

Pengembangan Aplikasi Simulasi Sesi *Speaking* Tes *IELTS* Berbasis *Android*

1st Hafid Ikhsan Arifin
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
hafidikhshan@student.telkom
university.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
setiacasie@telkomuniversity.
ac.id

3rd Astri Novianty
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
astrinov@telkomuniversity.a
c.id

4th Umar Ali Ahmad
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
umar@telkomuniversity.ac.i
d

Abstrak—*International English Language Testing System (IELTS)* adalah salah satu tes kemampuan bahasa Inggris yang diselenggarakan oleh Universitas *Cambridge, British Council*, dan *IDP Education Australia*. Dalam tes *IELTS* terdiri dari 4 sesi yaitu *Reading, Writing, Listening*, dan *Speaking*. Menurut statistik pada tahun 2022, sesi *Speaking* menempati posisi kedua dengan nilai terendah. Hal ini dikarenakan persiapan yang dilakukan untuk sesi *Speaking* ini cukup kompleks dan memakan waktu yang cukup lama. Saat ini terdapat simulasi yang diadakan oleh pihak resmi penyelenggara tes *IELTS*. Namun proses simulasi ini berbayar dan proses penilaian cukup lama. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah mesin yang dapat mengevaluasi simulasi sesi *Speaking* tes *IELTS* secara langsung dan akurat berdasarkan 4 matriks evaluasi yaitu *Fluency, Lexical, Grammar*, dan *Pronunciation*. Akan ada 4 model yang akan dikembangkan dan di setiap model akan merepresentasikan satu matriks evaluasi. Setiap model telah diuji dengan hasil matriks evaluasi pada model *Fluency* didapat nilai akurasi 99%, model *Lexical* didapat nilai akurasi 62%, model *Grammar* didapat nilai *loss* 0.562, dan model *Pronunciation* didapat nilai akurasi 82%. Hasil integrasi sistem dengan platform *Cloud* dan platform *Android* ini diharapkan dapat digunakan sebagai sarana pelatihan simulasi sesi *Speaking* tes *IELTS* yang dapat melakukan penilaian secara cepat dan murah.

Kata kunci— *IELTS, Speaking, Fluency, Lexical, Grammar, Pronunciation, Cloud, Android*

I. PENDAHULUAN

International English Language Testing System (IELTS) adalah sebuah tes kemampuan bahasa Inggris yang diselenggarakan oleh Universitas *Cambridge, British Council*, dan *IDP Education Australia* dan menjadi salah satu prasyarat wajib untuk seseorang yang ingin bekerja, belajar, atau bermigrasi ke negara yang menggunakan bahasa Inggris sebagai bahasa utama [1]. Dengan banyaknya manfaat yang diperoleh saat mendapatkan sertifikasi ini, maka banyak orang yang ingin mengikuti tes ini baik untuk kebutuhan akademik, umum, atau hanya sekedar ingin mendapatkan sertifikat ini. Tes ini terdiri dari 4 sesi yaitu *Reading, Writing, Listening*, dan *Speaking* [2]. Menurut statistik resmi yang dikeluarkan oleh pihak penyelenggara tes di tahun 2022, sesi *Speaking* menempati posisi kedua dengan nilai terendah [3]. Hal ini dikarenakan sesi *speaking* memerlukan proses pelatihan yang melelahkan dan kompleks karena perlunya untuk merekam percakapan diri sendiri ataupun mencari teman bicara untuk mendapatkan *feedback* dari pengucapan yang telah diucap. Terdapat beberapa matriks evaluasi pada sesi *Speaking* tes *IELTS* yaitu *Fluency, Lexical, Grammar*,

dan *Pronunciation* dan di setiap matriks evaluasi memiliki kriteria masing di setiap *band* [4].

Untuk mempersiapkan sebelum mengikuti tes *IELTS*, terdapat simulasi atau latihan yang diadakan oleh pihak resmi penyelenggara tes *IELTS*. Simulasi ini mencakup format tes dengan berbagai contoh pertanyaan dan jawaban tes, untuk membantu peserta dalam mempersiapkan tes *IELTS* yang sebenarnya [5]. Biaya untuk melakukan simulasi di website resmi *IELTS* berkisar sekitar Rp700.000 [6] dan hasil penilaian berupa *feedback* report akan diberikan lima hari setelah pengambilan simulasi tes [7]. Dari kedua hal tersebut, banyak peserta yang mengurungkan niatnya untuk mengambil simulasi tes dikarenakan biaya yang cukup mahal dan proses menunggu hasil penilaian cukup lama.

Dari permasalahan yang didapat, maka akan dikembangkan sebuah sistem yang dapat melakukan penilaian simulasi sesi *Speaking* tes *IELTS* secara langsung dan akurat. Untuk mengembangkan sistem ini akan digunakan sebuah sistem *Artificial Intelligence* dengan menggunakan teknologi *Deep Learning* dengan algoritma *RNN-LSTM* dan *Transformers* menggunakan model *DistilBERT, T5, Wav2Vec2* untuk mengevaluasi simulasi sesi *Speaking* tes *IELTS* berdasarkan 4 matriks evaluasi yaitu *Fluency, Lexical, Grammar*, dan *Pronunciation*. Akan ada 4 model yang akan dikembangkan di sistem ini dan di setiap model akan merepresentasikan satu matriks evaluasi tes *IELTS* sesi *Speaking*. Agar sistem ini dapat digunakan oleh pengguna dengan mudah, sistem ini akan terintegrasi dengan 2 sistem lainnya yaitu sistem *Cloud* dengan menggunakan *Flask* dan *GCP* untuk mengakses model dan melakukan penilaian, sistem *Android* yang juga terintegrasi dengan sistem *database* dan *UI/UX* sebagai *front-end* dari aplikasi. Hasil integrasi sistem keseluruhan ini akan menjadi sebuah aplikasi simulasi sesi *Speaking* tes *IELTS* berbasis *Android* yang diharapkan dapat digunakan sebagai sarana pelatihan simulasi sesi *Speaking* tes *IELTS* bagi pada calon peserta tes *IELTS* yang dapat melakukan penilaian secara cepat dan murah.

II. KAJIAN TEORI

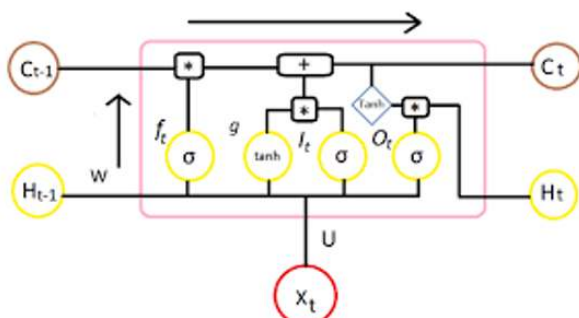
A. *IELTS*

International English Language Testing System (IELTS) adalah sebuah tes kemampuan bahasa Inggris internasional terpopuler di dunia yang biasa digunakan untuk keperluan bekerja, belajar, atau bermigrasi ke negara yang menggunakan bahasa Inggris sebagai bahasa utama [8]. Tes *IELTS* diselenggarakan dan dikembangkan oleh Universitas *Cambridge, British Council*, dan *IDP Education Australia*

[1]. Tes *IELTS* dapat digunakan untuk 2 tujuan yaitu untuk kebutuhan akademik atau kebutuhan umum. Tes *IELTS* ini terdiri dari 4 sesi yaitu *Reading*, *Writing*, *Listening*, dan *Speaking* [2]. Salah satu sesi yang memiliki nilai terendah yaitu sesi *Speaking*. Dalam sesi *Speaking*, terdapat 4 matriks evaluasi yaitu *Fluency*, *Lexical*, *Grammar*, dan *Pronunciation*.

