DECODE: JURNAL PENDIDIKAN TEKNOLOGI INFORMASI
P-ISSN: 2775-2984 | E-ISSN: 2775-1813

Vol. 6, No 1, Maret 2026, Hal: 79-93
Doi: http://dx.doi.org/10.51454/decode.v6i1.1563
https://journal.umkendari.ac.id/index.php/decode

Perbandingan Algoritma *Transformer* Dengan *Bi-Long Short-Term Memory* Untuk *Speech-To-Text*

Achmad Rizky Zulkarnain[1], Muhammad Ezar Al Rivan[1]

[1]Program Studi Informatika, Universitas Multi Data Palembang, Indonesia.

Artikel Info

Kata Kunci:

BiLSTM;
Ekstraksi Fitur;
Speech-to-text;
Transformer.

Keywords:

BiLSTM;
Feature Extraction;
Speech-to-text;
Transformer.

Riwayat Artikel:

Submitted: 15 Desember 2025
Accepted: 22 Februari 2026
Published: 24 Februari 2026

Abstrak: Penelitian ini bertujuan untuk membandingkan kinerja dua arsitektur *Speech-to-Text*, yaitu *Bidirectional Long Short-Term Memory* (BiLSTM) dan *Transformer*, dengan menggunakan dua jenis ekstraksi fitur akustik, yaitu *Log-Mel Spectrogram* dan *Filterbank Energies* (FBANK). Perbandingan ini dilakukan untuk menganalisis pengaruh kesesuaian antara arsitektur model dan representasi fitur terhadap performa sistem pengenalan suara otomatis. Pemilihan kedua arsitektur didasarkan pada perbedaan mekanisme pemrosesan sekuens, di mana BiLSTM memproses data secara dua arah untuk menangkap konteks temporal dari masa lalu dan masa depan, sedangkan Transformer memanfaatkan mekanisme *self-attention* yang mampu memproses keseluruhan urutan data secara paralel dan memahami konteks global. Kebaruan penelitian ini terletak pada evaluasi perbandingan yang dilakukan secara konsisten antara model BiLSTM dan *Transformer* dengan skema ekstraksi fitur yang digunakan agar menemukan kecocokan antara model dengan ekstraksi fitur, dengan tokenisasi yang sudah disesuaikan untuk masing-masing arsitektur, yaitu tokenisasi *word-level* pada BiLSTM dan tokenisasi *sub-word* berbasis *SentencePiece* pada *Transformer*, sehingga memberikan analisis kuantitatif yang lebih objektif terhadap pengaruh kesesuaian antara model dan jenis fitur akustik. Penelitian ini menggunakan pendekatan eksperimen kuantitatif dengan dataset *LibriSpeech* sebagai dataset utama. Proses penelitian meliputi ekstraksi fitur *audio*, pelatihan model menggunakan fungsi *loss Connectionist Temporal Classification* (CTC) dan *optimizer Adam*, serta evaluasi performa menggunakan metrik *Word Error Rate* (WER) dan *Character Error Rate* (CER). Hasil eksperimen menunjukkan bahwa pemilihan arsitektur model dan jenis fitur akustik memberikan pengaruh yang nyata terhadap performa sistem. Model BiLSTM menghasilkan performa yang lebih stabil pada seluruh kombinasi fitur, dengan nilai WER sekitar 29% pada *subset test-clean* dan berkisar antara 53%–55% pada *subset test-other*. Sementara itu, model Transformer menunjukkan performa terbaik ketika dipadukan dengan fitur *Log-Mel Spectrogram*, namun mengalami peningkatan WER yang signifikan saat menggunakan fitur FBANK.. Hasil yang sudah dijelaskan tadi menunjukkan bahwa kesesuaian antara arsitektur model dan jenis fitur sangat mempengaruhi kualitas transkripsi.

Abstract: *This study aims to compare the performance of two Speech-to-Text architectures, namely Bidirectional Long Short-Term Memory (BiLSTM) and Transformer, using two types of acoustic feature extraction, namely Log-Mel Spectrogram and Filterbank Energies (FBANK). This comparison was conducted to analyze the effect of the suitability between the model architecture and feature representation on the performance of the automatic speech recognition system. The selection of the two architectures is based on the difference in sequence processing mechanisms, where BiLSTM processes*

data bidirectionally to capture temporal context from the past and future, while Transformer utilizes a self-attention mechanism that is capable of processing the entire data sequence in parallel and understanding the global context. The novelty of this research lies in the consistent comparative evaluation between the BiLSTM and Transformer models with the feature extraction scheme used to find the match between the model and feature extraction, with tokenization that has been adjusted for each architecture, namely word-level tokenization -level tokenization for BiLSTM and SentencePiece-based sub-word tokenization for Transformer, thus providing a more objective quantitative analysis of the influence of the compatibility between the model and the type of acoustic features. This research uses a quantitative experimental approach with the LibriSpeech dataset as the main dataset. The research process includes audio feature extraction, model training using the Connectionist Temporal Classification (CTC) loss function and Adam optimizer, and performance evaluation using the Word Error Rate (WER) and Character Error Rate (CER) metrics. The experimental results show that the selection of model architecture and acoustic feature type has a significant effect on system performance. The BiLSTM model produced more stable performance across all feature combinations, with a WER value of around 29% on the test-clean subset and ranging from 53% to 55% on the test-other subset. Meanwhile, the Transformer model showed the best performance when combined with the Log-Mel Spectrogram feature, but experienced a significant increase in WER when using the FBANK feature. The results described above show that the compatibility between the model architecture and feature type greatly affects the transcription quality.

Corresponding Author:
Achmad Rizky Zulkarnain
Email: zulriz.22@gmail.com

PENDAHULUAN

Teknologi *speech-to-text* atau *automatic speech recognition* (ASR) terus mengalami perkembangan pesat seiring kemajuan pembelajaran mendalam (*deep learning*) dan model *end-to-end* modern, terutama dalam beberapa tahun terakhir. Sistem ASR berbasis *end-to-end* telah meningkatkan akurasi dan kemampuan pemrosesan konteks dengan lebih efisien dibandingkan pendekatan berbasis statistik tradisional, serta mampu menangani variasi lingkungan akustik dan model bahasa beragam (Ahlawat et al., 2025). Sistem konversi *speech-to-text* dapat menjadi teknologi yang sangat berguna memiliki banyak manfaat untuk mengubah suara menjadi teks hasil transkipsi. Salah satu manfaatnya yaitu membantu pembelajaran online dengan mengubah suara di video tersebut menjadi teks untuk membantu pemahaman materi yang lebih baik (Kamarullah, 2024). Meskipun begitu, masih terdapat tantangan seperti kualitas data suara, keberagaman aksen dan bahasa dari data tersebut, serta kemampuan algoritmanya dalam menghadapi *noise* (Lubis, 2025). Hal ini menunjukkan untuk teknologi *speech-to-text* masih membutuhkan pengembangan agar dapat mengatasi tantangan dari segi kualitas bahasa dan kemampuan arsitekturnya.

