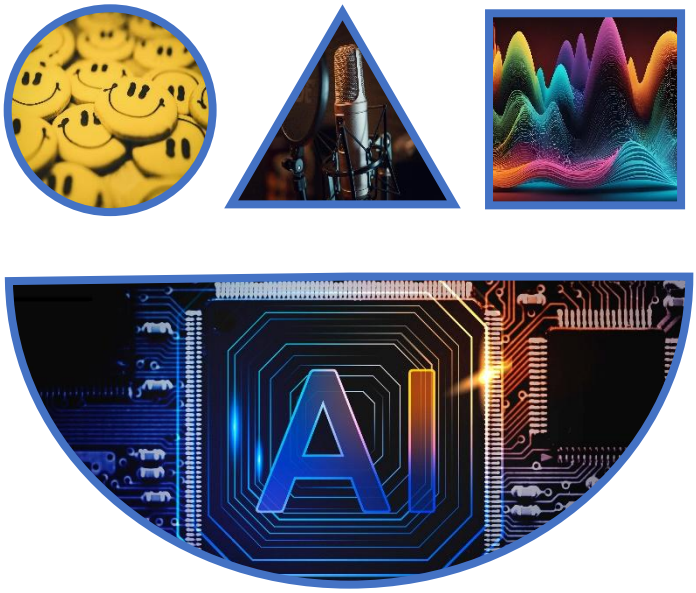
PENINGKATAN PERFORMA SISTEM KLASIFIKASI EMOSI SUARA DENGAN MEMPERTIMBANGKAN KETAHANAN *NOISE* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*



ZID IRSYADIN SARTONO WIJAOGY

D121201016

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2024

PENINGKATAN PERFORMA SISTEM KLASIFIKASI EMOSI SUARA DENGAN MEMPERTIMBANGKAN KETAHANAN *NOISE* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

ZID IRSYADIN SARTONO WIJAOGY
D121201016

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2024

PENINGKATAN PERFORMA SISTEM KLASIFIKASI EMOSI SUARA DENGAN MEMPERTIMBANGKAN KETAHANAN *NOISE* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

ZID IRSYADIN SARTONO WIJAOGY
D121201016

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Teknik Informatika

pada

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2024

SKRIPSI

PENINGKATAN PERFORMA SISTEM KLASIFIKASI EMOSI SUARA DENGAN MEMPERTIMBANGKAN KETAHANAN *NOISE* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

ZID IRSYADIN SARTONO WIJAOGY
D121201016

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Teknik Informatika pada 28 November 2024 dan dinyatakan telah memenuhi syarat kelulusan

pada

Program Studi Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknik
Universitas Hasanuddin
Makassar

Mengesahkan:

Pembimbing Utama,

Prof. Dr. Ir. Indrabayu, S.T., M.T.,
M.Bus.Sys., IPM, ASEAN. Eng.
NIP 19750716 200212 1 004

Pembimbing Pendamping,

Ir. Anugrayani Bustamin,
S.T., M.T.
NIP 19901201 201807 4 001

Mengetahui:
Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S.T., M.T.,
M.Bus.Sys., IPM, ASEAN. Eng.
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN SKRIPSI
DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "Peningkatan Performa Sistem Klasifikasi Emosi Suara dengan Mempertimbangkan Ketahanan *Noise* Menggunakan *Convolutional Neural Network*" adalah benar karya saya dengan arahan dari pembimbing Prof. Dr. Ir. Indrabayu, M.T., M.Bus.Sys., IPM, ASEAN. Eng. sebagai Pembimbing Utama dan Ir. Anugrayani Bustamin, S.T., M.T. sebagai Pembimbing Pendamping. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 28-11-2024

ZID IRSYADIN SARTONO WIJAOGY
NIM D121201016

UCAPAN TERIMA KASIH

Alhamdulillah, segala puji bagi Allah SWT atas segala rahmat, hidayah, dan kemudahan yang telah diberikan kepada saya dalam menyelesaikan skripsi ini dan menempuh seluruh proses pendidikan dengan baik.

Saya mengucapkan terima kasih yang sebesar-besarnya kepada Prof. Dr. Ir. Indrabayu, M.T., M.Bus.Sys. selaku Pembimbing Utama atas bimbingan, dukungan, dan arahan yang tak ternilai selama penyusunan skripsi ini. Terima kasih juga saya sampaikan kepada Ir. Anugrayani Bustamin, S.T., M.T., Pembimbing Pendamping, atas kesediaannya mendampingi saya dengan masukan dan wawasan yang sangat berarti.

Ucapan terima kasih yang tulus saya sampaikan kepada Beasiswa Karya Salemba Empat yang telah memberikan bantuan beasiswa sehingga saya dapat menempuh program pendidikan ini tanpa kendala finansial.

Saya juga ingin menyampaikan penghargaan yang mendalam kepada kedua orang tua saya Mujahidin Agus, Luluk Pian Utami, dan saudara-saudari saya Zid Jaysyaldin Sartono Wijaogy, Zid Zahfidin Sartono Wijaogy, Zidny Zhafira Sartono Wijaogy serta seluruh keluarga tercinta yang telah memberikan dukungan moral, doa, dan pengorbanan yang tiada henti, menjadi pilar kekuatan saya selama menjalani proses pendidikan ini. Kepada Azzahra Beladina Shaff, partner spesial yang selalu mendampingi dan memberikan dukungan penuh, saya ucapkan terima kasih atas motivasi dan semangat yang diberikan selama ini.

Tidak lupa, terima kasih saya sampaikan kepada para partner diskusi saya: Kak Rezky, Rama, Mekuma, Kevin, yang selalu siap memberikan masukan dan pandangan berharga. Terima kasih juga kepada Iksan, Agunawan, Sappe, Kak Sabda, Zahrayy, Ucha, Aldilah, Tri dan Deswi atas bantuan mereka dalam pembuatan *dataset* yang sangat membantu dalam penelitian ini.

Saya juga berterima kasih kepada platform dan aplikasi yang telah memudahkan proses penelitian dan analisis data yang saya lakukan.

Akhirnya, terima kasih yang sebesar-besarnya saya sampaikan kepada semua pihak yang tidak dapat saya sebutkan satu per satu atas segala dukungan dan bantuan yang telah diberikan. Semoga segala kebaikan dan kerja sama ini membawa manfaat dan keberkahan untuk kita semua.

.

ABSTRAK

ZID IRSYADIN SARTONO WIJAOGY. **Peningkatan Performa Sistem Klasifikasi Emosi Suara dengan Mempertimbangkan Ketahanan *Noise* Menggunakan *Convolutional Neural Network*** (dibimbing oleh Indrabayu dan Anugrayani Bustamin).

Latar belakang. Video *review* produk kini menjadi faktor penting yang memengaruhi keputusan konsumen, di mana analisis emosi suara dapat memberikan wawasan lebih mendalam terkait persepsi *reviewer*. Namun, gangguan *noise* dalam audio sering kali mengurangi akurasi sistem pengenalan emosi. **Tujuan**. Penelitian ini bertujuan untuk meningkatkan performa dan ketahanan model terhadap *noise* pada klasifikasi emosi suara dalam konteks video ulasan produk. **Metode**. Model yang digunakan adalah *Convolutional Neural Network* (CNN) yang dilatih menggunakan *dataset* suara dari berbagai sumber. Data diproses dengan teknik *augmentation* menggunakan simulasi *noise* melalui variasi *Amplitude Factor* (AF) dan ekstraksi fitur seperti MFCC, ZCR, mel spectrogram, chroma STFT, dan RMS. Evaluasi dilakukan pada data bersih dan data yang telah diinjeksi *noise* untuk mengukur ketahanan model terhadap gangguan akustik. **Hasil**. Hasil penelitian menunjukkan bahwa model mencapai akurasi pelatihan sebesar 99% pada semua *dataset*. Pada pengujian data yang diinjeksi *noise*, performa model bervariasi tergantung tingkat AF, dengan *Weighted Accuracy* (WA) tertinggi mencapai 99%. Temuan ini menunjukkan bahwa pelatihan dengan data yang mengandung *noise* dan variasi AF secara signifikan meningkatkan ketahanan model terhadap *noise* tanpa mengorbankan akurasi. Penelitian ini menekankan pentingnya *augmentation* data, pemilihan fitur, dan variasi AF dalam mengembangkan sistem pengenalan emosi yang tangguh untuk lingkungan berisik. **Kesimpulan**. Penerapan *augmentation*, pemilihan ekstraksi fitur tertentu serta pelatihan model dengan data *noisy* dengan berbagai level *Amplitude Factor* (AF) secara substansial meningkatkan ketahanan model dalam pengenalan emosi dari suara pada konteks video *review* produk.

Kata kunci: *robustness*; *deep learning*; sentimen pelanggan; *augmentation* audio; *signal processing*.

ABSTRACT

ZID IRSYADIN SARTONO WIJAOGY. **Enhancing the Performance of Speech Emotion Classification Systems by Considering Noise Robustness Using Convolutional Neural Networks** (supervised by Indrabayu dan Anugrayani Bustamin).

Background. Product's review video have become a key factor influencing consumer decisions, where speech emotion analysis provides deeper insights into reviewer perceptions. However, audio *noise* often reduces the accuracy of emotion recognition systems. **Aim.** This study aims to improve the performance and robustness of models against *noise* in speech emotion classification within the context of product review videos. **Method.** The model utilized is a Convolutional Neural Network (CNN) trained on audio datasets from various sources. The data was processed using augmentation techniques by simulating *noise* through variations in Amplitude Factor (AF) and feature extraction methods, including MFCC, ZCR, mel spectrogram, chroma STFT, and RMS. Evaluations were conducted on both clean and *noise*-injected data to measure the model's robustness against acoustic disturbances. **Results.** The study revealed that the model achieved a training accuracy of 99% across all datasets. When tested on *noise*-injected data, the model's performance varied depending on the AF level, with the highest Weighted Accuracy (WA) reaching 99%. These findings highlight that training with *noise*-augmented data and AF variations significantly enhances the model's robustness to *noise* while maintaining high predictive accuracy. The study underscores the importance of data augmentation, feature selection, and AF variations in developing robust emotion recognition systems for noisy environments. **Conclusion.** The application of augmentation techniques, specific feature extraction methods, and training model with noisy data with various Amplitude Factor (AF) significantly enhances the model's robustness in recognizing emotions from speech in the context of product review videos.

Keywords: robustness; deep learning; customer sentiment; audio augmentation; signal processing.

DAFTAR ISI

Halaman

DAFTAR TABEL

Nomor urut Halaman

DAFTAR GAMBAR

Nomor urut Halaman

DAFTAR LAMPIRAN

Nomor urut Halaman

Istilah	Arti dan Penjelasan
Amplitudo	besaran yang menggambarkan puncak tertinggi atau terendah dari gelombang dalam suatu sinyal, seperti sinyal suara, gelombang listrik, atau gelombang cahaya.
Augmentation	teknik yang digunakan untuk memperbanyak jumlah data pelatihan dengan cara membuat variasi dari data yang ada, tanpa menambah data baru.
Dimensionalitas	jumlah variabel atau fitur yang digunakan untuk merepresentasikan data dalam suatu ruang
Dropout	*teknik regulasi yang digunakan dalam neural networks (jaringan saraf tiruan) untuk mengurangi overfitting selama proses pelatihan.*
Filter Trapesium	jenis filter berbentuk trapesium yang digunakan dalam analisis spektrum untuk memetakan frekuensi sinyal ke dalam skala tertentu, seperti skala mel
Fitur Linguistik	elemen-elemen atau karakteristik dalam data bahasa yang digunakan untuk analisis atau pemrosesan dalam berbagai aplikasi, seperti natural *language processing* , *text mining*, atau *speech recognition*
Flattening	proses mengubah struktur data berukuran banyak dimensi menjadi satu dimensi (vektor).
Interpolasi	metode untuk memperkirakan nilai di antara dua titik data yang diketahui.
Konvergensi	proses ketika suatu algoritma secara bertahap mendekati solusi optimal atau mencapai nilai yang stabil setelah sejumlah iterasi.
Konvolusi	operasi matematika yang melibatkan dua fungsi untuk menghasilkan fungsi ketiga yang menggambarkan bagaimana bentuk satu fungsi dipengaruhi oleh bentuk fungsi lainnya.
Korpus	kumpulan teks atau data bahasa yang digunakan sebagai sumber untuk analisis linguistik atau pemrosesan bahasa alami (nlp).
Logaritmik	istilah yang menggambarkan hubungan matematis di mana suatu variabel berubah secara proporsional terhadap logaritma dari variabel lain.
Mel	satuan pengukuran yang digunakan untuk menggambarkan persepsi manusia terhadap frekuensi suara.

Multimodal	memproses dan menggabungkan informasi dari berbagai sumber atau modalitas yang berbeda, seperti teks, suara, gambar, dan gerakan.
Multivariat	analisis atau pemodelan yang melibatkan lebih dari satu variabel atau dimensi.
Noise	gangguan atau suara tidak diinginkan yang hadir dalam lingkungan perekaman
Non-Tonal	suara yang tidak memiliki nada dasar atau frekuensi fundamental yang teratur.
Onset	titik awal atau permulaan dari suatu peristiwa atau sinyal.
Para-Linguistik	aspek-aspek komunikasi verbal yang tidak terkait langsung dengan kata-kata atau struktur linguistik, tetapi lebih kepada bagaimana sesuatu dikatakan, seperti intonasi, nada suara, kecepatan bicara, volume, dan kualitas suara.
Pooling	teknik mengurangi dimensi atau ukuran representasi data, sambil mempertahankan fitur penting
Prosodi	elemen-elemen suprasegmental dalam sinyal suara yang mencakup pola ritme, intonasi, tekanan, dan durasi dalam ucapan.
Sentimen	pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu
Silabel	unit suara dalam bahasa yang terdiri dari satu atau lebih fonem dan diucapkan sebagai satu kesatuan ritmis.
Spectral	analisis dan representasi sinyal atau data berdasarkan frekuensi yang terkandung di dalamnya
Stokastik	proses atau sistem yang melibatkan ketidakpastian atau variabilitas acak.
Tonal	suara yang memiliki frekuensi fundamental yang jelas dan teratur, sehingga menghasilkan kesan nada tertentu
Windowing	teknik yang digunakan dalam pemrosesan sinyal untuk memotong atau membagi sinyal menjadi segmen-segmen kecil (dikenal sebagai *window*) sebelum dianalisis lebih lanjut.

Lambang/singkatan	Arti Dan Penjelasan
1D-CNN	One-Dimensional Convolutional Neural Network
ACNN	Attention-Based Convolutional Neural Network
AF	Amplitude Factor (Af)
DAW	Digital Audio Workstation
dB	Desibel
DCT	Discrete Cosinus Transform
DFT	Discrete Fourier Transform
CNN	Convolutional Neural Networks
Emo-DB	Berlin Emotional Database
ESC	Environmental Sound Classification
FCNN	Fully Convolutional Neural Network
FFT	Fast Fourier Transform
fmax	Frekuensi Maksimal
FN	False Negative
FP	False Positives
GRU	Gated Recurrent Unit
Hz	Hertz
IEMOCAP	Interactive Emotional Dyadic Motion Capture
IFFT	Inverse Fast Fourier Transform
KBBI	Kamus Besar Bahasa Indonesia
kHz	Kilohertz
LPC	Linear Predictor Coefficient
LSTM	Long Short-Term Memory
MFCCs	Mel-Frequency Cepstral Coefficients
MGD	Log-Mel Dan Delay Grup
ms	Milisecond
RAVDESS	Ryerson Audio-Visual Database Of Emotional Speech And Song
ReLU	Rectified Linear Unit
RMS	Root Mean Square
SER	Speech Emotion Recognition
SNR	Signal-To-*Noise* Ratio
STFT	Short-Time Fourier Transform
TN	True Negatives
TP	True Positives
WA	Weighted Accuracy
ZCR	Zero Crossing Rate

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa dekade terakhir, terjadinya revolusi teknologi, terutama di dunia internet, telah mengubah cara kita berinteraksi dengan lingkungan sekitar. Transformasi ini semakin diperkuat oleh platform multimedia, seperti YouTube, yang tidak hanya menjadi sumber hiburan tetapi juga menjadi pionir dalam revolusi konten digital (Dafit, 2023). YouTube telah berhasil menciptakan ekosistem yang memungkinkan pengguna dengan mudah menemukan dan berbagi berbagai jenis video, termasuk *review* produk. Peran signifikan YouTube dalam memengaruhi preferensi dan keputusan konsumen tercermin dalam survei oleh Nielsen awal tahun ini. Nielsen menemukan bahwa 52% konsumen berusia 35-49 tahun dan 49% orang berusia 18-34 tahun mengatakan bahwa mereka dipengaruhi untuk membeli produk yang mereka lihat dalam *streaming* konten video (Nielsen, 2021).

Dalam era di mana multimedia memainkan peran kunci, analisis sentimen menjadi semakin kompleks, terutama ketika berkaitan dengan *review* produk dalam format video. Penelitian ini menyoroti pentingnya tidak hanya menganalisis teks atau transkrip, tetapi juga dimensi suara sebagai ekspresi emosi yang kompleks. Suara manusia dalam video *review* produk tidak hanya menyampaikan informasi melalui kata-kata tertulis, tetapi juga melalui intonasi, ekspresi wajah, dan variasi suara lainnya (Ambar, 2017). Analisis emosi dari suara dalam video *review* produk membantu dalam memperkaya pemahaman dan memberikan nuansa yang lebih dalam tentang perasaan pembuat konten terhadap produk yang diulas. Hal ini sangat berguna bagi pembuat keputusan dan penonton untuk mendapatkan wawasan yang lebih lengkap dan akurat, sehingga mampu menghemat waktu dan mendapatkan informasi yang lebih relevan dalam lingkungan digital yang dipenuhi dengan berbagai sumber informasi.

Dalam pengembangan teknologi analisis sentimen, terutama yang berbasis *Speech Emotion Recognition* (SER) dalam *review* produk berbentuk video, faktor *noise* pada audio menjadi aspek krusial yang perlu diperhatikan. *Noise* dapat diartikan sebagai gangguan atau suara tidak diinginkan yang muncul di lingkungan perekaman, seperti suara kendaraan, mesin industri, atau elemen eksternal lainnya. Kehadiran *noise* dapat berdampak signifikan pada interpretasi emosi yang terdeteksi dari suara, sehingga dapat mengaburkan atau mengubah persepsi emosi yang sebenarnya. Dalam menghadapi masalah *noise* pada *Speech Emotion Recognition* (SER), salah satu metode yang digunakan adalah kombinasi algoritma yang baik dalam menghadapi *noise*, seperti dengan menggunakan algoritma yang fleksibel sehingga sistem tetap mampu mendeteksi emosi secara akurat meskipun kondisi *noise* berubah (Vásquez-Correa et al., 2015).

Selain itu, *Amplitude Factor* (AF) dan *Signal-to-Noise Ratio* (SNR) merupakan parameter penting yang mempengaruhi kualitas rekaman suara dan,

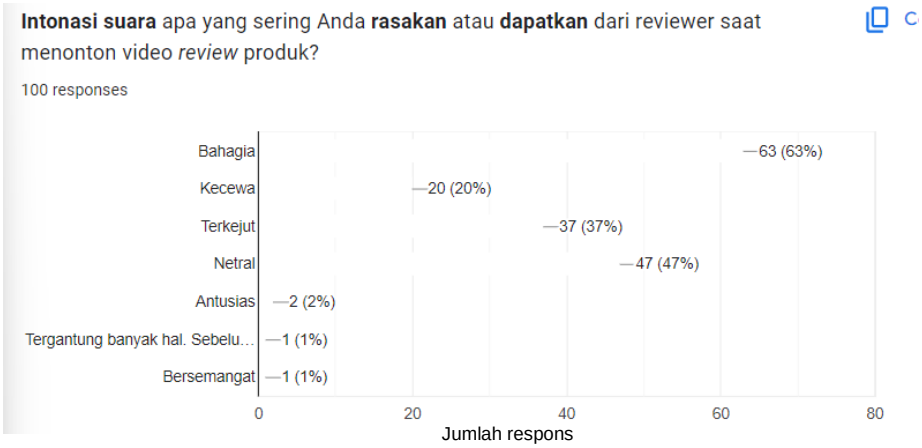
pada gilirannya, akurasi sistem *Speech Emotion Recognition* (SER). *Amplitude Factor* (AF) mengacu pada tingkat intensitas suara, di mana variasi amplitudo dapat mempengaruhi persepsi emosi yang terekam. SNR, yang merupakan perbandingan antara kekuatan sinyal (suara) dan kekuatan *noise*, sangat menentukan seberapa jelas suara yang diinginkan terdengar dibandingkan dengan *noise* latar. Tingkat SNR yang rendah menunjukkan bahwa *noise* lebih dominan dibandingkan sinyal, yang dapat mengaburkan atau mengubah persepsi emosi yang sebenarnya (Xu, Zhang, Yang, et al., 2020).

Beberapa penelitian sebelumnya telah mengkaji berbagai metode dalam *noisy speech emotion recognition*. Widiyanti & Endah (2018) mengeksplorasi penggunaan spektrum *log-mel* dan *Modification Grup Delay* (MGD) dengan arsitektur *Fully Convolutional Neural Network* (FCNN) yang dioptimalkan dengan *AutoKeras*, mencapai akurasi rata-rata 81% pada kondisi *clean speech* dan 76% pada kondisi *noisy speech* setelah penambahan *noise*. Xu et al. (2021) mengembangkan metode *Head Fusion* yang berbasis pada mekanisme *multi-head attention* untuk meningkatkan akurasi pengenalan emosi suara. Mereka menggunakan Attention-based *Convolutional Neural Network* (ACNN) dan melakukan eksperimen pada *dataset* IEMOCAP dan RAVDESS, menunjukkan peningkatan akurasi menjadi 76.18% (*Weighted Accuracy*) dan 76.36% (*Unweighted Accuracy*). Abdelhamid et al. (2022) mengusulkan pendekatan CNN+LSTM yang dioptimalkan dengan algoritma pencarian *fractal* stokastik, mencapai akurasi rata-rata lebih dari 98% pada *dataset* IEMOCAP, SAVEE, dan Emo-DB. Rayhan Ahmed et al. (2023) mengembangkan model *ensemble* yang terdiri dari 1D-CNN, LSTM, dan GRU dengan teknik data *augmentation*, menggunakan fitur spektrum seperti MFCCs, LMS, ZCR, *Chromagram*, dan RMS sebagai input, mencapai akurasi tertinggi 97% pada beberapa *dataset*. Integrasi penelitian-penelitian ini memberikan fondasi kuat untuk pengembangan lebih lanjut dalam bidang pengenalan emosi suara, khususnya dengan fokus pada optimasi *noise* adaptif untuk meningkatkan akurasi dan ketahanan model dalam berbagai kondisi lingkungan.