B. RNN-LSTM

Long short-term memory (LSTM) adalah salah satu jenis algoritma *Recurrent Neural Network* (RNN) dengan memori negara dan struktur sel *multilayer* [9]. Jenis RNN ini biasanya digunakan untuk mempelajari dan mengingat ketergantungan pada pola jangka panjang. Dengan menggunakan teknologi ini akan membantu untuk mengingat semua informasi yang ada di masa lalu dari periode tertentu. LSTM juga dapat berfungsi untuk menyimpan informasi dari waktu ke waktu. Informasi yang akan berguna untuk keperluan *time series* karena dapat mengingat *input* sebelumnya. Sel LSTM terdiri dari satu lapisan *input*, satu lapisan *output*, dan satu lapisan tersembunyi *self-connected*. Berikut adalah arsitektur layer LSTM.



Gambar 1. Arsitektur layer LSTM

Persamaan sel LSTM diperoleh sebagai berikut:

$$F_t = \sigma (X_t \times U_f + H_{t-1} \times W_f) \quad (1)$$

$$\hat{C}_t = \tanh (X_t \times U_c + H_{t-1} \times W_c) \quad (2)$$

$$I_t = \sigma (X_t \times U_i + H_{t-1} \times W) \quad (3)$$

$$O_t = \sigma (X_t \times U_o + H_{t-1} \times W_o) \quad (4)$$

$$C_t = F_t \times C_{t-1} + I_t \times \hat{C}_t \quad (5)$$

$$H_t = (O_t \times \tanh (C_t)) \quad (6)$$

Dimana X_t adalah *input*, H_{t-1} adalah *output* sel sebelumnya, C_{t-1} adalah memori sel sebelumnya, H_t adalah sel *output*, C_t adalah memori sel, dan W adalah bobot. Terdapat beberapa *library Python* yang memiliki algoritma RNN-LSTM seperti *Pytorch* dan *TensorFlow*.

C. DistilBERT

DistilBERT adalah salah satu model pra-pelatihan versi BERT dengan tujuan umum yang memiliki kemampuan 40% lebih kecil, 60% lebih cepat, yang mempertahankan 97% kemampuan pemahaman bahasa. *DistilBERT* menggunakan metode *Knowledge distillation* dimana sebuah teknik kompresi model kompak (*student*) dilatih untuk mereproduksi perilaku model yang lebih besar (*teacher*) atau

ansambel model. *Student* dilatih dengan *distillation loss* atas probabilitas target lunak *teacher* [10].

$$L_{ce} = \sum_i t_i \times \log(s_i) \quad (7)$$

Dimana t_i adalah probabilitas yang diperkirakan oleh *teacher* dan s_i adalah probabilitas yang diperkirakan oleh *student*. Model pra-pelatihan *DistilBERT* dapat diakses menggunakan *Hugging Face Transformers API library*. Pada *library Transformers* menyediakan beberapa model pra-pelatihan yang dapat digunakan untuk beberapa tugas tertentu.

D. T5

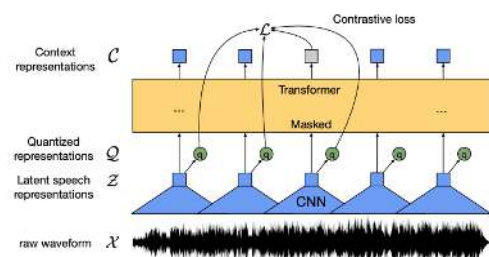
Text-To-Text Transfer Transformer atau biasa dikenal dengan *T5* adalah salah satu model *transfer learning* NLP yang dikembangkan oleh peneliti *Google* di mana *input* dan *output* selalu berupa *string* teks. Dengan model *T5* memungkinkan untuk menggunakan model, fungsi *loss*, dan hyperparameter yang sama pada tugas NLP apa pun, termasuk *machine translation*, *document summarization*, *question answering*, dan tugas klasifikasi (misalnya, *sentiment analysis*) [11]. Model pra-pelatihan *T5* menyediakan 5 jenis model yaitu *small*, *base*, *large*, 3B, dan 11B. Model pra-pelatihan *T5* dapat diakses menggunakan *Hugging Face Transformers API library*.

E. Happy Transformers

Happy Transformers adalah salah satu framework yang dapat memudahkan proses *fine-tune* dan melakukan inferensi dengan model NLP *Transformer*. *Happy Transformers* memiliki beberapa fitur yang dapat digunakan seperti untuk *text generation*, *question answering*, *word prediction*, *text-to-text*, *next sentence prediction*, klasifikasi teks dan klasifikasi token [12].

F. Wav2Vec2

Wav2Vec2 adalah sebuah *framework* untuk *self-supervised learning* sebuah representasi ucapan yang dikembangkan oleh peneliti di *Facebook* dimana dapat menutupi representasi laten dari bentuk gelombang mentah dan menyelesaikan tugas kontrasif atas representasi ucapan terkuantisasi. Model *Wav2Vec2* akan melakukan *encodes* audio ucapan melalui *multi-layer CNN* dan kemudian menutupi rentang representasi ucapan *laten* yang dihasilkan. Representasi *laten* selanjutnya masuk ke jaringan *Transformer* untuk membangun representasi kontekstual dan model dilatih melalui tugas kontrasif di mana *laten* sebenarnya harus dibedakan dari distraktor [13].



Gambar 2. Arsitektur model Wav2Vec2

Untuk melakukan *fine-tune* model *Wav2Vec2* dapat menggunakan fungsi *CTC loss*. Model pra-pelatihan *Wav2Vec2* dapat digunakan untuk *Speech Recognition* tanpa harus melakukan *fine-tune*. Selain untuk *Speech Recognition*, model *Wav2Vec2* juga dapat melakukan klasifikasi audio. Model pra-pelatihan *Wav2Vec2* menyediakan 2 jenis model yaitu *base* dan *large*. Model pra-pelatihan *Wav2Vec2* dapat diakses menggunakan *Hugging Face Transformers API library*.

G. Confusion Matrix

Confusion matrix adalah metode untuk menganalisis seberapa baik *classifier* digunakan untuk mengenali *tuple* dari kelas yang berbeda [14].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (8)$$

Akurasi adalah matriks untuk model klasifikasi yang mengukur jumlah prediksi yang benar sebagai persentase dari jumlah total prediksi yang dibuat [15].

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

Recall berfungsi untuk menunjukkan tingkat keberhasilan atau kekhususan untuk mengetahui informasi dengan benar tentang data suatu kelas [15].

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Fungsi *precision* adalah kesesuaian sistem antara permintaan informasi dan jawaban atas permintaan tersebut [14].

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

Fungsi untuk *f1-score* untuk melihat apakah model algoritma bekerja dengan baik [14]. Deskripsi rumus:

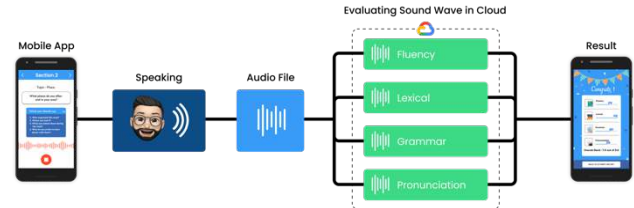
TP : *True Positive*
TN : *True Negative*
FP : *False Positive*
FN : *False Negative*

H. Loss

Salah satu matriks evaluasi yang sering digunakan pada pelatihan model *Machine Learning* adalah nilai *loss*. Nilai *loss* adalah sebuah *penalty* untuk prediksi yang buruk. Dengan kata lain *loss* adalah angka yang menunjukkan seberapa buruk prediksi model pada satu contoh. Jika prediksi modelnya sempurna, nilai *loss* akan nol dan jika tidak, nilai *loss* lebih besar [16]. Sehingga nilai *loss* terbaik adalah nilai *loss* dengan angka mendekati nol.