Long Short-Term Memory (LSTM) merupakan pengembangan dari *Recurrent Neural Network* (RNN) yang dirancang untuk menangani permasalahan *vanishing gradient* melalui mekanisme memori dan *gating*, sehingga mampu mempertahankan dependensi jangka panjang pada data sekuensial (Sherstinsky, 2023), namun arsitektur LSTM satu arah masih memiliki keterbatasan dalam menangkap konteks *global* secara menyeluruh karena pemrosesan data dilakukan secara sekuensial dan hanya bergantung pada informasi masa lalu (Dong et al., 2018). Untuk mengatasi keterbatasan pemodelan konteks satu arah, penelitian ini menggunakan arsitektur *Bidirectional Long Short-Term Memory* (BiLSTM). Berbeda dengan LSTM konvensional, BiLSTM memproses sekuens dalam dua arah sehingga mampu menangkap dependensi temporal jangka pendek maupun jangka panjang secara simultan.

Pendekatan ini terbukti efektif dalam berbagai tugas pengenalan suara karena memungkinkan model memahami konteks fonetik dan linguistik secara lebih menyeluruh (Feng, 2024).

Transformer merupakan arsitektur *deep learning* yang memanfaatkan mekanisme *self-attention* untuk memodelkan hubungan antar elemen dalam suatu urutan secara paralel, sehingga setiap token dapat mempertimbangkan konteks *global* tanpa bergantung pada proses rekursif (Rogers et al., 2020). *Self-attention* merupakan mekanisme utama dalam *Transformer* yang memungkinkan setiap elemen input menghitung relevansinya terhadap elemen lain dalam urutan yang sama, sehingga dependensi jarak jauh dapat dimodelkan secara efisien dan paralel Lalu memiliki arsitektur yang mampu memahami konteks global lebih baik karena mekanisme *self-attention* (Rogers et al., 2020). Dalam bidang pengenalan suara, arsitektur berbasis *Transformer* untuk *Automatic Speech Recognition* (ASR) telah berkembang dengan mengintegrasikan convolutional layer dan mekanisme self-attention dalam satu kerangka pemodelan. Convolutional layer berfungsi untuk mengekstraksi pola lokal serta karakteristik temporal jangka pendek dari sinyal akustik, seperti perubahan frekuensi dan dinamika spektral. Sementara itu, mekanisme self-attention memungkinkan model menangkap dependensi jangka panjang dan konteks global pada seluruh urutan fitur audio secara paralel, sehingga representasi akustik yang dihasilkan menjadi lebih komprehensif dan kontekstual (Loubser et al., 2024a), meskipun demikian, model berbasis *Transformer* memiliki kompleksitas parameter yang tinggi sehingga berpotensi mengalami *overfitting*, terutama ketika dilatih pada dataset berukuran kecil atau data dengan tingkat noise yang tinggi. Oleh karena itu, strategi regularisasi, augmentasi data, dan teknik normalisasi menjadi komponen penting dalam meningkatkan generalisasi model (Latif et al., 2025).

LibriSpeech merupakan salah satu dataset standar yang paling banyak digunakan dalam penelitian *Automatic Speech Recognition* (ASR) modern. Dataset ini terdiri dari rekaman suara berbahasa Inggris yang bersumber dari *audiobook* domain publik, dengan kualitas perekaman yang relatif tinggi serta variasi pembicara yang beragam. *LibriSpeech* umumnya dibagi ke dalam subset *clean* dan *other*, di mana subset *clean* memiliki tingkat kebisingan rendah, sedangkan subset *other* mengandung variasi *noise* dan kondisi akustik yang lebih kompleks. Pembagian ini memungkinkan evaluasi performa model ASR pada kondisi ideal maupun menantang secara akustik (Pratap et al., 2020). Dalam penelitian-penelitian ASR terkini, *LibriSpeech* sering digunakan sebagai *benchmark* utama untuk mengukur kemampuan generalisasi model, khususnya dalam menilai *Word Error Rate* (WER) pada berbagai arsitektur berbasis *deep learning* seperti RNN dan *Transformer*. Penggunaan *dataset* ini secara luas juga memungkinkan perbandingan yang objektif antar pendekatan karena konfigurasi data pelatihan, validasi, dan pengujian telah terstandarisasi (Baevski et al., 2020). Maka dari itu, *LibriSpeech* sangat cocok dijadikan *dataset* untuk penelitian ini.

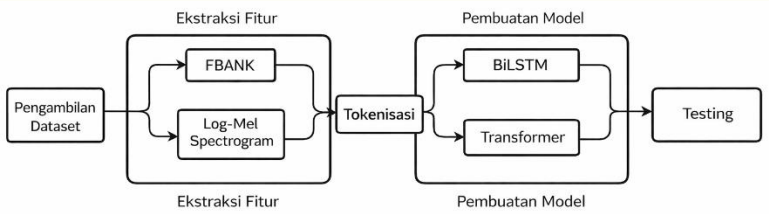
Terdapat penelitian yang membandingkan beberapa arsitektur *Convolutional Neural Network* (CNN) untuk *speech-to-text*. Penelitian yang dilakukan oleh (Kafle et al., 2024) membandingkan arsitektur BiLSTM, *Transformer*, dan kombinasi CNN-ResNet-BiLSTM untuk mengenali ucapan bahasa Nepal menggunakan *dastaset* OpenSLR dengan ekstraksi fitur yang digunakan *Mel Frequency Cepstral Coefficient* (MFCC) dengan hasil dari penelitian ini menunjukkan model *hybrid* (CNN-resNet-BiLSMT) lebih unggul dengan nilai CER (*Character Error Rate*) 17.06%, akurasi 82.94%. Sedangkan arsitektur *Transformer* mendapatkan CER 22.72% dan akurasi 77.28%, lalu BiLSTM dengan nilai CER 23.69% dan akurasi 76.31%. Penelitian yang dilakukan oleh (Zeyer et al., 2019) membandingkan LSTM dengan *Transformer* memakai *dataset Librispeech* yang dimana penelitian ini memberikan hasil yang dimana model *Transformer* memberikan performa lebih baik dari LSTM dalam tugas ASR *end-to-end*, dengan pelatihan yang lebih cepat dan stabil. Pada dataset *LibriSpeech* (1000h, 12.5 epoch), *Transformer* + *SpecAugment* + *LM-EOS* mencatatkan WER terbaik yaitu 2.51% (*dev-clean*) dan 7.42% (*test-other*), sedikit lebih baik dibandingkan LSTM + *SpecAugment* + *LM-EOS* yang mencatat 2.56% dan 7.46%. Di Switchboard (300h), *Transformer* mencapai WER 9.3% (SWB) dan 20.3% (CH), unggul dibanding LSTM (9.9% / 21.5%). Pada *TED-LIUM-v2* (20h), *Transformer* + *LM-EOS* juga memberikan WER terendah yaitu 10.3% (dev) dan 8.8% (test), mengalahkan LSTM yang mencatat 10.5% dan 9.0%. Meski *Transformer* cenderung lebih mudah overfitting, hal ini dapat diatasi dengan data *augmentation* seperti *SpecAugment* yang meningkatkan performa hingga 33% relatif, dibanding peningkatan 15% pada LSTM. Penelitian

lain yang membahas perbandingan arsitektur *Transformer* dengan RNN yang dilakukan oleh (Karita et al., 2019) yang dimana hasil penelitiannya menunjukkan arsitektur *Transformer* secara konsisten mengungguli RNN dalam aplikasi *end-to-end* ASR, ST, dan TTS.yang dimana pada ASR, *Transformer* mencatat WER 2.2% (*dev-clean)* dan 5.6% (*dev-other)* di *LibriSpeech*, lebih baik dari RNN yang dimana mencatat 3.3% (*dev-clean*) dan 10.8% (*dev-other*) serta di 13 dari 15 dataset. adapun penelitian yang yang membahas arsitektur BiLSTM dengan ekstraksi fitur *Mel-spectogram* dan MFCC oleh (Tami, 2020). Dari penelitian terdahulu yang telah dilakukan menunjukkan bahwa *Transformer* mengungguli hampir dari semua penelitian terdahulu tersebut. Tetapi belum ada menunjukkan perbandingan dengan BiLSTM yang dimana merupakan arsitektur yang mengatasi kekurangan dari LSTM.