Penelitian yang dilakukan oleh Bustamin et al. (2024) berjudul "Analisis Sentimen Terhadap Video Berdasarkan Pengenalan Emosi pada Data Audio" memiliki beberapa kelemahan yang signifikan yang mempengaruhi validitas dan generalisasi hasil penelitian. Keterbatasan data menjadi salah satu kelemahan utama, di mana penelitian ini menggunakan data primer dari 10 responden dan data sekunder dari 60 sampel video YouTube pada kanal GadgetIn. Jumlah data yang relatif kecil ini dapat menyebabkan kurangnya representatif dan risiko *overfitting*, di mana model terlalu terlatih pada data terbatas dan tidak berkinerja baik pada data baru atau tidak terlihat. Selain itu, variasi emosi yang tidak relevan juga menjadi masalah, terutama karena penelitian ini mencakup lima kelas emosi: *neutral*, *happy*, *surprise*, *disgust*, dan *disappointed*, dengan emosi *disgust* yang jarang ditemui dalam konteks video *review* produk. Kondisi ini dapat mengganggu performa model dalam aplikasi nyata. Kondisi *noise* yang tidak dipertimbangkan dalam penelitian ini juga merupakan kelemahan signifikan, karena model dilatih dalam kondisi tanpa *noise* atau dengan *noise* yang tidak bervariasi, sehingga kurang *robust* dan sulit diterapkan

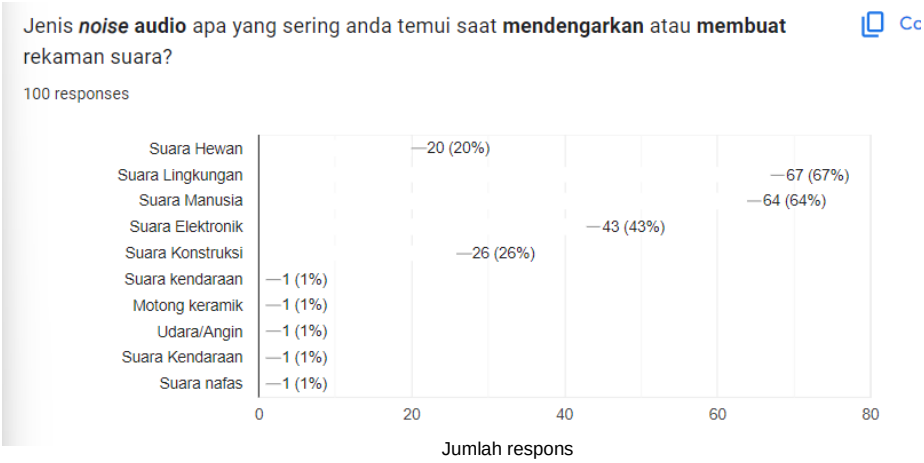
dalam kondisi nyata dimana *noise* suara bervariasi dan sering hadir (Xu, Zhang, Yang, et al., 2020). Selain itu, data *augmentation* yang dilakukan menggunakan metode *noise*, *time stretch*, *pitch, dan shift* untuk meningkatkan jumlah data dan variasi belum optimal dalam menangani variasi data sekunder, terlihat dari akurasi model data sekunder dengan *augmentation* (85%) yang masih lebih rendah dibandingkan model data primer dengan *augmentation* (94%).

Dalam konteks penelitian ini, telah dilakukan survei untuk merinci dan mengidentifikasi emosi-emosi yang cenderung muncul pada sentimen yang terkandung dalam video *review* produk. Hasil survei dari 100 responden dengan total 171 respons (*multi answer*) menunjukkan bahwa 63% respons merasakan emosi bahagia, 20% merasakan emosi kecewa, 47% merasakan emosi netral, dan 37% merasakan emosi terkejut. Emosi lain seperti antusias, tergantung banyak hal, dan bersemangat hanya disebutkan oleh sedikit responden. Dari hasil survei tersebut, tiga emosi utama yang dipilih untuk analisis lebih lanjut adalah bahagia, netral, dan kecewa. Ketiganya merepresentasikan kategori sentimen yang luas, yaitu positif, netral, dan negatif, yang menjadi fokus dalam penelitian ini.



Gambar 1. Distribusi intonasi suara yang sering dirasakan atau didapatkan dari responden saat menonton video *review* produk

Selain mengidentifikasi emosi-emosi yang cenderung muncul, survei awal yang dilakukan juga mengidentifikasi jenis atau tipe *noise* yang sering ditemui saat mendengarkan atau membuat rekaman suara. Hasil respons menunjukkan bahwa jenis *noise* paling sering ditemui adalah suara lingkungan (67%), diikuti oleh suara manusia (64%), suara elektronik (43%), suara konstruksi (26%), dan suara hewan (20%). Jenis *noise* lain seperti suara kendaraan, nafas, potong keramik, dan udara/angin hanya diidentifikasi oleh sedikit responden. Hasil ini menunjukkan bahwa sebagian besar responden menyadari adanya berbagai macam *noise* yang dapat mengganggu kualitas rekaman suara.



Gambar 2. Distribusi jenis *noise* audio yang sering ditemui saat mendengarkan atau membuat rekaman suara

Mayoritas responden (70.87%) menyatakan bahwa emosi yang mereka rasakan saat menonton video *review* produk mempengaruhi keputusan pembelian mereka. Selain itu, 90.29% responden menganggap bahwa rekaman suara yang jelas dan bebas dari gangguan sangat penting untuk memahami emosi yang disampaikan melalui suara seseorang. Hal ini menekankan pentingnya kualitas rekaman suara dalam penelitian ini, terutama dalam konteks pengenalan emosi suara yang akan dianalisis.

Hasil dari survei ini diharapkan dapat memberikan wawasan lebih lanjut tentang variasi emosi yang ditemui dalam konteks video *review* produk, serta variasi jenis *noise* yang sering kita temui ketika mendengar atau membuat rekaman suara. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi emosi yang difokuskan pada tiga kategori emosi utama, yaitu netral, bahagia, dan kecewa, dalam berbagai kondisi *noise* audio yang berbeda. Sistem ini akan dibangun dengan pendekatan *noise* untuk meningkatkan akurasi dan ketahanan terhadap berbagai tingkat gangguan *noise*, yang kerap ditemukan dalam rekaman audio. Selain itu, penelitian ini juga akan mengeksplorasi pengaruh *Amplitude Factor* (AF) terhadap performa sistem dalam mengklasifikasikan emosi pada berbagai kondisi *noise*. Pengetahuan ini diharapkan dapat memberikan pemahaman lebih mendalam tentang bagaimana AF memengaruhi ketahanan dan akurasi sistem dalam menghadapi tingkat gangguan *noise* yang berbeda-beda. Berdasarkan latar belakang tersebut, penulis mengambil penelitian dengan judul "**Peningkatan Performa Sistem Klasifikasi Emosi Suara dengan Mempertimbangkan Ketahanan *Noise* Menggunakan *Convolutional Neural Network***". Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam bidang analisis sentimen multimedia, khususnya dalam aplikasi pengenalan emosi suara yang andal dan tahan terhadap gangguan *noise*.

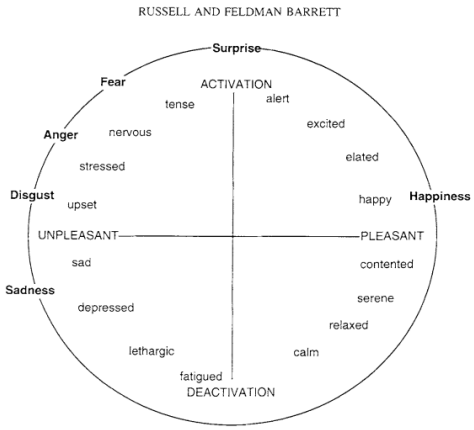
1.2 Landasan Teori

1.2.1 Emosi

Emosi adalah respons kompleks yang melibatkan pengalaman subjektif, respons fisiologis, dan ekspresi perilaku. Emosi memainkan peran penting dalam kehidupan sehari-hari manusia, mempengaruhi pikiran, tindakan, dan interaksi sosial. Emosi dapat dianggap sebagai reaksi terhadap stimulus yang penting bagi individu dan dapat bervariasi dari kegembiraan hingga kesedihan, kemarahan, ketakutan, dan banyak lagi (K. R. Scherer, 2005a).

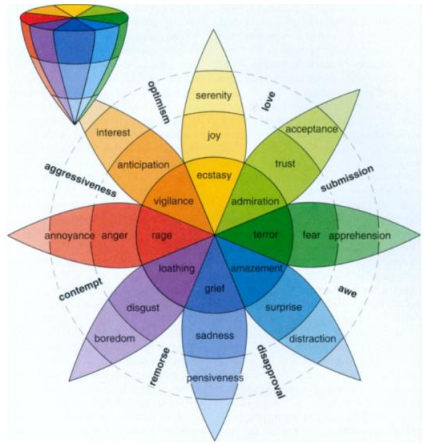
Ucapan manusia sendiri mengandung banyak informasi, termasuk fitur linguistik dan para-linguistik (Madanian et al., 2023). Fitur linguistik mengacu pada pola kualitatif dalam artikulasi manusia, seperti konten dan konteks, sedangkan fitur para-linguistik secara kuantitatif menggambarkan variasi dalam pengucapan pola linguistik (Madanian et al., 2023, sebagaimana dikutip dalam Anagnostopoulos et al., 2015; Zhao et al., 2019) . Dalam hal ini termasuk fitur prosodi (*subset* fitur para-linguistik) itu meliputi *pitch*, energi, artikulasi, dan fitur spektrum, seperti *Linear Predictor Coefficient* (LPC) dan *Mel-Frequency Cepstral Coefficients* (MFCCs) (Madanian et al., 2023, sebagaimana dikutip dalam Frant et al., 2017). Sebagai contoh, emosi kesedihan sering kali ditandai dengan standar deviasi *pitch* yang rendah dan kecepatan bicara yang lambat, sedangkan emosi marah cenderung memiliki standar deviasi *pitch* dan intensitas yang lebih tinggi serta kecepatan bicara yang cepat (K. R. Scherer, 2005).

Untuk memahami dan mengategorikan emosi, berbagai teori telah dikembangkan untuk memberikan kerangka kerja yang sistematis. Salah satu pendekatan yang dikenal luas adalah model dimensi emosi yang diusulkan oleh Russell dan Barrett (1999). Model ini memetakan berbagai emosi berdasarkan dua dimensi utama: *valensi* (tingkat menyenangkan atau tidak menyenangkannya suatu emosi) dan *arousal* (tingkat aktivasi atau intensitas emosi tersebut). Sumbu vertikal menggambarkan emosi dari aktivasi tinggi hingga deaktivasi, seperti tegang dan lelah. Sementara itu, sumbu horizontal menggambarkan tingkat valensi, dari tidak menyenangkan seperti sedih hingga menyenangkan seperti bahagia. Model ini memberikan perspektif tentang bagaimana emosi yang berbeda dapat dikategorikan dan saling berkaitan berdasarkan dua dimensi ini, sehingga memudahkan pemahaman mengenai spektrum emosi manusia.



Gambar 3. Dimensi *valence-arousal emotion* (Russell & Barrett, 1999a)

Selain model Russell & Barrett, Plutchik (2001) mengembangkan model yang dikenal sebagai Plutchik's *Wheel of Emotions*. Model ini menggambarkan delapan emosi dasar yang terhubung satu sama lain dalam bentuk roda emosi. Emosi-emosi dasar ini diatur secara berpasangan, seperti kegembiraan yang berlawanan dengan kesedihan, kepercayaan dengan jijik, dan kemarahan dengan ketakutan. Plutchik juga menggambarkan intensitas masing-masing emosi dengan menggunakan gradasi warna, di mana emosi dengan intensitas tinggi seperti ekstasi terletak di pusat roda, dan emosi dengan intensitas rendah seperti ketenangan berada di bagian luar roda. Salah satu fitur penting dari model ini adalah bahwa emosi dasar dapat bergabung untuk membentuk emosi kompleks, seperti cinta yang merupakan kombinasi dari kegembiraan dan kepercayaan. Model ini memberikan cara sistematis untuk memahami bagaimana emosi saling berkaitan dan dapat berubah dalam intensitas.



Gambar 4. Plutchik's *Wheel of Emotions* (Plutchik, 2001)

Pendekatan-pendekatan ini, baik dari Russell & Barrett maupun Plutchik, memberikan dasar yang penting untuk memahami emosi dalam berbagai konteks, termasuk dalam pengenalan emosi suara. Model dimensi emosi dari Russell & Barrett memungkinkan kita untuk melihat bagaimana aktivasi dan valensi dapat mempengaruhi persepsi emosional seseorang, sedangkan roda emosi Plutchik memberikan panduan tentang bagaimana emosi dasar dapat dikombinasikan dan diukur dalam hal intensitas dan keterkaitan antar emosi. Kedua model ini akan sangat relevan dalam memahami respons emosional terhadap video *review* produk, di mana emosi yang dirasakan oleh responden bisa diidentifikasi dan dianalisis menggunakan kerangka kerja tersebut.

1.2.2 Analisis sentimen

Menurut Kamus Besar Bahasa Indonesia (KBBI), analisis merupakan "penyelidikan terhadap suatu peristiwa (karangan, perbuatan, dan sebagainya) untuk mengetahui keadaan yang sebenarnya (sebab-musabab, duduk, perkaranya, dan sebagainya)". Sedangkan sentimen "merupakan pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu" (Kamus Besar Bahasa Indonesia (KBBI), n.d.). Dengan demikian, analisis sentimen adalah penyelidikan yang terdiri dari serangkaian aktivitas mengurai, membedakan, mencari kaitan, hingga menafsirkan pendapat atau pandangan yang didasarkan pada perasaan seseorang.

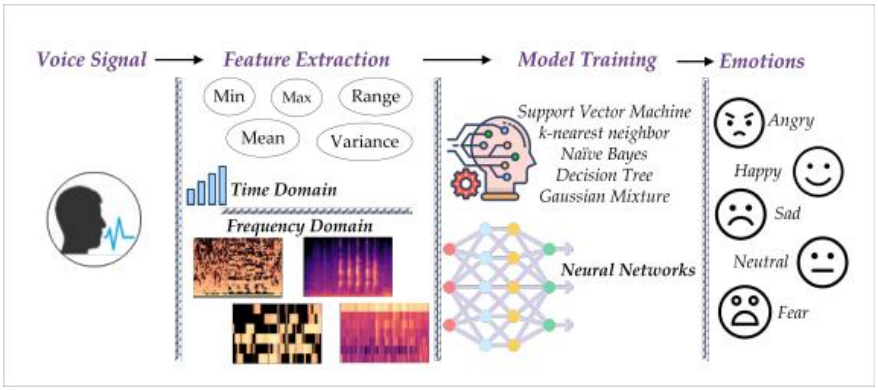
Analisis sentimen bertujuan untuk mengekstrak atribut atau fitur dari suatu teks atau dokumen yang biasanya diperoleh dari media sosial dan nantinya dapat dikategorikan apakah teks tersebut masuk ke dalam sentimen positif, netral, dan negatif (Fahrudin et al., 2022). Proses ini penting untuk memahami pola opini publik, tren, dan reaksi terhadap isu tertentu, serta untuk pengambilan keputusan yang lebih informatif.

1.2.3 *Speech emotion recognition*

Pengenalan emosi suara atau *Speech Emotion Recognition* (SER) merupakan bidang penelitian yang berfokus pada pengembangan sistem dan teknik untuk mendeteksi dan mengklasifikasikan emosi dari sinyal suara manusia (George & Muhamed Ilyas, 2024). SER bertujuan untuk meniru kemampuan manusia dalam memahami keadaan emosional seseorang berdasarkan analisis berbagai aspek suara, seperti *pitch*, intensitas, kecepatan bicara, dan fitur spektrum. Pendekatan konvensional dalam SER umumnya melibatkan beberapa tahapan. Pertama, sinyal suara diambil dan dilakukan ekstraksi fitur dalam dua domain utama, yaitu domain waktu dan domain frekuensi. Di domain waktu, fitur-fitur seperti nilai minimum, maksimum, rata-rata, dan variansi diekstrak. Sementara itu, di domain frekuensi, analisis *spectrogram* dilakukan untuk memahami distribusi frekuensi dari sinyal suara.

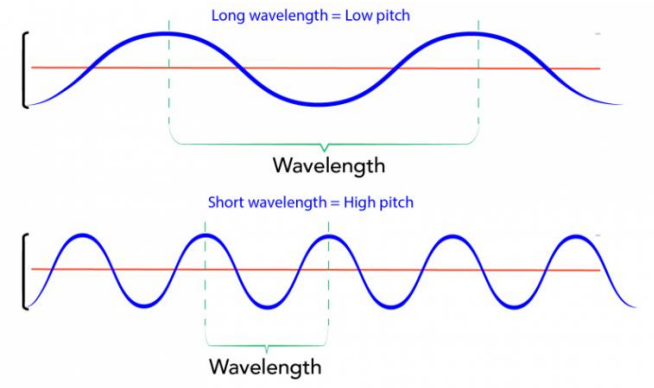
Setelah fitur-fitur ini diekstraksi, proses berikutnya adalah pelatihan model menggunakan algoritma *machine learning* seperti *support vector machine*, *k-nearest*

neighbor, *naïve bayes*, *decision tree*, *gaussian mixture*, dan *neural networks*. Model ini kemudian digunakan untuk mengklasifikasikan emosi menjadi berbagai kategori, seperti marah, bahagia, sedih, netral, dan takut. Proses ini memungkinkan komputer untuk secara otomatis memahami emosi yang terkandung dalam sinyal suara, mendekati kemampuan manusia dalam mengenali emosi dari percakapan sehari-hari.



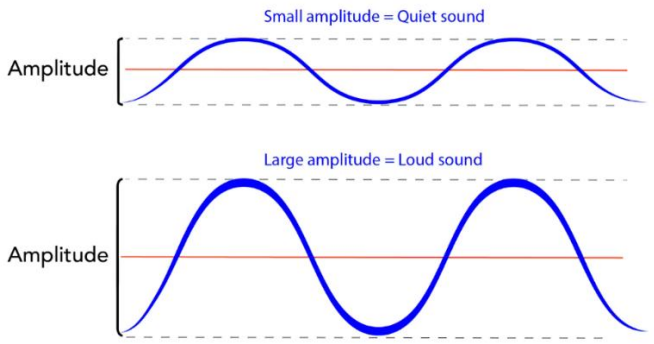
Gambar 5. Visualisasi proses SER menggunakan pendekatan konvensional (Alluhaidan et al., 2023)

Fitur prosodi. Fitur prosodi mengacu pada karakteristik yang terkait dengan pola intonasi, ritme, dan stres dalam bicara yang dapat memberikan informasi penting tentang keadaan emosional pembicara. Salah satu elemen prosodi yang penting adalah kontur *pitch*, yang menggambarkan perubahan *pitch* (frekuensi dasar suara) selama waktu tertentu dalam percakapan. *Pitch* yang lebih tinggi dihasilkan oleh gelombang suara dengan panjang gelombang lebih pendek, sedangkan *pitch* yang lebih rendah dihasilkan oleh gelombang dengan panjang gelombang lebih panjang. Variasi *pitch* sering kali mengindikasikan emosi; misalnya, *pitch* yang tinggi dan dinamis sering diasosiasikan dengan kegembiraan atau kemarahan, sementara *pitch* yang rendah dan datar cenderung mengindikasikan kesedihan atau ketenangan (Pell et al., 2009).



Gambar 6. Visualisasi hubungan panjang gelombang dengan *pitch* dalam sinyal suara (Saini, 2019)

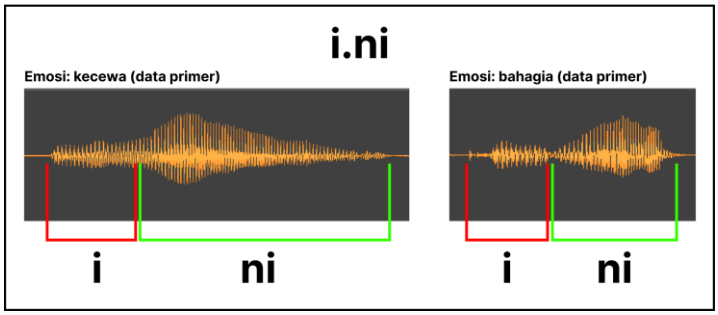
Selain kontur *pitch*, variasi amplitudo adalah fitur prosodi lain yang berperan penting dalam pengenalan emosi suara. Amplitudo mengukur intensitas suara atau seberapa kuat suara tersebut terdengar. Variasi dalam amplitudo dapat memberikan petunjuk mengenai stres atau penekanan dalam bicara. Amplitudo tinggi biasanya menunjukkan emosi kuat seperti kegembiraan atau kemarahan, sedangkan amplitudo rendah menunjukkan emosi seperti ketenangan atau kesedihan. Perubahan mendadak dalam amplitudo dapat menandakan transisi emosional yang intens, seperti kemarahan atau kejutan (K. R. Scherer, 2003).