III. METODE

A. Desain Sistem



Gambar 3. Desain sistem

Aplikasi simulasi tes *IELTS* sesi *speaking* ini akan dikembangkan menggunakan tiga teknologi utama, yaitu *Android*, *Artificial Intelligence*, dan *Cloud*. Proses penilaian akan dilakukan untuk setiap pertanyaan. Setelah aplikasi menampilkan dan membacakan pertanyaan kepada pengguna, kemudian untuk setiap jawaban pengguna, aplikasi akan langsung membuat berkas audio yang berisi jawaban pengguna dan mengirimkannya ke *Cloud* di *GCP* untuk dievaluasi. Sebelum mengevaluasi berkas audio, berkas audio akan disimpan di *Cloudinary*. Berkas audio yang disimpan di *Cloudinary* dapat digunakan untuk menilai tes *IELTS* sesi *speaking*. Dalam proses evaluasi, aplikasi akan mengikuti matriks evaluasi untuk sesi *speaking* tes *IELTS*, yaitu *Fluency*, *Grammar*, *Lexical*, dan *Pronunciation*. Sebelum dilakukan evaluasi, berkas audio akan diperiksa apakah terdapat suara dalam berkas audio menggunakan *VAD* dan pemeriksaan bahasa menggunakan *Whisper* dan *Langdetect*.

Untuk matriks evaluasi *Fluency*, akan digunakan model *RNN-LSTM* untuk mengklasifikasikan berkas audio ke dalam tingkat *CEFR*. Untuk audio yang memiliki durasi lebih dari 10 detik, akan dipotong menggunakan *Pydub* berdasarkan hasil *timestamp segmen* dari transkrip teks dari model *Speech Recognition Whisper*. Dari tingkat *CEFR* akan diubah menjadi tingkat berbicara *IELTS* sesuai dengan dokumen resmi *IELTS* [17]. Untuk menghitung nilai *Fluency* secara keseluruhan, perhitungan rata-rata akan digunakan, kemudian dibulatkan mengikuti aturan pembulatan skor *IELTS* [4]. Tabel I berisi konversi dari level *CEFR* ke dalam bentuk *IELTS Band*.

TABEL I. Konversi *CEFR* ke dalam bentuk *IELTS Band*

<i>CEFR</i>	<i>IELTS Band</i>
C2	9
C1	8
B2	6,5
B1	5
A2	4
A1	3

Untuk penilaian *Lexical*, akan menggunakan model *DistilBERT* untuk mengklasifikasikan teks transkrip dari *Speech Recognition Whisper* ke dalam tingkat *CEFR*. Dari tingkat *CEFR* akan diubah menjadi tingkat *IELTS* sesuai dengan dokumen resmi *IELTS* [17].

Untuk penilaian *Grammar*, akan menggunakan model *T5* untuk melakukan *GEC (Grammar Error Correction)* untuk setiap segmen teks transkrip dari *Speech Recognition Whisper*. Hasil *GEC* akan dibandingkan dengan teks

transkrip asli untuk mendapatkan persentase kesalahan *Grammar*. Selanjutnya, teks transkrip akan diperiksa apakah merupakan kalimat sederhana atau kompleks. Hasil persentase kesalahan dan pemeriksaan kalimat akan dihitung untuk mendapatkan tingkat berbicara *IELTS* sesuai dengan deskripsi resmi band berbicara *IELTS* [17].

TABLE II. *Grammar Evaluation*

<i>Grammar Evaluation</i>	<i>IELTS Band</i>
Jumlah kalimat hanya 1, kalimat sederhana, kesalahan <i>grammar</i> < 40%	2
Jumlah kalimat hanya 1, kalimat sederhana, kesalahan <i>grammar</i> ≥ 40%	3
Jumlah kalimat lebih dari 1, hanya kalimat sederhana, kesalahan <i>grammar</i> < 50%	4
Jumlah kalimat lebih dari 1, hanya kalimat sederhana, kesalahan <i>grammar</i> ≥ 50%	5
Jumlah kalimat lebih dari 1, jumlah kalimat sederhana dan kompleks sama, kesalahan <i>grammar</i> < 50%	
Jumlah kalimat lebih dari 1, jumlah kalimat sederhana lebih banyak dari kalimat kompleks, kesalahan <i>grammar</i> < 50%	
Jumlah kalimat lebih dari 1, jumlah kalimat sederhana dan kompleks sama, kesalahan <i>grammar</i> ≥ 50%	6
Jumlah kalimat lebih dari 1, jumlah kalimat sederhana lebih dari kalimat kompleks, kesalahan <i>grammar</i> ≥ 50%	
Jumlah kalimat hanya 1, kalimat kompleks, kesalahan <i>grammar</i> < 40%	7
Jumlah kalimat lebih dari 1, semua kalimat kompleks, kesalahan <i>grammar</i> < 40%	
Jumlah kalimat lebih dari 1, jumlah kalimat kompleks lebih banyak dari kalimat sederhana, kesalahan <i>grammar</i> < 50%	8
Jumlah kalimat hanya 1, kalimat kompleks, kesalahan <i>grammar</i> ≥ 40%	
Jumlah kalimat lebih dari 1, semua kalimat kompleks, kesalahan <i>grammar</i> ≥ 40% and < 80%	
Jumlah kalimat lebih dari 1, jumlah kalimat kompleks lebih banyak dari kalimat sederhana, kesalahan <i>grammar</i> ≥ 50%	9
Jumlah kalimat lebih dari 1, semua kalimat kompleks, kesalahan <i>grammar</i> ≥ 80%	

Untuk penilaian *Pronunciation*, akan menggunakan model *Wav2Vec2* untuk mengklasifikasikan berkas audio. Untuk audio yang memiliki durasi lebih dari 10 detik, akan dipotong menggunakan *Pydub* berdasarkan hasil *timestamp segmen* dari transkrip teks dari model *Speech Recognition Whisper*. Dari hasil klasifikasi akan diubah menjadi tingkat *IELTS* menggunakan konversi pada Tabel III. Untuk menghitung tingkat *Pronunciation* secara keseluruhan, perhitungan rata-rata akan digunakan, kemudian dibulatkan mengikuti aturan pembulatan skor *IELTS* [4].

TABLE III. *Pronunciation Evaluation*

<i>Pronunciation Level</i>	<i>IELTS Band</i>
<i>Beginner</i>	4
<i>Intermediate</i>	5
<i>Advance</i>	7
<i>Proficient</i>	9

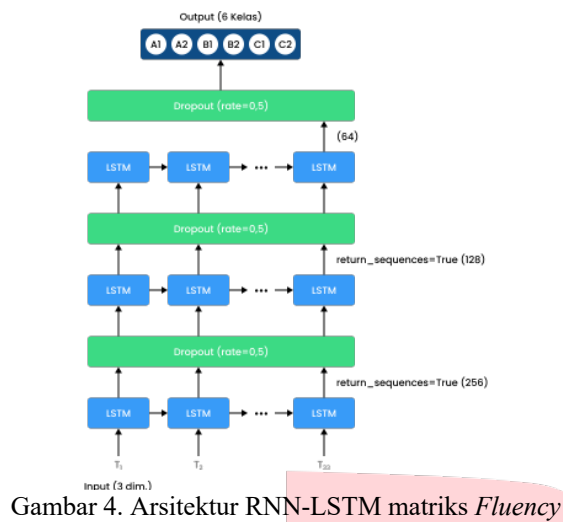
Setelah semua proses evaluasi selesai, hasil evaluasi akan dikirim kembali ke aplikasi *Android* sebagai *respons*. Selanjutnya, aplikasi akan menyimpan dan mengumpulkan data *respons* dari *Cloud* yang berisi skor evaluasi yang dihasilkan oleh *Machine Learning* ke dalam *Firestore Realtime Database* (FRTDB). Aplikasi akan mengambil data dari FRTDB dan menghitung hasil keseluruhan sesuai perhitungan resmi *band IELTS* untuk ditampilkan kepada pengguna.

B. Proses Pengembangan Sistem

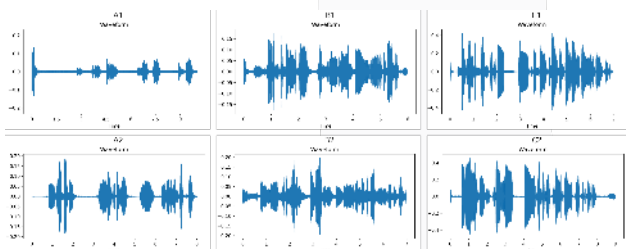
Proses pengembangan sistem *Machine Learning* pada aplikasi yang dikembangkan dibagi menjadi 4 bagian sesuai matriks evaluasi pada tes *IELTS* sesi *Speaking*.

a. Fluency Evaluation

Untuk matriks evaluasi *Fluency*, akan menggunakan model RNN-LSTM untuk mengklasifikasikan berkas audio ke dalam tingkat *CEFR*. Arsitektur model terdiri dari 3 lapisan tersembunyi LSTM dengan jumlah *neuron* 256, 128, dan 64 serta fungsi aktivasi "*relu*". Didalam arsitektur model juga terdapat lapisan *Dropout* setelah lapisan LSTM untuk mengurangi *overfitting* dalam *neural network* dengan nilai 0,5. Dan di akhir lapisan model, terdapat lapisan *output* dengan 6 *neuron* dan fungsi aktivasi "*softmax*". Berikut adalah gambar arsitektur model untuk matriks evaluasi *Fluency*.

