Terbit *online* pada laman : http://teknosi.fti.unand.ac.id/

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |

Artikel Penelitian

Automatic Speech Recognition for Javanese Language using Wav2Vec 2.0 with Finetuning

Johanes Setiawan [a], Ardytha Luthfiarta [b,], Adhitya Nugraha [c], Rismiyati [d], Bastiaans, Jessica Carmelita [e], Yohanes Deny Novandian [f]*

[a,b,c,e,f] *Universitas Dian Nuswantoro, Jl. Imam Bonjol No.207, Pendrikan Kidul, Kec. Semarang Tengah, Kota Semarang, Jawa Tengah 50131, Indonesia*
[d] *Universitas Diponegoro, Jl. Prof. Soedarto No.13, Tembalang, Kec. Tembalang, Kota Semarang, Jawa Tengah 50275, Indonesia*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 27 Oktober 2024
Revisi Akhir: 28 Agustus 2025
Diterbitkan *Online*: 30 April 2026

KATA KUNCI

Javanese Language,
Wav2Vec 2.0,
Speech to Text,
Deep Learning,
Finetuning

KORESPONDENSI

E-mail: ardytha.luthfiarta@dsn.dinus.ac.id *

A B S T R A C T

Penelitian ini bertujuan untuk mengembangkan sistem pengenalan suara untuk bahasa Jawa dengan memanfaatkan model Wav2Vec 2.0 melalui proses *finetuning*. Bahasa Jawa, sebagai salah satu bahasa daerah dengan lebih dari 80 juta penutur, memiliki tantangan tersendiri dalam pengenalan suara akibat keterbatasan data dan kompleksitas linguistiknya. Penelitian ini menggunakan dataset audio yang diambil dari OpenSLR dan diterapkan pada dua varian model, yaitu wav2vec2-base dan wav2vec2-large, yang masing-masing memiliki jumlah parameter 94,4 juta dan 315 juta. Proses *finetuning* dilakukan untuk meningkatkan akurasi sistem dalam mengenali variasi suara bahasa Jawa. Evaluasi dilakukan menggunakan metrik Word Error Rate (WER) dan evaluation loss, dengan hasil akhir menunjukkan bahwa model wav2vec2-base memiliki WER sebesar 15,02% dan model wav2vec2-large sebesar 15,57%. Hasil ini menunjukkan efektivitas pendekatan *finetuning* dalam meningkatkan performa pengenalan suara bahasa Jawa.

1. PENDAHULUAN

Pengenalan suara telah berkembang pesat terutama dalam ranah *speech to text* (STT), yang bertujuan untuk mengubah sinyal suara menjadi teks. Meskipun teknologi ini telah menunjukkan hasil yang luar biasa dalam bahasa global seperti Bahasa Inggris, namun masih diperlukan riset-riset lanjutan dalam domain bahasa lokal terutama bahasa daerah, seperti Bahasa Jawa, Sunda, Melayu, Batak dan bahasa lain sejenisnya [1]. Berdasarkan data Kementerian Pendidikan dan Kebudayaan survei terbaru 18 Maret 2023 dari Badan Pusat Statistik (BPS) menyebutkan bahwa Bahasa Jawa sebagai salah satu bahasa daerah dengan lebih dari 80 juta penutur di Indonesia. Bahasa Jawa merupakan salah satu penulisan Melayu kuno yang pertama kali diperkenalkan pada era kolonial Inggris [2], khususnya untuk tradisi agama dan budaya di wilayah Asia Tenggara [2], [3]. Namun, literatur menunjukkan bahwa sejak diperkenalkannya sistem penulisan modern berbasis huruf Romawi, bahasa Jawi menjadi kurang populer [2], [4], [5], [6]. Pengolahan Speech to text dengan bahasa jawa belum mendapatkan perhatian yang cukup dalam pengembangan model STT berbasis kecerdasan buatan.

Bahasa Jawa memiliki tantangan tersendiri dalam konteks pengenalan suara. Kendala utama adalah keterbatasan data suara yang tersedia dalam bahasa tersebut, serta kompleksitas linguistik yang unik. Hal ini menambah tingkat kesulitan dalam pengembangan model STT yang akurat dan adaptif terhadap variasi tersebut. Selain itu, struktur fonetik dan morfologis Bahasa Jawa cukup kompleks, dengan adanya tingkat tutur atau ngoko dan krama, yang semakin memperluas spektrum variasi

https://doi.org/10.25077/TEKNOSI.v12i1.2026.001-009

kata yang harus dipahami oleh model STT. Tantangan lainnya meliputi penyesuaian terhadap konteks dan gaya bicara informal, yang sering kali berbeda dari gaya formal yang mungkin lebih mudah dikenali oleh teknologi STT yang telah berkembang untuk bahasa yang lebih umum. Oleh karena itu, pengembangan model STT untuk Bahasa Jawa tidak hanya memerlukan data yang lebih representatif, tetapi juga pendekatan teknologi yang lebih adaptif dan inovatif.

Ekstraksi fitur menjadi salah satu tantangan penting dalam pengembangan sistem STT, terutama karena proses ini memerlukan pemahaman mendalam tentang struktur fonetik, morfologis, dan linguistik dari bahasa jawa. Namun, dalam kasus Bahasa Jawa yang memiliki keragaman dialek, tingkat tutur, serta kompleksitas morfologi, pendekatan berbasis fitur tradisional seperti fonetik dan morfologi tidak cukup representatif untuk menangkap variasi tersebut secara menyeluruh. Oleh karena itu, penulis memilih untuk menggunakan model yang langsung belajar dari sinyal suara mentah, seperti Wav2Vec 2.0, yang mampu secara langsung mengekstrak pola dari sinyal suara tanpa memerlukan tahap ekstraksi fitur manual. Pendekatan ini memungkinkan model untuk menangkap representasi yang lebih menyeluruh dari data suara, termasuk variasi linguistik yang sulit diakomodasi oleh pendekatan berbasis fitur. Dengan memanfaatkan pembelajaran langsung dari sinyal suara, model dapat lebih fleksibel dalam mengenali variasi gaya bicara, intonasi, dan konteks, yang sangat penting untuk Bahasa Jawa dengan berbagai macam keragamannya.