Gambar 7. Visualisasi variasi amplitudo (Saini, 2019)

Durasi silabel juga merupakan indikator penting dalam pengenalan emosi melalui suara. Durasi silabel mengukur lamanya waktu yang dibutuhkan untuk mengucapkan setiap silabel dalam suatu kalimat atau kata. Durasi silabel yang lebih panjang biasanya mengindikasikan emosi seperti kesedihan atau kebingungan, sedangkan durasi yang lebih pendek sering kali diasosiasikan dengan kegembiraan atau kemarahan (Pell et al.,

2009). Misalnya, dalam pengucapan kata "i.ni" dalam dua konteks emosi yang berbeda—kecewa dan bahagia—karakteristik durasi dan amplitudo dapat berbeda, seperti yang ditunjukkan dalam Gambar 8.



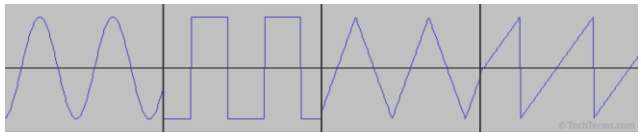
Gambar 8. Visualisasi gelombang suara untuk emosi kecewa dan bahagia

Secara keseluruhan, fitur prosodi seperti kontur *pitch*, variasi amplitudo, dan durasi silabel memberikan wawasan yang mendalam mengenai bagaimana emosi diekspresikan melalui suara. Kombinasi dari ketiga elemen ini membantu dalam membedakan antara emosi yang berbeda, seperti kegembiraan, kesedihan, kemarahan, dan ketenangan. Melalui analisis fitur prosodi, kita dapat memahami bagaimana emosi seseorang diartikulasikan dalam pola intonasi dan perubahan intensitas suara selama bicara.

Representasi visual sinyal audio. Representasi visual sinyal audio adalah teknik yang digunakan untuk mengubah data audio menjadi bentuk visual yang dapat dianalisis secara grafis. Metode ini penting dalam berbagai aplikasi seperti pengenalan suara, pengenalan emosi, analisis musik, dan lainnya. Ada beberapa metode representasi visual sinyal audio, diantaranya *waveform*, *spectrogram, Mel-Frequency Cepstral Coefficients* (MFCCss), *chroma features,* dan *mel spectrogram*.

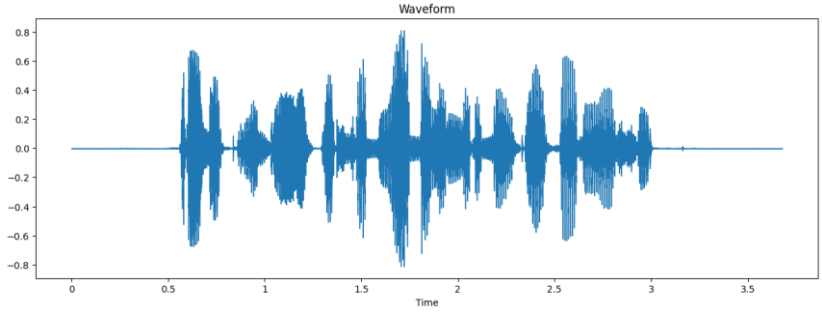
Waveform. Waveform adalah representasi visual dari sinyal audio atau listrik yang menggambarkan amplitudo (atau kekuatan sinyal) seiring waktu. Dalam editor audio seperti DAW (*Digital Audio Workstation*), bentuk gelombang digunakan untuk memberikan gambaran tentang bagaimana sinyal audio berosilasi dan berubah selama rekaman. Selain itu, *waveform* juga sering digunakan dalam pemantauan listrik untuk memvisualisasikan bagaimana tegangan atau arus berperilaku dalam sirkuit. Gelombang sinus, persegi, gigi gergaji, dan segitiga adalah beberapa bentuk gelombang yang umum digunakan dalam berbagai aplikasi, termasuk audio dan elektronik. Masing-masing bentuk gelombang memiliki karakteristik yang unik dalam menghasilkan suara atau sinyal elektronik. Misalnya, gelombang sinus memberikan sinyal yang halus dan murni,

sementara gelombang persegi menghasilkan harmonik yang kuat karena transisi tajam antara dua level.



Gambar 9. Visualisasi bentuk gelombang sinyal audio (Pickle, 2022)

Gelombang audio yang direkam memiliki sifat yang jauh lebih kompleks dibandingkan dengan bentuk gelombang sederhana seperti sinus atau persegi. Hal ini terjadi karena berbagai suara tumpang tindih, membentuk gelombang kompleks yang tidak berosilasi dengan pola yang mulus atau frekuensi yang konsisten. Ketika suara-suara dari berbagai sumber, seperti suara instrumen, vokal, dan latar belakang, direkam bersama, hasilnya adalah campuran frekuensi dan amplitudo yang berubah-ubah. Ini menghasilkan *waveform* yang tidak beraturan dan sangat bervariasi dari waktu ke waktu. Gambar 10 visualisasi dari gelombang suara yang direkam ini memungkinkan kita untuk melihat bagaimana amplitudo berubah seiring waktu, memberikan gambaran tentang intensitas suara di berbagai titik waktu dalam rekaman.

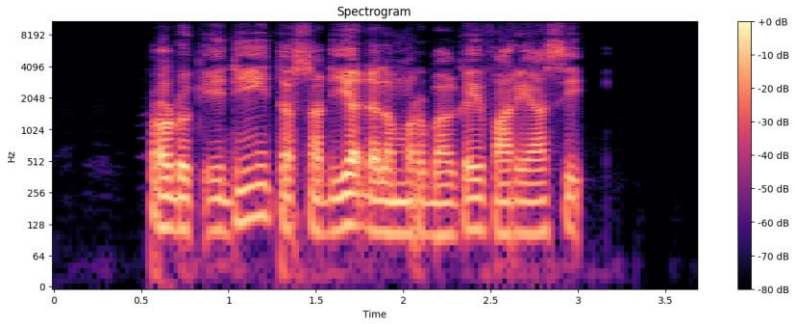


Gambar 10. Visualisasi *waveform* audio yang direkam

Spectrogram. Spectrogram representasi visual yang menunjukkan bagaimana frekuensi dari sinyal audio berubah seiring waktu. Untuk menghasilkan *spectrogram*, sinyal audio dipecah menjadi segmen-segmen kecil menggunakan *Short-Time Fourier Transform* (STFT). Proses ini memungkinkan sinyal yang awalnya berada dalam domain waktu dikonversi ke domain frekuensi, sehingga komponen frekuensi dari sinyal dapat dianalisis. Hasil dari STFT ini diplot pada *spectrogram*, di mana sumbu horizontal merepresentasikan waktu, sumbu vertikal merepresentasikan frekuensi, dan warna menunjukkan amplitudo atau intensitas sinyal pada frekuensi tertentu. Dengan visualisasi Gambar 11, *spectrogram* memberikan gambaran tiga dimensi mengenai sinyal audio

yang menampilkan bagaimana frekuensi dan amplitudo berubah selama periode waktu.

Défossez (2021) menjelaskan bahwa *spectrogram* digunakan untuk memisahkan sumber audio dengan menggabungkan representasi spektrum dan temporal. Selain itu ditunjukkan bahwa dengan menggunakan representasi *hybrid*, model dapat memilih representasi yang paling sesuai untuk bagian tertentu dari sinyal, meningkatkan akurasi dan fleksibilitas dalam analisis audio.

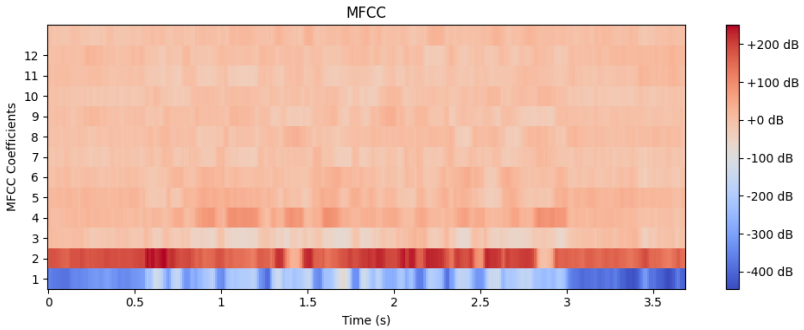


Gambar 11. Visualisasi *spectrogram* audio yang direkam

Spectrogram tidak hanya bermanfaat dalam melihat distribusi frekuensi dalam sinyal audio, tetapi juga dalam memvisualisasikan perubahan amplitudo pada frekuensi tertentu. Dengan menggunakan skala warna, amplitudo sinyal ditunjukkan dengan gradasi warna, dari warna gelap (amplitudo rendah) hingga warna terang (amplitudo tinggi). Proses ini penting dalam berbagai aplikasi, termasuk pengenalan pola suara dan pemisahan sumber audio, di mana frekuensi yang berbeda dapat diisolasi dan dianalisis secara mendalam.

*Mel-frequency cepstral coefficients (*MFCCs*). Mel-Frequency Cepstral Coefficients* (MFCCs) adalah salah satu teknik yang paling umum digunakan dalam pengolahan sinyal audio, terutama untuk pengenalan suara dan analisis emosi. MFCCs berfungsi untuk menangkap karakteristik penting dari sinyal audio dengan memetakan spektrum frekuensi ke skala *mel*, yang lebih sesuai dengan cara manusia mendengar dan memahami suara. Proses ekstraksi MFCCs melibatkan beberapa tahap penting, mulai dari normalisasi, *framing*, *windowing*, *Fast Fourier Transform* (FFT), dan penyaringan menggunakan *mel-filter bank*, hingga akhirnya menghitung energi logaritmik dan menerapkan *Discrete Cosine Transform* (DCT). Hasil akhirnya adalah representasi spektrum frekuensi sinyal audio dalam bentuk koefisien MFCCs, yang kemudian digunakan untuk analisis lebih lanjut. Teknik ini memungkinkan transformasi sinyal audio yang kompleks menjadi data visual yang dapat dianalisis grafis.

Menurut penelitian yang dilakukan oleh Turab et al. (2022) dan Wang & Shen (2023) MFCCs sangat efektif dalam aplikasi pengenalan suara dan emosi, karena koefisien ini merepresentasikan komponen frekuensi suara secara komprehensif, yang dapat digunakan untuk menganalisis spektrum frekuensi audio dengan lebih baik dibandingkan metode lainnya.



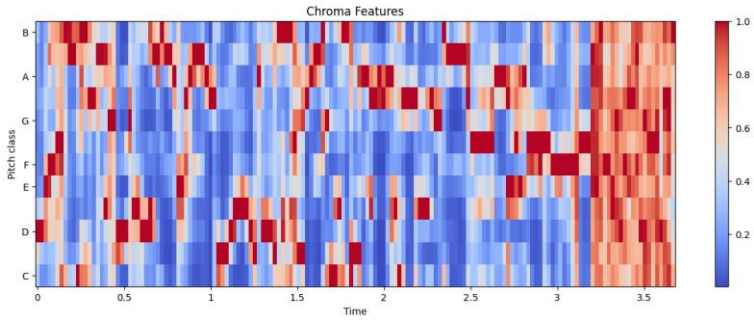
Gambar 12. Visualisasi *mel-frequency cepstral coefficients* (MFCCs)

Koefisien MFCCs yang dihasilkan memetakan energi dari komponen frekuensi pada skala *mel*, yang merefleksikan persepsi suara manusia lebih akurat. Pada sumbu horizontal (X), waktu diplot dalam satuan detik, menggambarkan bagaimana sinyal berubah seiring waktu. Sumbu vertikal (Y) menggambarkan koefisien MFCCs yang merepresentasikan frekuensi yang telah dipetakan ke skala *mel*. Intensitas warna pada grafik menunjukkan amplitudo atau kekuatan dari sinyal audio pada frekuensi tersebut. Warna merah menandakan frekuensi dengan energi tinggi, sedangkan warna biru menunjukkan frekuensi dengan energi rendah, memungkinkan identifikasi cepat dari perubahan signifikan dalam spektrum suara selama rekaman berlangsung.

Chroma features. Chroma features adalah teknik dalam pemrosesan sinyal audio yang digunakan untuk menangkap distribusi energi dalam 12 kelas *pitch* (nada) sepanjang waktu. Setiap kelas *pitch* ini sesuai dengan 12 nada dalam skala musik Barat, yaitu C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Teknik ini sangat berguna dalam aplikasi musik, karena mampu menangkap informasi yang berkaitan dengan harmoni dan *tonalitas* dari sinyal audio. Mempertimbangkan pengulangan nada pada berbagai oktaf, *chroma features* membantu menganalisis struktur harmonis dan melodi dalam musik dengan lebih efisien.

Penelitian yang dilakukan oleh Berenzweig et al. (2004) menunjukkan bahwa *chroma features* sangat efektif dalam pengenalan musik dan sinkronisasi audio, terutama untuk menyelaraskan rekaman audio dengan representasi melodi simbolis seperti *file* MIDI. Menurut (Ellis, 2007b), *chroma features* dirancang untuk mencerminkan konten melodi

dan harmonis, dengan mengurangi pengaruh dari instrumentasi dan tempo. Meskipun *chroma features* mungkin kurang efektif untuk klasifikasi artis dibandingkan dengan MFCCs, fitur ini menyediakan informasi yang berbeda dan independen dari fitur spektrum lainnya. Oleh karena itu, kombinasi antara *chroma features* dan MFCCs mampu meningkatkan akurasi klasifikasi secara signifikan dalam beberapa aplikasi musik.



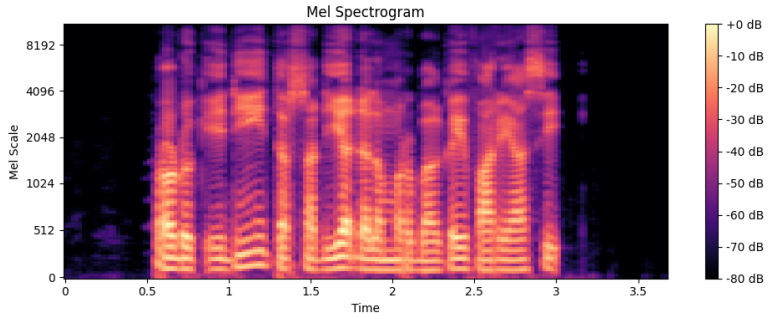
Gambar 13. Visualisasi *chroma features* menunjukkan distribusi energi dalam kelas *pitch* sepanjang waktu

Chroma features memungkinkan analisis yang lebih dalam tentang bagaimana harmoni dan melodi dalam sinyal audio terdistribusi, dengan sumbu horizontal (X) yang mewakili waktu dan sumbu vertikal (Y) yang menampilkan kelas *pitch* dari C hingga B. Intensitas energi dari setiap *pitch* divisualisasikan melalui warna, dengan warna merah menunjukkan *pitch* dengan energi tinggi, dan warna biru menunjukkan energi rendah. Aplikasi ini penting dalam penelitian musikologi dan pengembangan aplikasi musik digital, karena memberikan representasi yang lebih komprehensif dari aspek-aspek *tonal* dan harmonis.

Mel spectrogram. *Mel spectrogram* adalah teknik yang umum digunakan dalam pengolahan sinyal audio, yang memetakan spektrum frekuensi ke skala *mel* untuk memberikan representasi yang lebih sesuai dengan persepsi manusia terhadap suara. *Mel spectrogram* menangkap karakteristik penting dari sinyal audio melalui serangkaian proses. Langkah pertama adalah normalisasi sinyal audio, yang dilakukan untuk memastikan konsistensi dalam amplitudo dan mengurangi gangguan. Setelah itu, sinyal audio dibagi menjadi beberapa *frame* pendek untuk menangkap perubahan yang terjadi seiring waktu. Setiap *frame* kemudian diterapkan fungsi *windowing* (seperti *hamming* atau *hanning*) untuk mengurangi efek kebocoran *spektrum*, sebelum diubah dari domain waktu ke domain frekuensi menggunakan *Fast Fourier Transform* (FFT).

Frekuensi yang diperoleh dari FFT kemudian dipetakan ke skala *mel* menggunakan *mel filterbank*, yang terdiri dari filter berbentuk segitiga

yang didistribusikan secara logaritmik di sepanjang sumbu frekuensi. Energi dari setiap filter *mel* dihitung dan diubah ke skala logaritmik. Hasil akhirnya adalah *mel spectrogram*, yang memberikan representasi visual komprehensif dari spektrum frekuensi sinyal audio (Lambamo et al., 2023;(Shen et al., 2017).



Gambar 14. Visualisasi *mel spectrogram*, menunjukkan distribusi energi frekuensi dalam sinyal audio dipetakan ke skala *mel*

Pada *mel spectrogram*, sumbu horizontal (X) menunjukkan waktu dalam detik, menggambarkan bagaimana sinyal audio berubah seiring waktu. Sumbu vertikal (Y) menunjukkan frekuensi dalam Hertz (Hz), yang telah dipetakan ke skala *mel*, dengan rentang dari 0 Hz hingga sekitar 8 kHz. Intensitas atau amplitudo dari frekuensi ini divisualisasikan menggunakan skala warna. Warna kuning menunjukkan frekuensi dengan energi tinggi, sedangkan warna biru menunjukkan frekuensi dengan energi rendah. Dengan visualisasi ini, kita dapat melihat distribusi energi dalam frekuensi yang berbeda sepanjang durasi sinyal audio, yang sangat berguna dalam berbagai aplikasi audio seperti pengenalan suara dan analisis emosi.

***Amplitude factor* (AF).** *Amplitude Factor* (AF) merupakan sebuah nilai yang digunakan untuk menyesuaikan atau mengubah tingkat amplitudo suara atau sinyal. AF dihitung sebagai perbandingan antara dua amplitudo (amplitudo sinyal yang ingin diukur dibandingkan dengan amplitudo sinyal referensi) yang ditunjukkan pada persamaan (1) (Ponton, 2007).

$$AF = \frac{Amplitudo\ yang\ ingin\ diukur}{Amplitudo\ Referensi} \quad (1)$$

dimana, AF adalah *Amplitude Factor* (AF).

AF juga dapat digunakan sebagai bagian dari rasio amplitudo, yaitu pengukuran relatif antara dua amplitudo suara atau sinyal, yang biasanya

dinyatakan dalam desibel (dB). Rasio amplitudo ini memberikan informasi tentang perbedaan relatif dalam kekuatan atau level suara antara dua sinyal atau satu sinyal terhadap tingkat referensi tertentu. Persamaan untuk menghitung penguatan dalam dB ditunjukkan oleh persamaan (2).

$$G_{dB} = 20 \times log_{10}(Amplitude\ Factor) \quad (2)$$

dimana, G_{dB} adalah rasio amplitudo atau penguatan dalam desibel (dB).

AF digunakan dalam berbagai aplikasi pemrosesan sinyal untuk mengukur dan menyesuaikan amplitudo sinyal audio. Penggunaannya sangat penting dalam pengendalian kualitas suara, pengenalan suara, dan aplikasi komunikasi. Dalam konteks *Speech Emotion Recognition* (SER), AF membantu mengevaluasi ketahanan sistem terhadap *noise* dan memastikan keandalan model di berbagai kondisi sinyal.

Amplitude Factor (AF)	Decibels (dB)
0.0	-∞
0.1	-20.0
0.2	-13.98
0.3	-10.46
0.4	-7.96
0.5	-6.02
0.6	-4.44
0.7	-3.10
0.8	-1.94
0.9	-0.92
1.0	0.0

Gambar 15. Tabel konversi *amplitude factor* (AF) ke desibel (dB), menunjukkan hubungan antara AF dengan level desibel yang dihasilkan

Tabel di Gambar 15 menunjukkan konversi *Amplitude Factor* (AF) ke desibel (dB), di mana AF 1.0 setara dengan 0 dB (amplitudo referensi), dan nilai AF yang lebih kecil menghasilkan dB negatif, menunjukkan penurunan level energi. Sebagai contoh, AF 0.1 setara dengan -20 dB, yang menunjukkan bahwa sinyal mengalami penurunan energi yang signifikan dibandingkan dengan referensi.

***Signal-to-noise rasio* (SNR).** *Signal-to-Noise Ratio* (SNR) adalah ukuran yang digunakan untuk membandingkan tingkat sinyal yang diinginkan dengan tingkat *noise* latar belakang. SNR sering digunakan dalam berbagai bidang, seperti komunikasi data, rekayasa audio, radar, dan pemrosesan gambar, untuk mengevaluasi kualitas sinyal dan kinerja

sistem yang dinyatakan dalam desibel (dB). Persamaan SNR ditunjukkan pada persamaan (3) (George & Muhamed Ilyas, 2024; Xu et al., 2021).

$$SNR_{dB} = 20 \times log_{10}\left(\frac{A_{signal}}{A_{noise}}\right) \quad (3)$$

dimana A_{signal} merupakan Amplitudo *signal* dan A_{noise} adalah Amplitudo *noise*

SNR merupakan metrik penting untuk mengukur seberapa baik sinyal telah tercemar oleh kebisingan. SNR yang lebih tinggi menunjukkan sinyal yang lebih bersih dan lebih dominan dibandingkan dengan kebisingan, yang berarti lebih banyak informasi yang berguna tersedia dibandingkan dengan data yang tidak diinginkan (*noise*) (T. Zhou, 2013). Misalnya, dalam rekayasa audio, SNR yang tinggi menghasilkan kualitas suara yang lebih jelas dan bebas dari gangguan, yang meningkatkan pengalaman mendengarkan (Gorbunov, 2020, Suarez-Perez et al., 2018).

1.2.4 *Preprocessing*

Preprocessing adalah langkah penting dalam analisis sinyal audio untuk memastikan kualitas dan konsistensi data sebelum digunakan dalam model pembelajaran mesin atau aplikasi lainnya. Tahapan *preprocessing* meliputi *sampling rate conversion*, *normalization*, dan *noise reduction*, yang semuanya berperan dalam mempersiapkan data audio agar siap untuk diekstraksi fitur dan dianalisis lebih lanjut.

