TDNN-F + HMM VS TRANSFORMER UNTUK SPEECH RECOGNITION BAHASA INGGRIS PADA EDGE DEVICE

Azarya Aditya Krisna Moeljono*, Muhammad Jauharul Fuady

Departemen Teknik Elektro dan Informatika, Universitas Negeri Malang, Malang, Indonesia
*Corresponding author, email: azarya.aditya.2205356@students.um.ac.id

doi: 10.17977/um068.v5.i6.2025.2

Kata kunci
Automatic Speech Recognition (ASR)
Edge Computing
TDNN-F + HMM
Transformer
Performance Benchmark

Abstrak
Implementasi Automatic Speech Recognition (ASR) pada perangkat edge menghadapi tantangan rekayasa dalam menyeimbangkan akurasi transkripsi dengan efisiensi komputasi. Penelitian ini melakukan perbandingan kinerja secara empiris antara arsitektur ASR hibrida, yaitu Time-Delay Neural Network-Factorized + Hidden Markov Model (TDNN-F+HMM) yang diimplementasikan melalui Vosk, dengan arsitektur end-to-end modern, yaitu Transformer, yang diimplementasikan menggunakan model Whisper varian tiny. Pengujian dilakukan secara langsung pada perangkat edge Raspberry Pi 4 Model B menggunakan subset data dari korpus LibriSpeech, dengan mengevaluasi dua metrik utama: Word Error Rate (WER) untuk akurasi dan waktu eksekusi untuk efisiensi. Hasil eksperimen menunjukkan bahwa arsitektur Transformer (Whisper) secara konsisten mencapai akurasi yang lebih unggul, dengan skor WER rata-rata 0.096 dibandingkan 0.108 untuk Vosk, yang merepresentasikan penurunan tingkat kesalahan relatif sebesar 11.1%. Namun, dalam hal efisiensi, arsitektur TDNN-F+HMM (Vosk) terbukti secara signifikan lebih cepat, dengan waktu eksekusi rata-rata 4.043 detik, hampir 80% lebih cepat dibandingkan Whisper yang mencatatkan 7.290 detik. Studi ini menyimpulkan bahwa terdapat trade-off yang jelas antara kedua arsitektur: Whisper menawarkan akurasi yang lebih tinggi, sementara Vosk memberikan latensi yang jauh lebih rendah. Temuan ini memberikan panduan berbasis bukti yang esensial bagi para pengembang dalam memilih arsitektur ASR yang optimal sesuai dengan prioritas kasus penggunaan spesifik, baik itu untuk aplikasi yang menuntut presisi tinggi maupun yang memerlukan responsivitas waktu nyata.

1. Pendahuluan

Bahasa Inggris berfungsi sebagai lingua franca global dan memainkan peran sentral dalam komunikasi internasional (Bhandari & Ghimire, 2025). Dalam konteks transformasi digital, bahasa ini menjadi dasar penting bagi interoperabilitas dan mendorong inovasi lintas batas di bidang sains dan teknologi. Perpaduan penggunaan bahasa universal dengan kemajuan teknologi telah melahirkan paradigma interaksi manusia-komputer baru yang menempatkan suara sebagai medium input utama. Di pusat perubahan ini terdapat teknologi pengenalan ucapan otomatis (automatic speech recognition, ASR), sebuah subdisiplin pemrosesan bahasa alami yang memungkinkan mesin mentranskripsi ucapan manusia menjadi teks yang dapat diproses komputer (Alharbi et al., 2021). Integrasi ASR ke berbagai perangkat, mulai dari asisten virtual hingga sistem kontrol industri, tidak hanya meningkatkan aksesibilitas tetapi juga mengubah ekspektasi pengguna terhadap interaksi yang lebih mulus, responsif, dan kontekstual, sehingga menjadi fondasi bagi pengembangan aplikasi generasi berikutnya.

Seiring kebutuhan interaksi cerdas meningkat, arsitektur komputasi bergeser dari model cloud terpusat ke paradigma edge computing yang menanggapi tuntutan latency rendah, perlindungan privasi data yang lebih baik, dan kemampuan operasi luring yang andal untuk perangkat IoT dan sistem otomasi modern (Avasalcai, Zarrin, & Dustdar, 2022). Memindahkan beban komputasi ASR yang intensif ke perangkat edge menghadirkan tantangan teknis karena spektrumnya meliputi perangkat sederhana seperti mikrokontroler hingga Single-Board Computer yang lebih kuat; bahkan generasi perangkat terbaru dengan daya komputasi lebih besar tetap dibatasi oleh kendala termal, efisiensi energi, dan jejak memori yang ketat (Lyu, Yuan, Lu, & Zhang, 2022). Akibatnya, adanya

kesenjangan antara kompleksitas model ASR modern dan kebutuhan efisiensi operasional pada perangkat edge menjadi hambatan utama yang harus diatasi.