Dalam upaya mengatasi tantangan ini, diperlukan penerapan model kecerdasan buatan yang mampu belajar secara efisien dari data yang tersedia, meskipun dalam jumlah terbatas. Pendekatan seperti Wav2Vec 2.0 berpotensi menjadi solusi efektif, karena model ini dapat memanfaatkan data suara beserta transkripsi teks untuk memahami pola akustik dan linguistik secara efisien. Dengan menggunakan dataset yang telah disediakan secara terstruktur, model ini mengurangi ketergantungan pada proses pelabelan manual, sekaligus memaksimalkan pemanfaatan pasangan data audio dan teks.

Model Wav2Vec 2.0 dipakai di beberapa penelitian sebelumnya sebagai model yang digunakan untuk speech recognition yang dikonversi ke text atau dapat disebut Speech to Text (STT). Pada tahun 2023, diadakan eksperimen identifikasi audio menggunakan model Wav2Vec 2.0 terhadap 3 dataset yang merupakan kombinasi suara orang dewasa dan anak-anak dan didapatkan WER terendah di 2.91% [7]. Lalu pada tahun 2020 dilakukan eksperimen menggunakan model Wav2Vec 2.0 yang di *finetuning* dan digunakan pada dataset Librispeech berlabel tinggi yang berisi 960 jam data audio dengan WER di 1.8% [8]. Model Wav2Vec 2.0 juga dipakai di beberapa penelitian tentang proses STT melalui transkripsi atau pengenalan berbagai bahasa. Penelitian pengenalan bahasa Bengali pada 2022 yang merupakan bahasa sumber rendah menggunakan model Wav2Vec 2.0 yang telah di *finetuning* dan digunakan pada dataset Bengali Common Voice mencapai WER di 0.25% [9]. Kemudian Zhanibek et al. melakukan penelitian serupa pada 2023 menggunakan Wav2Vec 2.0 dengan dataset bahasa Kazakh yang merupakan bahasa sumber rendah juga dan didapatkan WER 8.7% [10]. Berbagai penelitian dan eksperimen yang telah dilakukan sebelumnya menunjukkan model Wav2Vec 2.0 memiliki performa yang baik terutama terhadap bahasa dengan dataset sumber rendah.

Dalam konteks pengenalan suara, penulis menggunakan model deep learning Wav2Vec 2.0 sebagai sarana pengolahan audio menjadi teks. Hal tersebut didukung dengan data audio bahasa jawa yang termasuk dalam kategori Low Resources Language [9]. Terdapat penelitian serupa menggunakan dataset Bahasa Jawa yang sama yang di mana peneliti tersebut mengggunakan model Wav2Vec 2.0-xls-r-300M dan Whisper [11][11]. Model wav2vec2-xls-r-300M menghasilkan Word Error Rate (WER) sebesar 21.99 tanpa menggunakan bantuan model bahasa tambahan seperti KenLM. Sementara itu, model Whisper diuji pada beberapa varian, yaitu Whisper base, small, medium, dan large-v2 dengan hasil evaluasi menunjukkan 28.57%, 18.84%, 15.97%, dan 13.77%. Pada sisi lain, peneliti Panji Arisaputra et al. melakukan *finetuning* pada model Automatic Speech Recognition (ASR) untuk meningkatkan performa model secara signifikan dengan menurunkan WER hingga 85%. Kemudian digunakan juga metode *Parameter Efficient Fine-Tuning* (PEFT) dan *Low-Rank Adaptation (LoRA)* untuk mengurangi kebutuhan komputasi dengan mengurangi jumlah parameter hingga <1% yang dilatih tanpa mengorbankan akurasi. *Finetuning* mempermudah model dalam mengenali pola suara bahasa bersumber daya rendah seperti bahasa Jawa, sehingga meningkatkan akurasi pengenalan suara secara efisien dan efektif pada data yang terbatas. Hasil akhir menunjukkan WER sebesar 13,77% pada model Whisper large-v2 setelah *finetuning*, jauh lebih baik dibandingkan WER 89,40% sebelum *finetuning*, membuktikan efektivitas metode yang diterapkan [11].

Berdasarkan hasil yang diperoleh dari penelitian sebelumnya, penulis melakukan eksperimen lebih lanjut dengan menggunakan model *wav2vec2*-base dan *wav2vec2-large* untuk meningkatkan performa dengan menurunkan nilai WER pada proyek *speech to text* bahasa Jawa. Penulis menggunakan finetuning untuk meningkatkan performa model karena kemampuannya untuk meningkatkan kinerja model dengan menggunakan jumlah data yang lebih kecil yang telah dilabeli secara spesifik [12], [13], [14]. Dengan menggunakan data suara yang lebih spesifik dan tertranskripsi dari bahasa Jawa, finetuning dapat secara signifikan menurunkan Word Error Rate (WER) dan meningkatkan akurasi dalam pengenalan suara bahasa tersebut [14]. Selain itu, finetuning membantu model beradaptasi dengan pola suara unik dan struktur linguistik bahasa Jawa, sehingga memberikan performa yang lebih optimal dalam mengenali dan memahami ucapan dalam bahasa ini, bahkan dengan jumlah data yang relatif terbatas [15].

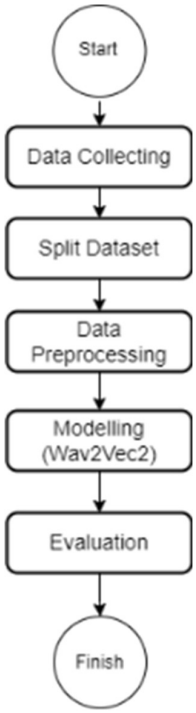
Penulis memilih model *wav2vec2-base* dan *wav2vec2-large* karena kedua model ini telah dilatih dengan data suara yang sangat luas, sehingga memiliki kemampuan dasar yang kuat dalam pengenalan suara. Model *wav2vec2-base* menawarkan keseimbangan antara ukuran dan kecepatan, sehingga cocok untuk pengujian dengan sumber daya komputasi yang lebih terbatas. Sementara itu, *wav2vec2-large*, dengan jumlah parameter yang lebih banyak, memiliki kapasitas yang lebih besar untuk memahami pola suara yang lebih kompleks, sehingga mampu memberikan hasil yang lebih akurat dalam pengenalan ucapan.