***Sampling rate*.** *Sampling rate* adalah jumlah sampel yang diambil per detik dari sinyal audio kontinu untuk membuat sinyal diskrit, yang diukur dalam Hertz (Hz). *Sampling rate* yang umum digunakan dalam audio adalah 44.1 kHz atau 48 kHz. Pemilihan *sampling rate* yang tepat sangat penting untuk mempertahankan kualitas sinyal audio sambil mengurangi ukuran data (John, 2024).

Proses *sampling rate* melibatkan transformasi sinyal ke domain frekuensi menggunakan *Fast Fourier Transform* (FFT), melakukan interpolasi pada spektrum frekuensi untuk menyesuaikan dengan *sampling rate* baru, dan kemudian menggunakan *Inverse* FFT (IFFT) untuk mengembalikan sinyal ke domain waktu dengan *sampling rate* baru. Persamaan untuk *sampling rate* adalah sebagai berikut:

$$y[n] = x(nT_s) \quad (4)$$

di mana $y[n]$ adalah sinyal diskrit, $x(nT_s)$ adalah sinyal kontinu, T_s merupakan interval sampel (1/*sampling rate*).

***Normalization*.** Proses mengubah skala amplitudo sinyal audio sehingga berada dalam rentang tertentu, biasanya antara -1 dan 1. Ini dilakukan untuk memastikan konsistensi dalam data audio dan untuk menghindari distorsi selama pemrosesan lebih lanjut (Farhan, 2024). *Normalization* juga membantu dalam menstandarkan level volume antara *file* audio yang berbeda.

Proses *normalization* melibatkan pencarian nilai amplitudo maksimum dari sinyal audio dan kemudian membuat skala seluruh sinyal audio dengan faktor yang membuat amplitudo maksimum menjadi 1 atau 0 dB. Persamaan untuk *normalization* adalah sebagai berikut:

$$y[n] = \frac{x[n]}{\max\left(|x[n]|\right.} \tag{5}$$

di mana $y[n]$ adalah sinyal dinormalisasi, $x[n]$ adalah sinyal asli, dan $max(|x[n]|)$ adalah nilai absolut maksimum dari sinyal asli.

***Noise reduction*.** *Noise reduction* proses menghilangkan atau mengurangi *noise* dari sinyal audio untuk meningkatkan kualitas sinyal. Teknik ini penting untuk memastikan bahwa fitur-fitur yang diekstraksi dari sinyal audio murni berasal dari sumber suara yang diinginkan dan bukan dari *noise*.

Proses *noise reduction* melibatkan transformasi sinyal audio ke domain frekuensi menggunakan FFT, membuat profil *noise* dari segmen audio yang hanya berisi *noise*, mengurangi profil *noise* dari spektrum audio, dan menggunakan *inverse* FFT untuk mengembalikan sinyal ke domain waktu tanpa *noise*. Persamaan untuk *noise reduction* dengan metode spektrum adalah sebagai berikut:

$$Y(f) = X(f) - N(f) \tag{6}$$

di mana $Y(f)$ adalah spektrum sinyal yang telah direduksi *noise*, $X(f)$ adalah spektrum sinyal asli, dan $N(f)$ adalah spektrum *noise*.

Noise injection. *Noise injection* merupakan proses penambahan *noise* yang direkam dari lingkungan nyata ke sinyal suara asli (Xu et al., 2021; Xu, Zhang, & Khan, 2020; Q. Zhou et al., 2021). Rumus *noise injection* ditunjukkan pada persamaan berikut.

$$y(t) = x(t) + \beta \cdot n_{env}(t) \tag{7}$$

di mana $y(t)$ adalah sinyal suara yang diberi *noise*, $x(t)$ adalah sinyal suara asli. $n_{env}(t)$ merupakan *noise* lingkungan, dan β adalah amplitudo *noise*.

***Data augmentation*.** *Data augmentation* adalah teknik yang digunakan untuk meningkatkan keragaman data pelatihan dengan menambahkan variasi pada data yang ada (George & Muhamed Ilyas, 2024). Teknik ini penting dalam pembelajaran mesin karena membantu mengatasi masalah *overfitting*, terutama ketika *dataset* asli berukuran kecil atau tidak representatif dari berbagai kondisi dunia nyata. Dalam konteks *noisy speech emotion recognition*, data *augmentation* digunakan untuk membuat model lebih *robust* terhadap berbagai jenis *noise* dan perubahan dalam sinyal suara (Rayhan Ahmed et al., 2023). Teknik data *augmentation* yang digunakan dalam penelitian ini yaitu, *white noise injection*, *shift*, *pitch*, dan *stretch* dengan detail sebagai berikut.

1. *Noise*

   Noise adalah teknik menambahkan *noise* secara sengaja ke dalam data audio selama pelatihan model. Teknik ini bertujuan untuk meningkatkan ketahanan model terhadap berbagai jenis *noise* yang bersumber dari data *noise* yang telah dikumpulkan. Teknik *noise* yang digunakan pada penelitian ini adalah *white noise*. *White noise* merupakan proses panambahan *noise* acak dengan distribusi spektrum yang rata ke sinyal suara (Wei et al., 2020). Rumus *white noise* ditunjukkan pada Persamaan (8).

   $$y(t) = x(t) + \alpha \cdot n(t) \tag{8}$$

   di mana $y(t)$ adalah sinyal suara yang diberi *noise*, $x(t)$ adalah sinyal suara asli, $n(t)$ merupakan *white noise*, dan α adalah amplitudo *noise*.

2. *Shift*

   Shift adalah teknik *augmentation* di mana sinyal suara digeser dalam waktu untuk menyimulasikan variasi dalam waktu mulai sinyal. Ini dilakukan dengan memindahkan semua sampel audio dengan sejumlah waktu tertentu. *Shift* membuat model untuk menjadi lebih *robust* terhadap variasi posisi temporal dalam sinyal audio dan meningkatkan generalisasi model dengan memaksa model untuk mengenali fitur audio tanpa tergantung pada waktu tertentu. Proses *shift* ditunjukkan pada Persamaan (9).

   $$y(t) = x(t + \delta t) \tag{9}$$

   di mana $y(t)$ adalah sinyal suara yang telah digeser, $x(t)$ adalah sinyal suara asli, δt adalah jumlah waktu pergeseran.

3. *Pitch*

   Pitch adalah teknik *augmentation* di mana *pitch* (frekuensi dasar) dari sinyal audio diubah tanpa mengubah durasi sinyal. Ini dilakukan dengan menggeser frekuensi semua komponen harmonik dalam sinyal. *Pitch*

membantu model untuk menjadi lebih *robust* terhadap variasi *pitch* dalam data audio. Proses *pitch* ditunjukkan pada Persamaan (10).

$$y(t) = x(t) \cdot 2^{\frac{\Delta f}{12}} \tag{10}$$

di mana $y(t)$ adalah sinyal suara yang telah digeser, $x(t)$ adalah sinyal suara asli, Δf adalah perubahan *pitch* dalam *semitone*.

4. *Stretch*
   Stretch adalah teknik *augmentation* di mana durasi sinyal audio diubah tanpa mengubah *pitch*. Ini dilakukan dengan mengubah kecepatan pemutaran sinyal audio. *Stretch* membantu model untuk menjadi lebih *robust* terhadap variasi kecepatan bicara dan meningkatkan kemampuan model untuk mengenali emosi dalam berbagai kecepatan ucapan yang berbeda. Proses *stretch* ditunjukkan pada Persamaan (11).

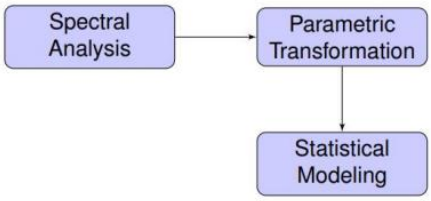
$$y(t) = x(\alpha t) \tag{11}$$

di mana $y(t)$ adalah sinyal suara yang telah digeser, $x(t)$ adalah sinyal suara asli, α adalah faktor peregangan yang menentukan seberapa besar perubahan durasi.

1.2.5 Feature Extraction

Ekstraksi fitur (*feature extraction)* adalah proses mengidentifikasi dan menyaring karakteristik penting dari data mentah yang relevan untuk analisis lebih lanjut. Dalam konteks pengolahan sinyal audio, ekstraksi fitur bertujuan untuk mengubah data audio mentah menjadi representasi yang lebih bermakna dan kompak, yang memudahkan proses analisis dan pengenalan oleh algoritma pembelajaran mesin. Sehingga sederhananya, ekstraksi fitur merupakan proses untuk menonjolkan karakteristik paling dominan dan mendiskriminasikan dari sebuah sinyal (Sharma et al., 2020).

Operasi dasar ekstraksi fitur melibatkan analisis spektrum, transformasi para metrik, dan pemodelan statistik. Pada tahap analisis spektrum, sinyal dianalisis dalam domain frekuensi untuk mengidentifikasi komponen-komponen frekuensi yang signifikan. Selanjutnya, di tahap transformasi parametrik, sinyal diubah menjadi parameter digital yang lebih mudah diolah melalui operasi seperti diferensiasi dan deret rangkaian. Terakhir, pemodelan statistik menggunakan parameter sinyal untuk membentuk representasi yang lebih akurat dari karakteristik sinyal suara (Mazumder & Salam, 2019). Meskipun ekstraksi fitur membantu menyaring informasi yang tidak diperlukan, ada risiko kehilangan informasi penting selama proses penyaringan (More, 2016).



Gambar 16. Operasi dasar dari proses *feature extraction*, mencakup analisis spektrum, transformasi parametrik, dan pemodelan statistik (Mazumder & Salam, 2019)

Ekstraksi fitur-fitur penting dari sinyal audio ucapan merupakan salah satu langkah terpenting dalam kegiatan yang berkaitan dengan SER. Ekstraksi yang tepat dari fitur-fitur krusial dapat meningkatkan kinerja model dalam hal akurasi SER. Secara tradisional, diamati bahwa fitur-fitur buatan tingkat rendah mengandung petunjuk emosional yang signifikan tentang ucapan dan, dengan rekayasa fitur yang tepat, bekerja dengan baik dengan arsitektur CNN 1D (Zhang et al., 2018; Zhao et al., 2019b). Berbagai penelitian sebelumnya telah menggunakan fitur-fitur seperti *Zero Crossing Rate* (ZCR), *Mel-Frequency Cepstral Coefficients* (MFCCs), *chromagram, root-mean square*, dan *mel spectrogram* untuk *speech emotion recognition* . Misalnya, penelitian oleh Rayhan Ahmed et al. (2023) menunjukkan bahwa MFCCs, ZCR, *chroma*, RMS, secara efektif dapat menangkap karakteristik spektrum suara yang relevan dengan emosi.

Pemilihan fitur dalam penelitian ini didasarkan pada keberhasilan penelitian sebelumnya dan relevansi fitur tersebut dalam menangkap aspek emosional dari sinyal suara. Selain itu, fitur-fitur tersebut juga efisien secara komputasi, memberikan informasi dasar yang relevan, *robust* terhadap *noise* (George & Muhamed Ilyas, 2024) dan variasi, mudah diinterpretasikan, telah terbukti efektif dalam penelitian sebelumnya, dan dapat bekerja sangat baik dengan model pembelajaran mesin yang lebih kompleks. Sehingga penelitian ini akan menggunakan lima fitur spektrum yang berbeda yaitu ZCR, MFCCs, RMS, *Chroma* STFT, dan *mel spectrogram*. Perincian dari fitur yang diekstraksi diberikan di bawah ini.

***Zero crossing rate* (ZCR).** *Zero-Crossing Rate* (ZCR) adalah metrik yang mengukur frekuensi perubahan tanda pada sinyal audio dari positif ke negatif atau sebaliknya dalam sebuah *frame* tertentu (O'Shaughnessy, 2023). Secara matematis, ZCR dihitung dengan membagi jumlah perubahan tanda sinyal dengan panjang *frame* tersebut (Sharma et al., 2020). Dengan kata lain, ZCR menunjukkan berapa kali sinyal melintasi titik nol dalam satu interval detik (Torres-García et al., 2022). ZCR untuk *frame* ke-i dengan panjang N didefinisikan oleh persamaan (12) dengan fungsi *sgn* pada persamaan (13) (Giannakopoulos & Pikrakis, 2014).

$$Z(i) = \frac{1}{2N} \sum_{n=1}^{N} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (12)$$

$$sgn[x_i(n)] = \begin{cases} 1, x_i(n) > 0 \\ 0, x_i(n) = 0 \\ -1, x_i(n) < 0 \end{cases} \quad (13)$$

dimana, $Z(i)$ merupakan *Zero-Crossing Rate* untuk *frame* ke-i, N adalah jumlah sampel dalam *frame*, $x_i(n)$ adalah nilai sinyal audio pada sampel ke-n dalam *frame* ke-i, dan sgn adalah fungsi tanda (*sign function*), yang mengembalikan jika nilai positif, -1 jika nilai negatif, dan 0 jika sama dengan nol.

***Chroma* STFT.** *Chroma Short-Time Fourier Transform* (*Chroma* STFT) adalah teknik ekstraksi fitur dalam pemrosesan sinyal audio yang digunakan untuk menangkap distribusi energi dari sinyal audio dalam 12 kelas *pitch* (nada) sepanjang waktu (Das et al., 2020). Setiap kelas *pitch* dalam *Chroma* STFT sesuai dengan 12 nada dalam skala musik Barat (C, C#, D, D#, E, F, F#, G, G#, A, A#, B). Teknik ini sangat berguna dalam berbagai aplikasi, termasuk pengenalan emosi suara (*Speech Emotion Recognition*, SER), karena mampu menangkap informasi harmonis dan tonal dari sinyal audio (Ellis & Poliner, 2007; Muller, 2021).

Proses dari *Chroma* STFT dimulai dari *preprocessing*, yang melibatkan segmentasi sinyal audio menjadi *frame-frame* pendek, biasanya 20-40 milidetik. Tujuannya adalah untuk memastikan bahwa setiap *frame* menangkap informasi frekuensi yang cukup kecil untuk mendeteksi perubahan cepat dalam sinyal audio, namun cukup besar untuk mendapatkan resolusi frekuensi yang baik (Ellis, 2007a). Setelah sinyal audio dibagi menjadi *frame-frame*, setiap *frame* kemudian mengalami *Short-Time Fourier Transform* (STFT). STFT mengubah sinyal dari domain waktu ke domain frekuensi. Persamaan untuk mendapatkan STFT ditunjukkan pada persamaan (146) (Leiber et al., 2023):

$$X(t, \omega) = \sum_{n=0}^{N-1} x[n] \cdot w[n-t] \cdot e^{-j\omega n} \quad (14)$$

di mana $X(t, \omega)$ adalah hasil STFT pada waktu t dan frekuensi ω, $x[n]$ adalah sinyal audio, dan $\mathrm{w[n-t]}$ adalah fungsi *window*. Fungsi *window* ω digunakan untuk memastikan bahwa hanya sebagian kecil dari sinyal yang dianalisis pada setiap waktu, yang memungkinkan deteksi perubahan frekuensi dalam waktu yang singkat, e adalah basis dari logaritma natural, dan j adalah unit imajiner yang memenuhi persamaan $j^2 = -1$.

Langkah berikutnya adalah pemetaan frekuensi ke dalam *bin chroma*. Setiap bin *chroma* mewakili satu dari 12 *semitone* dalam satu oktaf musik. Untuk melakukan ini, bin frekuensi dari hasil STFT dipetakan ke 12 bin *chroma* berdasarkan kelas *pitch* mereka. Proses ini melibatkan penjumlahan magnitudo dari semua frekuensi yang sesuai dengan *chroma* tertentu yang ditunjukkan pada persamaan (7) (Shah et al., 2019):

$$C_k(t) = \sum_{\omega \epsilon B_k} |X(t, \omega)| \tag{15}$$

di mana $X(t, \omega)$ merupakan hasil STFT pada waktu t dan frekuensi ω, $C_k(t)$ adalah energi *chroma* untuk bin *chroma* ke-k pada waktu t. B_k set bin frekuensi yang sesuai dengan bin *chroma* ke-k. Dengan cara ini, harmonik dari nada dasar yang sama akan dikumpulkan dalam bin yang sama, menghilangkan informasi tentang oktaf tetapi mempertahankan informasi tentang kelas *pitch*.

Setelah pemetaan *chroma*, hasilnya dapat mengalami *post-processing* untuk meningkatkan ketahanan dan keandalan fitur *chroma*. Salah satu teknik yang umum digunakan adalah normalisasi, yang menyamakan skala magnitudo dari semua bin *chroma* sehingga mereka dapat dibandingkan dengan lebih mudah (Muller, 2021).

***Root mean square* (RMS).** *Root Mean Square* (RMS) adalah nilai statistik yang digunakan untuk mengukur magnitudo rata-rata dari sinyal (Rayhan Ahmed et al., 2023). Dalam konteks analisis audio, RMS memberikan indikasi tentang energi atau amplitudo rata-rata dari sinyal audio dalam setiap *frame* waktu (Mashhadi & Osei-Bonsu, 2023). RMS sering digunakan untuk menggambarkan kekuatan sinyal audio, yang relevan dalam berbagai aplikasi seperti pengenalan musik, deteksi emosi, dan segmentasi audio. Rumus *root mean square* ditunjukkan pada persamaan (16) (McFee et al., 2015).

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^{N} x[n]^2} \tag{16}$$

di mana $x[n]$ merupakan nilai sampel ke- n dalam *frame*, dan N adalah jumlah total sampel dalam *frame* tersebut.

***Mel spectrogram*.** *Mel spectrogram* adalah representasi spektrum dari sinyal audio yang diubah ke dalam skala *mel*, yang mencerminkan persepsi frekuensi oleh pendengaran manusia (Q. Zhou et al., 2021). Skala *mel* memberikan resolusi yang lebih tinggi untuk frekuensi rendah dibandingkan

frekuensi tinggi, sesuai dengan karakteristik pendengaran manusia yang lebih sensitif terhadap perubahan frekuensi rendah. Proses pembuatan *mel spectrogram* (McFee et al., 2015) adalah sebagai berikut:

1. *Short-Time Fourier Transform* (STFT)
   Langkah pertama adalah mengubah sinyal audio dari domain waktu ke domain frekuensi menggunakan STFT. Persamaan STFT ditunjukkan pada persamaan (17) (McFee et al., 2015).

$$X(t,\omega) = \sum_{n=0}^{N-1} x[n] \cdot w[n-t] \cdot e^{-j\omega n} \tag{17}$$

   di mana $X(t,\omega)$ adalah hasil STFT pada waktu t dan frekuensi ω, $x[n]$ adalah sinyal audio, dan $\mathrm{w[n-t]}$ adalah fungsi *window*. Fungsi *window* ω digunakan untuk memastikan bahwa hanya sebagian kecil dari sinyal yang dianalisis pada setiap waktu, yang memungkinkan deteksi perubahan frekuensi dalam waktu yang singkat, e adalah basis dari logaritma natural, dan j adalah unit imajiner yang memenuhi persamaan $j^2 = -1$.

2. *Power Spectrum*
   Setelah mendapatkan STFT, langkah berikutnya adalah menghitung spektrum kekuatan dengan menguadratkan magnitudo dari STFT. Persamaan STFT ditunjukkan pada persamaan (18) (McFee et al., 2015).

$$P(t,\omega) = |X(t,\omega)|^2 \tag{18}$$

   di mana $P(t,\omega)$ adalah spektrum kekuatan pada waktu t dan frekuensi ω, $|X(t,\omega)$ adalah magnitudo dari STFT.

3. *Mel filterbank*
   Untuk memetakan spektrum kekuatan ke skala *mel*, digunakan *Mel filterbank*. *Mel filterbank* terdiri dari serangkaian filter berbentuk segitiga yang masing-masing mencakup rentang frekuensi tertentu. Persamaan *Mel filterbank* ditunjukkan pada persamaan (19) sedangkan skala *mel* ditunjukkan pada persamaan (20) (McFee et al., 2015).

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \tag{19}$$

   Mel filterbank diterapkan pada spektrum daya:

$$M(t,m) = \sum_{\omega} P(t,\omega) \cdot H_m(\omega) \tag{20}$$

di mana $M(t,m)$ adalah *mel spectrogram* pada waktu t dan indeks *mel* m, $H_m(\omega)$ adalah respons filter *mel* ke-m pada frekuensi ω.

4. Transformasi Logaritmik
   Mengonversi *mel spectrogram* ke skala logaritmik untuk mendapatkan representasi *log-mel spectrogram*. Persamaan STFT ditunjukkan pada persamaan (21) (McFee et al., 2015).

$$Log - Mel(t,m) = \log\left(M(t,m) + \epsilon\right) \tag{21}$$

   di mana $M(t,m)$ adalah *mel spectrogram* pada waktu t dan indeks *mel* m, ϵ adalah konstanta kecil untuk menghindari logaritma dari nol.

***Mel-frequency cepstral coefficients* (MFCCs).** *Mel-Frequency Cepstral Coefficient (*MFCCs*)* adalah metode ekstraksi fitur yang digunakan untuk pemrosesan sinyal suara yang bertujuan untuk menangkap karakteristik spektrum dari sinyal audio yang mendekati persepsi pendengaran manusia. Menurut Rayhan Ahmed et al. (2023), suara manusia disaring melalui bentuk saluran vokal yang melibatkan elemen seperti lidah dan gigi, yang berbeda pada setiap individu. Struktur dari elemen-elemen ini menentukan karakteristik unik dari suara seseorang. Pengukuran yang akurat dari bentuk ini mencerminkan fonem yang dihasilkan. Bentuk ini terwujud dalam amplop spektrum daya jangka pendek, yang diwakili oleh MFCCs. Fitur ini sering digunakan dalam penelitian *speech emotion recognition* karena kemampuannya untuk merepresentasikan perbedaan fonem yang dihasilkan oleh bentuk saluran vokal yang berbeda (Abdel-Hamid, 2020; Nantasri et al., 2020).