Gambar 4. Arsitektur RNN-LSTM matriks *Fluency*

Model dilatih dengan *dataset* yang dibuat sendiri. *Dataset* ini dibuat dari kumpulan video simulasi sesi *speaking* tes *IELTS* di YouTube yang dipotong menjadi 5-10 detik dan kemudian diberi label *CEFR* berdasarkan nilai *band* yang tertera dalam deskripsi video. proses augmentasi data untuk menyeimbangkan jumlah data di setiap kelas menggunakan metode *Oversampling*. Proses augmentasi data yang dilakukan termasuk penambahan *noise Gaussian*, penyerapan udara, membatasi sinyal audio, dan memotong keheningan di awal dan akhir audio. Setelah proses augmentasi data, total 4560 data diperoleh. *Dataset* akan dilakukan *preprocessing* dengan menghitung MFCC dengan jumlah *mel* sebanyak 30, mencari ZCR, RMSE, dan SF. *Dataset* akan dibagi menjadi 3 bagian untuk pelatihan, pengujian, dan validasi.

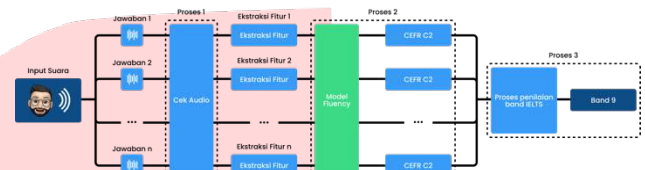
Gambar 5. Contoh *dataset* untuk model *Fluency*

Model RNN-LSTM dilatih untuk dapat melakukan klasifikasi kelancaran dalam mengucapkan sebuah kalimat (*fluency*) untuk dapat menghasilkan kelas *band speaking IELTS*. Untuk dapat menghasilkan kelas *band speaking IELTS* file audio yang akan diolah akan masuk kedalam proses pemotongan *segment* audio. Untuk mengetahui *segment* audio akan menggunakan teknologi *library Faster Whisper* menggunakan model dengan ukuran "*small.en*". *Faster Whisper* akan memberikan informasi berupa *timestamp* setiap kata yang diucapkan dan *segment* pada file audio. Jika dalam satu file audio terdapat lebih dari 1 *segment*, maka file audio akan dilakukan pemotongan menggunakan *library Pydub* sesuai dengan *timestamp* setiap *segment*.

Proses selanjutnya setelah mendapatkan *segment* audio dan pemotongan *segment* audio adalah proses *preprocessing*. Proses *preprocessing* yang dilakukan sama dengan proses *preprocessing* pada tahap pembuatan model yaitu dengan

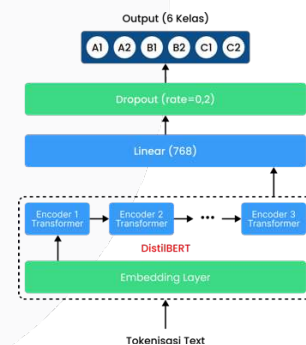
melakukan ekstraksi fitur MFCC pada file audio dengan jumlah *mel* 30 dan mengambil nilai ZCR, RMSE, dan SF.

Setelah dilakukan proses *preprocessing* selanjutnya data akan masuk ke dalam model untuk dilakukan klasifikasi. Model akan memberikan keluaran berupa hasil klasifikasi setiap data yang masuk ke model. Model dilatih menggunakan *dataset* untuk klasifikasi kategori *CEFR*, sehingga perlu dilakukan konversi untuk mendapatkan hasil *band speaking IELTS* sesuai pada Tabel I. Hasil konversi setiap data selanjutnya akan masuk ke dalam proses perhitungan nilai akhir *band speaking IELTS*. Untuk proses perhitungan nilai akhir *band speaking IELTS*, akan digunakan perhitungan rata-rata dengan pembulatan 0,5. Berikut adalah gambar alur proses evaluasi matriks *Fluency*.

Gambar 6. Proses evaluasi matriks *Fluency*

b. Lexical Evaluation

Untuk matriks evaluasi *Lexical*, akan menggunakan model *DistilBERT* untuk mengklasifikasikan teks transkrip dari *Speech Recognition Whisper* ke dalam tingkat *CEFR*. API *Hugging Face* digunakan untuk *fine-tuning* model *pre-train base DistilBERT*. Untuk *Speech Recognition*, digunakan versi "*small.en*" bahasa Inggris dari *Whisper*. Untuk penelitian ini, model *DistilBERT* akan digunakan untuk mengklasifikasikan teks, sehingga lapisan *output* akan memiliki lapisan *linear* dan lapisan *output* dengan 6 neuron. Berikut adalah gambar arsitektur model untuk matriks evaluasi *Lexical*.

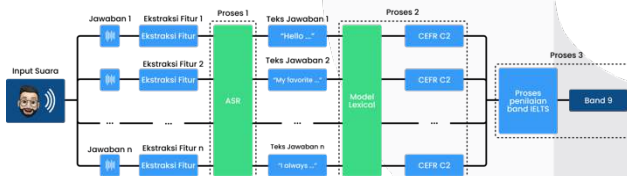
Gambar 7. Arsitektur *DistilBERT* matriks *Lexical*

Model dilatih dengan *dataset Lexical* yang telah dipreproses. beberapa *dataset*, yaitu *CEFR-SP*, *Dataset CEFR* dari *Kaggle*, *CERD*, dan *Dataset CEFR* dari *Hugging Face*. Beberapa *dataset* ini digunakan karena terdapat perbedaan jumlah data yang terlalu jauh antara kelas-kelas data. *Dataset* yang diperoleh akan dipreproses dengan menghapus simbol, kata-kata, dan sebagian tanda baca yang tidak diperlukan. Data yang akan digunakan adalah data yang memiliki jumlah kata kurang dari 500 kata. Setelah proses *pre-processing*, *dataset* dengan 6 kelas *CEFR* diperoleh total 5526 data. *Dataset* akan dibagi menjadi 3 bagian untuk pelatihan, pengujian, dan validasi.

text (string)	label (class label)
"But in France it was republicans who were much kinder to centralize authority and to politicize Brittany Alsace or Corsica than royalty had ever been"	3 (R2)
"Parish councils are run by volunteer councillors who are elected to serve for four years and are not paid"	3 (R2)
"Your trail exits on the near side of the bridge but your route beginning with the more demanding part of the circuit plunges into the bush at the bridges west end"	6 (A1)
"Instead European countries in particular Germany and Scandinavia focus on the provision of youth services for young people discussed in the next chapters"	3 (R2)
"In TCM theory damage to kidney functions directly affects the ear"	3 (R2)
"After Lafferty had been sworn in and was seated Phil approached the witness box"	3 (R2)
"One day one second I might close the shutter on the perfect photograph There was always the chance so long as there was film in my camera Finish one load another and keep looking with."	5 (C2)
"Giovanni recognizes that effective change occurs through action"	3 (R2)
"In not Jimmy Carter Clinton said firmly"	1 (A2)
"Our relations are very much better than they were a few years ago"	2 (R1)
"Congested and Confused When I was young a nose had few choices when it came to cold remedies"	3 (R2)
"Just after the plane took off she realized she had left her presentation in the strangers car and as a result she lost her bid with the investors"	1 (A2)
"Not actually he said"	3 (R2)
"Resolved to go along with the majority despite her personal misgivings Phil had been to buy a laptop She had been able to get a ten percent discount which impressed Chen though Lily felt."	5 (C2)
"Our final margin over all other parties was four seats"	4 (C1)
"The use of complementary colors is an important aspect of aesthetically pleasing art and graphic design"	4 (C1)