Hal ini menunjukkan celah penelitian dalam perbandingan antara arsitektur BiLSTM dengan *Transformer* yang memiliki kelebihannya masing-masing, maka dalam penelitian ini akan membandingkan performa arsitektur BiLSTM dan *Tranformer* dengan ekstraksi fitur *Log-Mel Spectogram* dan dan *Filter Bank Energies* (FBANK) dengan dataset *LibriSpeech*.

METODE

Penelitian ini menggunakan desain eksperimen berbasis komputasi untuk membandingkan dua model *Speech-to-Text*, yaitu BiLSTM dan *Transformer*, serta membandingkan dua ekstraksi fitur, yaitu *Log-mel Spectogram* dan FBANK. Algoritma ini dipilih karena kedua model bekerja dengan cara yang berbeda, BiLSTM membaca data suara secara urut dari awal hingga akhir, sedangkan Transformer menggunakan mekanisme self-attention yang dapat melihat keseluruhan urutan secara bersamaan. Dan juga kedua ekstraksi fitur ini dipilih karena, *Log-Mel Spectrogram* adalah representasi suara yang sudah dipadatkan ke dalam skala logaritmik sehingga pola frekuensi lebih mudah dibaca oleh model. Sementara itu, FBANK (*Filterbank Energies*) merupakan fitur berupa energi dari setiap filter mel yang masih lebih mentah dan detail. Fitur ini sering digunakan pada sistem ASR karena mampu mempertahankan informasi frekuensi yang penting. Desain penelitian ini dipilih karena kedua model memiliki cara kerja yang berbeda. Dengan membandingkan keduanya, penelitian ini dapat melihat model mana yang lebih cocok untuk jenis fitur tertentu. Pendekatan eksperimen ini cocok digunakan karena performa kedua model dapat diukur secara jelas menggunakan metrik standar, yaitu *Word Error Rate* (WER) dan *Character Error Rate* (CER). Dapat dilihat prosedur desain eksperimen pada Gambar 1.



Gambar 1. Alur penelitian Model BiLSTM dan *Transformer*

Gambar 1. menunjukkan alur metodologi yang digunakan dalam penelitian ini untuk melakukan pengenalan suara (*Automatic Speech Recognition*) menggunakan dua arsitektur model, yaitu BiLSTM dan *Transformer*. Proses dimulai dari pengambilan dataset audio dan transkripsi teks dari korpus *LibriSpeech*. Setelah data diperoleh, tahap berikutnya adalah ekstraksi fitur akustik untuk mengubah sinyal audio mentah menjadi representasi numerik berupa *Log-Mel Spectrogram* dan FBANK, yang kemudian akan digunakan sebagai masukan bagi model. Selanjutnya dilakukan proses tokenisasi, di mana model BiLSTM menggunakan tokenisasi word-level sedangkan Transformer menggunakan tokenisasi *sub-word* berbasis *SentencePiece* agar lebih fleksibel dalam menangani variasi kata. Setelah fitur dan tokenisasi siap, dilakukan pembentukan model dengan menyusun arsitektur BiLSTM dan *Transformer* sesuai konfigurasi yang telah ditetapkan. Model tersebut kemudian dilatih menggunakan algoritma *CTC Loss* untuk mencocokkan urutan suara dengan teks referensi. Tahap akhir adalah evaluasi dan *testing* kinerja model dengan menghitung nilai *Word Error Rate* (WER) dan *Character Error Rate* (CER) untuk menilai tingkat akurasi serta kemampuan masing-masing model dalam mengonversi

audio menjadi teks. Proses ini memberikan gambaran menyeluruh mengenai langkah-langkah yang ditempuh dalam penelitian serta hubungan antar setiap tahapan dalam pipeline ASR yang dibangun.

Dataset

Pada penelitian ini, *dataset* yang digunakan adalah *dataset LibriSpeech* yang dimana memiliki 2 tipe *subset* yang berbeda, yaitu *subsets clean* yang berisi data yang memiliki WER kecil dan *other* yang memiliki WER relatif lebih besar dan untuk tujuan data *training*, data uji, atau data validasi. Dataset ini memiliki aksen yang banyak dan memiliki *impact* penting untuk tingkat tinggi dan rendah WER (Pratap et al., 2020). *Subset* yang dipakai bisa dilihat pada Tabel 1. Yang memiliki fungsinya masing-masing pada setiap *subset*-nya pada berikut :

Tabel 1. *Dataset LibriSpeech* yang dipakai

	Total Durasi	**Fungsi**
train-clean-100	100,6 Jam	Data pelatihan clean dengan durasi hingga 100 jam
Dev-clean	5,4 Jam	Data validasi *clean*
Dev-other	5,12 – 5,4 jam	Data validasi berisi data *noise*
Test-clean	5,4 jam	Data testing *clean*
test-other	5,10 – 5,12 jam	Data testing data *noise*

Ekstraksi Fitur

1. Log-Mel Spectogram

Mel spectogram adalah spektrum frekuensi linier yang diubah ke dalam skala *mel* atau persepsi manusia yang dilakukan melalui filter bank segitiga (Shehab et al., 2024), sedangkan *log-mel spectogram* adalah log yang diaplikasikan di hasil *mel spectogram* (Abduh et al., 2020). Fungsi tradisional dari *frequency to mel scale* (Gao et al., 2023) yaitu :

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad \text{(Gao et al., 2023)} \quad (1)$$

Setelah konfersi dari *frequency* ke *mel scale* sudah tercapai, kita bisa membuat *mel filter bank* yang dimana persamaan yang digunakan adalah :

$$\begin{cases} 0, jika\ k < f_{m-1}\ and\ k > f_{m+1} \\ \dfrac{k - f_{m-1}}{f_m - f_{m-1}}, jika\ f_{m-1} \le k \le f_m \\ \dfrac{f_{m+1} - k}{f_{m+1} - f_m}, jika\ f_m \le k \le f_{m+1} \end{cases} \quad \text{(Gao et al., 2023)} \quad (2)$$

Yang dimana f adalah *frequency* dari *mel-scale*nya, dan m adalah total filter per-*band*. Setelah itu, filter tersebut akan di *multiplied* dengan hasil *spectogram* yang diperoleh persamaan diatas untuk memperoleh *mel scale spectogram*, dengan fungsi yang digunakan yaitu :

$$MS\,(x) = \text{Spectogram}(x)\ \odot B_m(k) \quad \text{(Gao et al., 2023)} \quad (3)$$

Setelah itu, kita perlu mengubah *spectogram* ke *dB unit* dengan melakukan operasi algoritma untuk mendapatkan hasil dari *log-mel spectogram*, fungsi yang digunakan dapat dilihat sebagai berikut:

$$LogMelSpec = 10 * \log_{10}(MS(x)) - 10 * \log_{10}(ref) \quad (4)$$

ref merupakan *reference value* berdasarkan dari mana skala relatif *amplitude* $MS(x)$ (Abduh et al., 2020). Dari fungsi (4) diatas hasil dari *log-mel spectogram* dapat ditemukan.