MFCCs tidak hanya merepresentasikan energi dalam frekuensi yang relevan dengan pendengaran manusia tetapi juga mencakup informasi penting yang berkaitan dengan intonasi, kejelasan, dan emosi dalam ucapan, menjadikannya alat yang sangat berharga dalam berbagai aplikasi pengenalan dan analisis suara. Proses ekstraksi fitur adalah sebagai berikut.

1. *Pre-emphasis.* Proses ini melibatkan penerapan filter *high-pass* pada sinyal audio untuk memperkuat komponen frekuensi tinggi. *Pre-emphasis* bertujuan untuk mengurangi dampak penurunan energi frekuensi tinggi yang terjadi selama perekaman. Persamaan *pre-emphasis* ditunjukkan pada persamaan (22) (McFee et al., 2015):

$$y[t] = x[t] - \alpha \cdot x[t-1] \text{ untuk } 0.9 < \alpha < 1 \tag{22}$$

   dengan $y[t]$ adalah sinyal setelah *Pre-emphasis*, $x[t]$ adalah sinyal asli, nilai α kurang lebih 0.97 yang merupakan koefisien *Pre-emphasis*.

2. *Framing*. Sinyal audio dibagi menjadi beberapa *frame* pendek, biasanya dengan durasi 20-30 milidetik. *Framing* diperlukan karena sinyal audio dianggap stasioner dalam interval waktu yang pendek. Tumpang tindih antara *frame* biasanya sekitar 50% untuk memastikan kontinuitas informasi. Persamaan *Framing* ditunjukkan pada persamaan (23) (McFee et al., 2015). Jika sinyal memiliki panjang N dan *frame* memiliki panjang M, maka *frame* ke-k dapat didefinisikan sebagai:

$$x_k[m] = x[k \cdot H + m] \tag{23}$$

di mana $x_k[m]$ adalah *frame* ke-k, k adalah indeks *frame*, H adalah *hop size* (jumlah sampel antara awal *frame* satu dengan *frame* berikutnya).

3. *Windowing*. Setiap *frame* diaplikasikan fungsi jendela (misalnya, jendela *hamming*) untuk mengurangi efek tepi yang dapat menyebabkan distorsi spektrum. Fungsi jendela membantu dalam hal penghalusan transisi antara *frame*. Persamaan *windowing* ditunjukkan pada persamaan (24) (McFee et al., 2015):

$$x_w[m] = x[m] \cdot w[m] \tag{24}$$

dimana $x_w[m]$ adalah *frame* yang telah melalui proses *windowing*, $w[m]$ adalah fungsi *window*, seperti *hanning* atau *hamming*.

4. *Discrete Fourier Transform (DFT)*. DFT digunakan untuk mengubah sinyal dari domain waktu ke domain frekuensi, yang memberikan representasi spektrum frekuensi dari sinyal. Untuk efisiensi komputasi, DFT biasanya diimplementasikan dengan algoritma *Fast Fourier Transform* (FFT). Persamaan *Discrete Fourier Transform* (DFT) ditunjukkan pada persamaan (25) (McFee et al., 2015).

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \ e^{-j\frac{2\pi}{N}kn} \tag{25}$$

dimana N adalah jumlah sampel, $x[n]$ adalah sinyal dalam domain waktu, dan $X[k]$ adalah sinyal dalam domain frekuensi.

5. *Mel Filterbank* . Spektrum frekuensi hasil DFT diaplikasikan pada *Mel filterbank*. Skala *mel* adalah skala frekuensi non-linear yang lebih sesuai dengan cara telinga manusia mendengar suara. Filter *mel* biasanya berupa filter trapesium yang diposisikan pada skala *mel*. Persamaan *mel filterbank* ditunjukkan pada persamaan (26) sedangkan skala *mel* ditunjukkan pada persamaan (27) (McFee et al., 2015).

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \quad (26)$$

dimana f adalah frekuensi dalam Hz.

mel filterbank diterapkan pada spektrum daya:

$$M(t,m) = \sum_{\omega} P(t,\omega) \cdot H_m(\omega) \quad (27)$$

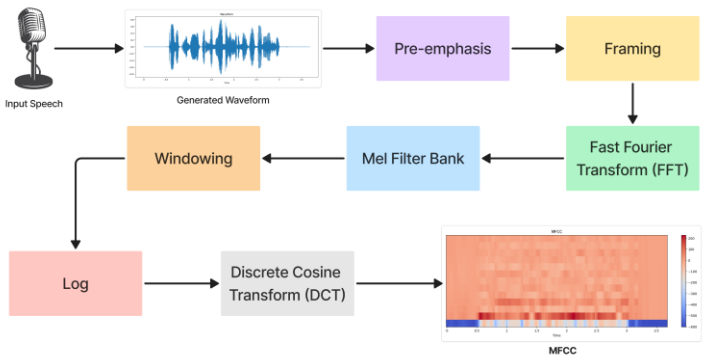
di mana $M(t,m)$ adalah *mel spectrogram* pada waktu t dan indeks *mel* m, $H_m(\omega)$ adalah respons filter *mel* ke-m pada frekuensi ω.

6. *Log* energi. Energi dari setiap filter *mel* diambil logaritmanya untuk menghasilkan log energi spektrum *mel*. Penggunaan logaritma ini untuk mencerminkan sifat persepsi manusia yang logaritmik terhadap intensitas suara.

7. *Discrete Cosine Transform (DCT).* DCT diterapkan pada log energi spektrum *mel* untuk menghasilkan koefisien MFCCs. DCT membantu dalam pemadatan informasi dengan memfokuskan energi ke beberapa koefisien pertama. Persamaan DCT ditunjukkan pada persamaan (28).

$$C[k] = \sum_{n=0}^{N-1} \log(S[n]) \cos\left(\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right) \quad (28)$$

dimana $C[k]$ adalah koefisien DCT ke-k yang merupakan koefisien MFCCs yang dihasilkan oleh log energi spektrum *mel*, $S[n]$ adalah energi spektrum *mel* pada indeks ke- n dan fungsi kosinus digunakan dalam transformasi dan melakukan pemetaan dari domain waktu ke domain frekuensi.



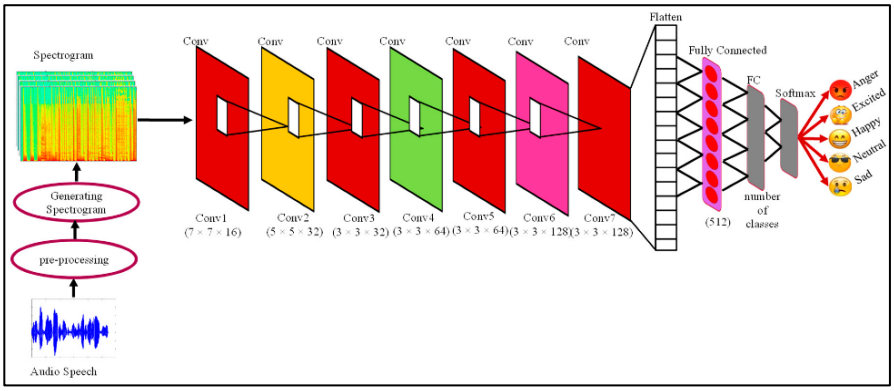
Gambar 17. Gambar ilustrasi dari proses ekstraksi fitur MFCCs

1.2.6 *Convolutional neural network* (CNN)

*Convolutional Neural Network*s (CNN) telah digunakan secara luas dalam pengenalan emosi dari suara, di mana model ini efektif dalam mengekstraksi fitur-fitur emosional yang signifikan untuk klasifikasi emosi suara. Beberapa penelitian, termasuk karya oleh Ahmed et al. (2023), telah menunjukkan bagaimana CNN dapat secara akurat mengklasifikasikan emosi dalam suara melalui ekstraksi dan analisis fitur spektrum dan temporal. Model-model ini mampu menangkap nuansa emosional yang kompleks yang terkandung dalam dinamika suara, yang membuktikan keunggulan CNN dalam tugas pengenalan emosi (Abdelhamid et al., 2022; Aini et al., 2021; Alzubaidi et al., 2021; Rayhan Ahmed et al., 2023; Wijayasingha & Stankovic, 2021; Xu et al., 2021). Penelitian lain oleh Das et al. (2020) juga mendukung temuan ini yang menunjukkan bahwa penggunaan CNN dalam mengidentifikasi emosi dari fitur suara memberikan tingkat akurasi yang lebih tinggi dibandingkan dengan teknik-teknik *machine learning* tradisional (Das et al., 2020). *Convolutional Neural Network*s (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk mengolah data dengan struktur *grid*, seperti gambar dan sinyal suara. Dalam konteks *Speech Emotion Recognition* (SER), CNN digunakan untuk mengekstraksi fitur-fitur emosional dari sinyal suara dan mengklasifikasikannya ke dalam berbagai kategori emosi (Rayhan Ahmed et al., 2023). Potensi CNN untuk mempelajari berbagai pola spektrum-temporal telah membuatnya sangat cocok untuk klasifikasi suara lingkungan (Das et al., 2020).

CNN memiliki kemampuan luar biasa dalam mengekstraksi fitur penting dari data suara yang telah diubah menjadi representasi visual seperti *spectrogram*. Lapisan konvolusi pada CNN secara efektif dapat mengidentifikasi dan menyoroti pola-pola spesifik yang berhubungan dengan emosi dari sinyal suara. Seperti penelitian yang dilakukan oleh Rayhan Ahmed et al. (2023), fitur-fitur seperti *Mel-Frequency Cepstral Coefficients* (MFCCs), *log-mel spectrogram*, dan lainnya sangat penting karena mereka menangkap informasi frekuensi yang relevan dengan emosi. Penelitian tersebut juga menunjukkan bahwa fitur-fitur buatan tingkat rendah yang diproses oleh CNN dapat memberikan petunjuk emosional yang signifikan dari ucapan.

CNN memiliki keunggulan dalam hal generalisasi, terutama ketika digunakan dengan teknik data *augmentation*. Teknik ini membantu model untuk belajar dari variasi data yang lebih luas, sehingga meningkatkan kemampuan model untuk mengenali emosi dari berbagai kondisi dan *dataset* yang berbeda. Dengan data *augmentation* seperti penambahan *noise*, *shifting pitch*, dan *stretching*, model CNN dapat menjadi lebih *robust* terhadap variasi input (Das et al., 2020).



Gambar 18. Contoh arsitektur CNN dalam deteksi emosi suara (Mustaqeem & Kwon, 2020)

Selain itu, CNN memiliki struktur komponen yang fleksibel yang dapat dikombinasikan dengan berbagai algoritma lain untuk meningkatkan akurasi. Fleksibilitas ini memungkinkan integrasi dengan arsitektur lain seperti LSTM, GRU, dan berbagai teknik data *augmentation* untuk mengoptimalkan kinerja dalam pengenalan emosi dari sinyal suara. CNN mampu mengekstraksi fitur penting dari data dengan efisien, memanfaatkan kekuatan komponen-komponennya untuk memberikan representasi yang kaya dan mendetail dari input data. Komponen-komponen utama pada CNN yaitu *convolutional layer*, *activation function*, *pooling layer*, dan *fully connected layer* (Taye, 2023). Detail komponen utama CNN dijelaskan di bawah ini.

***Convolutional layer*.** *Convolutional layer* dalam SER digunakan untuk mengekstraksi fitur-fitur emosional dari sinyal suara yang telah diubah menjadi representasi visual seperti *log-mel-spectrogram*. Lapisan ini menangkap pola lokal dalam data suara yang berkaitan dengan berbagai emosi.

Lapisan ini menggunakan filter (*kernel*) untuk melakukan operasi konvolusi pada input, seperti *log-mel-spectrogram* dari sinyal suara. Filter diterapkan pada input data, bergerak melintasi input dengan langkah tertentu (*stride*), dan pada setiap posisi, *dot product* antara elemen filter dan elemen input dihitung. Hasil operasi ini adalah peta fitur yang mencerminkan fitur lokal seperti intonasi, nada, dan perubahan amplitudo yang berkaitan dengan emosi. Operasi konvolusi ini menghasilkan peta fitur yang menangkap pola lokal dalam data suara, mengurangi parameter model, dan memungkinkan model untuk dioptimalkan(Q. Zhou et al., 2021). Persamaan untuk *convolutional layer* ditunjukkan pada persamaan (29) (Q. Zhou et al., 2021).

$$x_j^n = f\left(\sum_{i \in M_j} x_i^{n-1} * k_{ij}^n + b_j^n\right) \quad (29)$$

di mana x_j^n adalah peta fitur *output*, x_i^{n-1} adalah peta fitur input, M_j adalah area terpilih dalam layar, $n-1$, k_{ij}^n adalah parameter bobot, b_j^n merupakan bias dan f merupakan fungsi aktivasi

***Activation function*.** Fungsi aktivasi dalam SER memperkenalkan *non-linearitas* ke dalam model, memungkinkan jaringan saraf untuk mempelajari representasi yang lebih kompleks dan non-linear dari data suara. Setelah operasi konvolusi, hasilnya diterapkan fungsi aktivasi. Fungsi aktivasi yang sering digunakan adalah ReLU (*Rectified Linear Unit*), yang mengubah nilai negatif menjadi nol dan mempertahankan nilai positif. Persamaan untuk *activation function* ditunjukkan pada persamaan (30) (Taye, 2023).

$$f(x) = \max(0, x) \quad (30)$$

di mana $f(x)$ adalah fungsi aktivasi (ReLU), x adalah *Output* dari operasi konvolusi pada titik tertentu dalam peta fitur. Fungsi ini mengubah semua nilai negatif menjadi nol, mempertahankan hanya nilai positif.

***Pooling layer*.** *Pooling layer* dalam SER digunakan untuk mengurangi *dimensionalitas* peta fitur sambil mempertahankan fitur-fitur penting yang berkaitan dengan emosi dalam data suara (D. Scherer et al., 2010). *Pooling layer* mengambil jendela kecil dari peta fitur dan menghitung nilai agregat, seperti nilai maksimum dalam kasus *max pooling*. Operasi *pooling* memilih nilai maksimum dalam jendela *pooling*, mengurangi resolusi spasial peta fitur, dan membuat jaringan lebih tahan terhadap pergeseran dan distorsi kecil dalam input. Persamaan untuk *pooling layer* (*max pooling*) ditunjukkan pada persamaan (31) (Nanos, 2024).

$$P(i,j) = \max_{m,n}(I(i+m, j+n)) \quad (31)$$

di mana $P(i,j)$ adalah nilai maksimum dalam jendela *pooling* pada posisi i,j, I adalah input (peta fitur), dan m,n adalah posisi relatif dalam jendela *Pooling*.

***Fully connected Layer*.** *Fully connected layer* dalam SER menghubungkan setiap neuron di lapisan sebelumnya ke setiap neuron di lapisan berikutnya, digunakan untuk klasifikasi berdasarkan fitur-fitur yang diekstraksi. *Output* dari *pooling layer* diubah menjadi vektor satu dimensi

dalam proses yang disebut *flattening*. Setiap elemen vektor terhubung ke setiap neuron di *fully connected layer* melalui bobot dan bias. Input ke neuron dihitung sebagai kombinasi linear dari input vektor dengan bobot dan bias, kemudian diterapkan fungsi aktivasi untuk mendapatkan *output* akhir. Persamaan *fully connected layer* ditunjukkan pada persamaan (32) (Q. Zhou et al., 2021).

$$y_j = f\left(\sum_{i=1}^{N} x_i * w_{ij} + b_j\right) \tag{32}$$

di mana x_i adalah layar input, N adalah jumlah *node* lapisan input, w_{ij} merupakan bobot antara tautan x_i dan y_j, b_j adalah bias, dan f adalah fungsi aktivasi.

1.2.7 Evaluasi model

Evaluasi model adalah langkah penting dalam *deep learning* untuk mengukur kinerja model yang telah dilatih, salah satu metode evaluasi model yaitu metrik evaluasi. Metrik evaluasi memberikan wawasan tentang seberapa baik model tersebut dalam memprediksi data yang belum pernah dilihat sebelumnya. Evaluasi yang tepat dan analisis pengenalan emosi suara dalam lingkungan alami adalah hal yang esensial. Literatur telah menyarankan beberapa metrik evaluasi untuk mengukur pengenalan emosi suara dalam kondisi berisik. Metrik evaluasi yang paling umum digunakan adalah akurasi, *confusion matrix*, *precision*, *recall*, dan *F1-score* (George & Muhamed Ilyas, 2024). Dalam penelitian ini, metrik evaluasi yang akan digunakan adalah *confusion matrix* dan akurasi yang merupakan metrik dasar dalam melakukan evaluasi, serta turunan dari akurasi yaitu *weighted accuracy* (WA) yang digunakan ketika *dataset* per emosi yang dimiliki tidak seimbang (Xu et al., 2021).

***Weighted accuracy* (WA).** *Weighted accuracy* mengacu pada akurasi dari semua sampel, dengan mempertimbangkan distribusi kelas dalam *dataset*. Ini membantu dalam kasus ketidakseimbangan kelas dengan memberikan bobot lebih pada kelas yang memiliki lebih banyak sampel. Rumus persamaan dari akurasi ditunjukkan pada persamaan (33).

$$Weighted\ Accuracy = \sum_{i=1}^{k} \frac{N_i}{N} \times \frac{TP_i}{TP_i + FN_i} \tag{33}$$

di mana k adalah jumlah kelas, N_i adalah jumlah sampel dalam kelas i, N adalah jumlah total sampel dalam *dataset*, TP_i adalah jumlah *True Positives* untuk kelas i, dan FN_i adalah jumlah *False Negatives* untuk kelas i.

Weighted accuracy juga mencerminkan distribusi kelas yang sebenarnya dalam *dataset*, sehingga lebih realistis dalam kondisi dunia nyata di mana distribusi kelas tidak merata.

Confusion matrix. *Confusion matrix* adalah tabel yang digunakan untuk mengevaluasi kinerja algoritma klasifikasi. Matriks ini memperlihatkan perbandingan antara prediksi model dan hasil aktual dari data yang diuji, yang memungkinkan untuk melihat kesalahan dan keberhasilan klasifikasi secara lebih rinci yang dibuat oleh model, dengan merinci kelas-kelas yang diprediksi. *Confusion matrix* mencakup:

- *True Positives* (TP)**:** Jumlah prediksi benar untuk kelas positif.
- *True Negatives* (TN)**:** Jumlah prediksi benar untuk kelas negatif.
- *False Positives* (FP)**:** Jumlah prediksi salah di mana kelas negatif diprediksi sebagai positif.
- *False Negatives* (FN)**:** Jumlah prediksi salah di mana kelas positif diprediksi sebagai negatif.

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Meningkatkan akurasi dan ketahanan *noise* pada sistem klasifikasi emosi dari data suara.
2. Menganalisis pengaruh *Amplitude Factor* (AF) terhadap performa sistem metode pengenalan emosi dalam kondisi dengan tingkat *noise* yang berbeda.

1.3.2 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Menghasilkan model klasifikasi emosi yang lebih *robust* terhadap *noise*, sehingga meningkatkan akurasi pengenalan emosi dalam berbagai kondisi lingkungan.
2. Memberikan kontribusi pada pengembangan teknik *adaptive noise* dalam bidang pengenalan emosi berbasis suara.
3. Menyediakan korpus suara bahasa Indonesia yang mencakup berbagai tingkat kebisingan, untuk mendukung penelitian lebih lanjut.
4. Memfasilitasi analisis emosi yang lebih akurat pada video ulasan produk, meskipun dalam kondisi audio yang tidak ideal.

BAB II

METODE PENELITIAN

2.1 Tempat dan Waktu

Penelitian dilakukan mulai sejak disetujuinya proposal penelitian pada November 2023 sampai Oktober 2024. Penulis melakukan penelitian ini di Studio Audio AIMP, Laboratorium *Artificial Intelligence,* Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin seperti ditunjukkan pada Gambar 19.

Gambar 19. Studio audio AIMP

2.2 Benda Uji dan Alat

Benda dan alat yang digunakan dalam penelitian ini meliputi perangkat lunak dan perangkat keras berikut:

1. Perangkat lunak
    a. Operating System: Windows 11 64-bit
    b. Jupyter Lab
    c. Visual Studio Code
    d. Microsoft Excel
    e. Microsoft Word
    f. Microsoft Powerpoint
    g. Microsoft Edge
    h. Notion
    i. Audacity Dekstop
    j. Mendeley
    k. Figma
    l. Canva
2. Perangkat keras
    a. VivoBook ASUS Laptop X415JP, Intel Core i3-1005G1, NVIDIA GeForce MX330, RAM 12GB, SSD 512GB

   b. ASUS ROG, Intel Core i7-7700HQ, NVIDIA GTX 1080 8GB, RAM 16GB, SSD 1TB
   c. Microphone BM800
   d. Audio Mixer Maonocaster E2
   e. Stnad arm Nb-35
3. Bahasa pemrograman
   a. *Python* v3.11.0
4. Library
   a. Google-colab v1.0.0 oleh tim Google Colaboratory berguna untuk integrasi antara Google Colab dan Google Drive
   b. Librosa v0.10.0.post2 oleh tim Librosa Development berguna untuk analisis audio yang mencakup fungsi dalam mengekstrak informasi dari audio seperti spektrum daya, *pitch, beat, onset*, dan lain-lain.
   c. Pandas v1.5.3 oleh tim The Pandas Development berguna untuk manipulasi dan analisis data.
   d. Numpy v1.22.4 oleh Travis E. Oliphant et al. berguna untuk menyediakan dukungan *array* dan matriks besar fungsi matematis operasi pada array tersebut.
   e. Seaborn v0.12.2 berguna untuk menyediakan antarmuka tingkat tinggi untuk menggambar grafik statistik yang informatif dan menarik.
   f. Matplotlib v3.7.1 oleh John D. Hunter & Michael Droettboom berguna untuk visualisasi data yang digunakan untuk menggambar grafik di *Python*.
   g. Sklearn atau scikit-learn v1.2.2 adalah pustaka *machine learning* untuk *Python*. Berguna menyertakan algoritma untuk klasifikasi, regresi, pengelompokan, pemrosesan *pra*-data, pemilihan dan evaluasi model, dan lain-lain.
   h. Keras v2.12.0 oleh tim Keras yang berguna untuk *deep learning* yang menyediakan antarmuka tingkat tinggi untuk pembuatan dan pelatihan model jaringan saraf.