 https://doi.org/10.25077/TEKNOSI.v12i1.2026.001-009

Kedua model ini juga didukung dengan teknik finetuning menggunakan CTC Loss (Connectionist Temporal Classification), yang memungkinkan model untuk beradaptasi secara efektif terhadap dataset bahasa tertentu. Keunggulan ini memudahkan penyesuaian model untuk meningkatkan performa pengenalan suara dalam berbagai bahasa, termasuk bahasa yang mungkin kurang terwakili dalam pelatihan awal. Dengan finetuning, model dapat lebih cepat dan akurat dalam mengenali ucapan, bahkan ketika dihadapkan dengan variasi aksen atau intonasi. Hal ini membuat kedua model tersebut sangat cocok untuk diterapkan pada berbagai skenario aplikasi, termasuk dalam kondisi real-time dan lingkungan dengan kualitas audio yang bervariasi.

Penelitian ini melibatkan lima tahapan utama: (1) Pengumpulan Data, yaitu mengumpulkan dataset audio berbahasa Jawa; (2) Preprocessing Audio, yakni melakukan transformasi data mentah seperti normalisasi dan resampling data menjadi 16kHZ (3) Penggunaan Tokenizer dengan metode Connectionist Temporal Classification (CTC) untuk pemetaan suara ke teks; (4) Penerapan model Wav2Vec 2.0 untuk menghasilkan representasi suara; dan (5) Post-processing, yaitu konversi hasil model menjadi teks yang akurat.

2. METODE



Gambar 1. Metode Penelitian

Pada penelitian ini, penulis menggunakan metode pengembangan model ucapan-ke-teks berbasis Wav2Vec 2.0 untuk bahasa Jawa. Data dikumpulkan dari OpenSLR berupa rekaman audio pria dan wanita dalam format .wav, kemudian dilakukan pembagian dataset dengan rasio 90% data pelatihan dan 10% data pengujian. Setelah itu, dilakukan pengolahan data dengan proses preprocessing seperti resampling menjadi 16kHz dan normalisasi menggunakan Wav2Vec2Processor. Proses selanjutnya meliputi penghapusan karakter tidak relevan dari transkripsi teks dan ekstraksi fitur audio. Model Wav2Vec2ForCTC digunakan dengan teknik Connectionist Temporal Classification (CTC) untuk menghubungkan sinyal audio dengan transkripsi teks, di mana model dilatih menggunakan optimizer AdamW dan learning rate yang disesuaikan. Evaluasi model dilakukan menggunakan metrik Word Error Rate (WER) untuk mengukur akurasi prediksi teks dari audio, serta *evaluation loss*. Hasil pelatihan diharapkan mampu meningkatkan performa transkripsi otomatis bahasa Jawa.

2.1. Data Collecting

Penulis melakukan pengumpulan dan pemrosesan data sebagai langkah awal ketika mengembangkan model ucapan-ke-teks berbasis Wav2Vec 2.0 untuk bahasa Jawa. Pada langkah ini, data yang kuat dari berbagai sumber dikumpulkan dan dijadikan dasar pelatihan model. Pada penelitian ini, kami memperoleh data dari OpenSLR, sebuah repositori sumber terbuka yang menyediakan data linguistik beranotasi untuk berbagai bahasa (https://www.openslr.org/41). Data yang digunakan adalah rekaman audio bahasa Jawa yang tersedia dalam dua arsip terpisah yaitu jv_id_female.zip dan jv_id_male.zip. Arsip ini berisi file audio dalam format .wav dari suara wanita dan pria. Arsip jv_id_female.zip berisi total 2865 file dan arsip jv_id_male.zip berisi total 2959 file, termasuk rekaman audio dan file indeks yang berfungsi sebagai metadata. File indeks bertindak sebagai kunci utama yang menghubungkan setiap rekaman audio ke transkripsi teksnya. File ini disusun dalam format TSV dan berisi informasi penting seperti nama file, panjang, dan transkripsi kalimat yang diucapkan dalam rekaman. File audio

dalam arsip ini memiliki ukuran dan panjang yang bervariasi, sehingga menyediakan beragam data pelatihan untuk membantu melatih model dalam memahami berbagai macam intonasi dan kecepatan bicara.

Proses pengumpulan data dimulai dengan mengunduh dan mengekstrak data dari OpenSLR ke direktori lokal sehingga jalur pemrosesan dapat mengakses semua data secara efisien. Selanjutnya, file indeks line_index.tsv untuk setiap arsip dimuat ke dalam dataframe menggunakan pustaka Pandas. Langkah ini dilakukan agar dapat mengaitkan setiap rekaman audio dengan transkrip yang sesuai. Data rekaman suara perempuan dan laki-laki kemudian digabungkan menjadi kumpulan data besar dan dipecah menjadi data pelatihan serta pengujian untuk melatih model Wav2Vec 2.0. Dalam pendekatan ini, menggunakan data yang dikumpulkan dan diolah untuk melatih model Wav2Vec 2.0 yang diharapkan dapat menghasilkan model ucapan-ke-teks yang efektif untuk bahasa Jawa.

2.2. Split Dataset

Setelah data berhasil dikumpulkan, langkah berikutnya adalah membagi dataset untuk melatih model Wav2Vec 2.0 secara optimal. Pada tahap ini, dataset yang terdiri dari rekaman suara pria dan wanita dibagi menjadi dua bagian dengan rasio 90% data pelatihan dan 10% data pengujian, yang dilakukan secara acak untuk menghindari bias dan memastikan representasi variasi yang ada dalam dataset. Pembagian data dilakukan dengan memastikan bahwa data pengujian tidak pernah dilihat oleh model selama pelatihan, evaluasi kinerja model dapat dilakukan secara objektif pada data baru. Setelah proses pembagian selesai, dataset yang telah terpisah ini siap untuk melalui tahap selanjutnya, yaitu preprocessing, guna mempersiapkannya dalam format yang sesuai untuk melatih model Wav2Vec 2.0.