2. FBANK

FBANK adalah metode ekstraksi fitur yang menggunakan *filter bank mel* (*gaussian*) untuk menghitung energi spektral dalam tiap *sub-band*nya, yang dimana menggunakan rumus dasar konversi frekuensi ke skala *mel* dan filter *triangular* (Zhang & Wu, 2019). FBANK menggunakan filter *triangle* berdasarkan *mel scale* untuk mensimulasikan persepsi dari telinga manusia, yang dimana memiliki

karakterisitik *non-linier frequency* yang memilki respon dan lebih sensitif terhadap frekuensi rendah (Peng et al., 2025). Untuk penelitian ini, digunakan fungsi-fungsi yang digunakan oleh (Peng et al., 2025) untuk mencari frekuensi yang tepat yaitu:

$$Mel = 2595 \,.\log_{10}(\frac{f}{700} + 1) \qquad \text{(Zhang \& Wu, 2019)} \qquad (5)$$

Dimana f merupakan *linier frequency* dalam satuan Hz, dan fungsi *filter triangle* untuk menemukan *isometric trangular* untuk menemukan *value scale mel* dengan fungsi sebagai berikut :

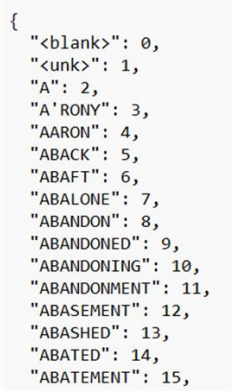
$$H_{mel}(k) = \begin{cases} \frac{k-f(m-1)}{f(m)-f(m-1)}, & jika\ f(m-1) \le k \le f(m) \\ 0, & sebaliknya \\ \frac{f(m-1)-k}{f(m+1)-f(m)}, & jika\ f(m) \le k \le f(m+1) \end{cases} \qquad \text{(Zhang \& Wu, 2019)} \qquad (6)$$

$$f = 700 \,.\left(10^{\frac{Mel}{2595}} - 1\right) \text{(Zhang \& Wu, 2019)} \qquad (7)$$

Dimana m adalah frekuensi dari *mel-central* dari m^{th} yang digunakan pada fungsi (6) dengan $f(m-1)$ sebagai *lower mel frequence*, $f(m)$ sebagai *center mel frequence*, dan $f(m+1)$ sebagai *higher mel frequence*. Adapun fungsi (7) digunakan untuk menghasilkan *triangle filter* yang bertujuan untuk mengubah kembali menjadi *linier frequency* (Peng et al., 2025).

Tokenisasi

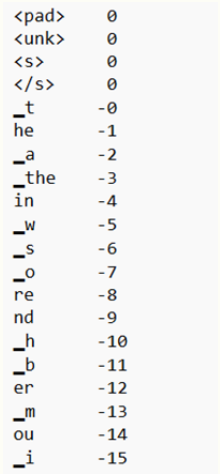
Proses tokenisasi dilakukan untuk mengubah teks transkripsi menjadi urutan token numerik yang dapat diproses oleh model. Pada penelitian ini digunakan dua metode tokenisasi yang disesuaikan dengan karakteristik masing-masing arsitektur, yaitu word-level dan *sub-word*. Pada model BiLSTM, digunakan tokenisasi *word-level*, di mana setiap kata direpresentasikan sebagai satu token. Pendekatan ini sesuai untuk model berbasis RNN karena arsitektur berurutan seperti BiLSTM sensitif terhadap panjang input dan performanya lebih stabil jika ukuran *vocabulary* tidak terlalu besar. Penelitian pada sistem ASR berbasis RNN menunjukkan bahwa penggunaan *vocabulary* terbatas dapat mengurangi panjang urutan dan menekan risiko *vanishing gradient*, sehingga model lebih mudah dilatih (Hannun et al., 2019).Contoh tokenisasi *word-level* untuk model BiLSTM dapat diliat pada Gambar 2.



```
{
  "<blank>": 0,
  "<unk>": 1,
  "A": 2,
  "A'RONY": 3,
  "AARON": 4,
  "ABACK": 5,
  "ABAFT": 6,
  "ABALONE": 7,
  "ABANDON": 8,
  "ABANDONED": 9,
  "ABANDONING": 10,
  "ABANDONMENT": 11,
  "ABASEMENT": 12,
  "ABASHED": 13,
  "ABATED": 14,
  "ABATEMENT": 15,
```


Gambar 2. Tokenisasi *word-level* untuk model BiLSTM

Sedangkan, model *Transformer* pada penelitian ini menggunakan tokenisasi *sub-word* berbasis *SentencePiece*. Pendekatan ini lebih fleksibel dalam menangani kata baru *(out-of-vocabulary*) dan mampu memecah kata panjang menjadi unit yang lebih kecil. Berbagai penelitian terbaru menunjukkan bahwa tokenisasi *sub-word* secara konsisten memberikan WER lebih rendah pada sistem ASR modern berbasis *Transformer* dibandingkan *word-level* atau *char-level*, terutama pada bahasa dengan struktur morfologi kompleks. Selain itu, evaluasi komprehensif terhadap tokenisasi pada arsitektur *self-attention* menunjukkan bahwa metode seperti BPE atau *Unigram SentencePiece* menghasilkan representasi yang lebih padat dan stabil sehingga cocok untuk transformer encoder yang memproses urutan secara paralel (Baneras-Roux et al., 2024). Contoh tokenisasi *sub-word* menggunakan *SentencePiece* untuk model *Transformer* dapat diliat pada Gambar 3.



Gambar 3. Tokenisasi *sub-word* untuk model *Transformer*

Dengan demikian, tokenisasi *word-level* lebih sesuai untuk BiLSTM yang bekerja secara sekuensial, sedangkan tokenisasi *sub-word* menjadi pilihan ideal untuk *Transformer* karena lebih efisien, mampu menangani OOV, dan mendukung performa pengenalan suara yang lebih baik.

Pembentukan arsitektur model

Pada tahap ini, kedua model BiLSTM dan *Transformer* dibangun berdasarkan konfigurasi arsitektur yang telah ditentukan untuk memastikan proses pelatihan berlangsung secara konsisten dan dapat dibandingkan secara objektif. Setiap model dirancang untuk memproses fitur akustik (*Log-Mel Spectrogram* dan FBANK) yang telah diekstraksi pada tahap sebelumnya dan menghasilkan prediksi urutan token yang sesuai dengan transkripsi referensi.