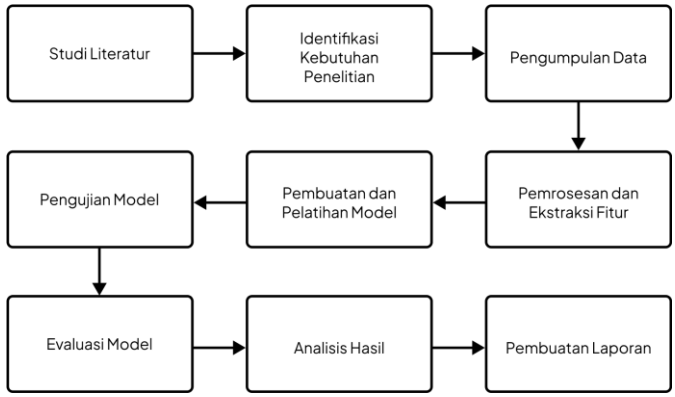
2.3 Tahapan penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang dirancang secara sistematis dan komprehensif. Tahapan dimulai dengan studi literatur, yang bertujuan untuk memahami konsep-konsep yang relevan serta penelitian terdahulu yang berkaitan dengan topik ini. Setelah itu, dilakukan identifikasi kebutuhan penelitian, di mana permasalahan dan tujuan penelitian ditentukan. Tahap selanjutnya adalah pengumpulan data, di mana data yang akan dianalisis dikumpulkan dari berbagai sumber.

Setelah data terkumpul, dilakukan pemrosesan dan ekstraksi fitur untuk mengubah data mentah menjadi representasi yang lebih kompak dan bermakna. Kemudian, pada tahap pembuatan dan pelatihan model, model dikembangkan dan

dilatih menggunakan data yang telah diproses. Setelah model selesai dibuat, dilakukan pengujian model untuk mengevaluasi kinerjanya dalam mengklasifikasikan atau memprediksi data baru.

Tahapan terakhir mencakup evaluasi model, di mana hasil pengujian dianalisis untuk menentukan seberapa baik model bekerja. Setelah itu, dilakukan analisis hasil, di mana data dari evaluasi model diinterpretasikan untuk menjawab tujuan penelitian. Terakhir, seluruh proses penelitian dirangkum dalam pembuatan laporan yang mendokumentasikan setiap tahap dan hasil dari penelitian ini.



Gambar 20. Tahapan penelitian

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini merupakan gabungan beberapa jenis kumpulan data yang berbeda. Teknik pengambilan data pada penelitian ini dibagi menjadi data primer, sekunder dan *noisy.*

2.4.1 Data primer

Data primer yang digunakan dalam penelitian ini adalah data yang diperoleh langsung dari rekaman suara emosi ucapan berbahasan Indonesia.

> ***Data collection*.** Kriteria responden yaitu berumur minimal 17 tahun, memiliki kemampuan berbahasa Indonesia yang baik dengan aksen baku, mampu menunjukkan emosi (bahagia, kecewa, dan netral) melalui ucapannya dan memiliki suara serta artikulasi huruf yang jelas. Responden berjumlah 10 orang yang terdiri dari 5 responden perempuan dan 5 responden laki-laki. Data responden berdasarkan jenis kelamin, usia, dan latar belakang ditampilkan pada Tabel 1.

Tabel 1. Data responden perekaman suara

Urutan Responden	Jenis Kelamin	Usia	Profesi
Responden 1	L	22	Mahasiswa
Responden 2	P	22	Mahasiswa
Responden 3	L	21	Mahasiswa
Responden 4	P	22	Mahasiswa
Responden 5	L	19	Mahasiswa
Responden 6	P	21	Mahasiswa
Responden 7	L	21	Mahasiswa
Responden 8	P	21	Mahasiswa
Responden 9	L	22	Mahasiswa
Responden 10	P	20	Mahasiswa

Setiap responden melakukan rekaman langsung di studio audio, dalam ruangan kedap suara. Suara responden ditangkap menggunakan *microphone* saluran mono yang terhubung ke *mixer* audio serta komputer. Visualisasi dan pengolahan rekaman suara responden dilakukan menggunakan aplikasi Audacity *dekstop*. Adapun pengaturan resolusi ditampilkan pada Tabel 2.

Tabel 2. Resolusi audio yang direkam

Channel	***Sample rate***	***Bit depth***	**Format**
Mono	48 kHz	16 bit	.wav

Pemilihan resolusi didasarkan pada penelitian pembuatan *dataset* (Jackson & ul haq, 2011; Livingstone & Russo, 2018; Pichora-Fuller & Dupuis, 2020) yang sering digunakan pada penelitian *speech emotion recognition* (Abdelhamid et al., 2022; Li et al., 2022; Rayhan Ahmed et al., 2023; Xu et al., 2021; Xu, Zhang, Yang, et al., 2020) dengan deskripsi berikut:

- **Menggunakan kanal perekaman mono**: Pemilihan kanal perekaman mono memastikan bahwa suara direkam dari satu sumber, yang umum dalam perekaman ucapan. Ini mengurangi kompleksitas sinyal dan fokus pada suara utama tanpa campuran saluran lain, sehingga lebih mudah dianalisis dan diproses dalam penelitian pengenalan emosi suara..
- ***Sample rate* 48 kHz:** Standar dalam industri musik dan perekaman audio berkualitas tinggi, *sample rate* 48 kHz digunakan untuk menangkap detail frekuensi tinggi dengan lebih baik. Ini memastikan

bahwa rekaman memiliki resolusi temporal yang cukup untuk analisis spektrum yang akurat, mendukung pengenalan detail halus dalam intonasi dan ekspresi emosi.

- ***Bit depth* 16 bit**: *Bit depth* 16 bit menyediakan rentang dinamis yang lebih besar, memungkinkan rekaman untuk menangkap detail audio yang lebih halus dan variasi dinamis yang lebih luas. Hal ini penting untuk memastikan bahwa ekspresi emosional dalam suara dapat direkam dengan kejelasan dan akurasi yang tinggi, tanpa kehilangan detail penting akibat keterbatasan rentang dinamis.
- **Format audio .wav**: Format .wav adalah format tanpa kompresi yang menjaga kualitas asli dari rekaman. Ini penting untuk penelitian yang memerlukan analisis detail dari sinyal audio karena tidak ada informasi yang hilang selama proses kompresi. Format .wav memungkinkan penyimpanan dan pemrosesan data audio dengan kualitas terbaik, memastikan integritas data selama analisis lebih lanjut.

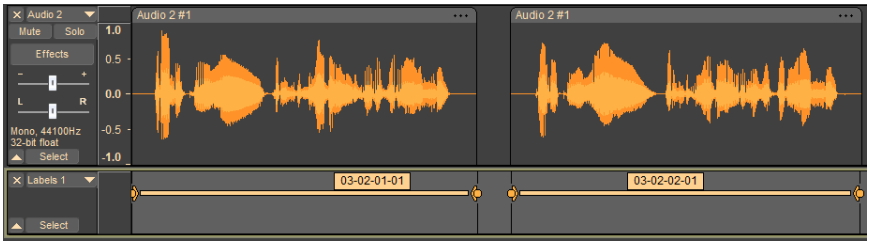
Setiap responden akan mengucapkan kalimat yang sama yaitu **"Produk ini tersedia dalam berbagai variasi"** dengan 3 kelas emosi yaitu netral, bahagia dan kecewa pada intensitas normal dan kuat sebanyak 3 kali perulangan. Alasan pemilihan kalimat tersebut antara lain:

1. Netralitas Konten: Kalimat ini netral dan tidak mengandung emosi yang jelas, sehingga memungkinkan responden untuk mengekspresikan emosi yang ditentukan (netral, bahagia, kecewa) dengan lebih mudah.
2. Keseragaman Pengujian: Menggunakan kalimat yang sama untuk semua responden dan semua kondisi emosi memastikan konsistensi dalam data yang dikumpulkan, yang penting untuk analisis yang akurat.
3. Kompleksitas Sederhana: Kalimat ini cukup sederhana namun cukup panjang untuk menangkap variasi dalam intonasi, nada, dan ekspresi emosi.
4. Relevansi Kontekstual: Kalimat ini memiliki relevansi kontekstual yang memungkinkan berbagai interpretasi emosional, membuatnya ideal untuk penelitian pengenalan emosi.

Data preprocessing. Setelah direkam, data audio akan menjalani proses pembersihan dan pengolahan sebelum digunakan untuk pelatihan model. Tahap ini melibatkan normalisasi volume dan penghapusan *noise* untuk memastikan kualitas audio yang optimal. Beberapa *library* yang digunakan dalam proses ini termasuk Librosa untuk analisis dan ekstraksi fitur audio, Pydub untuk manipulasi audio, serta model *enhance* audio untuk peningkatan kualitas rekaman. Langkah-langkah ini penting untuk menghasilkan data yang bersih dan konsisten, yang mendukung performa model dalam mengenali emosi dari ucapan dengan lebih baik.

Data labeling. Setelah data audio diproses, langkah berikutnya adalah melakukan proses *labeling* untuk memberikan label atau nama pada setiap segmen audio. Setiap label diberikan berdasarkan durasi tertentu, yang biasanya berkisar antara 3 hingga 5 detik per segmen. Proses ini dilakukan menggunakan *software* Audacity, di mana audio direpresentasikan dalam bentuk *waveform*, dan fitur *labeling* Audacity dimanfaatkan untuk memberikan nama atau kategori pada setiap bagian audio. Label ini berfungsi untuk mengidentifikasi segmen audio sesuai dengan klasifikasi yang diinginkan.

Setelah proses pelabelan selesai, dilakukan ekspor ganda di mana setiap *file* audio disimpan berdasarkan label yang telah diberikan. Setiap *file* audio diberi nama unik yang terdiri dari empat bagian, dipisahkan oleh tanda strip. Penamaan ini memudahkan identifikasi berdasarkan kategori atau tingkat yang relevan, seperti kelas emosi, intensitas emosi, pengulangan, dan responden. Aturan penamaan *file* tersebut dijelaskan dalam Tabel 3.



Gambar 21. Contoh pelabelan data menggunakan *software* audacity. Setiap segmen audio dilabeli dengan menggunakan fitur *labeling* yang disediakan oleh Audacity

Tabel 3. Aturan penamaan dari *file* perekaman

Identifier	**Dekskripsi**
Kelas Emosi	(01) = netral, (02) = bahagia, (03) = kecewa
Intensitas Emosi	(01) = normal, (02) = kuat
Pengulangan	(01) pengulangan pertama, (02) pengulangan kedua, (03) pengulangan ketiga
Responden	(01-10) responden, (01) sampai (10). (01 – 05) laki-laki, (06 – 10) perempuan.

2.4.2 Data sekunder

Data sekunder dalam penelitian ini terbagi menjadi 3 jenis yaitu data RAVDESS (*Ryerson Audio-Visual Database of Emotional Speech and Song*) yang berasal dari Livingstone & Russo (2018), data IndoWaveSentiment yang berasal dari Bustamin et al. (2024) , dan data emosi kecewa yang bersumber dari data primer.

RAVDESS *(ryerson* audio-*visual database of emotional speech and song)*. *The Ryerson Audio-Visual Database of Emotional Speech and Song* (RAVDESS) adalah *dataset multimodal* yang dirancang untuk penelitian pengenalan emosi dari ucapan dan lagu. *Dataset* ini terdiri dari 24 aktor profesional (12 perempuan dan 12 laki-laki) yang mengekspresikan emosi melalui ucapan dan nyanyian dalam aksen Amerika Utara yang netral. Setiap aktor menyampaikan dua pernyataan yang secara leksikal cocok dalam berbagai emosi. Emosi yang diekspresikan dalam *dataset* ini mencakup tenang, bahagia, sedih, marah, takut, terkejut, dan jijik. Setiap emosi dilakukan pada dua tingkat intensitas, yaitu normal dan kuat, kecuali untuk emosi netral yang hanya memiliki satu tingkat intensitas (Livingstone & Russo, 2018).

Dataset RAVDESS di unduh melalui *link* yang disediakan pada penelitian (Livingstone & Russo, 2018). Lalu akan dilakukan pemilahan terhadap *dataset* sehingga yang akan diunduh hanya *dataset* audio saja. Fail yang telah diunduh akan dipindahkan ke folder yang terpisah untuk memudahkan akses dan pengolahan lebih lanjut. Proses ini memastikan bahwa data yang akan digunakan itu hanya data audio dengan emosi bahagia dan netral yang diperlukan pada penelitian. Sedangkan untuk resolusi dari *dataset* RAVDESS itu ditunjukkan pada Tabel 4.

Tabel 4. Resolusi audio *speech dataset ravdess*

Sample rate	***Bit depth***	**Format**
48 kHz	16 bit	.wav

IndoWaveSentiment *dataset*. IndoWaveSentiment *dataset* adalah kumpulan data yang dirancang untuk penelitian pengenalan emosi suara dalam bahasa Indonesia. *Dataset* ini dikembangkan untuk mengatasi keterbatasan data bahasa Indonesia dalam penelitian pengenalan emosi suara. *Dataset* ini terdiri dari rekaman suara dari 10 responden (5 laki-laki dan 5 perempuan) yang mengekspresikan lima emosi: netral, bahagia, terkejut, jijik, dan kecewa. Setiap responden mengucapkan kalimat "Kualitas HP ini cukup bagus" dengan berbagai ekspresi emosi pada dua tingkat intensitas (normal dan kuat).

IndoWaveSentiment *dataset* diakses dan diunduh melalui izin dari peneliti yang mengembangkan *dataset* tersebut. Setelah mendapatkan izin, *dataset* diunduh dari sumber yang telah disediakan. *Dataset* ini kemudian dipilah untuk memastikan hanya *file* audio yang sesuai dengan kebutuhan penelitian, yaitu emosi bahagia dan netral. *File* audio yang telah diunduh kemudian dipindahkan ke folder terpisah untuk memudahkan akses dan pengolahan lebih lanjut. Proses ini memastikan bahwa data yang digunakan dalam penelitian hanya mencakup rekaman dengan emosi

bahagia dan netral, sesuai dengan tujuan penelitian. Resolusi data dari *dataset* ini ditunjukkan pada Tabel 5 ini.

Tabel 5. Resolusi audio *speech* IndoWaveSentiment *dataset*

Sample rate	***Bit depth***	**Format**
16 kHz	32 bit	.wav

Primer *join* sekunder. Data Primer *Join* Sekunder merupakan data primer dengan emosi kecewa yang dialokasikan untuk menutupi kekurangan dari data emosi kecewa pada *dataset* RAVDESS. Resolusi data dari primer *join* sekunder ditunjukkan pada ini Tabel 6.

Tabel 6. Resolusi audio *dataset* primer *join* sekunder

Sample rate	***Bit depth***	**Format**
48 kHz	16 bit	.wav

Gadgetin *Dataset*. Gadgetin *dataset* berasal dari IndoWaveSentiment. Gadgetin *dataset* merupakan data yang diperoleh dari video YouTube kanal GadgetIn yang mengandung beberapa sampel emosi suara dalam video tersebut. Data Gadgetin diolah dibagi menjadi 5 kelas emosi (*neutral*, *happy*, *surprise*, *disgust*, *disappointed*). Setiap label memiliki panjang durasi 3 hingga 5 detik.

Tabel 7. Resolusi audio *dataset gadgetin*

Sample rate	***Bit depth***	**Format**
16 kHz	32 bit	.wav

2.4.3 Data *noise*

Data *Noise* yang akan digunakan dalam penelitian ini yaitu ESC-50 *dataset*. ESC-50 *dataset* adalah kumpulan data yang terdiri dari rekaman audio dari 50 kelas suara lingkungan yang berbeda, dengan masing-masing kelas terdiri dari 40 rekaman. *Dataset* ini dikembangkan untuk penelitian dalam pengenalan suara lingkungan dan klasifikasi suara (Piczak, 2015). Dalam penelitian ini, diambil lima jenis suara *noise* berdasarkan survei awal yang telah dilakukan yang ditunjukkan pada Gambar 2, yaitu suara lingkungan, suara manusia, suara konstruksi, suara hewan, dan suara elektronik.

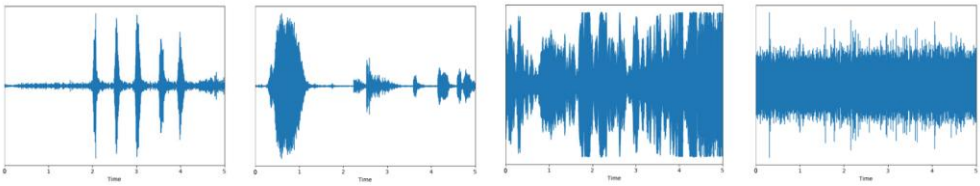
Proses pengumpulan data *noise* dimulai dengan mengakses dan mengunduh *dataset* ESC-50 dari sumber resmi, kemudian memilah data untuk memilih kelas suara yang relevan dengan hasil survei, yaitu suara lingkungan,

manusia, konstruksi, hewan, dan elektronik. Setelah pemilahan, sampel suara paling representatif dipilih untuk setiap kategori melalui pengecekan kualitas suara guna memastikan kesesuaian dengan standar penelitian. Sampel yang terpilih kemudian dipindahkan ke folder terpisah yang dikelompokkan berdasarkan kategori *noise* untuk memudahkan akses dan pengolahan data lebih lanjut. Selain itu akan dipilih juga satu *noise* tambahan di tiap-tiap kelas sebagai pembanding dan pembuatan data *testing*.

Tabel 8. Empat kategori dari *noise* pada ESC-50 (Xu et al., 2021)

Kategori	**Pemisahan**	**Fluktuasi**
Diskrit-Pendek (D-Pe)	Jelas	Jelas
Diskrit-Panjang (D-P)	Jelas	Tidak Jelas
Kontinu-Pendek (C-Pe)	Tidak Jelas	Jelas
Kontinu-Panjang (C-P)	Tidak Jelas	Tidak Jelas

Dilakukan juga *reclassify* dari tiap jenis-jenis *noise* yang akan digunakan dengan metode yang berasal dari (Xu et al., 2021) yang membagi jenis tersebut kedalam 4 kategori yang ditunjukkan pada Tabel 8 berdasarkan apakah ada pemisahan yang jelas di antara segmen *noise* dan apakah ada fluktuasi yang jelas dan terputus-putus.

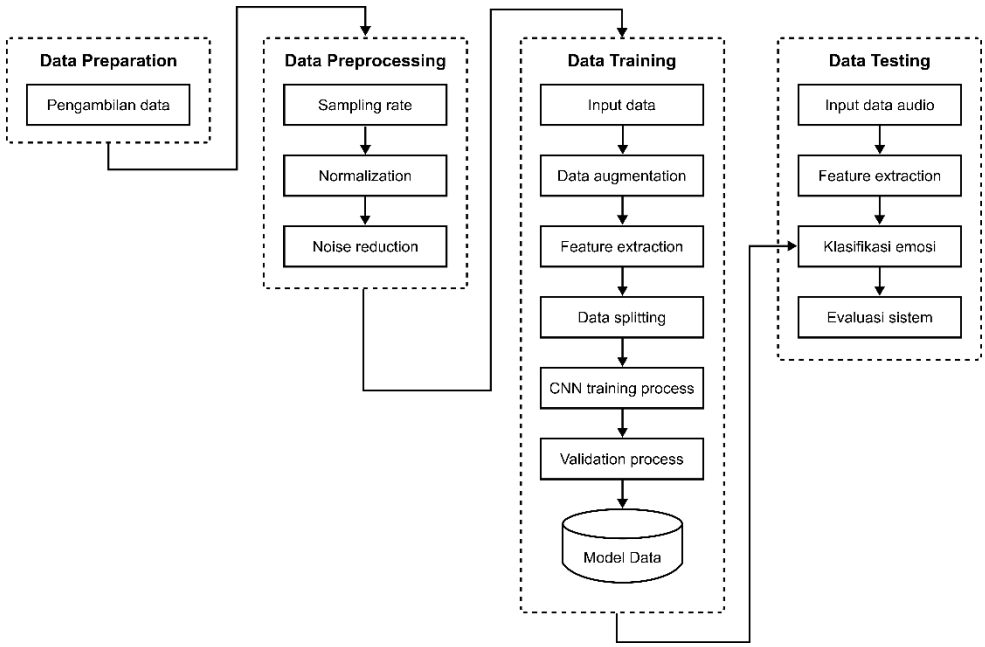


(a) D-Pe (b) D-P (c) C-Pe (d) C-P

Gambar 22. Contoh *waveform* tiap-tiap kategori *noise*

2.5 Perancangan dan Implementasi Sistem

Perancangan sistem memberikan langkah-langkah teoritis dan konseptual yang memberikan gambaran dari sistem yang akan dibuat. Proses sistem dalam penelitian ini ditunjukkan pada Gambar 23.



Gambar 23. Alur perancangan sistem. Proses perancangan sistem dimulai dari data *preparation*, lalu data *preprocessing*, lalu akan melalui proses *training* serta *testing* sebagai proses akhir

2.5.1 Data *preparation*

Pada tahap data *preparatioin*, dilakukan pengambilan data yang akan digunakan selama proses *training* maupun *testing*. Data yang akan diambil terbagi atas 4, yaitu data primer yang bersumber dari perekaman audio, data sekunder yang bersumber dari peneliti sebelumnya (Bustamin et al., 2024; Livingstone & Russo, 2018) , data primer yang melengkapi data sekunder dan data *noise* (Piczak, 2015).