Gambar 8. Contoh dataset untuk modek Lexical

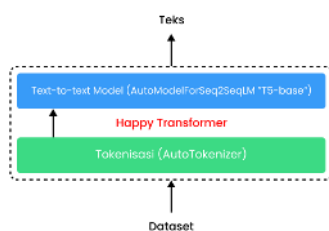
Model *DistilBERT* dilatih untuk dapat melakukan klasifikasi *vocabulary* baik itu banyak jenis *vocabulary* yang dipakai hingga ketepatan penggunaan *vocabulary* dalam sebuah kalimat untuk dapat menghasilkan kelas *band speaking IELTS*. Untuk dapat menghasilkan kelas *band speaking IELTS* file audio akan masuk kedalam proses ASR untuk mendapatkan *transcribe* teks hasil ASR pada file audio yang dikirim. Untuk mendapatkan *transcribe* teks hasil ASR akan menggunakan teknologi *library Faster Whisper* menggunakan model *Whisper* dengan ukuran "small.en". Hasil *transcribe* selanjutnya akan masuk ke dalam model *lexical* untuk dilakukan klasifikasi. Model *lexical* akan memberikan keluaran berupa hasil klasifikasi dari teks hasil ASR berdasarkan kelas *CEFR* yang masuk ke model. Hasil klasifikasi yang memiliki presentase tertinggi akan menjadi hasil akhir klasifikasi *CEFR*. Model dilatih menggunakan dataset untuk klasifikasi kategori *CEFR*, sehingga perlu dilakukan konversi untuk mendapatkan hasil *band speaking IELTS* sesuai pada Tabel I. Hasil konversi adalah nilai akhir *band speaking IELTS* aspek *lexical*. Berikut adalah gambar alur proses evaluasi matriks *Lexical*.



Gambar 9. Proses evaluasi matriks Lexical

c. Grammar Evaluation

Untuk matriks evaluasi *Grammar*, akan menggunakan model *T5* untuk melakukan *GEC* pada teks transkrip dari *Speech Recognition Whisper* dengan menggunakan *Happy Transformers* untuk proses *fine-tuning* model. Berikut adalah gambar arsitektur model untuk matriks evaluasi *Grammar*.



Gambar 10. Arsitektur T5 matriks Grammar

Model dilatih dengan dataset *Grammar* yang telah dipreproses. Dataset yang digunakan adalah *Dataset C4 200M* berisi 200 juta data untuk *GEC*. Karena dataset ini cukup besar, hanya 125.000 data yang akan digunakan untuk melatih model *GEC*. Dataset yang diperoleh akan *preprocessing* dengan menambahkan awalan "grammar:" untuk data output dan membagi dataset menjadi 2 bagian untuk pelatihan dan pengujian.

input (string)	output (string)
"Birnath is for \$7,094 this morning, which CafeBook says."	"Birnath goes for \$7,894 this morning, according to CafeBook."
"The effect of widespread due targets was face up attack position monsters on the field."	"1. the effect of "widespread due" targets two face up attack position monsters on the field."
"Tax on sales of shares for non residents are set at 21% for 2016 and 20% in 2015 payable on sales (initially)..."	"Capital Gains tax on the sale of properties for non-residents is set at 21% for 2016 and 20% in 2015 payable."
"Much many brands and sellers still in the market."	"Many brands and sellers still in the market."
"This is is the latest maintenance release of Samba 3.6"	"this is is the latest maintenance release of Samba 3.6."
"Fairy Dr. Mel, T in the Endfather: no just look, but my outfit" for taking the part as godfather."	"Fairy Dr. Mel, T in the Endfather: No just a look, but my outfit for taking the role as godfather."
"Watch as this Dodge Challenger Hellcat gets snowed by a Tesla Model S - with the drag strip."	"Watch as this Dodge Challenger Hellcat gets snowed by a Tesla Model S at the drag strip."
"However, these devices have been proven to help consumers during another company his information."	"However, these devices are proven to help consumers while another company that information."
"Every cloud has a silver lining and it's just possible that we were beaten before the off as the first three now."	"Every cloud has a silver lining and it's just possible that we were beaten before the off as the first three have."
"Workless involved's supporting doc the movement."	"Get involved and help the movement!"
"Mark Miller said in a post on Instagram that he and fellow driver Kimmey Jeffries have confirmed the..."	"On Monday, driver Mark Miller said in a post on Instagram he and fellow driver Kimmey Jeffries confirm."
"Once done, you could pay for your pumpkins and then leave or take advantage of the great stuff in there."	"Once done, you could pay for your pumpkins and then leave or take advantage of the great stuff there."

Gambar 11. Contoh dataset untuk model GEC

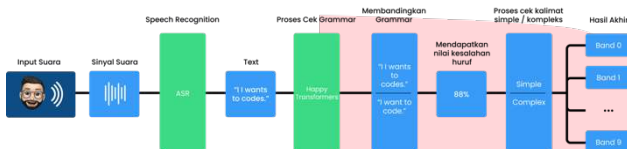
Model *T5* dilatih untuk dapat melakukan melakukan koreksi kesalahan *grammar* sebuah kalimat. Untuk mendapatkan koreksi kesalahan *grammar* sebuah kalimat, file audio akan masuk kedalam proses ASR untuk mendapatkan *transcribe* teks hasil ASR pada file audio yang dikirim. Untuk mendapatkan *transcribe* teks hasil ASR akan menggunakan teknologi *library Faster Whisper* menggunakan model *Whisper* dengan ukuran "small.en". Hasil *transcribe* selanjutnya akan masuk ke dalam model *NLTK* untuk dilakukan analisis teks untuk mendeteksi jumlah kalimat pada hasil *transcribe* ASR. Untuk proses penilaian akan dilakukan setiap kalimat dari hasil analisis teks menggunakan model *NLTK*. Di setiap kalimat yang didapat akan masuk kedalam model *GEC* untuk dilakukan koreksi kesalahan *grammar* pada setiap kalimat. Hasil model *GEC* akan dihitung kesalahannya dengan cara membandingkan teks sebelum masuk ke *GEC* dan sesudah menggunakan *library Errant*.

Evaluasi ini juga menggunakan model bahasa Inggris *NLTK* untuk mendapatkan *tag* kata dalam kalimat yang berfungsi untuk mengklasifikasikan jenis kalimat apakah kalimat sederhana atau kalimat kompleks. Untuk kalimat sederhana, hanya terdapat 1 klausa atau 1 kata kerja, sedangkan untuk kalimat kompleks akan terdapat lebih dari 1 klausa atau kata kerja. Untuk mengklasifikasikan jenis kalimat, model *NLTK* akan melakukan *tagging* kata dalam kalimat dan sistem akan mencari *ROOT* dan *VERB* token dalam kalimat. Jika sistem mendapatkan hanya 1 token *ROOT* atau *VERB*, maka jenis kalimatnya adalah kalimat sederhana. Jika sistem mendapatkan lebih dari 1 token *ROOT* atau *VERB*, jenis kalimatnya adalah kalimat kompleks.

Word	Index	Pos	Tag	Parent	children
he	0	PRON	nsubj	['eats']	['']
eats	1	VERB	ROOT	['']	['he', 'cheese', ',', 'but', 'eat']
cheese	2	NOUN	dobj	['eats']	['']
,	3	PUNCT	punct	['eats']	['']
but	4	CCONJ	cc	['eats']	['']
he	5	PRON	nsubj	['eat', 'eats']	['']
wo	6	AUX	aux	['eat', 'eats']	['']
n't	7	PART	neg	['eat', 'eats']	['']
eat	8	VERB	conj	['eats']	['he', 'wo', 'n't', 'cream', '.']
ice	9	NOUN	compound	['cream', 'eat', 'eats']	['']
cream	10	NOUN	dobj	['eat', 'eats']	['ice']
.	11	PUNCT	punct	['eat', 'eats']	['']

Gambar 12. Tagging kata dengan model Bahasa Inggris NLTk

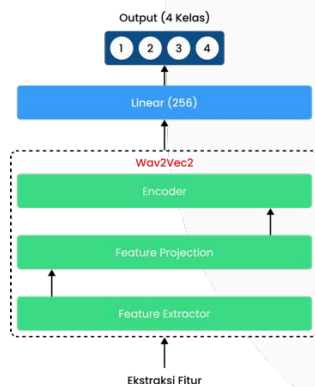
Hasil perhitungan kesalahan *Grammar* dan jenis kalimat akan dievaluasi untuk mendapatkan *IELTS band* sesuai pada Tabel II. Berikut adalah gambar alur proses evaluasi matriks *Grammar*.