Secara historis, solusi dominan dalam domain pengenalan ucapan otomatis adalah arsitektur hibrida yang mengintegrasikan komponen jaringan saraf dalam dari deep learning dengan model statistik klasik (Mao, Tao, Zhang, Ching, & Lee, 2019). Salah satu implementasi umum adalah gabungan Time-Delay Neural Network dengan Hidden Markov Model (TDNN-HMM), di mana TDNN berperan sebagai model akustik yang mengekstraksi representasi fitur hierarkis dari sinyal audio, sedangkan HMM memodelkan sekuens temporal unit fonetik dan, dengan bantuan leksikon serta model bahasa, melakukan dekoding untuk menghasilkan transkripsi akhir (Ing, Pascual, & Dimzon, 2022). Meskipun arsitektur ini menunjukkan kinerja baik dalam berbagai kondisi, ia memiliki keterbatasan mendasar: proses dekoding yang sering menggunakan pencarian Viterbi bersifat komputasi-intensif dan sekuensial (Sun et al., 2017), serta kesulitan dalam menangani dependensi kontekstual jangka panjang yang penting untuk mengatasi ambiguitas dalam ucapan (Li et al., 2024).

Sebagai respon terhadap keterbatasan arsitektur hibrida, paradigma end-to-end muncul sebagai alternatif yang efektif, dengan arsitektur Transformer sebagai salah satu perwakilan utama; Transformer, awalnya dikembangkan untuk pengolahan bahasa alami, menunjukkan kinerja tinggi ketika diadaptasi untuk ASR karena mekanisme self-attention yang memungkinkan model menilai relevansi setiap segmen input terhadap segmen lain dalam satu sekuens sehingga menangkap dependensi kontekstual lokal dan global secara lebih efektif (Rahali & Akhloufi, 2023; Gong, Lai, Chung, & Glass, 2022). Berbeda dengan model berbasis recurrent neural network yang memproses data secara sekuensial, Transformer memproses seluruh input secara paralel, menawarkan keuntungan komputasi yang dapat dimanfaatkan oleh akselerator perangkat keras modern, dan kapasitas untuk menyederhanakan pipeline ASR menjadi satu jaringan tunggal menjadikannya pendekatan yang potensial untuk implementasi yang lebih efisien dan akurat dalam tugas pengenalan ucapan (Loubser, De Villiers, & De Freitas, 2024).

Meskipun Transformer menawarkan keunggulan dalam memodelkan dependensi kontekstual jangka panjang melalui mekanisme self-attention serta munculnya varian model yang lebih ringan, adopsinya pada perangkat edge sering terhambat oleh persepsi bahwa arsitektur ini memerlukan sumber daya komputasi besar (Lee et al., 2024). Sebaliknya, arsitektur hibrida telah mengalami optimisasi ekstensif untuk efisiensi, termasuk penggunaan faktor dekomposisi pada TDNN-F untuk mengurangi beban komputasi (Povey et al., 2018). Kondisi ini menimbulkan dilema praktis mengenai apakah varian ringan dari arsitektur modern dapat melampaui arsitektur tradisional yang telah teruji dan dioptimalkan untuk lingkungan terbatas. Karena bukti empiris yang secara langsung membandingkan trade-off akurasi dan latensi kedua pendekatan pada perangkat edge masih terbatas, evaluasi eksperimental yang terkontrol diperlukan untuk memberikan panduan berbasis bukti bagi pengambilan keputusan arsitektural.

Oleh karena itu, penelitian dilakukan untuk mengisi kesenjangan tersebut dengan melakukan perbandingan kinerja secara empiris antara arsitektur TDNN-F+HMM dengan arsitektur Transformer dalam konteks speech recognition Bahasa Inggris pada perangkat edge. Tujuan utamanya adalah untuk menentukan superioritas arsitektural berdasarkan dua metrik penting dalam perangkat edge, yaitu akurasi transkripsi, yang diukur secara objektif melalui Word Error Rate (WER), dan efisiensi pemrosesan, yang diukur melalui waktu eksekusi. Dengan menyediakan data perbandingan yang empiris dan terkontrol, studi ini berkontribusi pada pemahaman yang lebih dalam mengenai kelayakan dan performa masing-masing arsitektur. Hasil dari penelitian ini diharapkan dapat memberikan panduan praktis dan rekomendasi berbasis data bagi para praktisi dan peneliti dalam memilih arsitektur ASR yang optimal, guna mengakselerasi adopsi teknologi interaksi suara yang andal dan responsif pada perangkat edge generasi berikutnya.

2. Metode Penelitian

2.1. Automatic Speech Recognition

Automatic Speech Recognition (ASR) adalah cabang komputasi yang mengembangkan teknik untuk mentranskripsi ucapan manusia menjadi teks dengan tujuan memperlancar interaksi manusia–mesin; prosesnya umum dimulai dengan ekstraksi fitur dari sinyal audio digital, seperti

Mel-Frequency Cepstral Coefficients (MFCCs), yang merepresentasikan karakteristik spektral relevan bagi pendengaran manusia. Setelah fitur diekstraksi, sistem melakukan pemodelan akustik untuk memetakan sekuens fitur ke unit linguistik (misalnya fonem) dan pemodelan bahasa untuk menilai probabilitas rangkaian kata agar terbentuk kalimat yang koheren secara tata bahasa. Secara arsitektural, sistem ASR modern terbagi menjadi dua paradigma utama: sistem hibrida yang menggabungkan komponen pemodelan terpisah dan sistem end-to-end yang memetakan input audio langsung ke keluaran teks dalam satu jaringan saraf tunggal (Malik, Malik, Mehmood, & Makhdoom, 2021).