2.5.2 Data *preprocessing*

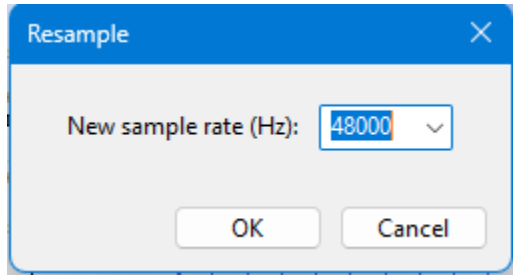
Data *preprocessing* adalah tahap awal dalam pengolahan data yang bertujuan untuk menyiapkan data mentah menjadi bentuk yang lebih sesuai dan optimal untuk analisis atau pelatihan model. Dalam konteks pemrosesan audio, *preprocessing* mencakup berbagai teknik untuk membersihkan dan mengubah data suara mentah agar dapat digunakan secara efektif dalam model pembelajaran mesin atau analisis lainnya. Berikut adalah langkah-langkah umum dalam *preprocessing* data audio.

1. *Sampling Rate*

   Pada tahap ini, dilakukan penyesuaian *sampling rate* pada setiap *file* audio untuk memastikan konsistensi, yaitu sebesar 48 kHz. Proses penyesuaian ini dilakukan dengan beberapa langkah utama. Pertama, *file* audio dimuat ke dalam perangkat lunak Audacity. Setelah *file* terbuka,

dilakukan pengaturan *Project Rate* pada Audacity ke 48000 Hz, yang merupakan nilai *sampling rate* yang akan digunakan pada seluruh *dataset*.

Selanjutnya, konversi *sampling rate* diterapkan pada *track* audio dengan menggunakan fitur *resample* untuk memastikan setiap *file* mengikuti nilai *sampling rate* yang diinginkan. Dengan cara ini, sinyal audio dikonversi ke 48 kHz tanpa mengubah karakteristik dasar sinyal. Setelah proses konversi selesai, *file* audio diekspor dan disimpan dalam format yang sesuai untuk digunakan dalam analisis.

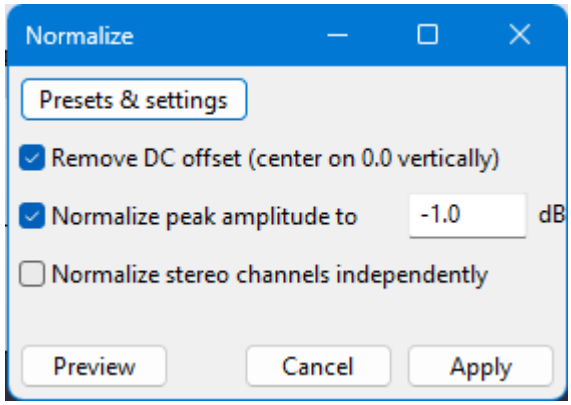


Gambar 24.Tampilan fitur *resample* pada audacity

Proses penyesuaian *sampling rate* ini memastikan bahwa seluruh *file* dalam *dataset* memiliki resolusi suara yang seragam, sehingga memudahkan proses analisis serta meningkatkan kualitas dan konsistensi data untuk tahap-tahap pemrosesan selanjutnya.

2. *Normalization*

Pada tahap ini, dilakukan normalisasi amplitudo pada setiap *file* audio untuk memastikan konsistensi nilai amplitudo di seluruh *dataset*. Proses normalization dilakukan menggunakan perangkat lunak Audacity. Langkah pertama dalam normalisasi adalah membuka *file* audio dalam Audacity dan memastikan seluruh bagian *track* audio terpilih. Selanjutnya, proses normalisasi amplitudo diterapkan dengan menggunakan fitur *Normalize*, di mana *peak amplitude* disesuaikan menjadi -1 dB. Penyesuaian ini bertujuan untuk menjaga amplitudo puncak dari sinyal audio tetap dalam batas aman, sehingga memberikan sedikit *headroom* untuk menghindari *clipping* atau distorsi pada pemrosesan lebih lanjut.



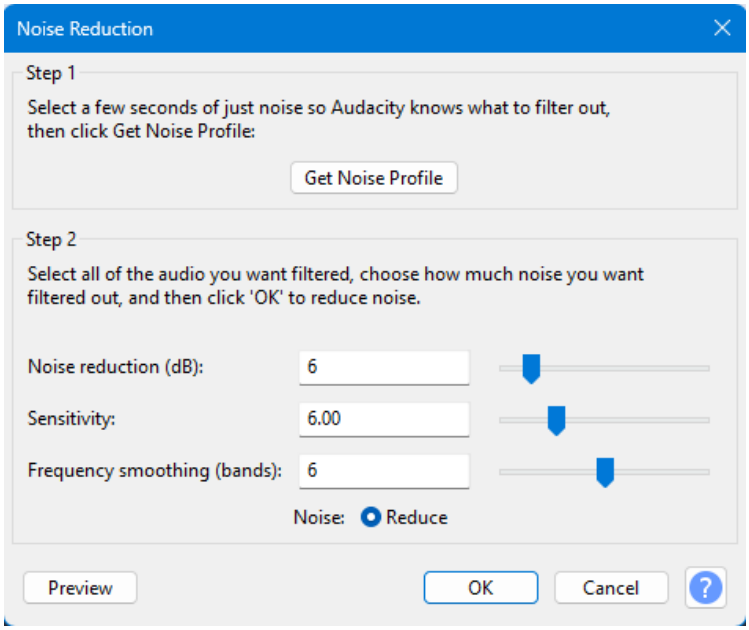
Gambar 25.Tampilan fitur *normalize* pada audacity

Normalisasi ini membantu mengurangi variasi amplitudo yang ekstrem di antara *file* audio, sehingga seluruh *dataset* memiliki skala amplitudo yang seragam dan konsisten. Proses ini diharapkan dapat mempermudah analisis dan meningkatkan akurasi dalam pelatihan model, karena setiap *file* audio disajikan dengan amplitudo yang telah dinormalisasi secara merata.

3. *Noise Reduction*

Pada tahap ini, diterapkan teknik *noise reduction* untuk mengurangi atau menghilangkan suara latar yang tidak diinginkan dari setiap *file* audio. Proses ini dilakukan dengan menggunakan perangkat lunak Audacity untuk meningkatkan kualitas sinyal audio sehingga fitur yang relevan dapat diekstraksi dengan lebih jelas.

Langkah pertama adalah mengidentifikasi bagian dari audio yang hanya berisi *noise* latar, seperti desisan atau dengungan, yang nantinya akan digunakan sebagai *noise profile*. Setelah *noise profile* ditentukan, seluruh *track* audio dipilih, dan efek *noise reduction* diterapkan dengan mengacu pada profil *noise* yang telah diambil.



Gambar 26. Tampilan fitur *noise reduction* pada audacity

Pengaturan intensitas *reduction* diatur untuk mengoptimalkan hasil tanpa menghilangkan bagian sinyal utama, sehingga hanya *noise* latar yang direduksi secara signifikan. Melalui proses ini, kualitas sinyal utama menjadi lebih jelas, memungkinkan data audio yang lebih bersih dan seragam untuk digunakan pada tahap-tahap analisis dan pelatihan model selanjutnya.

2.5.3 Data *training*

Pada tahap data *training*, dilakukan beberapa proses untuk membentuk sebuah sistem dengan model yang dilatih agar mampu menghasilkan model yang terbaik dalam melakukan klasifikasi emosi. Adapun proses yang dilakukan pada tahap ini adalah sebagai berikut:

1. Input data

   Data yang akan digunakan dalam proses *training* secara umum ada 2 yaitu data *clean* (bersih dari *noise* audio) dan data *noisy* (data *clean* yang dinjeksi *noise*). Kedua jenis data tersebut merupakan gabungan dari data primer dan sekunder yang nantinya akan digunakan dalam berbagai skenario pelatihan.

2. Data *augmentation*

   Sebelum dilakukan data *augmentation* pada sistem, terlebih dahulu dilakukan proses *noise injection. Noise injection* adalah teknik menambahkan *noise* ke dalam data audio untuk pelatihan model, dengan tujuan meningkatkan ketahanan model terhadap berbagai jenis *noise* yang bersumber dari *noise dataset*. Dalam penelitian ini, *noise injection* dilakukan

dengan menambahkan lima sampel dari jenis *noise* yang terpilih ke dalam *clean dataset*, di mana amplitudo *noise* diatur menggunakan *Amplitude Factor* (AF). AF mulai dari 0,1 hingga 1,0, dengan kenaikan bertahap sebesar 0,1, yang merepresentasikan perbandingan amplitudo *noise* terhadap sinyal asli.

Proses *noise injection* dilakukan dengan langkah-langkah sebagai berikut:

1. Penyesuaian Volume *Noise*

   AF diterapkan pada sinyal *noise* sebelum ditambahkan ke sinyal *clean*. Pengaturan ini dilakukan dengan mengonversi *AF* menjadi penyesuaian desibel (dB) menggunakan kode berikut:

```python
# Adjust noise volume based on the amplitude factor
    if amplitude_factor > 0:
        adjustment_dB = 20 * np.log10(amplitude_factor)
        noise = noise + adjustment_dB
        print(f"Adjusted noise by {adjustment_dB} dB")
    else:
        # If amplitude factor is 0, we don't add any noise
        noise = AudioSegment.silent(duration=len(audio))
        print(f"No noise added as AF is {amplitude_factor}")
```

   Gambar 27. Fungsi implementasi AF pada *noise injection*

   Dalam kode ini, *amplitude_factor* menentukan berapa kali *noise* harus diperkuat atau dilemahkan sebelum ditambahkan. Misalnya, jika *AF* bernilai 0.5, maka sinyal *noise* akan dikurangi sebesar -6 dB sebelum ditambahkan ke sinyal utama. Sebaliknya, nilai *AF* di atas 1 akan meningkatkan volume *noise*.

2. Sinkronisasi Durasi antara Sinyal Utama dan *Noise*

   Untuk memastikan *noise* dapat diterapkan secara seragam di sepanjang durasi audio, durasi sinyal *noise* disesuaikan dengan sinyal *clean*. Jika durasi *noise* lebih pendek dari sinyal *clean*, *noise* akan diulangi (looped) hingga panjangnya sesuai dengan durasi sinyal utama. Jika durasi *noise* lebih panjang, maka sinyal akan dipotong sesuai panjang sinyal *clean*.

3. Penggabungan Sinyal Utama dan *Noise*

   Setelah sinyal *noise* diatur berdasarkan durasi dan *AF*, sinyal *noise* ditumpangkan (*overlay*) ke sinyal utama. Hasilnya adalah sinyal audio baru yang mengandung *noise* latar dengan intensitas yang diatur.

4. Penyimpanan Hasil

   File audio hasil injeksi *noise* disimpan dalam struktur folder yang diatur berdasarkan jenis *noise* dan nilai *AF*. Hal ini bertujuan untuk menjaga keteraturan data dan mempermudah pengelolaan selama proses pelatihan model.

5. Perhitungan Signal-to-*Noise* Ration (SNR)

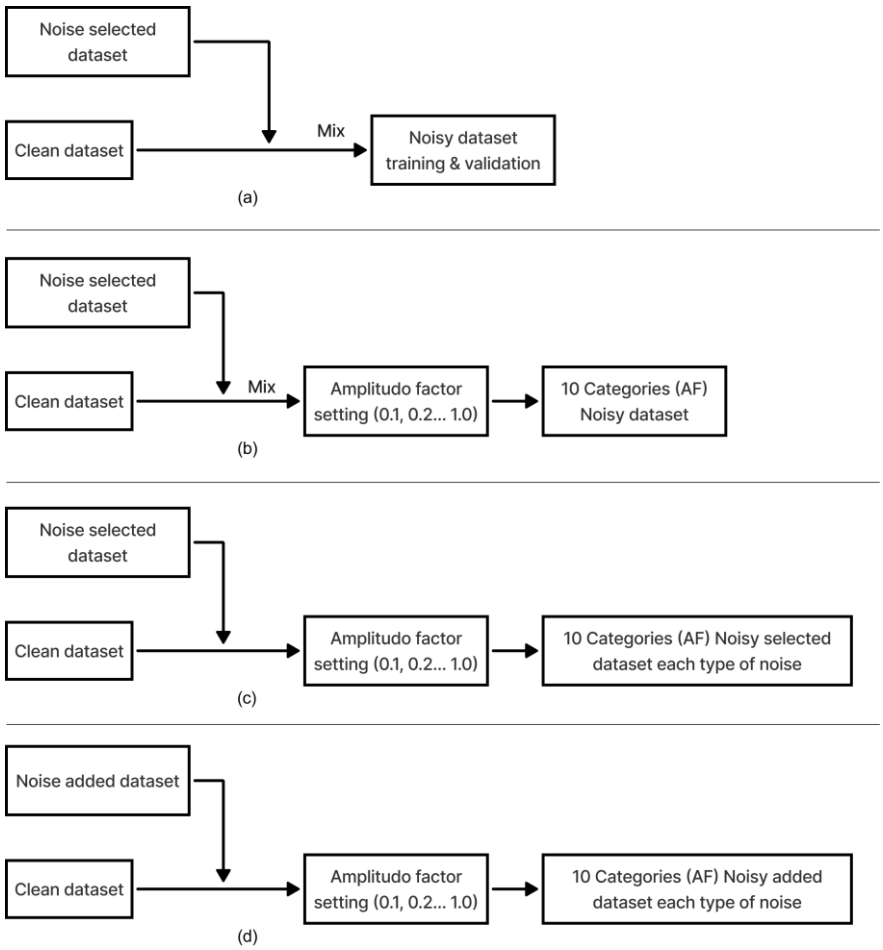
Dilakukan juga perhitungan SNR pada *dataset* noisy untuk melihat seberapa besar perbandingan tingkat sinyal dengan *noise*, menggunakan fungsi yang ditunjukkan pada gambar berikut:

```
def calculate_snr(clean_signal, noisy_signal):
    min_length = min(len(clean_signal), len(noisy_signal))
    clean_signal, noisy_signal = clean_signal[:min_length], noisy_signal[:min_length]
    noise = noisy_signal - clean_signal
    return 10 * np.log10(np.mean(clean_signal**2) / np.mean(noise**2))
```

Gambar 28. Fungsi perhitungan SNR

Fungsi di atas digunakan untuk menghitung nilai *Signal-to-Noise Ratio* (SNR) dengan membandingkan kekuatan sinyal utama terhadap *noise* pada data audio. Prosesnya dimulai dengan memastikan kedua sinyal, yaitu sinyal asli (*clean signal*) dan sinyal yang sudah bercampur *noise* (*noisy signal*), memiliki panjang yang sama dengan memotong keduanya sesuai panjang minimum. Selanjutnya, *noise* dihitung sebagai selisih antara sinyal bercampur *noise* dan sinyal asli. Dengan data tersebut, fungsi kemudian menghitung nilai SNR yang menunjukkan seberapa dominan sinyal utama dibandingkan *noise*. Hasil akhirnya berupa nilai SNR dalam satuan desibel (dB), yang membantu menilai kualitas sinyal audio secara kuantitatif.

Proses *noise injection* dilakukan dengan beberapa skenario, Gambar 29 menunjukkan empat skenario eksperimen yang dilakukan dengan berbagai pengaturan AF dan jenis *noise*.



Gambar 29. Diagram pengaturan eksperimen *noise injection*: (a) Injeksi lima jenis *noise* pada *dataset* primer atau sekunder untuk menghasilkan *dataset* campuran. (b) Injeksi dengan pengaturan AF pada lima jenis *noise* menghasilkan *dataset* campuran. (c) Pengaturan AF yang bervariasi untuk menghasilkan 10 kategori *noisy dataset* pada setiap jenis *noise* terpilih. (d) Pengaturan AF yang bervariasi untuk menghasilkan 10 kategori *noisy added dataset* pada setiap jenis *noise* tambahan

Gambar 29 merangkum empat pengaturan eksperimen untuk *noise injection*. Masing-masing skenario menguji efek penambahan *noise* dengan *amplitude factor* yang bervariasi pada *dataset* primer dan sekunder, baik dengan *noise* terpilih maupun tambahan, untuk menghasilkan *noisy dataset* dalam berbagai kategori intensitas. Teknik ini digunakan untuk mengukur seberapa baik model dapat bertahan terhadap gangguan *noise* dan bagaimana hal itu memengaruhi performa sistem.

Untuk meningkatkan keragaman data pelatihan dan membuat model lebih *robust*, penelitian ini akan menerapkan berbagai teknik data

augmentation dalam sistem. Teknik-teknik ini meliputi *noise*, *shift*, *pitch*, dan *stretch*. Berikut adalah detail implementasi dari masing-masing teknik *augmentation* yang digunakan:

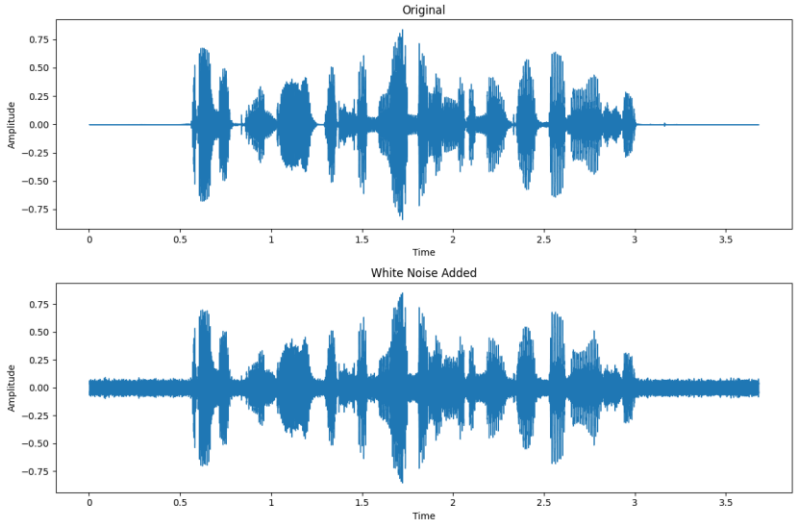
a. *Noise*

Teknik ini melibatkan penambahan *white noise* pada sinyal audio asli selama pelatihan model. Fungsi *noise* yang ditunjukkan pada Gambar 30 menghasilkan *white noise* yang terdistribusi secara normal dengan menyesuaikan amplitudo *white noise* menggunakan *Amplitude Factor* (AF) yang telah ditetapkan berdasarkan penelitian sebelumnya (Mashhadi & Osei-Bonsu, 2023). Nilai AF 0,035 dipilih untuk memastikan bahwa tingkat *white noise* yang ditambahkan tidak terlalu mengganggu sinyal asli, sambil tetap membuat model lebih tangguh dalam menangani variasi *noise* (Burnwal, 2020; Bustamin et al., 2024; Hamid, 2023).

Pada Gambar 30, ditunjukkan contoh *waveform* dari sinyal audio yang telah ditambahkan *white noise*. Perubahan amplitudo acak terlihat di sepanjang sinyal, yang menunjukkan tambahan *noise*. Dengan teknik ini, model dilatih untuk bekerja lebih baik dalam kondisi berisik, yang umum di dunia nyata.

```
1  def noise(data):
2      noise_amp = 0.035*np.random.uniform()*np.amax(data)
3      data = data + noise_amp*np.random.normal(size=data.shape[0])
4      return data
```

Gambar 30. Kode fungsi *noise* dalam *python* untuk data *augmentation* audio



Gambar 31. *Waveform* dari sinyal audio asli (atas) dan sinyal yang telah ditambahkan *white noise* (bawah)

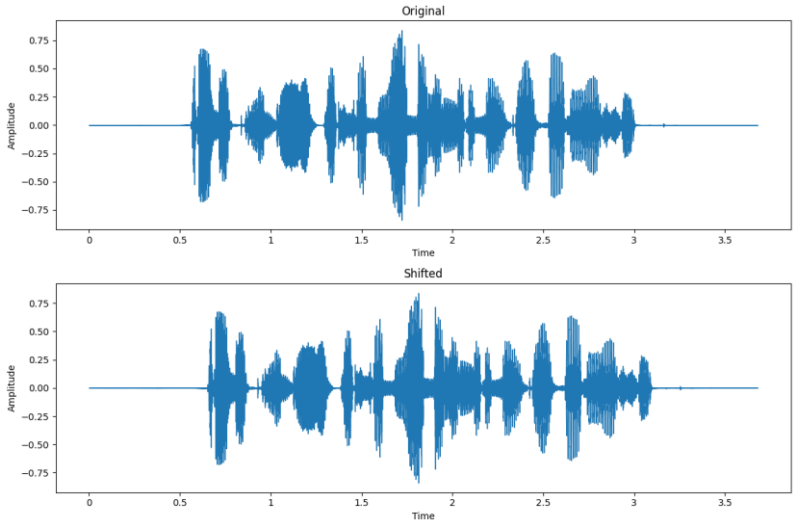
b. Shift

Shift dalam data *augmentation* adalah teknik menggeser sinyal audio dalam domain waktu untuk menyimulasikan variasi pada titik awal dan akhir sinyal audio. Teknik ini bertujuan untuk membuat model lebih tangguh terhadap variasi kecil dalam waktu mulai dan berakhirnya sinyal, yang sering terjadi dalam rekaman audio yang tidak terkontrol. Gambar 32 menunjukkan kode fungsi *shift* yang digunakan dalam data *augmentation* audio. Fungsi ini menggunakan *shift_range*, yang dihitung dengan mengalikan nilai acak dari distribusi uniform dengan faktor 1000 untuk menghasilkan nilai dalam ms. Rentang pergeseran ini didasarkan pada eksperimen (Burnwal, 2020; Bustamin et al., 2024; Hamid, 2023), yang menunjukkan bahwa variasi ini cukup untuk menyimulasikan kondisi nyata tanpa merusak struktur sinyal asli secara signifikan.

