6] “TIMIT Acoustic-Phonetic Continuous Speech Corpus - Linguistic Data Consortium.” Accessed: Nov. 08, 2024. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[7] “Common Voice.” Accessed: Mar. 14, 2024. [Online]. Available: https://www.kaggle.com/datasets/mozillaorg/common-voice/data

[8] “Mel Frequency Cepstral Coefficient - an overview | ScienceDirect Topics.” Accessed: Jun. 12, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/mel-frequency-cepstral-coefficient

[9] M. G. Christensen, *Introduction to Audio Processing*. 2019. doi: 10.1007/978-3-030-11781-8.

[10] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, “A critical look at the current train/test split in machine learning,” 2021, [Online]. Available: http://arxiv.org/abs/2106.04525

[11] “Fold Cross Validation - an overview | ScienceDirect Topics.” Accessed: Jun. 12, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/fold-cross-validation

[12] S. Akhtar *et al.*, “Improving mispronunciation detection of arabic words for non-native learners using deep convolutional neural network features,” *Electron.*, vol. 9, no. 6, pp. 1–17, 2020, doi: 10.3390/electronics9060963.

[13] K. López-de-Ipiña *et al.*, “On the analysis of speech and disfluencies for automatic detection of Mild Cognitive Impairment,” *Neural Comput. Appl.*, vol. 32, no. 20, pp. 15761–15769, 2020, doi: 10.1007/s00521-018-3494-1.

[14] Y. Sato and Y. Bao, “Identification of 3D Lip Shape during Japanese Vowel Pronunciation Using Deep Learning,” *Appl. Sci.*, vol. 12, no. 9, 2022, doi: 10.3390/app12094632.

[15] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, *Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide*. 2023. doi: 10.1007/978-981-19-5170-1.