2.2. Time-Delay Neural Network - Factorized + Hidden Markov Model (TDNN-F + HMM)

Arsitektur TDNN-F + HMM merepresentasikan pendekatan hibrida dalam ASR yang menggabungkan dua komponen fungsional yang berbeda. Time-Delay Neural Network (TDNN) berperan sebagai model akustik yang, sebagai varian jaringan saraf konvolusional, dirancang untuk menangkap konteks temporal dengan memproses beberapa frame audio secara bersamaan sehingga mampu menghasilkan prediksi keadaan fonetik yang lebih akurat. Tambahan huruf F (Factorized) menunjukkan penggunaan dekomposisi matriks untuk mereduksi jumlah parameter dan beban komputasi. Hidden Markov Model (HMM) berfungsi sebagai model sekuensial yang menerima probabilitas keluaran dari TDNN dan memodelkan urutan keadaan fonetik menjadi urutan kata paling mungkin dengan bantuan kamus pelafalan dan model bahasa untuk menghasilkan transkripsi akhir (Kipyatkova, 2017).

Dalam penelitian ini, implementasi model ASR menggunakan Vosk, sebuah toolkit pengenalan suara offline yang menyediakan antarmuka pemrograman aplikasi (API) untuk mengakses model ASR pra-latih yang dibangun di atas framework Kaldi; Kaldi merupakan kerangka kerja yang mapan dalam riset ASR dan mendukung berbagai arsitektur jaringan saraf dalam termasuk varian hibrida seperti TDNN-F+HMM, sehingga pendekatan berbasis Kaldi melalui Vosk memungkinkan penerapan model kompleks secara efisien dan andal dalam lingkungan pengujian yang terkontrol (Xu, Hu, Liu, & Meng, 2022).

2.3. Transformer

Transformer adalah arsitektur jaringan saraf yang mengandalkan mekanisme self-attention untuk memproses data sekuensial (Chitty-Venkata, Emani, Vishwanath, & Somani, 2022). Keunggulan utamanya adalah kemampuan untuk mengolah seluruh data input secara paralel, memungkinkannya menangkap hubungan kontekstual jangka panjang secara efektif. Dalam konteks ASR, arsitektur Transformer diadopsi sebagai representasi dari paradigma end-to-end. Arsitektur ini memetakan sekuens fitur audio langsung ke sekuens teks melalui sebuah jaringan saraf tunggal yang terintegrasi. Struktur intinya terdiri dari dua komponen utama: encoder dan decoder. Encoder bertugas memproses seluruh input fitur audio dalam bentuk Mel-spectrogram dan menghasilkan serangkaian representasi yang kaya akan konteks. Kemampuan ini dicapai melalui mekanisme self-attention, yang memungkinkan setiap frame untuk menimbang relevansinya terhadap semua frame lain dalam satu sekuens. Selanjutnya, decoder secara autoregressive menghasilkan teks keluaran kata demi kata. Pada setiap langkah prediksi, decoder memperhatikan representasi dari encoder untuk menentukan kata berikutnya yang paling probabel.

Implementasi arsitektur Transformer dalam penelitian ini diwujudkan melalui Whisper, sebuah sistem ASR multiguna yang dikembangkan oleh OpenAI (Radford et al., 2022). Whisper merupakan model yang telah melalui proses pelatihan pada kumpulan data yang masif dan beragam, serta ditawarkan dalam serangkaian varian dengan kompleksitas yang berbeda. Rangkaian model ini mencakup varian large yang menargetkan akurasi maksimal hingga varian tiny yang secara spesifik dirancang untuk efisiensi tinggi pada perangkat terbatas. Sejalan dengan fokus penelitian, varian model tiny dipilih untuk mengevaluasi kinerja arsitektur Transformer dalam skenario yang menuntut efisiensi komputasi.

2.4. Edge Device

Edge device merujuk pada kelas perangkat keras yang melakukan pemrosesan data secara lokal, di "tepi" jaringan, sebagai alternatif dari pemrosesan terpusat di cloud computing (Palermo et al.,

2025). Paradigma edge computing ini dirancang untuk menjawab tantangan aplikasi modern seperti kebutuhan akan latensi rendah, jaminan privasi data, dan fungsionalitas luring. Secara umum, perangkat edge beroperasi dengan keterbatasan sumber daya, mencakup daya komputasi, kapasitas memori (RAM), dan konsumsi energi. Keterbatasan ini menjadi tantangan rekayasa yang signifikan, khususnya untuk menjalankan model kecerdasan buatan yang kompleks. Penelitian ini menggunakan Raspberry Pi 4 Model B sebagai platform pengujian yang representatif untuk perangkat edge. Pemilihan platform ini didasarkan pada posisinya sebagai perangkat yang sangat populer dan representatif untuk prototipe aplikasi AI di lingkungan edge, sehingga perbandingan kinerja menjadi krusial untuk memahami trade-off antara akurasi dan efisiensi pada perangkat dengan karakteristik tersebut.

2.5. Dataset

Dataset yang digunakan dalam penelitian ini adalah LibriSpeech ASR corpus, yang diperoleh dari platform repositori data Kaggle di bawah lisensi CC BY-SA 4.0. LibriSpeech merupakan korpus yang umum digunakan sebagai tolok ukur dalam komunitas riset ASR (. Dataset ini bersumber dari koleksi buku audio domain publik berbahasa Inggris dengan total 1000 jam pelatihan, dengan karakteristik utama berupa kualitas audio yang baik (Panayotov, Chen, Povey, & Khudanpur, 2015).