1. Arsitektur BiLSTM

BiLSTM merupakan salah satu dari variasi LSTM yang dimana ide awalnya berasal dari proses *foward training* dan *backward training* dari 2 LSTM yang saling terhubung menuju *output layer*, dari struktur ini bisa semakin memahami konteks dari masa depan dan masa lalu dari setiap poin di *input layer* (Rao et al., 2017). Representasi dua arah ini terbukti efektif pada tugas ASR konvensional karena mampu mempertahankan konteks jangka panjang pada urutan akustik (Hannun et al., 2019). Arsitektur ini umumnya terdiri dari beberapa lapisan BiLSTM, dilengkapi dengan dropout untuk mencegah *overfitting* dan lapisan *fully-connected* untuk menghasilkan prediksi token. Model dioptimalkan dengan *Connectionist Temporal Classification* (CTC), yang memungkinkan pencocokan urutan *audio* dan teks tanpa memerlukan alignment *frame*-per-*frame*. Hasil akhir berupa probabilitas token pada setiap langkah waktu, yang kemudian di *decode* menggunakan CTC *greedy decoding* atau *beam search*.

2. Arsitektur Transformer

Transformer dirancang dengan struktur *encoder-only*, yang terdiri dari beberapa blok *encoder* dengan komponen utama berupa *multi-head self-attention* dan *feed-forward network*. Mekanisme *self-attention* memungkinkan model memperhatikan seluruh posisi dalam urutan secara paralel, sehingga *Transformer* lebih efisien dalam mempelajari hubungan konteks jarak jauh dibandingkan model RNN. Penelitian terkini menunjukkan bahwa pendekatan *self-attention* memberikan peningkatan akurasi pada berbagai sistem ASR modern (Karita et al., 2019).Komponen utama arsitektur Transformer yang dimana seperti, *Positional Encoding* untuk menyisipkan informasi urutan ke dalam *embedding*, *Multi-Head Self-Attention* untuk menangkap hubungan antar *frame* akustik, *Feed-Forward Layer* untuk *transformasi non-linear*, *Layer Normalization* dan *residual connection* untuk stabilitas pelatihan,
CTC Loss sebagai *objective function* untuk menyelaraskan *output* dengan target token. Token hasil tokenisasi *sub-word* digunakan sebagai target sehingga *embedding* yang dihasilkan lebih padat dan mudah dipelajari oleh model. Arsitektur ini memperlihatkan blok *encoder decoder* yang sudah ditunjukkan pada gambar 3 yang menunjukkan WER lebih rendah dibandingkan model konvensional (Moritz et al., 2020).

Hyperparameter dan Proses Pelatihan

Kedua model dilatih menggunakan menggunakan *hyperparameter* yang dapat dilihat pada tabel2.

Tabel 2. *Hyperparameter training* BiLSTM dan *Transformer*

	BiLSTM	***Transformer***
Epoch	20	20
Learning rate	3e-4	1e-4
Optimizer	Adam	Adam
Batch size	2	8
Loss function	CTC Loss	CTC Loss
Gradient clipping	5.0	5.0

Selama pelatihan, diterapkan mekanisme *early stopping* berdasarkan performa pada data validasi guna mengurangi risiko *overfitting*. Nilai *loss*, WER, dan CER dipantau pada setiap *epoch* untuk mengevaluasi stabilitas pembelajaran dan mendeteksi kemungkinan degradasi kinerja. Setelah model selesai dilatih, dilakukan pengujian menggunakan subset *test-clean* dan *test-other* dari *LibriSpeech* untuk menilai kemampuan generalisasi model dalam berbagai kondisi akustik.

Evaluasi dan Testing

Word Error Rate (WER) adalah metrik yang mengukur akurasi sistem *voice recognition* atau teks yang didasarkan pada *levenshtein distance* (Varod et al., 2021). *Character Error Rate* (CER) yang memiliki fungsi metrik yang hampir sama dengan WER tetapi yang dihitung per-karakter daripada WER yang menghitung per-kata (Varod et al., 2021). Persamaan untuk mendapatkan nilai tersebut bisa dilihat rumus umum WER (8) dan CER (9) sebagai berikut :

$$WER = \frac{S + D + I}{N} \tag{8}$$

$$CER = \frac{S_c + D_c + I_c}{N_c} \tag{9}$$

Dimana :
S = jumlah penggantian kata
D = jumlah penghapusan kata
I = jumlah penambahan kata
N = jumlah kata
S_c = jumlah penggantian karakter
D_c = jumlah penghapusan karakter
I_c = jumlah penambahan karakter
N_c = jumlah karakter

Kedua metrik ini digunakan untuk mendapatkan hasil validasi dan pengujian dari kedua Algoritma dan kedua ekstraksi fitur penelitian ini. Penelitian sebelumnya menunjukkan bahwa performa *Automatic Speech Recognition* (ASR) sangat dipengaruhi oleh arsitektur model, jenis fitur akustik, dan karakteristik dataset (Hung et al., 2018), melaporkan bahwa pada dataset Aurora-4, model GMM-HMM dengan MFCC menghasilkan WER 31,70%, sedangkan DNN-HMM dengan FBANK mencapai 27,33%, yang menunjukkan bahwa pemilihan fitur memiliki peran penting dalam meningkatkan ketahanan terhadap *noise*. Pada pendekatan berbasis *recurrent neural network* (Zeyer et al., 2017)menggunakan BLSTM dengan fitur 50-dim VTLN *Gammatone* dan memperoleh WER 17,1% pada *Switchboard*, namun performanya menurun menjadi 52,8% pada *Babel Javanese*. Sementara itu, *Transformer* skala kecil yang diuji oleh (Loubser et al., 2024b) pada LibriSpeech menunjukkan perbedaan signifikan antara data bersih dan bising, dengan WER 28,91% pada *dev-clean* dan 51,85% pada *dev-other*.

Meskipun berbagai arsitektur telah dievaluasi, sebagian besar penelitian tersebut berfokus pada satu jenis fitur atau satu pendekatan model secara terpisah. Perbandingan langsung antara arsitektur sekuensial seperti BiLSTM dan arsitektur berbasis *self-attention* seperti Transformer dengan

menggunakan konfigurasi fitur yang berbeda masih relatif terbatas. Selain itu, kajian yang secara sistematis menganalisis pengaruh kombinasi jenis fitur (Log-Mel dan FBANK) terhadap dua arsitektur yang berbeda dalam satu kerangka eksperimen yang konsisten belum banyak dilaporkan. Oleh karena itu, penelitian ini menawarkan kebaruan dengan melakukan evaluasi komparatif antara BiLSTM dan *Transformer* menggunakan dua jenis representasi fitur akustik dalam pengaturan eksperimen yang seragam, sehingga dapat memberikan pemahaman yang lebih komprehensif mengenai kesesuaian antara arsitektur model dan karakteristik fitur terhadap kualitas transkripsi. Untuk hasil dari penelitian sebelumnya yang sudah dijelaskan dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian ASR penelitian terdahulu

Peneliti	Arsitektur	Fitur	Dataset	WER
(Hung et al., 2018)	GMM-HMM	MFCC	Aurora-4	31.70%
(Hung et al., 2018)	DNN-HMM	FBANK	Aurora-4	27.33%
(Zeyer et al., 2017)	BLSTM	50-dim VTLN Gammatone	Switchboard (300h)	17.1 %
(Zeyer et al., 2017)	BLSTM	50-dim VTLN Gammatone	Babel Javanese	52.8 %
(Loubser et al., 2024b)	Tranformer small scale	Log-mel	LibriSpeech (dev-clean)	28.91 %
(Loubser et al., 2024b)	Tranformer small scale	Log-mel	LibriSpeech (dev-other)	51.85 %
(Loubser et al., 2024b)	Tranformer small scale	Log-mel	LibriSpeech (test-clean)	30.14 %
(Loubser et al., 2024b)	Tranformer small scale	Log-mel	LibriSpeech (test-other)	53.75 %