Gambar 13. Proses evaluasi matriks *Grammar*

d. Pronunciation Evaluation

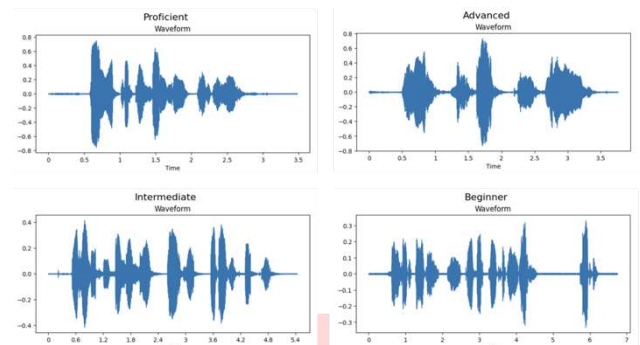
Untuk matriks evaluasi *Pronunciation*, akan menggunakan model *Wav2Vec2* untuk mengklasifikasikan *file* audio. API *Hugging Face* digunakan untuk *fine-tuning* model *pre-train wav2vec2-large-xlsr-53-english* dari *Jonatas Grosman*. Untuk penelitian ini, model *Wav2Vec2* akan digunakan untuk mengklasifikasikan *file* audio, sehingga lapisan *output* akan memiliki lapisan *linear* dan lapisan *output* dengan 4 *neuron*. Model dilatih dengan *dataset Pronunciation* yang telah dipreproses. Berikut adalah gambar arsitektur model untuk matriks evaluasi *Pronunciation*.



Gambar 14. Arsitektur *Wav2Vec2* matriks *Pronunciation*

Model dilatih dengan *dataset Pronunciation* yang telah dipreproses. *Dataset* yang akan digunakan adalah *dataset speechocean762*. Dalam penelitian ini, akan menggunakan penilaian tingkat kalimat. Karena penilaian yang diberikan pada *dataset* dalam bentuk nilai dari 1 hingga 10, maka akan dilakukan pemrosesan *dataset*. Pemrosesan *dataset* dilakukan dengan menyederhanakan penilaian *dataset* menjadi 4 kelas, yaitu 0 - 4 menjadi kelas *beginner*, 5 - 6 menjadi kelas *intermediate*, 7 - 8 menjadi kelas *advance*, 9 - 10 menjadi kelas *proficient*. *Dataset* juga dilakukan proses augmentasi data untuk menyeimbangkan jumlah data di setiap kelas menggunakan metode *Oversampling*. Proses augmentasi data yang dilakukan termasuk penambahan *noise Gaussian*, penyerapan udara, membatasi sinyal audio, dan

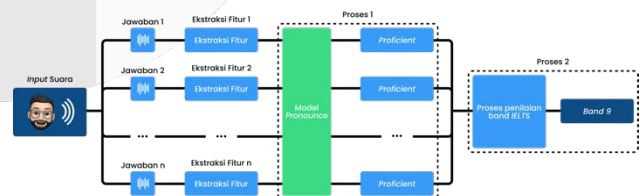
memotong keheningan di awal dan akhir audio. Setelah proses augmentasi data, diperoleh total 4560 data. *Dataset* akan dibagi menjadi 3 bagian untuk pelatihan, pengujian, dan validasi.



Gambar 15. Contoh *dataset* untuk model *Pronunciation*

Model *Wav2Vec2* dilatih untuk dapat melakukan klasifikasi pengucapan (*pronunciation*) dalam sebuah kalimat untuk dapat menghasilkan kelas *band speaking IELTS*. Untuk dapat menghasilkan kelas *band speaking IELTS file* audio akan masuk kedalam proses pemotongan *segment* audio. Untuk mengetahui *segment* audio akan menggunakan teknologi *library Faster Whisper* menggunakan model dengan ukuran "*small.en*". *Faster Whisper* akan memberikan informasi berupa *timestamp* setiap kata yang diucapkan dan *segment* pada *file* audio. Jika dalam satu *file* audio terdapat lebih dari 1 *segment*, maka *file* audio akan dilakukan pemotongan menggunakan *library Pydub* sesuai dengan *timestamp* setiap *segment*.

Setelah mendapatkan *segment* audio dan pemotongan *segment* audio, selanjutnya data akan masuk ke dalam model untuk dilakukan klasifikasi. Model akan memberikan keluaran berupa hasil klasifikasi setiap data yang masuk ke model. Model dilatih menggunakan *dataset* untuk klasifikasi penilaian *pronounce* yang sudah diolah, sehingga perlu dilakukan konversi untuk mendapatkan hasil *band speaking IELTS* sesuai pada Tabel III. Hasil konversi setiap data selanjutnya akan masuk ke dalam proses perhitungan nilai akhir *band speaking IELTS*. Untuk proses perhitungan nilai akhir *band speaking IELTS*, akan digunakan perhitungan rata-rata dengan pembulatan 0,5. Berikut adalah gambar alur proses evaluasi matriks *Pronunciation*.



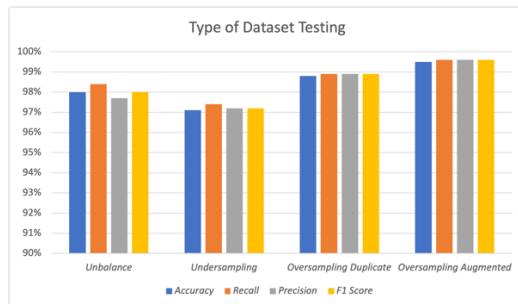
Gambar 16. Proses evaluasi matriks *Pronunciation*

IV. HASIL DAN PEMBAHASAN

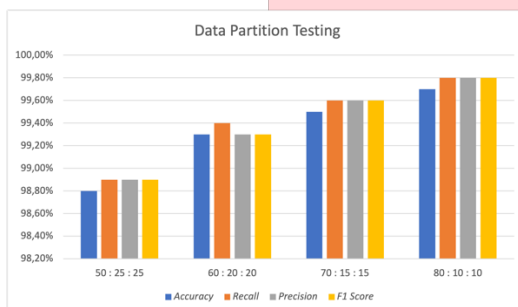
Terdapat 4 pengujian sistem *Machine Learning* pada aplikasi yang dikembangkan sesuai dengan matriks evaluasi tes *IELTS* sesi *Speaking*. Dan disetiap sistem akan ada beberapa pengujian. Berikut adalah pengujian di setiap sistem pada aplikasi yang dikembangkan.

A. Pengujian model RNN-LSTM *Fluency*

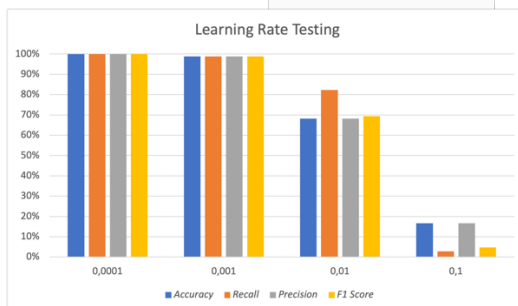
Untuk model *Fluency* terdapat 5 skenario pengujian. Skenario pengujian pertama adalah jenis *dataset*, partisi data, *learning rate*, *batch size*, dan *epochs*.