Untuk tujuan eksperimen ini, digunakan sebuah subset data yang terdiri dari 2000 sampel audio. Sampel-sampel ini dipilih dari partisi test-clean, sebuah bagian dari korpus LibriSpeech yang secara khusus dialokasikan untuk pengujian. Guna memastikan konsistensi dan validitas perbandingan, setiap file audio melalui tahap pra-pemrosesan standar. Tahap ini mencakup konversi sinyal audio menjadi format single-channel (mono) dan penyesuaian frekuensi sampling menjadi 16 kHz, sesuai dengan persyaratan input standar untuk model Vosk maupun Whisper. Contoh data dapat dilihat pada Tabel 1.

Tabel 1. Contoh Data LibriSpeech

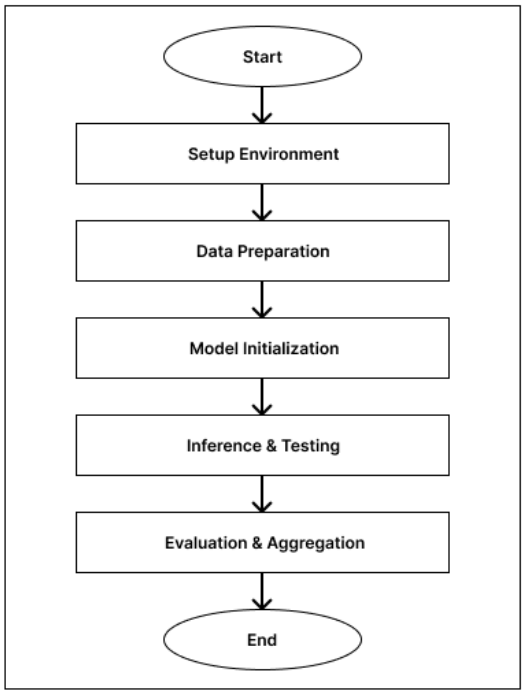
Audio	Transkrip
84-121123-0000.Flac	Go do you hear.
84-121123-0001.Flac	But in less than five minutes the staircase groaned beneath an extraordinary weight
174-50561-0016.Flac	Once more the singer plays and the ladies dance but one by one they fall asleep to the drowsy music and then the singer steps into the ring and unlocks the tower and kisses the emperor's daughter
2803-161169-0017.Flac	When your hands or lips are cracked and rough from the cold does your mother ever put on glycerin to heal them

2.6. Desain Ekperimen

Penelitian ini memanfaatkan serangkaian konfigurasi perangkat keras dan lunak untuk menjalankan keseluruhan proses eksperimen. Visual Studio Code dimanfaatkan sebagai Integrated Development Environment (IDE) utama untuk pengembangan kode dalam bahasa Python. Untuk kebutuhan prototipe dan analisis data interaktif, digunakan fungsionalitas Jupyter Notebook. Library inti yang menjadi subjek perbandingan meliputi Vosk dan Whisper (tiny) sebagai model ASR. Proses ini didukung oleh pydub untuk manipulasi audio, jiwer untuk kalkulasi metrik Word Error Rate (WER) secara akurat, serta pandas dan matplotlib untuk analisis dan visualisasi data.

Dari sisi perangkat keras, pengembangan kode dan analisis awal dilakukan pada sebuah workstation (Laptop Acer Swift 3 Infinity 4 dengan CPU Intel Core i7-1165G7 dan 16GB RAM). Namun, untuk memastikan relevansi hasil dengan skenario target, seluruh pengujian kinerja termasuk pengukuran akurasi dan latensi dijalankan secara eksklusif pada perangkat edge. Perangkat edge yang digunakan adalah Raspberry Pi 4 Model B dengan spesifikasi: Prosesor Broadcom BCM2711 (Quad-core Cortex-A72 ARM v8 64-bit @ 1.8GHz), memori 8GB LPDDR4-3200 SDRAM, dan penyimpanan berbasis kartu micro-SD 64 GB. Pemilihan platform ini didasarkan pada posisinya sebagai perangkat yang sangat populer dan representatif untuk prototipe aplikasi AI di lingkungan edge, sehingga perbandingan kinerja menjadi krusial untuk memahami trade-off antara akurasi dan efisiensi pada perangkat dengan karakteristik tersebut.

Alur kerja eksperimen dalam penelitian ini dirancang secara sistematis, mencakup beberapa tahapan utama mulai dari penyiapan lingkungan hingga evaluasi hasil akhir, yang diilustrasikan pada Gambar 1.



Gambar 1. Alur Kerja Eksperimen

Tahap awal penelitian ini berfokus pada pembentukan lingkungan komputasi yang terstandarisasi dan dapat direproduksi. Proses ini diawali dengan instalasi semua pustaka perangkat lunak yang diperlukan, termasuk library speech recognition seperti Vosk dan Whisper, utilitas pemrosesan audio seperti Librosa dan Pydub, serta pustaka analisis data dan evaluasi metrik seperti Pandas dan Jiwer. Secara simultan, bagian ini secara otomatis mengunduh dan melakukan dekompresi model-model pre-trained yang akan digunakan. Langkah awal ini memastikan bahwa semua dependensi terpenuhi dan lingkungan pengujian sepenuhnya konsisten sebelum eksekusi eksperimen inti.