HASIL DAN PEMBAHASAN

Proses analisis pada penelitian ini dilakukan melalui beberapa tahapan berurutan, mulai dari persiapan ulang dataset, ekstraksi fitur, tokenisasi, pelatihan model, hingga evaluasi performa. Setiap kombinasi arsitektur dan jenis fitur diuji menggunakan dua metrik utama, yaitu *Word Error Rate* (WER) dan *Character Error Rate* (CER), untuk menentukan konfigurasi yang memberikan hasil paling optimal. Tahap awal dimulai dengan ekstraksi fitur menggunakan dua pendekatan, yaitu *Log-Mel Spectrogram* dan FBANK (*Filterbank Energies*). Tahap ini memastikan seluruh berkas audio memiliki bentuk representasi numerik yang seragam sehingga siap digunakan pada proses pelatihan baik untuk model Bi-LSTM maupun *Transformer*.

Pada tahap tokenisasi, pendekatan *word-level* digunakan pada model BiLSTM sedangkan, model *Transformer* menggunakan pendekatan *sub-word* seperti *SentencePiece*. Setelah seluruh proses persiapan selesai, dataset kemudian digunakan untuk melatih dua arsitektur berbeda, yaitu BiLSTM dan *Transformer*, untuk melihat pengaruh variasi fitur dan tokenisasi terhadap performa pengenalan suara. Proses training dilaksanakan secara bertahap dengan memanfaatkan mekanisme early stopping untuk mencegah overfitting, di mana pelatihan akan berhenti ketika peningkatan performa pada data validasi tidak lagi signifikan. Setiap iterasi pelatihan menghasilkan nilai loss dan akurasi yang dipantau untuk mengevaluasi stabilitas serta kemampuan model dalam mempelajari pola akustik. Seluruh prosedur ini dilakukan secara konsisten pada kedua model, sehingga hasil akhir dapat dibandingkan secara objektif berdasarkan metrik WER dan CER. Pada tabel 5 hingga tabel 8 merupakan jabaran hasil training.

Tabel 5. Hasil Training BiLSTM dengan Fitur FBANK

Epoch	Test-clean (WER/CER)	Test-Other (WER/CER)
1	99.04%/95.54%	98.52%/96.54%
2	88.89%/72.46%	85.24%/67.24%
3	62.16%/45.69%	76.06%/57.22%
4	49.99%/35.35%	69.85%/52.37%
5	43.61%/30.38%	66.49%/49.26%
6	38.40%/25.64%	63.04%/45.73%
7	35.68%/23.42%	60.04%/42.56%
8	34.29%/22.39%	59.08%/41.92%
9	33.07%/21.72%	57.57%/40.41%
10	32.13%/21.12%	56.95%/40.61%
11	32.91%/21.76%	56.10%/40.23%
12	31.41%/20.44%	55.30%/38.71%
13	31.43%/20.42%	54.77%/38.53%
14	30.31%/19.82%	54.32%/37.82%
15	29.88%/19.28%	53.87%/38.08%
16	29.84%/19.26%	53.36%/37.32%
17	29.99%/19.47%	53.54%/37.59%
18	29.66%/19.25%	53.36%/37.32%
19	29.71%/19.23%	53.31%/37.28%
20	29.64%/19.21%	53.30%/37.28%

Tabel 6. Hasil Training BiLSTM dengan Fitur Log-Mel

Epoch	Test-clean (WER/CER)	Test-Other (WER/CER)
1	96.44%/88.19%	99.49%/95.57%
2	64.72%/50.35%	79.37%/60.30%
3	53.18%/40.16%	69.29%/50.08%
4	46.35%/33.66%	65.87%/47.60%
5	43.82%/31.67%	66.94%/49.01%
6	39.55%/28.32%	63.34%/46.60%
7	37.34%/26.22%	61.68%/44.26%
8	35.29%/25.09%	64.10%/47.40%
9	33.94%/23.70%	57.31%/40.27%
10	32.47%/22.52%	57.73%/41.58%
11	31.80%/22.19%	55.64%/38.97%
12	31.14%/21.61%	55.83%/39.32%
13	30.32%/20.92%	54.79%/38.19%
14	29.96%/20.71%	54.58%/37.70%
15	29.80%/20.56%	53.81%/38.01%
16	29.52%/20.48%	53.07%/37.53%
17	29.59%/20.25%	52.10%/36.36%
18	29.60%/20.45%	52.09%/36.79%
19	29.51%/20.35%	51.47%/35.83%
20	29.52%/20.38%	51.18%/36.50%

Tabel 7. Hasil Training ***Transformer*** dengan Fitur FBANK

Epoch	Test-clean (WER/CER)	Test-Other (WER/CER)
1	100.00%/100.00%	100.00%/100.00%
2	100.18%/89.65%	104.94%/86.44%
3	102.79%/89.01%	102.79%/89.23%
4	103.01%/90.09%	104.47%/93.24%
5	102.85%/89.81%	104.33%/92.83%
6	102.36%/85.16%	102.04%/88.31%
7	101.69%/85.20%	103.54%/87.93%
8	103.09%/85.38%	103.06%/85.93%
9	102.77%/85.78%	102.47%/85.42%
10	104.49%/85.92%	102.30%/85.13%
11	102.69%/84.61%	102.48%/84.32%
12	102.54%/84.23%	102.36%/84.36%
13	103.07%/84.14%	102.33%/83.96%
14	99.74%/82.19%	100.39%/82.98%
15	90.73%/74.33%	93.20%76.65%
16	85.12%/70.29%	86.26%/69.91%
17	81.56%/69.20%	82.55%/
18	78.03%/63.62%	80.07%/66.91%
19	75.11%/59.62%	76.00%/61.13%
20	72.40%/56.73%	73.33%/57.40%

Tabel 8. Hasil Training ***Transformer*** dengan Fitur Log-Mel

Epoch	Test-clean (WER/CER)	Test-Other (WER/CER)
1	100.00%/100.00%	100.00%/100.00%
2	99.77%/95.36%	99.94%/98.92%
3	99.56%/96.14%	99.34%/96.73%
4	99.63%/95.66%	99.66%/97.12%
5	98.47%/90.16%	98.77%/92.08%
6	95.17%/85.26%	93.58%/82.84%
7	79.38%/66.00%	83.15%/69.12%
8	69.15%/54.70%	77.13%/62.12%
9	62.02%/46.89%	72.61%/57.07%
10	56.75%/41.04%	68.83%/52.66%
11	52.83%/37.45%	66.45%/50.18%
12	49.73%/34.06%	63.84%/47.05%
13	47.37%/32.12%	61.44%/45.07%
14	45.37%/29.89%	60.07%/42.89%
15	43.74%/28.24%	58.85%/41.89%
16	42.36%/26.47%	57.16%/39.92%
17	41.32%/25.73%	55.76%/38.83%
18	39.71%/24.98%	55.03%/38.54%
19	38.65%/23.59%	53.65%/36.62%
20	38.36%/23.03%	53.69%/36.38%

Berdasarkan hasil evaluasi, model BiLSTM menunjukkan pola konvergensi yang stabil dan konsisten pada kedua jenis fitur. Penurunan WER dan CER terjadi secara signifikan pada 5–10 *epoch* pertama, yang mengindikasikan proses pembelajaran awal yang efektif dalam menangkap pola akustik. Setelah epoch ke-15, kurva penurunan mulai melandai, dengan WER *test-clean* mencapai 29,64% (FBANK) dan 29,52% (*Log-Mel*) pada *epoch* ke-20. Hal ini menunjukkan bahwa model telah mendekati titik konvergensi dan mengalami perbaikan yang semakin kecil di akhir pelatihan. Selain itu, selisih performa antara *test-clean* dan *test-other* relatif konsisten, menandakan bahwa BiLSTM memiliki kemampuan generalisasi yang cukup stabil terhadap variasi kondisi akustik, meskipun data *test-other* tetap lebih menantang.