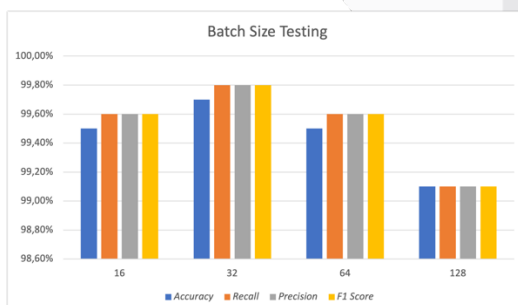
Gambar 17. Pengujian jenis *dataset* model *Fluency*



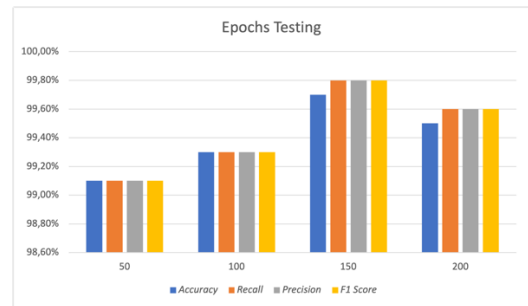
Gambar 18. Pengujian partisi *dataset* model *Fluency*



Gambar 19. Pengujian *learning rate* model *Fluency*



Gambar 20. Pengujian *batch size* model *Fluency*

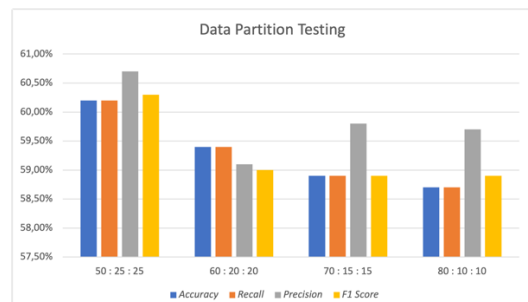


Gambar 21. Pengujian *epochs* model *Fluency*

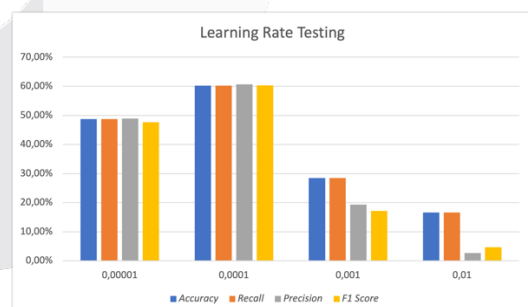
Dari hasil pengujian, model terbaik untuk model *Fluency* diperoleh pada jenis *dataset Augmented Oversampling*, partisi data dengan konfigurasi 80% untuk pelatihan, 10% untuk setiap data uji dan validasi, serta *hyperparameter learning rate* sebesar 0,0001, *batch size* 32, dan jumlah *epochs* 150 dengan nilai matriks evaluasi *accuracy*, *recall*, *precision*, dan *F1 scores* masing-masing sebesar 99,7%, 99,8%, 99,8%, dan 99,8%.

B. Pengujian model *DistilBERT Lexical*

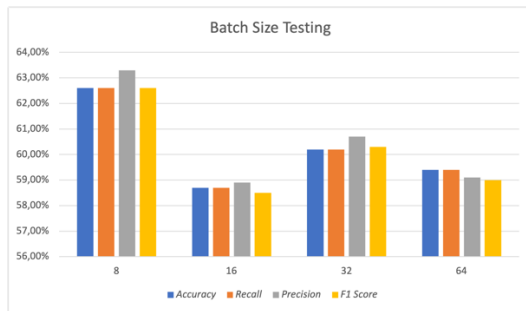
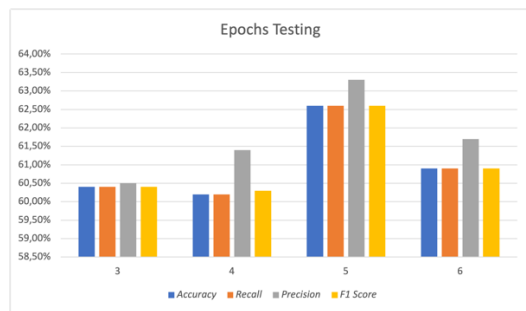
Untuk model *Lexical*, terdapat 4 skenario pengujian. Skenario pengujian pertama adalah partisi data, *learning rate*, *batch size*, dan *epochs*.



Gambar 22. Pengujian partisi *dataset* model *Lexical*



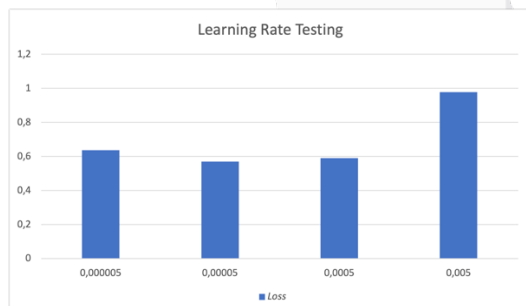
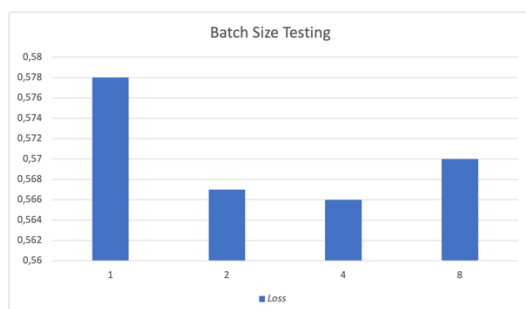
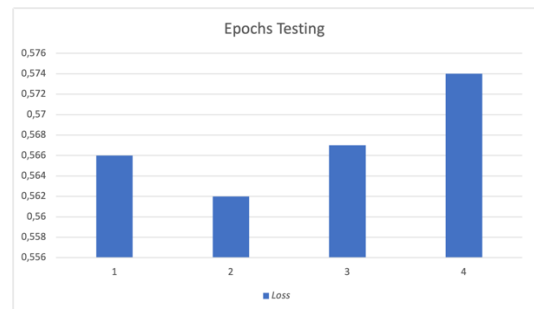
Gambar 23. Pengujian *learning rate* model *Lexical*

Gambar 24. Pengujian *batch size* model *Lexical*Gambar 25. Pengujian *epochs* model *Lexical*

Dari hasil pengujian, model terbaik diperoleh untuk model *Lexical* pada partisi data dengan konfigurasi 50% untuk pelatihan, 25% untuk masing-masing data uji dan validasi, serta *learning rate* sebesar 0,0001, *batch size* 8, dan jumlah *epochs* 5, dengan nilai matriks evaluasi *accuracy*, *recall*, *precision*, dan *F1 scores* adalah 62,6%, 62,6%, 63,3%, dan 62,6%.

C. Pengujian model *T5 Grammar*

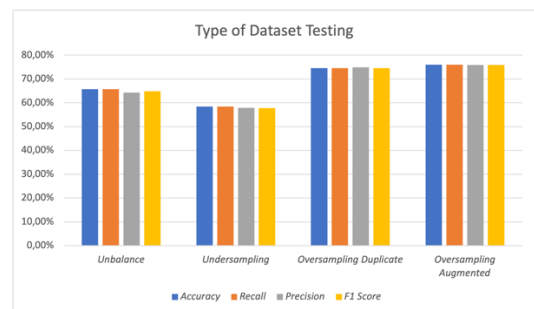
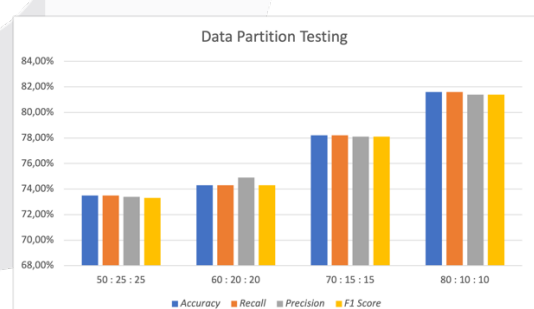
Untuk model *Grammar*, terdapat 3 skenario pengujian. Skenario pengujian pertama adalah *learning rate*, *batch size*, dan *epochs*.