Pada tahap data preparation, persiapan korpus data untuk pengujian dilakukan dengan memanfaatkan sub-set test-clean dari dataset LibriSpeech. Bagian ini dirancang untuk memindai struktur direktori dataset secara rekursif, mengidentifikasi file transkripsi referensi (.trans.txt), dan memetakan setiap transkripsi ke file audio .flac yang sesuai. Untuk memastikan reprodusibilitas dan efisiensi komputasi, digunakan metode pengambilan sampel (sampling) terprogram. Dengan menggunakan seed acak yang tetap, sebuah sub-set yang terdiri dari 500 sampel audio dipilih secara konsisten, sehingga menjamin bahwa setiap proses pengujian menggunakan data yang identik untuk perbandingan.

Tahap inisialisasi model melibatkan penyiapan model speech recognition dan metrik evaluasi. VoskRecognizer dan WhisperRecognizer, dibuat untuk mengenkapsulasi logika spesifik dari masing-masing model. Kelas VoskRecognizer mencakup fungsionalitas untuk mengonversi format audio dari FLAC ke WAV untuk model Vosk dengan mekanisme fallback. Sementara itu, WhisperRecognizer memuat model tiny langsung dari cache. Secara paralel, sistem evaluasi disiapkan dengan mendefinisikan fungsi normalisasi teks untuk standarisasi output termasuk mengubah ke huruf kecil dan menghapus tanda baca serta menginisialisasi fungsi untuk menghitung metrik Word Error Rate (WER).

Tahap Inferensi dan Pengujian merupakan inti dari eksekusi eksperimen, di mana proses transkripsi dilakukan secara iteratif. Sistem mengulang setiap file audio dalam sub-set data yang telah disiapkan sebelumnya. Untuk setiap file, proses inferensi dijalankan secara terpisah pada model Vosk dan Whisper. Selama setiap proses transkripsi, dua data krusial direkam: hasil

transkripsi teks yang dihasilkan oleh model dan waktu eksekusi yang diperlukan, yang diukur dengan presisi tinggi. Hasil transkripsi ini kemudian langsung dibandingkan dengan teks referensi untuk menghitung skor WER. Seluruh hasil, termasuk teks referensi, transkripsi, waktu, dan skor metrik, disimpan secara sistematis.

Tahap akhir adalah analisis dan sintesis hasil. Semua data mentah yang terkumpul dari tahap pengujian dikonsolidasikan ke dalam sebuah struktur data terorganisir menggunakan pustaka Pandas. Data ini kemudian diagregasi dengan mengelompokkannya berdasarkan model speech recognition. Melalui proses ini, metrik kinerja utama seperti rata-rata WER, dan waktu pemrosesan rata-rata per file dihitung untuk setiap model. Hasil agregat ini tidak hanya memberikan ringkasan statistik yang kuantitatif mengenai kinerja komparatif kedua model, tetapi juga menjadi dasar untuk pembuatan visualisasi data, seperti diagram batang, yang memfasilitasi interpretasi hasil secara intuitif dan komprehensif.

2.7. Metrik Evaluasi

Evaluasi kinerja kuantitatif dalam penelitian ini berfokus pada dua metrik fundamental: akurasi model, yang diukur dengan Word Error Rate (WER), dan efisiensi komputasi, yang diukur melalui Waktu Eksekusi. Word Error Rate (WER) digunakan sebagai metrik kuantitatif untuk mengevaluasi akurasi sistem. WER adalah metrik standar dalam penelitian ASR yang mengukur persentase kesalahan pada level kata antara teks hasil transkripsi dan teks referensi. Perhitungan metrik ini didasarkan pada formula WER=(S+D+I)/N, di mana S adalah jumlah substitusi, D adalah jumlah delesi, I adalah jumlah insersi, dan N adalah jumlah total kata pada teks referensi. Nilai WER yang lebih rendah merepresentasikan tingkat kesalahan yang lebih kecil dan, dengan demikian, akurasi yang lebih tinggi.

Formula:

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

Keterangan:

- S: Substitusi (Substitutions): Jumlah kata yang salah dikenali oleh model (misalnya, referensi "kucing" menjadi "kuncing").
- D: Delesi (Deletions): Jumlah kata pada teks referensi yang dihilangkan atau tidak terdeteksi oleh model.
- I: Insersi (Insertions): Jumlah kata yang ditambahkan oleh model yang sebenarnya tidak ada pada teks referensi.
- N: Jumlah Kata Referensi (Number of Words): Jumlah total kata dalam transkrip referensi yang asli.

Metrik kedua adalah Waktu Eksekusi, yang berfungsi sebagai proksi untuk efisiensi komputasi dan latensi sistem. Metrik ini didefinisikan sebagai total waktu riil (wall-clock time) dalam satuan detik yang diperlukan model untuk memproses satu sampel audio secara lengkap, dari saat data diumpankan hingga transkripsi teks dihasilkan. Pengukuran ini krusial untuk menilai kelayakan implementasi model pada edge device, di mana responsivitas yang cepat seringkali menjadi persyaratan utama. Nilai yang dilaporkan merupakan rata-rata aritmatika dari waktu eksekusi yang dicatat dari keseluruhan 2000 sampel audio.

Formula:

$$Waktu\ Eksekusi = Waktu\ Selesai - Waktu\ Mula\text{i}$$

Keterangan:

- Waktu Mulai (start_time): Titik waktu (timestamp) yang dicatat sesaat sebelum file audio diumpankan ke model untuk diproses.
- Waktu Selesai (end_time): Titik waktu (timestamp) yang dicatat segera setelah model selesai memproses audio dan menghasilkan output teks transkripsi.