2.3. Data Preprocessing

Pada tahap *preprocessing* audio dalam penelitian *speech to text*, memiliki beberapa langkah penting yang dilakukan untuk mengubah data audio mentah menjadi format yang siap digunakan oleh model Wav2Vec 2.0. Proses ini mencakup pemrosesan audio mentah serta persiapan transkripsi teks. Setiap dataset file audio yang telah dikumpulkan dan diolah memanfaatkan library torchaudio untuk mengubah sinyal audio yang disimpan dalam format file (.wav) menjadi array numerik yang merepresentasikan bentuk gelombang audio (waveform). Array ini merupakan representasi langsung dari amplitudo sinyal audio dalam domain waktu, dimana setiap elemen mewakili amplitudo sinyal audio pada suatu titik waktu. Representasi ini adalah dasar bagi model Wav2Vec 2.0 dalam memahami pola-pola suara yang terkandung dalam sinyal audio.

Pada setiap file audio yang terdapat pada dataset memiliki sampling rate yang berbeda-beda, maka dari itu penulis melakukan proses resampling agar sampling rate tersebut menjadi seragam yaitu diset dengan value 16000Hz, sesuai kebutuhan standar dari model Wav2Vec 2.0. Resampling ini dilakukan dengan menggunakan library librosa. Secara matematis, proses resampling audio dapat dijelaskan melalui interpolasi dan decimation. Jika *x(t)* merupakan sinyal audio asli, maka sinyal resampled *y(t)* pada frekuensi baru dihitung dengan,

$$y(t) = x\left(\frac{t \cdot f_{new}}{f_{old}}\right)$$

di mana f_{new} adalah laju sampel baru (16kHz), dan f_{old} adalah laju sampel asli. Proses ini melibatkan interpolasi nilai-nilai sinyal di antara titik-titik asli untuk mendapatkan sinyal yang telah disesuaikan. Sampling rate yang sudah disamakan tersebut sangat penting, karena model Wav2Vec 2.0 dilatih untuk bekerja secara optimal pada audio dengan sampling rate 16000Hz, sehingga setiap sinyal audio harus disesuaikan dengan frekuensi tersebut untuk menjaga konsistensi data input. Jika tidak dilakukannya resampling, laju sampel yang berbeda dapat mempengaruhi kinerja model, karena representasi frekuensi yang diharapkan tidak akan cocok dengan data yang diberikan. Kemudian pada tahap preprocessing dilakukan juga normalisasi audio untuk memastikan bahwa intensitas sinyal audio tidak terlalu bervariasi antara satu file dengan file lainnya. Proses normalisasi tersebut dilakukan oleh Wav2Vec2Processor, yang merupakan bagian dari library transformers.

$$x_{norm}(t) = \frac{x(t)}{max(|x(t)|)}$$

Jika *x(t)* adalah sinyal audio asli, maka normalisasi dilakukan dengan membagi setiap nilai dalam sinyal dengan nilai maksimum absolutnya. Normalisasi berguna untuk menjaga agar nilai-nilai amplitudo berada dalam rentang yang dapat diterima oleh model serta meminimalisir perbedaan yang signifikan antara sinyal audio. Dengan demikian, data input menjadi lebih konsisten yang mempengaruhi stabilitas dalam pelatihan model.

Di sisi lain, berdampingan dengan setiap file audio dilakukan pembersihan transkrispi teks untuk menghapus karakter-karakter yang tidak relevan seperti tanda baca atau simbol khusus, yang tidak diperlukan dalam konteks pelatihan model *speech to text*. Fungsi remove_special_character tersebut didukung oleh *regular expressions* atau regex, yang berguna untuk menyaring karakter-karakter tersebut, sehingga hanya huruf abjad dan spasi yang disimpan dalam teks. Hal ini memastikan bahwa model hanya belajar dari informasi yang relevan dan tidak terganggu oleh karakter yang tidak penting. Kemudian untuk data audio yang sudah dinormalisasi, dilakukan ekstraksi fitur audio dengan bantuan Wav2Vec2Processor dari library transformers. Sinyal audio yang berbentuk gelombang akan diubah menjadi vektor fitur yang lebih terstruktur dan informatif.

$$y[i] = \sum_{j=0}^{k-1} x[i-j] \cdot w[j]$$

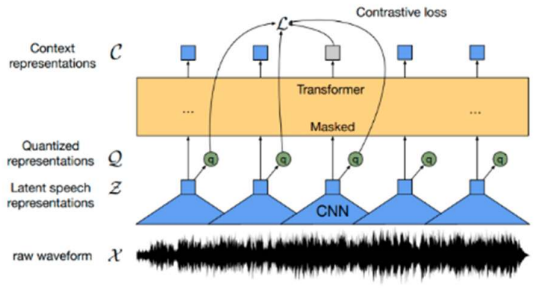
di mana *x[i]* adalah sinyal input, *w[j]* adalah kernel konvolusi, dan *k* adalah ukuran kernel.Proses ini mencakup penerapan lapisan-lapisan konvolusi yang bertujuan untuk mengekstrak fitur penting dari sinyal audio, seperti pola frekuensi yang khas untuk kata-kata atau fonem tertentu. Vektor fitur ini mewakili representasi numerik dari sinyal audio dalam format yang lebih mudah dipahami oleh model untuk tujuan pelatihan. Setelah itu, dilakukan transkripsi teks ditokenisasi menjadi urutan ID numerik yang diolah dengan menggunakan algoritma Connectionist Temporal Classification (CTC) pada modelling.

 https://doi.org/10.25077/TEKNOSI.v12i1.2026.001-009

2.4. *Modelling*

Wav2Vec 2.0 merupakan model yang dikhususkan untuk tugas pengenalan suara ke dalam text yang dikembangkan oleh Facebook AI. Penulis menggunakan model ini dikarenakan eksperimen yang dihasilkan pada dataset (*test*) dengan Librispeech mendapatkan metrik *Word Error Rate* (WER) sebesar 1.8. Keunggulan ini menjadikan Wav2Vec 2.0 sebagai fondasi yang kuat untuk membangun sistem *speech to text* yang akurat, *robust*, dan efisien.

Arsitektur pada model ini dirancang oleh Facebook AI dan memanfaatkan arsitektur yang cukup kompleks seperti Latent Feature Encoder, Quantization Module, dan Context Network. Berikut adalah visualisasi berupa gambar pada arsitektur Wav2Vec 2.0 yang bisa dilihat pada Gambar 2.