Sebaliknya, model *Transformer* memperlihatkan dinamika pelatihan yang berbeda. Pada fitur FBANK, model mengalami kesulitan konvergensi pada fase awal, dengan WER bertahan di atas 100% hingga beberapa *epoch* pertama dan hanya turun menjadi 72,40% (*test-clean*) pada epoch ke-20. Hal ini mengindikasikan bahwa kombinasi arsitektur Transformer dengan FBANK kurang optimal dalam konfigurasi eksperimen ini, dikarenakan sensitivitas model terhadap representasi fitur atau keterbatasan data pelatihan. Namun, pada fitur Log-Mel, Transformer menunjukkan pola penurunan yang lebih progresif, terutama setelah epoch ke-7, dengan WER mencapai 38,36% (*test-clean*) dan 53,69% (*test-other*) pada epoch ke-20. Meskipun demikian, performa akhir masih berada di bawah BiLSTM, yang mengindikasikan bahwa dalam skenario ini arsitektur berbasis *recurrent* lebih stabil dibandingkan pendekatan *self-attention* murni.

Berdasarkan pelatihan yang telah dianalisis sebelumnya, hasil pengujian akhir pada Tabel 9 semakin memperjelas perbedaan karakteristik kedua arsitektur. Model BiLSTM terbukti sebagai arsitektur yang paling stabil dan konsisten pada seluruh kombinasi fitur. Pada subset *test-clean*, BiLSTM dengan FBANK dan Log-Mel masing-masing menghasilkan WER sebesar 29,34**%** dan 29,51%, yang merupakan performa terbaik dibandingkan konfigurasi lainnya. Pada subset *test-other*, tingkat kesalahan meningkat menjadi 55,86% (FBANK) dan 53,74% (*Log-Mel*), namun tetap lebih rendah dibandingkan model *Transformer* pada konfigurasi yang sama. Perbedaan kecil antara kedua fitur menunjukkan bahwa FBANK sedikit lebih unggul pada data bersih, sedangkan *Log-Mel* memberikan generalisasi yang lebih baik pada kondisi yang lebih beragam.

Tabel 9. Hasil *Testing* BiLSTM dan *Transformer*

	Fitur	**Test-clean (WER/CER)**	**Test-Other (WER/CER)**
BiLSTM	FBANK	29.34% / 18.94%	55.86% / 39.22%
BiLSTM	Log-Mel	29.51% / 20.25%	53.74% / 38.46%
Transformer	Log-Mel	38.01% / 23.38%	55.65% / 38.54%
Transformer	FBANK	62.22% / 46.96%	76.25% / 59.53%

Sebaliknya, performa *Transformer* sangat dipengaruhi oleh jenis fitur yang digunakan. Kombinasi Transformer dengan Log-Mel memberikan hasil terbaik di antara konfigurasi Transformer, dengan WER **38,01%** pada *test-clean* dan 55,65% pada *test-other*. Namun, ketika menggunakan FBANK, performanya menurun secara signifikan hingga mencapai 62,22% pada *test-clean* dan 76,25% pada *test-other*. Hal ini mengindikasikan bahwa arsitektur berbasis *self-attention* cenderung lebih sensitif terhadap fitur dan kurang memiliki bias sekuensial eksplisit seperti pada model RNN. Secara keseluruhan, hasil pengujian ini menegaskan bahwa BiLSTM lebih *robust* dan stabil pada konfigurasi eksperimen yang digunakan, sementara *Transformer* menunjukkan potensi yang baik tetapi memerlukan representasi fitur yang lebih sesuai untuk mencapai performa yang lebih optimal.

KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa karakteristik fitur dan kesesuaian dengan arsitektur model sangat menentukan kualitas transkripsi suara. BiLSTM terbukti sebagai model yang paling efektif untuk skala data yang terbatas, dengan performa yang stabil baik pada fitur FBANK maupun *Log-Mel*. Keberhasilan BiLSTM menunjukkan bahwa model berurutan dengan bias temporal masih memiliki keunggulan dalam mempelajari representasi akustik sederhana, terutama ketika ukuran dataset tidak terlalu besar dan kompleksitas struktur bahasa masih dapat ditangani oleh tokenisasi *word-level*.

Sementara itu, model Transformer sebenarnya memiliki potensi yang besar, tetapi membutuhkan fitur yang lebih rapi dan terstruktur agar bisa bekerja dengan maksimal. Hasil pengujian menunjukkan bahwa *Transformer* lebih cocok menggunakan fitur *Log-Mel* dibandingkan FBANK, karena mekanisme self-attention bekerja lebih baik ketika pola suara sudah diringkas dan lebih jelas. Walaupun pada penelitian ini performa *Transformer* belum melebihi BiLSTM, arah peningkatannya menunjukkan bahwa model ini bisa menghasilkan akurasi yang lebih tinggi jika dilatih dengan data yang lebih banyak, menggunakan teknik regularisasi tambahan, dan didukung oleh pengaturan tokenisasi sub-word yang lebih tepat.

SARAN

Penelitian ini membuka peluang penelitian lebih lanjut, disarankan pada eksplorasi optimalisasi *Transformer* melalui penambahan ukuran model dan *epoch* yang dijalankan, penerapan *learning schedule* yang lebih baik, dan eksplorasi teknik ekstraksi fitur data untuk memperkaya variasi akustik. Selain itu, penjelajahan teknik pretrained seperti *wav2vec 2.0* atau HuBERT berpotensi meningkatkan performa secara signifikan, terutama dalam kondisi akustik yang bising atau tidak terstruktur. Dari sisi aplikasi, temuan penelitian ini dapat menjadi dasar dalam pengembangan ASR untuk bahasa dengan sumber daya terbatas, di mana pemilihan fitur dan arsitektur yang tepat dapat mengurangi kebutuhan data pelatihan dalam jumlah besar. Dengan demikian, penelitian ini tidak hanya memberikan gambaran empiris mengenai efektivitas BiLSTM dan *Transformer*, tetapi juga menekankan pentingnya kesesuaian fitur-model sebagai fondasi pengembangan ASR yang lebih efisien dan adaptif di masa mendatang.