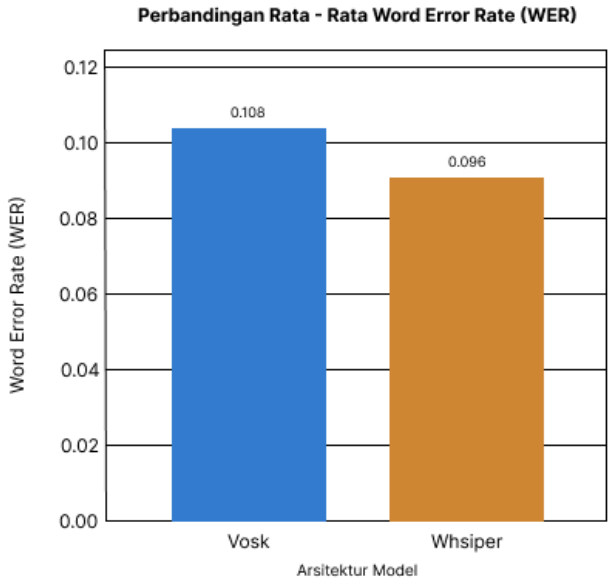
3. Hasil dan Pembahasan

Serangkaian eksperimen perbandingan telah dilakukan untuk mengevaluasi kinerja antara arsitektur TDNN-F+HMM yang diimplementasikan melalui Vosk, dengan arsitektur Transformer yang diimplementasikan melalui Whisper. Seluruh pengujian dijalankan pada perangkat edge Raspberry Pi 4 Model B untuk mensimulasikan skenario penggunaan di dunia nyata. Analisis berfokus pada dua metrik utama: akurasi transkripsi, yang diukur melalui Word Error Rate (WER), dan efisiensi komputasi, yang diukur melalui waktu eksekusi. Data kuantitatif yang disajikan menjadi dasar untuk pembahasan mendalam mengenai keunggulan, kelemahan, dan implikasi praktis dari masing-masing arsitektur dalam konteks komputasi edge.

Tabel 2. Rata-rata Perhitungan WER dan Waktu Eksekusi

Pengujian ke-	WER Vosk	WER Whisper	Waktu Eksekusi Vosk	Waktu Eksekusi Whisper
1	0.105	0.095	3.981	7.155
2	0.107	0.095	4.105	7.274
5	0.116	0.097	4.123	7.332
6	0.108	0.097	4.199	7.518
9	0.115	0.096	4.027	7.412
10	0.107	0.097	3.964	7.281

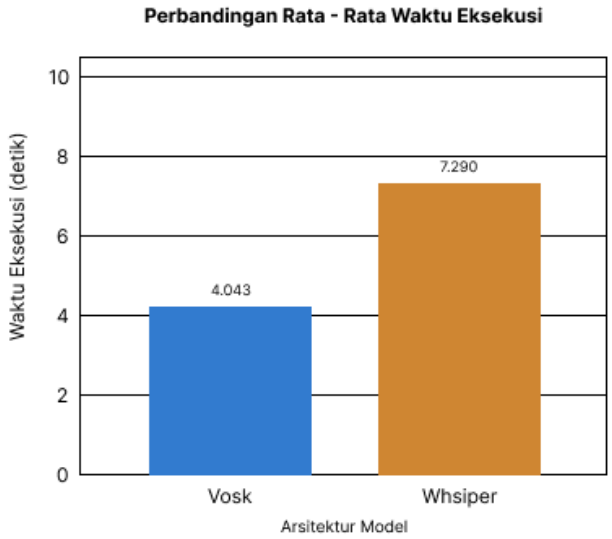
Data yang disajikan pada Tabel 2 menunjukkan dua tren kinerja yang konsisten dan berlawanan di antara kedua arsitektur. Dari segi akurasi, model Whisper secara konsisten mencapai skor Word Error Rate (WER) yang lebih rendah di setiap pengujian, dengan nilai berkisar antara 0.095 hingga 0.097. Hal ini mengindikasikan tingkat kesalahan transkripsi yang lebih kecil dan stabilitas kinerja yang tinggi. Sebaliknya, model Vosk menunjukkan WER yang sedikit lebih tinggi dan lebih bervariasi, berkisar antara 0.105 hingga 0.116. Namun, dalam hal efisiensi komputasi, Vosk menunjukkan keunggulan yang signifikan. Waktu eksekusinya secara konsisten berada di sekitar 4 detik, sementara Whisper membutuhkan waktu hampir dua kali lipat, rata-rata di atas 7 detik, untuk memproses sampel audio yang sama.



Gambar 2. Perbandingan WER

Gambar 2 memvisualisasikan perbandingan akurasi rata-rata antara kedua arsitektur setelah diuji pada keseluruhan sampel data. Hasil agregat menunjukkan keunggulan arsitektur Whisper dalam hal akurasi transkripsi. Model Whisper tiny berhasil mencapai skor WER rata-rata sebesar 0.096, sedangkan model Vosk mencatatkan skor WER 0.108. Meskipun perbedaan absolutnya terlihat kecil, yaitu 0.012, ini merepresentasikan penurunan tingkat kesalahan relatif sekitar 11.1% yang dicapai oleh Whisper. Keunggulan ini menunjukkan bahwa arsitektur Whisper, bahkan pada varian modelnya yang paling ringan, mampu menghasilkan transkripsi yang lebih presisi dibandingkan dengan pendekatan hibrida dalam kondisi pengujian ini. Temuan ini menegaskan

bahwa untuk tugas yang memprioritaskan keakuratan teks, model Whisper memberikan hasil yang lebih andal.