Gambar 2. Arsitektur Wav2Vec 2.0 [8]

Feature Encoder bertugas untuk mengekstraksi fitur dari sebuah audio melalui proses konvolusi dan menghasilkan *feature vectors*, kemudian *Quantization Module* bertugas untuk mengubah fitur menjadi representasi distrik, dan *Context Network* bertugas untuk menangkap konteks dan fitur dari representasi sebelum diteruskan ke modul Transformers yang bertugas untuk mempelajari pola dan informasi dari data audio dan teks secara mendalam. Juga adanya modul *Connectionist Temporal Classification* (CTC) yang membantu Wav2Vec 2.0 untuk membantu memahami pada suara – teks secara mendalam.

Connectionist Temporal Classification (CTC) merupakan sebuah *neural network* yang berfokus untuk menangani masalah pada pengenalan suara orang yang berbicara dengan intonasi dan pelafalan yang berbeda – beda dan memudahkan model untuk menyesuaikan diri dengan urutan *input audio* dan urutan teks pada dataset. CTC mengacu pada hasil dan penilaian, dan tidak tergantung pada struktur jaringan saraf yang mendasari [10][Zhanibek.K]. Proses bekerjanya yakni dengan mengiris audio input menjadi urutan X = [x1, x2, ..., xT], dan teks transkripsi output menjadi urutan Y = [y1, y2, ..., yT]. Rasio dan panjang X dan Y dapat bervariasi, tetapi tujuan algoritma CTC adalah untuk menyimpulkan pemetaan yang mungkin antara X dan Y [16][Arash.D].

$$p(Y|X) = \sum_{A\epsilon\ X,Y} \prod_{t=1}^{T} p_t(a_t|X)$$

Rumus diatas yang dibahas menghitung probabilitas urutan target Y (teks transkripsi) diberikan urutan input X (fitur suara), dengan mempertimbangkan semua alignment (penjajaran) antara X dan Y. Alignment ini memberikan fleksibilitas pemetaan antara input dan target, termasuk kemungkinan token kosong atau repetisi. Proses ini dihitung melalui produk probabilitas di setiap langkah waktu, dan probabilitas total dihitung dengan menjumlahkan semua alignment yang mungkin. CTC loss dihitung sebagai negatif log-likelihood dari probabilitas urutan target, dan digunakan untuk mengukur perbedaan antara prediksi model dan target sebenarnya seperti yang digambarkan pada rumus <dibawah>.

$$-log(p(Y|X))$$

Selanjutnya, model Wav2Vec 2.0 digunakan untuk pelatihan dan finetuning dalam proyek *speech to text* bahasa Jawa. Model ini adalah versi pretrained Wav2Vec 2.0 yang telah dilatih pada dataset audio umum. Melalui proses *finetuning*, model diadaptasi untuk mengenali pola suara, intonasi, dan karakteristik unik bahasa Jawa. Input audio diproses melalui Wav2Vec2Processor menjadi vektor fitur yang merepresentasikan sinyal audio. Transkripsi teks yang menyertai audio juga diproses menjadi urutan ID numerik untuk mempermudah model dalam mengenali teks.

Pada tahap pelatihan, CTC loss berfungsi sebagai algoritma utama untuk mencocokkan sinyal audio dengan transkripsi teks. Model memprediksi probabilitas pada setiap frame audio dan menghitung selisih antara prediksi dengan transkripsi sebenarnya. AdamW digunakan sebagai optimizer untuk memperbarui parameter model secara bertahap, sementara learning rate diatur secara dinamis dengan scheduler untuk memastikan kestabilan. Selama pelatihan, model divalidasi menggunakan dataset terpisah, dan performanya diukur dengan metrics seperti Word Error Rate (WER). Setelah pelatihan selesai, model yang di-fine-tune mampu mengenali pola suara bahasa Jawa dengan lebih akurat dibandingkan model pretrained awal. Model ini kemudian dapat digunakan pada data audio bahasa Jawa lainnya dengan performa yang lebih baik.

Penggabungan Wav2Vec 2.0 dengan CTC dalam sistem pengenalan suara memberikan sinergi yang kuat. Dengan Wav2Vec 2.0 yang menyajikan fitur audio yang kaya dan terstruktur, ditambah dengan kemampuan CTC dalam menyusun pemetaan yang efisien antara audio dan teks, sistem ini dapat menghasilkan hasil yang lebih baik, terutama dalam konteks bahasa tertentu. Penulis melakukan finetuning model Wav2Vec 2.0 menggunakan dataset bahasa Jawa untuk meningkatkan akurasi transkripsi dalam konteks lokal. Hal ini penting karena karakteristik fonetik dan struktur linguistik bahasa Jawa bisa sangat berbeda dari bahasa yang lebih umum seperti bahasa Inggris. Dengan finetuning, model dapat belajar dari data spesifik bahasa Jawa, menghasilkan representasi yang lebih relevan dan meningkatkan performa sistem dalam pengenalan suara yang lebih akurat dan kontekstual.

2.5. *Evaluation*

Untuk memproses logits dan menghitung kinerja model, beberapa proses penting dilakukan pada tahap post-processing output model. Proses ini dimulai dengan mengolah logits yang dihasilkan oleh model, yang merupakan output mentah dari lapisan akhir model. Setelah post-processing, langkah pertama

adalah menggunakan fungsi argmax untuk mendapatkan prediksi terbaik dari output logits di setiap posisi. Rumus argmax yang digunakan adalah:

$$pred\_ids = arg\,max(pred\_logits, axis = -1)$$

Dengan menggunakan argmax, indeks logits yang memiliki nilai tertinggi pada setiap posisi dalam urutan prediksi dapat dipilih. Rumus ini membantu menentukan huruf atau kata mana yang diprediksi model di tiap tempat dalam urutan suara karena setiap indeks menunjukkan kata atau karakter yang akan diterjemahkan tokenizer. Setelah mendapatkan indeks sebagai hasil dari prediksi, langkah berikutnya adalah mengubah indeks tersebut menjadi teks dengan menggunakan tokenizer. Rumus yang digunakan untuk melakukan ini adalah:

$$pred\_str = processor.batch\_decode(preds\_ids)$$

Menurut rumus, pred_ids yang dibuat dari fungsi argmax kemudian diproses oleh tokenizer, yang mengubah indeks menjadi karakter teks. Decoding akan mengubah indeks menjadi karakter "a" jika itu menunjukkan huruf "a". Untuk membuat pengolahan data yang besar lebih mudah dan lebih cepat, proses ini dilakukan secara bertahap. Penanganan token padding juga penting dalam proses ini. Padding token digunakan selama pelatihan dan evaluasi untuk menyamakan panjang input dalam setiap batch. Namun, saat menghitung metrik kinerja, token padding ini tidak dihitung. Akibatnya, nilai ini diabaikan atau digantikan dengan nilai minus 100 selama perhitungan kesalahan. Rumus untuk menghindari padding adalah sebagai berikut:

$$pred.label\_ids[pred.label\_ids = -100 = processor.tokenizer.pad\_token\_id$$

Dengan menggunakan rumus ini, setiap kemunculan padding diganti dengan nilai minus seratus. Ini menunjukkan bahwa posisi tersebut tidak diambil dalam perhitungan metrik seperti Word Error Rate (WER). Pada tahap ini, metrik Word Error Rate (WER) digunakan; ini dihitung dengan menghitung kesalahan antara prediksi teks dan teks referensi. Rumus WER adalah:

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

Di mana, S menunjukkan jumlah kata yang salah digantikan (substitutions), D menunjukkan jumlah kata yang dihapus (deletions), I menunjukkan jumlah kata yang salah disisipkan (insertions), dan N menunjukkan jumlah total kata dalam referensi teks. Rumus ini menghitung kesalahan yang terjadi selama decoding dibandingkan dengan teks referensi, memberikan hasil kinerja model dalam memprediksi teks secara akurat. Semakin rendah nilai WER, semakin baik model dalam membuat prediksi suara yang akurat.

3. HASIL

3.1. Hasil Modelling

Pada tahap ini, penulis berhasil mengembangkan dan mengevaluasi model *speech to text* menggunakan arsitektur Wav2Vec 2.0, yang berfokus pada dialek bahasa Jawa. Metode yang digunakan dalam pengembangan ini adalah menggunakan pendekatana *finetuning*, dimana model Wav2Vec 2.0 terbukti efisien dalam memanfaatkan kemampuan pre-trained model, yang mampu mempercepat proses adaptasi terhadap dataset spesifik dan meminimalisir risiko *overfitting*.

Penulis menggunakan model *wav2vec2-base* dengan jumlah sebesar 94,4 juta parameter sedangkan model *wav2vec2-large* dengan jumlah sebesar 315 juta parameter. Kedua model dilatih dengan dataset bahasa Jawa, dan dilakukan eksperimen dengan parameter yang konsisten untuk menjaga stabilitas pelatihan dan memaksimalkan kemampuan model yang ada. Berikut adalah parameter yang digunakan dalam *finetuning* yang disajikan pada Tabel 1.

Tabel 1. Parameter *Finetuning*

	Parameter			
Model	Size (M)	learning_rate	batch_size	epochs
wav2vec2-base	94.4	0.0001	8	65
wav2vec2-large	315	0.0001	8	50

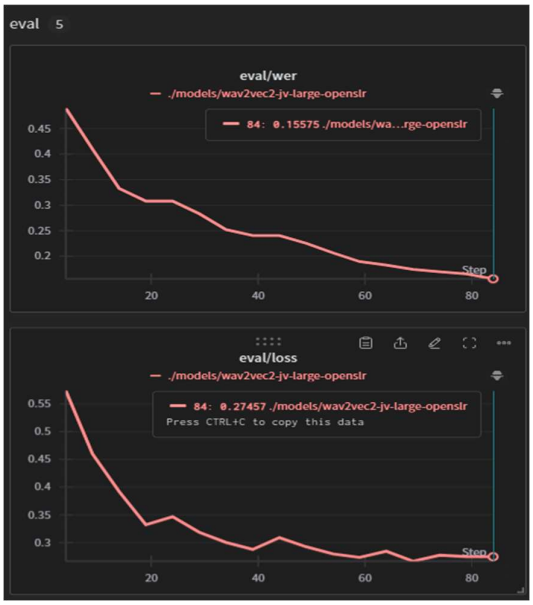
Dengan diberikan teknik *freezing layer* pada kedua model untuk menjaga representasi fitur pada kedua model yang telah dipelajari supaya tidak mempelajari ulang pada pola suara, penulis juga melakukan evaluasi kepada kedua model menggunakan metrik *Word Error Rate* (WER) sebagai indikator utama untuk menilai tingkat kesalahan dalam transkripsi suara ke teks, yang menjadi tolak ukur utama performa model. Selain itu, *evaluation loss* juga digunakan untuk menilai stabilitas dan keakuratan pembelajaran model. Berikut adalah hasil evaluasi kedua model yang disajikan pada Tabel 2.

Tabel 2. Hasil Evaluasi dan WER

Model	**Eval WER**	**Eval loss**
wav2vec2-base	15,02 %	0,2843
wav2vec2-large	15,57 %	0,2745

Dari Tabel 2, terlihat bahwa model *wav2vec2-base* memiliki kemampuan memahami suara bahasa Jawa dengan cukup baik, terbukti pada saat validasi menggunakan dataset bahasa Jawa dengan diukur menggunakan metrik WER. Sedangkan pada model *wav2vec2-large*, *evaluation loss* yang didapatkan cukup rendah daripada *wav2vec2-base*, sehingga model *wav2vec2-large* memahami konteks suara dan transkripsi pada dataset bahasa Jawa dengan cukup baik. Gambar metrik WER dan *evaluation loss* yang diperoleh dari pelatihan kedua model ini juga dapat dilihat pada Gambar 3 dan Gambar 4 yang menunjukkan perbandingan performa antara kedua model setelah *finetuning*.

https://doi.org/10.25077/TEKNOSI.v12i1.2026.001-009



Gambar 3. Evaluation loss dan WER – wav2vec2-large

Gambar 4. Evaluation loss dan WER – wav2vec2-base

3.2. Hasil Evaluasi

Penulis melakukan serangkaian uji coba pada kedua model, dengan memberikan suara dengan dialek Bahasa Jawa sebanyak 3 suara, sebagai pengukuran apakah kedua model dapat memberikan prediksi berupa teks yang sesuai dari *input* suara atau tidak. Hasil evaluasi dapat dilihat pada Tabel 3.