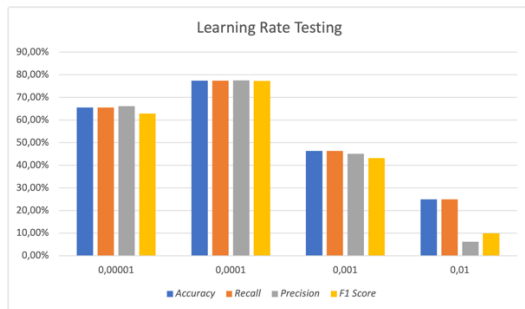
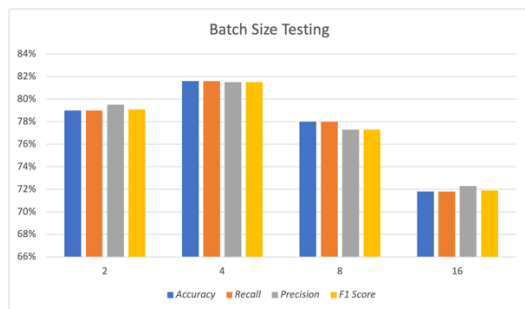
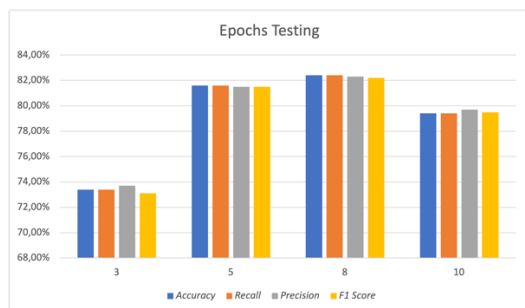
Gambar 26. Pengujian *learning rate* model *Grammar*Gambar 27. Pengujian *batch size* model *Grammar*Gambar 28. Pengujian *epochs* model *Grammar*

Dari hasil pengujian, model terbaik diperoleh untuk model *T5 GEC* pada partisi data dengan konfigurasi data pelatihan sebesar 80% dan data uji sebesar 20%, dengan *learning rate* 0,00005, *batch size* 4, dan jumlah *epochs* 2, dengan nilai matriks evaluasi loss sebelum pelatihan model sebesar 1,902 dan setelah pelatihan model dilakukan diperoleh 0,562.

D. Pengujian model *Wav2Vec2 Pronunciation*

Untuk model *Pronunciation*, terdapat 5 skenario pengujian. Skenario pengujian pertama adalah jenis *dataset*, partisi data, *learning rate*, *batch size*, dan *epochs*.

Gambar 29. Pengujian jenis *dataset* model *Pronunciation*Gambar 30. Pengujian partisi *dataset* model *Pronunciation*

Gambar 31. Pengujian *learning rate* model *Pronunciation*Gambar 32. Pengujian *batch size* model *Pronunciation*Gambar 33. Pengujian *epochs* model *Pronunciation*

Dari hasil pengujian, model terbaik untuk model *Pronunciation* diperoleh dalam jenis *dataset Augmented Oversampling*, partisi data dengan konfigurasi 80% untuk pelatihan, 10% untuk masing-masing data uji dan validasi, serta *hyperparameter learning rate* 0,0001, *batch size* 4, dan jumlah *epoch* 8 dengan nilai matriks evaluasi *accuracy*, *recall*, *precision*, dan *F1 scores* adalah 82,4%, 82,4%, 82,3%, dan 82,2%.

V. KESIMPULAN

Berdasarkan hasil penelitian studi, pengembangan aplikasi Simulasi tes *IELTS* sesi *Speaking* telah berhasil dikembangkan pada platform *Android*. Pada proses penilaian pada Simulasi tes *IELTS* sesi *Speaking* dilakukan sesuai dengan tes resmi *IELTS* sesi *Speaking* dengan menggunakan 4 matriks evaluasi yaitu *Fluency*, *Lexical*, *Grammar*, dan *Pronunciation*. Setiap matriks evaluasi akan menggunakan 1 model yang sudah dilatih sesuai dengan fungsi utama model tersebut dan menggunakan jenis *dataset*, partisi *dataset*, dan *hyperparameter* yang terbaik untuk model tersebut. Keempat model sudah diuji dan menghasilkan nilai akurasi 99% untuk model RNN-LSTM pada matriks evaluasi *Fluency*, nilai akurasi 62% untuk model *DistilBERT* pada matriks evaluasi

Lexical, nilai *loss* 0,562 untuk model *T5* pada matriks evaluasi *Grammar*, dan nilai akurasi 82% untuk model *Wav2Vec2* pada matriks evaluasi *Pronunciation*. Dengan pengembangan aplikasi Simulasi tes *IELTS* sesi *Speaking* dengan platform *Android*, diharapkan dapat mengurangi masalah yang telah dijelaskan dan menjadi solusi sebagai platform pelatihan untuk sesi *Speaking* tes *IELTS* yang memungkinkan penilaian yang cepat, akurat, dan murah.

REFERENSI

- [1] Universitas Cambridge; British Council; IDP Education Australia, "What is IELTS?," [Online]. Available: <https://www.ielts.org/about-ielts/what-is-ielts>. [Diakses 17 Oktober 2022].
- [2] A. Hashemi dan S. Daneshfar, "A Review of the IELTS Test: Focus on Validity, Reliability, and Washback," *Indonesian Journal of English Language Teaching and Applied Linguistics*, vol. 3, no. 1, pp. 42-43, 2018.
- [3] IELTS, "Test taker performance 2022," [Online]. Available: <https://www.ielts.org/for-researchers/test-statistics/test-taker-performance>. [Diakses 1 Agustus 2023].
- [4] IELTS, "IELTS scoring in detail," [Online]. Available: <https://www.ielts.org/for-organisations/ielts-scoring-in-detail>. [Diakses 1 Agustus 2023].
- [5] Universitas Cambridge; British Council; IDP Education Australia, "IELTS practice test," [Online]. Available: <https://www.ielts.org/usa/ielts-practice-test>. [Diakses 12 November 2022].
- [6] Universitas Cambridge; British Council; IDP Education Australia, "IELTS Academic – Practice Test 1 (Timed)," IELTS Progress Check, [Online]. Available: <https://www.ieltsprogresscheck.com/product/ielts-academic-practice-test-1-timed/>. [Diakses 12 November 2022].
- [7] Universitas Cambridge; British Council; IDP Education Australia, "What is IELTS Progress Check?," IELTS Progress Check, [Online]. Available: <https://www.ieltsprogresscheck.com/#:~:text=What%20is%20IELTS%20Progress,%20Check%3F>. [Diakses 12 November 2022].
- [8] IDP Education, "Apa itu IELTS?," [Online]. Available: <https://www.idp.com/indonesia/ielts/what-is-ielts/>. [Diakses 1 Agustus 2023].
- [9] K. Smagulova dan A. P. James, "A survey on LSTM memristive neural network architectures and applications," *The European Physical*, p. 2313–2324, 2019.
- [10] V. Sanh, L. Debut, J. Chaumond dan T. Wolf, "DistilBERT, a distilled version of BERT smaller, faster, cheaper, and lighter," *CoRR*, 2019.
- [11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li dan P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-

- to-Text Transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.
- [12] E. Fillion dan T. Brownlow, “About,” Happy Transformers, [Online]. Available: <https://happytransformer.com/>. [Diakses 1 Agustus 2023].
- [13] A. Baevski, H. Zhou, A. Mohamed dan M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” *CoRR*, 2020.
- [14] J. Han, M. Kamber dan J. Pei, “Data Mining Concepts and Techniques,” dalam *Elsevier*, USA, 2011.
- [15] M. Azhari, Z. Situmorang dan R. Rosnelly, “Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4 5 Random Forest SVM dan Naive Bayes,” *Jurnal Media Informatika Budidarma*, vol. 5, pp. 640-651, 2021.
- [16] Google Developers, “Descending into ML: Training and Loss,” Google, [Online]. Available: <https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss>. [Diakses 1 Agustus 2023].
- [17] IELTS, “IELTS in CEFR scale,” IELTS, [Online]. Available: <https://www.ielts.org/about-ielts/ielts-in-cefr-scale>. [Diakses 30 July 2023].