Gambar 3. Perbandingan Waktu Eksekusi

Berbanding terbalik dengan metrik akurasi, Gambar 3 dengan jelas menunjukkan superioritas arsitektur hibrida Vosk dalam hal efisiensi komputasi pada perangkat edge. Model Vosk mampu menyelesaikan tugas transkripsi dengan waktu eksekusi rata-rata hanya 4.043 detik per sampel audio. Sebaliknya, model Whisper tiny membutuhkan waktu rata-rata 7.290 detik, hampir 80% lebih lambat dibandingkan Vosk. Perbedaan kinerja yang signifikan ini menyoroti efektivitas pendekatan Vosk untuk inferensi berbasis CPU pada perangkat dengan sumber daya terbatas. Hasil ini mengindikasikan bahwa untuk aplikasi yang memerlukan pemrosesan cepat dan latensi rendah, arsitektur yang diimplementasikan oleh Vosk menawarkan solusi yang jauh lebih responsif dan efisien dibandingkan dengan arsitektur yang digunakan oleh Whisper tiny pada perangkat edge Raspberry Pi 4.

Analisis hasil eksperimen secara keseluruhan mengungkapkan adanya trade-off antara akurasi dan efisiensi komputasi pada kedua arsitektur saat diimplementasikan di perangkat edge. Kinerja Whisper menunjukkan bahwa pendekatan arsitektur modern mampu memberikan tingkat akurasi yang lebih tinggi. Skor WER yang lebih rendah secara konsisten di seluruh dataset pengujian mengindikasikan bahwa model ini lebih andal dalam menghasilkan transkripsi yang sesuai dengan referensi ground-truth. Keunggulan ini sangat berharga dalam skenario di mana presisi teks adalah prioritas utama. Namun, keunggulan akurasi ini dicapai dengan mengorbankan kecepatan pemrosesan. Waktu eksekusi yang lebih lama menunjukkan bahwa arsitektur ini menuntut beban komputasi yang lebih tinggi, yang menjadi tantangan signifikan pada perangkat keras dengan sumber daya terbatas seperti Raspberry Pi 4, di mana efisiensi seringkali menjadi faktor pembatas.

Di sisi lain, keunggulan kecepatan Vosk yang signifikan menunjukkan bahwa arsitektur hibrida tetap menjadi pilihan yang sangat kompetitif untuk aplikasi pada perangkat edge. Kemampuannya untuk memproses audio dengan latensi yang jauh lebih rendah menjadikannya sangat cocok untuk skenario penggunaan yang menuntut responsivitas tinggi. Karakteristik kinerja ini mengindikasikan bahwa arsitektur Vosk sangat dioptimalkan untuk eksekusi cepat pada CPU, yang merupakan komponen pemrosesan utama pada sebagian besar perangkat edge. Namun, efisiensi ini datang dengan sedikit penurunan akurasi jika dibandingkan dengan Whisper. Tingkat kesalahan kata yang sedikit lebih tinggi merupakan konsekuensi yang harus dipertimbangkan ketika memilih arsitektur ini, terutama jika aplikasi yang dikembangkan sangat sensitif terhadap kesalahan transkripsi sekecil apa pun, sehingga menyoroti dilema rekayasa yang fundamental.

Implikasi praktis dari temuan ini sangat signifikan bagi para praktisi dan pengembang yang menargetkan aplikasi ASR pada perangkat edge. Pilihan antara Vosk dan Whisper bukanlah tentang mana yang secara absolut lebih baik, melainkan tentang arsitektur mana yang paling sesuai dengan prioritas dan batasan kasus penggunaan spesifik. Untuk aplikasi yang memprioritaskan akurasi

transkripsi tertinggi dan dapat mentolerir latensi yang lebih tinggi seperti transkripsi offline untuk notulensi rapat atau analisis audio maka Whisper tiny merupakan pilihan yang lebih unggul. Sebaliknya, untuk aplikasi yang menuntut responsivitas waktu nyata (real-time) dan efisiensi daya seperti asisten suara interaktif, sistem perintah dan kontrol, atau aplikasi IoT kecepatan superior dan jejak komputasi yang lebih ringan dari Vosk menjadikannya pilihan yang jauh lebih praktis dan layak. Studi ini memberikan bukti kuantitatif yang jelas untuk memandu keputusan rekayasa dalam menavigasi trade-off fundamental ini.