Terlihat bahwa pada kedua model dapat memberikan prediksi yang cukup dari input suara, dan dapat memahami konteks serta dialek pada suara bahasa Jawa.

Tabel 3. Evaluasi model menggunakan data suara

Model	Audio	Ground Truth	Predicted
wav2vec2-base	audio1.wav	Chiao lan Sharipov nglapurake jarak sing aman saka panyurung sikap sing bener.	cia olan saripof nglaporake jarak sing aman saka panyurung sitap sing beneru
	audio2.wav	Kembar Otter wis njajal ndharat ing Kokoda wingi nalika Maskapai Penerbangan PNG CG4684, nanging wis batal sepisan.	kembar otter wis njajal ndarat ingko kodhal wingi nalika mas kapai penerbangan pletteren letter gletter cler gle papat enemnou wolu papat nanging wis batal sebisan
	audio3.wav	Ani lagi nggoreng tempe.	ani lagi go reeng telim te
wav2vec2-large	audio1.wav	Chiao lan Sharipov nglapurake jarak sing aman saka panyurung sikap sing bener.	diao lan sarrypof ngelaporake jarak sing aman saka panyurung sikab sing bener
	audio2.wav	Kembar Otter wis njajal ndharat ing Kokoda wingi nalika Maskapai Penerbangan PNG CG4684, nanging wis batal sepisan.	kembar otter wis njadi alndarat engko kodawingi nalika maskabai penerbangan piterand gletter kletter kletter papat enemnou wolu papat nanging wis batal spisan
	audio3.wav	Ani lagi nggoreng tempe.	ani lagi nggo reyeng teyembik

4. PEMBAHASAN

4.1. Eksperimen Finetuning

Penulis melakukan pelatihan model *speech to text* dengan menggunakan arsitektur Wav2Vec 2.0 yang telah *pre-trained*, sebuah pendekatan yang dikenal sebagai *finetuning*. Dimana ditujukan sebagai efisiensi dan efektifitas pada waktu serta sumber daya yang ada. Dengan ini, penulis memanfaatkan kemampuan penuh pada model Wav2Vec 2.0 yang memiliki pengetahuan yang sudah ada dan mempercepat proses adaptasi pada dataset Bahasa Jawa. Selain itu, *finetuning* memungkinkan untuk meminimalisir adanya *overfitting* dan terbukti efektif dalam menangkap fitur akustik pada pola suara. Selain mempercepat waktu pelatihan, proses ini juga meningkatkan akurasi dan konteks pada bahasa yang spesifik, menjadikan pilihan yang efektif dibandingkan membangun model dari awal.

Dalam eksperimen ini, penulis menggunakan dua varian model Wav2Vec 2.0, yakni *wav2vec2-base* dengan 94,4 juta parameter dan *wav2vec2-large* dengan 315 juta parameter. Kedua model tersebut dilatih menggunakan dataset berbahasa Jawa yang terdiri dari 5.824 sampel, yang dibahas lebih lanjut pada Bab 2.1. Untuk

memastikan performa yang optimal, kedua model di *finetune* dengan parameter yang dapat dilihat pada Tabel 1. Hal ini bertujuan agar proses pelatihan lebih stabil dan memungkinkan model untuk belajar secara lebih optimal dalam menangkap fitur akustik dari dataset yang diberikan.

Selama proses *finetuning* berlangsung, penulis menerapkan metode teknik *freeze layer* di *feature encoder* pada kedua model. Ini ditujukan untuk mempertahankan representasi fitur yang telah dipelajari oleh model pada tahap *pre-training*, sehingga model tidak perlu mempelajari kembali fitur dasar suara dari nol, melainkan langsung fokus pada penyesuaian terhadap karakteristik khusus dari dataset berbahasa Jawa. Ini memberikan keuntungan dalam mempercepat proses pelatihan dan mengurangi penggunaan komputasi yang berlebihan.

Setelah proses *finetuning* selesai, penulis mengevaluasi performa kedua model menggunakan dua metrik yaitu *Word Error Rate* dan *loss*. Seperti yang ditampilkan pada Tabel 2, model wav2vec2-base berhasil mencapai WER sebesar 15,02% dan *evaluation loss* sebesar 0.2843. Sementara pada model wav2vec2-large, berhasil mencapai WER sebesar 15,57% dan *evaluation loss* sebesar 0,2745. Perbedaan metrik kedua model pada WER dan *evaluation loss* menunjukkan bahwa kedua model mampu mempelajari karakteristik bahasa Jawa dengan baik.

5. KESIMPULAN

Dapat disimpulkan bahwa pendekatan *finetuning* menggunakan *pre-trained model* Wav2Vec 2.0 untuk domain bahasa Jawa telah memberikan hasil yang signifikan dalam tugas *speech to text*, terbukti pada kinerja kedua model cenderung mirip, dengan selisih WER dan *evaluation loss* yang relatif kecil. Hal ini menunjukkan bahwa kedua model memiliki kemampuan yang kuat dalam menangkap fitur pada suara bahasa Jawa, meskipun ukuran dan kompleksitas model berbeda.

Proses *finetuning* dengan parameter yang diatur secara cermat, memberikan keseimbangan antara akurasi dan efisiensi waktu pelatihan. Selain itu, teknik *freezing* pada lapisan *feature encoder* terbukti membantu menjaga representasi fitur yang sudah terlatih, yang kemudian diterapkan secara lebih efektif pada domain bahasa baru. Temuan ini menggaris bawahi bahwa *finetuning* pada model *pre-trained* tidak hanya mempercepat proses pelatihan tetapi juga meningkatkan efektivitas model.

Namun demikian, pada penelitian ini penggunaan dataset yang masih terbatas dan tingkat WER yang diperoleh pada penelitian ini berada di kisaran ~15%, membuat adanya ruang untuk peningkatan performa pada model di penelitan selanjutnya.