DAFTAR PUSTAKA

Abduh, Z., Nehary, E. A., Abdel Wahed, M., & Kadah, Y. M. (2020). Classification Of Heart Sounds Using Fractional Fourier Transform Based Mel-Frequency Spectral Coefficients And Traditional Classifiers. *Biomedical Signal Processing and Control*, *57*, 101788. https://doi.org/10.1016/j.bspc.2019.101788

Ahlawat, H., Aggarwal, N., & Gupta, D. (2025). Automatic Speech Recognition: A Survey Of Deep Learning Techniques And Approaches. *International Journal of Cognitive Computing in Engineering*, *6*, 201–237. https://doi.org/10.1016/j.ijcce.2024.12.007

Baneras-Roux, T., Rouvier, M., Wottawa, J., & Dufour, R. (2024). A Comprehensive Analysis of Tokenization and Self-Supervised Learning in End-to-End Automatic Speech Recognition applied on French Language. *HAL Open Science*.

Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Computation and Language*. https://doi.org/10.48550/arXiv.2006.11477

Dong, L., Xu, S., & Xu, B. (2018). Speech-Transformer: A No-Recurrence Sequence-To-Sequence Model For Speech Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2018-April*, 5884–5888. https://doi.org/10.1109/ICASSP.2018.8462506

Feng, Y. (2024). Intelligent Speech Recognition Algorithm In Multimedia Visual Interaction Via Bilstm And Attention Mechanism. *Neural Computing and Applications*, *36*(5), 2371–2383. https://doi.org/10.1007/s00521-023-08959-2

Gao, D., Tang, X., Wan, M., Huang, G., & Zhang, Y. (2023). EEG Driving fatigue detection based on log-Mel spectrogram and convolutional recurrent neural networks. *National Library of Medicine, Mar 9:17:1136609.*. https://doi.org/10.3389/fnins.2023.1136609

Hannun, A., Lee, A., Xu, Q., & Collobert, R. (2019). Sequence-to-Sequence Speech Recognition with Time-Depth Separable Convolutions. *Computation and Language*. https://doi.org/10.48550/arXiv.1904.02619

Hung, J. W., Lin, J. S., & Wu, P. J. (2018). Employing Robust Principal Component Analysis For Noise-Robust Speech Feature Extraction In Automatic Speech Recognition With The Structure Of A Deep Neural Network. *Applied System Innovation*, *1*(3), 1–14. https://doi.org/10.3390/asi1030028

Kafle, A., Rajlawat, J., Shah, N., Paudel, N., & Thapa, B. (2024). Advancements in Nepali Speech Recognition : A Comparative Study of BiLSTM , Transformer , and Hybrid Models. *International Journal on Engineering Technology*, *2*(1), 96–105. https://doi.org/10.3126/injet.v2i1.72525

Kamarullah, R. (2024). Implementasi Teknologi Speech To Text Dalam Transkripsi Pembelajaran Online Dengan Algoritma Dynamic Time Warping. http://repository.unas.ac.id/id/eprint/10885

Karita, S., Order, A., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Enrique, N., Soplin, Y., Yamamoto, R., Wang, X., Watanabe, S., Yoshimura, T., & Zhang, W. (2019). A Comparative Study On Transformer Vs Rnn In Speech Applications. *IEEE Xplore*, *9*(2), 449–456. https://doi.org/10.1109/ASRU46091.2019.9003750

Latif, S., Zaidi, S. A. M., Cuayáhuitl, H., Shamshad, F., Shoukat, M., Usama, M., & Qadir, J. (2025). Transformers In Speech Processing: Overcoming Challenges And Paving The Future. *Computer Science Review*, *58*, 100768. https://doi.org/10.1016/j.cosrev.2025.100768

Loubser, A., De Villiers, P., & De Freitas, A. (2024a). End-To-End Automated Speech Recognition Using A Character Based Small Scale Transformer Architecture. *Expert Systems with Applications*, *252*. https://doi.org/10.1016/j.eswa.2024.124119

Lubis, N., Siambaton, M. Z., & Aulia, R. (2025). Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN). *Sudo Jurnal Teknik Informatika,* 3(3), 113–126. https://doi.org/10.56211/sudo.v3i3.583

Manohar, K., A R, J., & Rajan, R. (2023). Improving Speech Recognition Systems For The Morphologically Complex Malayalam Language Using Subword Tokens For Language Modeling. *Eurasip Journal on Audio, Speech, and Music Processing*, *2023*(1). https://doi.org/10.1186/s13636-023-00313-7

Moritz, N., Hori, T., Roux, J. Le, & Electric, M. (2020). Streaming Automatic Speech Recognition With The Transformer Model. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6074–6078. https://doi.org/10.1109/ICASSP40776.2020.9054476

Peng, C., Zhang, Y., Lu, J., Lv, D., & Xiong, Y. (2025). A Bird Vocalization Classification Method Based on Improved Adaptive Wavelet Threshold Denoising and Bidirectional FBank. *Research Square*, 1–19. https://doi.org/10.21203/rs.3.rs-4181087/v1

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., & Collobert, R. (2020a). MLS: A Large-Scale Multilingual Dataset For Speech Research. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2020-Octob*, 2757–2761. https://doi.org/10.21437/Interspeech.2020-2826

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., & Collobert, R. (2020b). MLS: A Large-Scale Multilingual Dataset For Speech Research. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2020-Octob*, 2757–2761. https://doi.org/10.21437/Interspeech.2020-2826

Rao, G., Huang, W., Feng, Z., & Cong, Q. (2017). LSTM With Sentence Representations For Document-Level Sentiment Classification. *Neurocomputing*, *308*, 49–57. https://doi.org/10.1016/j.neucom.2018.04.045

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020a). A Primer In Bertology: What We Know About How Bert Works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. https://doi.org/10.1162/tacl_a_00349

Shehab, S. A., Mohammed, K. K., Darwish, A., & Hassanien, A. E. (2024). Deep Learning And Feature Fusion-Based Lung Sound Recognition Model To Diagnoses The Respiratory Diseases. *Soft Computing*, *28*(19), 11667–11683. https://doi.org/10.1007/s00500-024-09866-x

Sherstinsky, A. (2023). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physical D: Nonlinear Phenomena, 404, March 2020,* 132306. https://doi.org/10.1016/j.physd.2019.132306

Tami, M. A. (2020). Speech To Text Menggunakan Algoritma Deep Bidirectional LSTM. http://repository.unsri.ac.id/id/eprint/31499

Varod, V. S., Jokisch, O., Sinha, Y., & Geri, N. (2021). A cross-language study of speech recognition systems for English, German, and Hebrew. *Journal of Applied Knowledge Management (OJAKM)*, *9*(1), 1–15. https://dx.doi.org/10.36965/ojakm.2021.9(1)1-15

Zeyer, A., Bahar, P., Irie, K., Schluter, R., & Ney, H. (2019). A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, 8–15. https://doi.org/10.1109/ASRU46091.2019.9004025

Zeyer, A., Doetsch, P., Voigtlaender, P., Schlüter, R., & Ney, H. (2017). A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition. *IEEE*. https://doi.org/10.1109/ICASSP.2017.7952599

Zhang, T., & Wu, J. (2019). Discriminative Frequency Filter Banks Learning With Neural Networks. *Eurasip Journal On Audio, Speech, And Music Processing*, *2019*(1), 1–16. https://doi.org/10.1186/s13636-018-0144-6