4. kesimpulan

Penelitian ini secara sistematis mengevaluasi dan membandingkan kinerja dua paradigma arsitektur Automatic Speech Recognition (ASR) yaitu arsitektur hibrida TDNN-F+HMM yang diimplementasikan melalui Vosk dan arsitektur end-to-end Transformer yang diimplementasikan melalui Whisper tiny, dalam konteks komputasi pada perangkat edge dengan sumber daya terbatas, yaitu Raspberry Pi 4 Model B. Hasil penelitian secara definitif menunjukkan adanya trade-off fundamental antara akurasi transkripsi dan efisiensi komputasi yang melekat pada masing-masing pendekatan. Arsitektur Transformer, bahkan dalam varian modelnya yang paling ringan, berhasil mencapai superioritas dalam akurasi transkripsi, dengan skor Word Error Rate (WER) rata-rata 0.096. Hal ini mengkonfirmasi kemampuannya dalam menghasilkan transkripsi yang lebih presisi, menjadikannya pilihan yang lebih andal untuk aplikasi di mana keakuratan teks merupakan prioritas utama. Namun, keunggulan akurasi ini dicapai dengan biaya latensi yang lebih tinggi, dengan waktu eksekusi rata-rata 7.290 detik, yang dapat menjadi faktor pembatas dalam skenario penggunaan yang menuntut responsivitas cepat. Sebaliknya, arsitektur hibrida TDNN-F+HMM menunjukkan keunggulan yang signifikan dalam efisiensi komputasi, dengan waktu eksekusi rata-rata 4.043 detik, hampir 80% lebih cepat dibandingkan Whisper. Kinerja ini menegaskan kelayakannya untuk aplikasi yang sensitif terhadap latensi, seperti sistem perintah dan kontrol atau asisten suara interaktif. Implikasi praktis dari temuan ini adalah bahwa tidak ada satu arsitektur yang superior secara absolut untuk semua kasus penggunaan di perangkat edge. Pilihan arsitektur harus didasarkan pada analisis kebutuhan spesifik aplikasi: Whisper lebih cocok untuk tugas yang mengutamakan akurasi, sementara Vosk lebih unggul untuk tugas yang mengutamakan kecepatan dan responsivitas. Studi ini memberikan panduan berbasis bukti kuantitatif bagi para pengembang untuk membuat keputusan rekayasa yang tepat dalam menavigasi dilema antara akurasi dan efisiensi dalam pengembangan aplikasi ASR generasi berikutnya.

References

Alharbi, S., et al. (2021). Automatic speech recognition: Systematic literature review. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3112535

Avasalcai, C., Zarrin, B., & Dustdar, S. (2022). EdgeFlow—Developing and deploying latency-sensitive IoT edge applications. *IEEE Internet of Things Journal, 9*(5), 3877–3888. https://doi.org/10.1109/JIOT.2021.3101449

Bhandari, S. R., & Ghimire, S. (2025). Expanding horizon of English language as a lingua franca.

Chitty-Venkata, K. T., Emani, M., Vishwanath, V., & Somani, A. K. (2022). Neural architecture search for transformers: A survey. *IEEE Access, 10*, 108374–108412. https://doi.org/10.1109/ACCESS.2022.3212767

Gong, Y., Lai, C.-I., Chung, Y.-A., & Glass, J. (2022). SSAST: Self-supervised audio spectrogram transformer. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*(10), 10699–10709. https://doi.org/10.1609/aaai.v36i10.21315

Ing, J. A. Y., Pascual, R. M., & Dimzon, F. D. (2022). A hybrid TDNN-HMM automatic speech recognizer for Filipino children's speech. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)* (pp. 1–6). https://doi.org/10.1109/IICAIET55139.2022.9936815

Kipyatkova, I. (2017). Experimenting with hybrid TDNN/HMM acoustic models for Russian speech recognition. In A. Karpov, R. Potapova, & I. Mporas (Eds.), *Speech and Computer* (pp. 362–369). Springer International Publishing.

Lee, J., Bahk, I., Kim, H., Jeong, S., Lee, S., & Min, D. (2024). An autonomous parallelization of transformer model inference on heterogeneous edge devices. In *Proceedings of the 38th ACM International Conference on Supercomputing (ICS '24)* (pp. 50–61). Association for Computing Machinery. https://doi.org/10.1145/3650200.3656628

Li, Y., Gan, J., Lin, X., Qiu, Y., Zhan, H., & Tian, H. (2024). DS-TDNN: Dual-stream time-delay neural network with global-aware filter for speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 32*, 2814–2827. https://doi.org/10.1109/TASLP.2024.3402072

Loubser, A., De Villiers, P., & De Freitas, A. (2024). 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

Lyu, B., Yuan, H., Lu, L., & Zhang, Y. (2022). Resource-constrained neural architecture search on edge devices. *IEEE Transactions on Network Science and Engineering, 9*(1), 134–142. https://doi.org/10.1109/TNSE.2021.3054583

Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). Automatic speech recognition: A survey. *Multimedia Tools and Applications, 80*(6), 9411–9457. https://doi.org/10.1007/s11042-020-10073-7

Mao, S., Tao, D., Zhang, G., Ching, P. C., & Lee, T. (2019). Revisiting hidden Markov models for speech emotion recognition. In *ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6715–6719). https://doi.org/10.1109/ICASSP.2019.8683172

Palermo, F., et al. (2025). Advancements in context recognition for edge devices and smart eyewear: Sensors and applications. *IEEE Access, 13*, 57062–57100. https://doi.org/10.1109/ACCESS.2025.3555426

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5206–5210). https://doi.org/10.1109/ICASSP.2015.7178964

Povey, D., et al. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *INTERSPEECH 2018* (pp. 3743–3747). https://doi.org/10.21437/Interspeech.2018-1417

Rahali, A., & Akhloufi, M. A. (2023). End-to-end transformer-based models in textual-based NLP. *AI, 4*(1). https://doi.org/10.3390/ai4010004

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

Sun, M., et al. (2017). Compressed time delay neural network for small-footprint keyword spotting. In *INTERSPEECH 2017* (pp. 3607–3611). https://doi.org/10.21437/Interspeech.2017-480

Xu, J., Hu, S., Liu, X., & Meng, H. (2022). Towards green ASR: Lossless 4-bit quantization of a hybrid TDNN system on the 300-hr Switchboard corpus. *arXiv*. https://doi.org/10.48550/arXiv.2206.11643.