DAFTAR PUSTAKA

[1] S. Novitasari, A. Tjandra, S. Sakti, and S. Nakamura, "Cross-Lingual Machine Speech Chain for Javanese, Sundanese, Balinese, and Bataks Speech Recognition and Synthesis," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.02128

[2] Z. Othman, N. Abdullah, Z. Razak, and M.-Y. Mohd-Yusoff, "Speech to Text Engine for Jawi Language," 2014.

[3] Z. A. Othman, Z. Razak, N. A. Abdullah, and Mohd. Y. Z. B. Mohd. Yusoff, "Jawi Character Speech-to-Text Engine Using Linear Predictive and Neural Network for Effective Reading," in *2009 Third Asia International Conference on Modelling & Simulation*, IEEE, 2009, pp. 348–352. doi: 10.1109/AMS.2009.94.

[4] N. M. Diah, M. Ismail, S. Ahmad, and S. A. S. Syed Abdullah, "Jawi on Mobile devices with Jawi wordsearch game application," in *CSSR 2010 - 2010 International Conference on Science and Social Research*, 2010, pp. 326–329. doi: 10.1109/CSSR.2010.5773793.

[5] "JPP 10 Nik Rosila ART 10 (161-172)".

[6] H. A. A. H. Shitiq and R. Mahmud, "Using an edutainment approach of a Snake and Ladder game for teaching Jawi script," in *2010 International Conference on Education and Management Technology*, IEEE, Nov. 2010, pp. 228–232. doi: 10.1109/ICEMT.2010.5657667.

[7] R. Jain, A. Barcovschi, M. Y. Yiwere, D. Bigioi, P. Corcoran, and H. Cucu, "A WAV2VEC2-Based Experimental Study on Self-Supervised Learning Methods to Improve Child Speech Recognition," *IEEE Access*, vol. 11, pp. 46938–46948, 2023, doi: 10.1109/ACCESS.2023.3275106.

[8] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," 2020. [Online]. Available: https://github.com/pytorch/fairseq

[9] st HAZ Sameen Shahgir, nd Khondker Salman Sayeed, and rd Tanjeem Azwad Zaman, "Applying wav2vec2 for Speech Recognition on Bengali Common Voices Dataset," 2022. [Online]. Available: https://huggingface.co/docs/transformers/index

[10] Z. Kozhirbayev, "Kazakh Speech Recognition: Wav2vec2.0 vs. Whisper," *Journal of Advances in Information Technology*, vol. 14, no. 6, pp. 1382–1389, 2023, doi: 10.12720/jait.14.6.1382-1389.

[11] P. Arisaputra, A. T. Handoyo, and A. Zahra, "XLS-R Deep Learning Model for Multilingual ASR on Low-Resource Languages: Indonesian, Javanese, and Sundanese."

[12] H. Liu *et al.*, "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning," May 2022, [Online]. Available: http://arxiv.org/abs/2205.05638

[13] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, "On the Effectiveness of Parameter-Efficient Fine-Tuning," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.15583

[14] R. S. A. Pratama and A. Amrullah, "Analysis Of Whisper Automatic Speech Recognition Performance On Low Resource Language," *Jurnal Pilar Nusa Mandiri*, vol. 20, no. 1, pp. 1–8, Mar. 2024, doi: 10.33480/pilar.v20i1.4633.

[15] R. Jain, A. Barcovschi, M. Y. Yiwere, D. Bigioi, P. Corcoran, and H. Cucu, "A WAV2VEC2-Based Experimental Study on Self-Supervised Learning

https://doi.org/10.25077/TEKNOSI.v12i1.2026.001-009

Methods to Improve Child Speech Recognition," *IEEE Access*, vol. 11, pp. 46938–46948, 2023, doi: 10.1109/ACCESS.2023.3275106.

[16] A. Dabiri, "Improving accuracy of speech recognition for low resource accents Testing the performance of fine-tuned Wav2vec2 models on accented Swedish," 2023.

BIODATA PENULIS

Johanes Setiawan
Seorang mahasiswa S1 Teknik Informatika di Universitas Dian Nuswantoro (UDINUS). Dan seorang asisten peneliti di "Bengkel Koding", yang merupakan sebuah program di bawah Fakultas Ilmu Komputer UDINUS. Fokus penelitian saya adalah Deep Learning, dan Pemrosesan Bahasa Alami (NLP).

Ardytha Luthfiarta, M.Kom,
Saat ini bekerja sebagai Dosen di Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Ia merupakan lulusan Magister Rekayasa Perangkat Lunak dan Sistem Cerdas dari Universitas Teknikal Malaysia Malacca. Ia memiliki ketertarikan yang besar dalam bidang Penelitian dan Pendidikan di bidang Kecerdasan Buatan, Data Mining, Pemrosesan Bahasa Alami, dan Deep Learning.

Adhitya Nugraha, S.Kom, M.CS
Beliau lahir di Palangkaraya, Indonesia, pada Maret 1987. Beliau menerima gelar Sarjana Ilmu Komputer dari Universitas Dian Nuswantoro (UDINUS), Semarang, Indonesia, pada tahun 2010, dan Magister Ilmu Komputer dari Universitas Teknikal Malaysia Melaka (UTeM), bidang Intelijen pada tahun 2012. Saat ini beliau bekerja sebagai Dosen di UDINUS. Bidang penelitiannya adalah Jaringan Komputer dan Kecerdasan Buatan.

Bastiaans, Jessica Carmelita
Seorang mahasiswa S1 Teknik Informatika di Universitas Dian Nuswantoro (UDINUS). Dan seorang asisten peneliti di "Bengkel Koding", yang merupakan sebuah program di bawah Fakultas Ilmu Komputer UDINUS. Fokus penelitian saya adalah Deep Learning, dan Pemrosesan Bahasa Alami (NLP).

Yohanes Deny Novandian
Seorang mahasiswa S1 Teknik Informatika di Universitas Dian Nuswantoro (UDINUS). Dan seorang asisten peneliti di "Bengkel Koding", yang merupakan sebuah program di bawah Fakultas Ilmu Komputer UDINUS. Fokus penelitian saya adalah Deep Learning, Pemrosesan Bahasa Alami (NLP) dan Pemrosesan Gambar.