Pada Gambar 33, terlihat *waveform* dari sinyal audio yang telah digeser dalam domain waktu. Pergeseran ini membantu melatih model untuk mengenali emosi tanpa tergantung pada posisi temporal tertentu dalam sinyal audio, membuat model lebih fleksibel dalam menghadapi variasi kecil pada waktu rekaman.

```python
def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)
```

Gambar 32. Kode fungsi *shift* dalam *python* untuk data *augmentation* audio



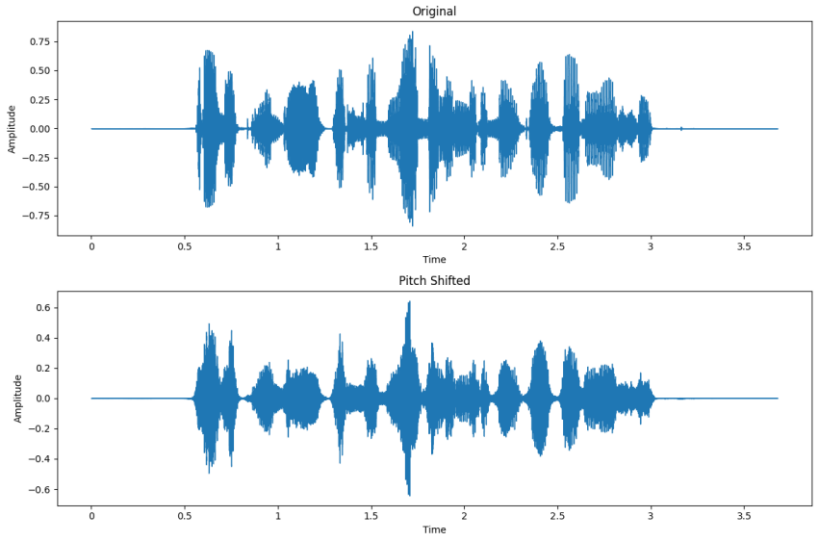
Gambar 33. *Waveform* dari sinyal audio yang telah digeser dalam domain waktu

c. *Pitch*

Pitch dalam data *augmentation* adalah teknik yang digunakan untuk mengubah frekuensi dasar (*pitch*) dari sinyal audio tanpa mengubah durasi sinyal. Teknik ini berguna untuk membuat model lebih tangguh terhadap variasi *pitch*, yang sering ditemukan dalam rekaman vokal atau instrumen. Pada Gambar 34, ditunjukkan kode fungsi *pitch* dalam *Python*, menggunakan pustaka *Librosa.effects.pitch_shift*. Fungsi ini menggeser *pitch* berdasarkan parameter *pitch_factor*, yang dalam kode ini diatur ke 0,7 *semitone*. Pengaturan ini dipilih berdasarkan eksperimen yang dilakukan dalam penelitian sebelumnya (Burnwal, 2020; Bustamin et al., 2024; Hamid, 2023), yang menunjukkan bahwa variasi *pitch* dalam rentang ini dapat meningkatkan ketahanan model terhadap variasi *pitch* alami dalam data audio. *sampling_rate* adalah laju sampel dari sinyal audio asli yang diperlukan oleh fungsi *pitch_shift* untuk melakukan perubahan *pitch* dengan tepat.

```python
def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(y=data, sr=sampling_rate, n_steps=pitch_factor)
```

Gambar 34. Kode fungsi *pitch* dalam *python* untuk data *augmentation* audio, menggunakan *Librosa.effects.pitch_shift* untuk mengubah *pitch* sinyal berdasarkan parameter *pitch_factor*



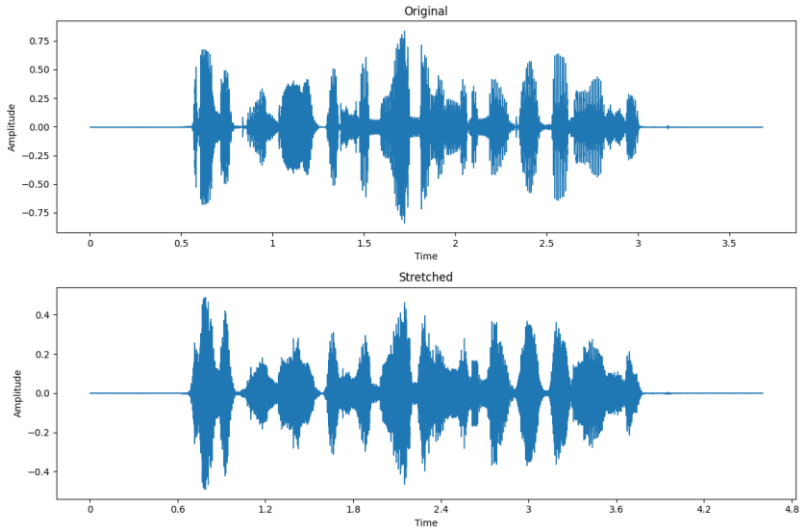
Gambar 35. *Waveform* dari sinyal audio yang telah mengalami perubahan *pitch*

d. *Stretch*

Stretch dalam data *augmentation* adalah teknik yang mengubah durasi sinyal audio tanpa mengubah *pitch*. Teknik ini digunakan untuk membuat model lebih tangguh terhadap variasi kecepatan bicara atau tempo musik dalam data audio. Gambar 36 menunjukkan kode fungsi *stretch* dalam *Python* menggunakan pustaka *Librosa.effects.time_stretch*. Fungsi ini mengubah durasi sinyal audio berdasarkan *rate* yang ditentukan. Dalam kode ini, *rate* diatur ke 0,8, yang berarti durasi sinyal audio diperpanjang sebesar 25%, sementara *pitch*-nya tetap tidak berubah. Pengaturan ini dipilih berdasarkan eksperimen sebelumnya (Burnwal, 2020; Bustamin et al., 2024; Hamid, 2023), yang menunjukkan bahwa variasi kecepatan dalam rentang ini efektif dalam meningkatkan ketahanan model terhadap variasi kecepatan bicara yang alami.

```
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(y=data, rate=rate)
```

Gambar 36. Kode fungsi *stretch* dalam *python* untuk data *augmentation* menggunakan *Librosa.effects.time_stretch*, di mana durasi sinyal diperpanjang berdasarkan parameter *rate*



Gambar 37. *Waveform* dari sinyal audio yang telah diperpanjang tanpa mengubah *pitch*

3. *Feature exraction*

Proses ekstraksi fitur adalah tahap penting dalam mempersiapkan data audio untuk model pembelajaran mesin. Dalam penelitian ini, fitur-fitur yang diekstraksi meliputi *Zero Crossing Rate* (ZCR), *Mel-Frequency Cepstral Coefficients* (MFCCs), *Chroma Short-Time Fourier Transform* (*Chroma* STFT), *Root Mean Square* (RMS), dan *mel spectrogram*. Setiap fitur memiliki peran khusus dalam menangkap karakteristik penting dari sinyal audio yang relevan untuk pengenalan emosi dari suara.

a. *Zero crossing rate*

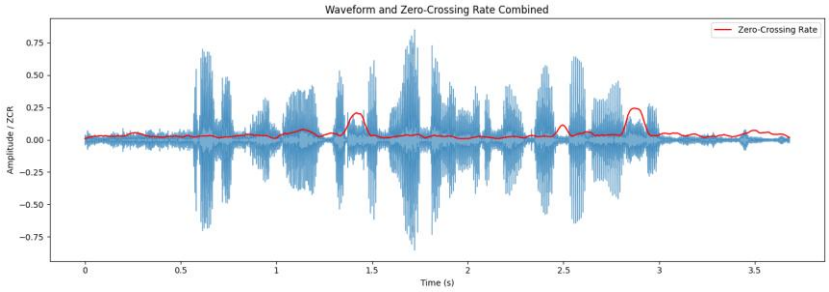
Pada penelitian ini, proses ekstraksi fitur *Zero Crossing Rate* (ZCR) menggunakan *library* Librosa dimulai dengan *preprocessing* sinyal audio, di mana sinyal dibagi menjadi beberapa *frame* kecil dengan panjang 25 ms dan tumpang tindih sebesar 10 ms. Hal ini bertujuan agar analisis dilakukan pada segmen-segmen yang relatif stabil. Setelah sinyal dibagi menjadi *frame-frame*, ZCR dihitung untuk setiap *frame* menggunakan fungsi Librosa*.feature.zero_crossing_rate*. Proses perhitungan ini menghasilkan vektor fitur yang digunakan sebagai

masukan untuk algoritma pembelajaran mesin dalam klasifikasi emosi suara.

Proses ekstraksi ZCR dari setiap *frame* audio ditunjukkan pada Gambar 38, di mana sinyal audio dibagi menjadi *frame-frame* dengan panjang 2048 sampel dan tumpang tindih 512 sampel. Hasil ZCR dari setiap *frame* kemudian disimpan dalam variabel yang akan digunakan untuk proses lebih lanjut. Sementara itu, pada Gambar 39, ditunjukkan hasil visualisasi ZCR yang diterapkan pada sinyal ucapan. Bagian-bagian dengan amplitudo rendah atau fluktuasi tinggi memiliki nilai ZCR yang lebih tinggi, menandakan ketidakstabilan atau *noise*, sedangkan bagian dengan amplitudo tinggi menunjukkan ZCR yang lebih rendah, menandakan suara yang lebih stabil (Sharma et al., 2020).

```python
import librosa

# Memuat sinyal audio
y, sr = librosa.load('path_to_audio_file.wav', sr=None)

# Membagi sinyal audio menjadi frame-frame
frame_length = 2048  # sekitar 25 ms dengan sr = 44100 Hz
hop_length = 512  # sekitar 10 ms

# Menghitung ZCR untuk setiap frame
zcr = librosa.feature.zero_crossing_rate(y, frame_length=frame_length, hop_length=hop_length)

# Menyimpan nilai ZCR
zcr_values = zcr[0]
```

Gambar 38. Kode ekstraksi ZCR menggunakan Librosa dengan *frame* 2048 sampel dan tumpang tindih 512 sampel



Gambar 39. Visualisasi ZCR untuk sinyal ucapan, di mana garis merah menggambarkan nilai ZCR yang lebih tinggi pada bagian sinyal dengan amplitudo rendah atau fluktuasi tinggi

b. *Mel-Frequency Cepstral Coefficients (*MFCCs*)*

Pada penelitian ini, proses ekstraksi fitur *Mel-Frequency Cepstral Coefficients* (MFCCs) dilakukan dengan menggunakan *library* Librosa. Proses ini diawali dengan *pre-emphasis* pada sinyal audio untuk memperkuat komponen frekuensi tinggi. Selanjutnya, sinyal dibagi menjadi *frame-frame* kecil dengan panjang 25 ms dan tumpang tindih 10

ms agar analisis dilakukan pada segmen-segmen yang relatif stasioner. Setiap *frame* kemudian dikenakan fungsi *windowing* untuk mengurangi efek tepi, dilanjutkan dengan penerapan *Discrete Fourier* Transform (DFT) untuk mengubah sinyal dari domain waktu ke domain frekuensi. Hasil dari DFT ini dipetakan ke skala *mel* menggunakan filter bank *mel*, yang kemudian dihitung energinya dan diterapkan logaritma untuk mengubahnya ke skala logaritmik. *Discrete Cosine Transform* (DCT) kemudian diterapkan pada hasil logaritmik tersebut untuk menghasilkan koefisien MFCCs.

Setiap frame menghasilkan vektor fitur MFCCs, yang mewakili karakteristik penting dari sinyal audio, terutama informasi terkait frekuensi yang relevan dengan persepsi manusia. Implementasi kode untuk ekstraksi MFCCs menggunakan Librosa ditunjukkan pada Gambar 40, di mana proses mulai dari pemuatan sinyal hingga perhitungan MFCCs dilakukan. Sementara itu, visualisasi MFCCs yang dihasilkan dari sinyal audio dapat dilihat pada Gambar 12.

```
1
2  import librosa
3
4  # Memuat sinyal audio
5  y, sr = librosa.load('path_to_audio_file.wav', sr=None)
6
7  # Pre-emphasis
8  y_preemphasized = librosa.effects.preemphasis(y)
9
10 # Membagi sinyal audio menjadi frame-frame
11 frame_length = 2048  # sekitar 25 ms dengan sr = 44100 Hz
12 hop_length = 512  # sekitar 10 ms
13
14 # Menghitung MFCC untuk setiap frame
15 mfcc = librosa.feature.mfcc(y=y_preemphasized, sr=sr, n_mfcc=13, hop_length=hop_length, n_fft=frame_length)
16
17 # Menyimpan nilai MFCC
18 mfcc_values = mfcc.T
```

Gambar 40. Kode untuk ekstraksi MFCCs menggunakan Librosa. Sinyal audio diproses mulai dari *pre-emphasis*, pembagian *frame*, hingga perhitungan MFCCs dengan hasil disimpan dalam variabel *mfcc_values*

c. *Mel Spectrogram*

Pada penelitian ini, ekstraksi *mel spectrogram* dilakukan dengan menggunakan *library* Librosa. Proses dimulai dengan memuat sinyal audio, di mana sinyal tersebut dibagi menjadi *frame-frame* dengan panjang 25 ms dan tumpang tindih sebesar 10 ms. Setelah itu, *Short-Time Fourier Transform* (STFT) diterapkan pada setiap *frame* untuk mengubah sinyal dari domain waktu ke domain frekuensi. Hasil dari STFT kemudian dipetakan ke skala *mel* menggunakan filter *mel*-bank. Energi dari filter *mel* ini kemudian diubah ke skala logaritmik untuk menghasilkan *mel spectrogram*. Setiap *frame* menghasilkan vektor fitur yang disimpan untuk analisis lebih lanjut.

Implementasi kode ekstraksi *mel spectrogram* ini ditunjukkan pada Gambar 41, yang menjelaskan bagaimana sinyal audio diolah menggunakan Librosa. Proses dari pemuatan sinyal hingga konversi ke

skala logaritmik dilalui untuk menghasilkan representasi *mel spectrogram*. Contoh visualisasi *mel spectrogram* yang dihasilkan dari sinyal audio dapat dilihat pada Gambar 14, yang menunjukkan bagaimana spektrum frekuensi sinyal terdistribusi dalam skala *mel* pada berbagai segmen waktu.

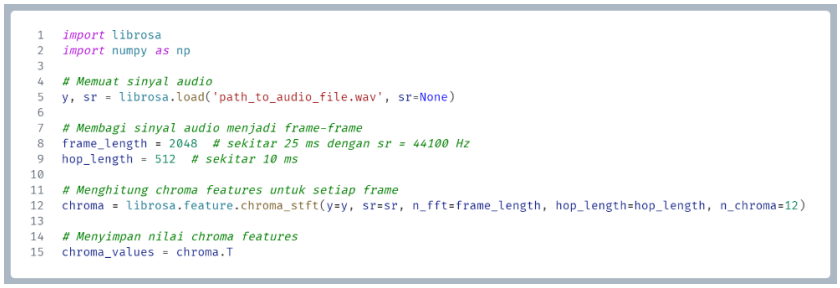
```python
import librosa
import numpy as np

# Memuat sinyal audio
y, sr = librosa.load('path_to_audio_file.wav', sr=None)

# Membagi sinyal audio menjadi frame-frame
frame_length = 2048  # sekitar 25 ms dengan sr = 44100 Hz
hop_length = 512  # sekitar 10 ms

# Menghitung Mel Spectrogram untuk setiap frame
S = librosa.feature.melspectrogram(y=y, sr=sr, n_fft=frame_length, hop_length=hop_length, n_mels=128, fmax=8000)

# Mengonversi Mel Spectrogram ke skala logaritmik (dB)
S_dB = librosa.power_to_db(S, ref=np.max)

# Menyimpan nilai Mel Spectrogram
mel_spectrogram_values = S_dB.T
```

Gambar 41. Kode untuk ekstraksi *mel spectrogram* menggunakan librosa. Sinyal audio diproses melalui STFT, dipetakan ke skala *mel*, dan dikonversi ke skala logaritmik, dengan hasil disimpan dalam variabel *mel_spectrogram_values*

d. *Chroma Short-Time Fourier Transform* (*Chroma* STFT)

Pada penelitian ini, proses ekstraksi fitur *Chroma Short-Time Fourier Transform* (*Chroma* STFT) dilakukan dengan menggunakan *library* Librosa. Proses ini dimulai dengan memuat sinyal audio dan membaginya menjadi beberapa *frame* dengan panjang 25 ms dan tumpang tindih 10 ms. Setelah sinyal terbagi menjadi *frame-frame* kecil, *Short-Time Fourier Transform* (STFT) diterapkan pada setiap frame untuk mengubah sinyal dari domain waktu ke domain frekuensi. Hasil dari STFT kemudian dipetakan ke 12 kelas *chroma* menggunakan fungsi Librosa*.feature.chroma_stft*. Hasil dari ekstraksi ini disimpan dalam bentuk vektor fitur yang akan digunakan untuk analisis lebih lanjut.

Pada Gambar 42, dijelaskan proses ekstraksi *Chroma* STFT menggunakan Librosa, dimulai dari pemuatan sinyal audio hingga penerapan STFT dan pemetaan frekuensi ke 12 kelas *chroma*. Contoh visualisasi dari hasil *Chroma* STFT dapat dilihat pada Gambar 13, di mana distribusi energi dalam kelas *pitch* (nada) terlihat jelas dalam visualisasi.



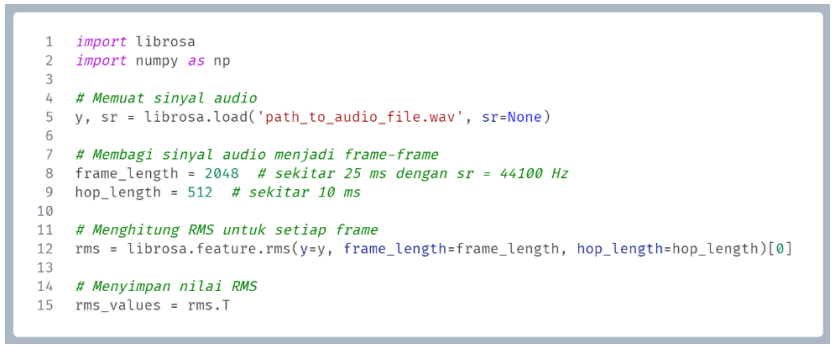
```python
import librosa
import numpy as np

# Memuat sinyal audio
y, sr = librosa.load('path_to_audio_file.wav', sr=None)

# Membagi sinyal audio menjadi frame-frame
frame_length = 2048  # sekitar 25 ms dengan sr = 44100 Hz
hop_length = 512  # sekitar 10 ms

# Menghitung chroma features untuk setiap frame
chroma = librosa.feature.chroma_stft(y=y, sr=sr, n_fft=frame_length, hop_length=hop_length, n_chroma=12)

# Menyimpan nilai chroma features
chroma_values = chroma.T
```

Gambar 42. Kode untuk ekstraksi chroma STFT menggunakan librosa. Sinyal audio diproses menggunakan STFT dan dipetakan ke 12 kelas *chroma*, dengan hasil disimpan dalam variabel *chroma_values*

e. *Root mean square* (RMS)

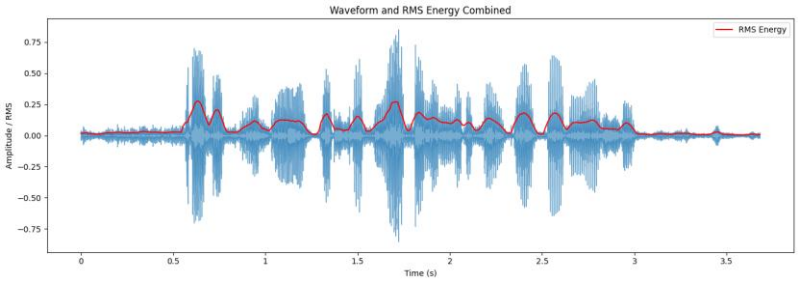
Pada penelitian ini, proses ekstraksi fitur *Root Mean Square* (RMS) dilakukan dengan menggunakan *library* Librosa. Ekstraksi ini dimulai dengan memuat sinyal audio dan membaginya menjadi beberapa *frame* dengan panjang 25 ms dan tumpang tindih 10 ms. Setelah sinyal terbagi menjadi *frame-frame* kecil, RMS dihitung untuk setiap *frame* menggunakan fungsi Librosa*.feature.rms*. Nilai RMS yang diperoleh dari setiap *frame* mewakili rata-rata energi atau intensitas sinyal audio dalam *frame* tersebut. Hasil perhitungan RMS kemudian disimpan sebagai vektor fitur yang akan digunakan untuk analisis lebih lanjut.

Gambar 43 menampilkan kode untuk proses ekstraksi RMS dengan menggunakan Librosa. Proses ini melibatkan pembagian sinyal audio menjadi *frame-frame* kecil, dan penghitungan RMS dilakukan untuk setiap *frame*. Gambar 44 menunjukkan visualisasi kombinasi antara *waveform* sinyal audio dan RMS *energy*, yang menggambarkan bagaimana amplitudo dan energi sinyal bervariasi sepanjang waktu.



```python
import librosa
import numpy as np

# Memuat sinyal audio
y, sr = librosa.load('path_to_audio_file.wav', sr=None)

# Membagi sinyal audio menjadi frame-frame
frame_length = 2048  # sekitar 25 ms dengan sr = 44100 Hz
hop_length = 512  # sekitar 10 ms

# Menghitung RMS untuk setiap frame
rms = librosa.feature.rms(y=y, frame_length=frame_length, hop_length=hop_length)[0]

# Menyimpan nilai RMS
rms_values = rms.T
```

Gambar 43. Kode untuk ekstraksi RMS menggunakan Librosa. Sinyal audio diproses dalam *frame-frame*, dan RMS dihitung untuk setiap *frame*, dengan hasil disimpan dalam variabel *rms_values*



Gambar 44. Visualisasi kombinasi antara *waveform* dan RMS *energy*. Garis merah menggambarkan RMS *energy*, menunjukkan rata-rata energi per *frame* audio

4. Data *splitting*

Dalam penelitian ini, dua skenario utama pembagian data diterapkan untuk melatih dan mengevaluasi model pengenalan emosi dari sinyal suara.

a. Skenario 1: Pembagian Data Dasar

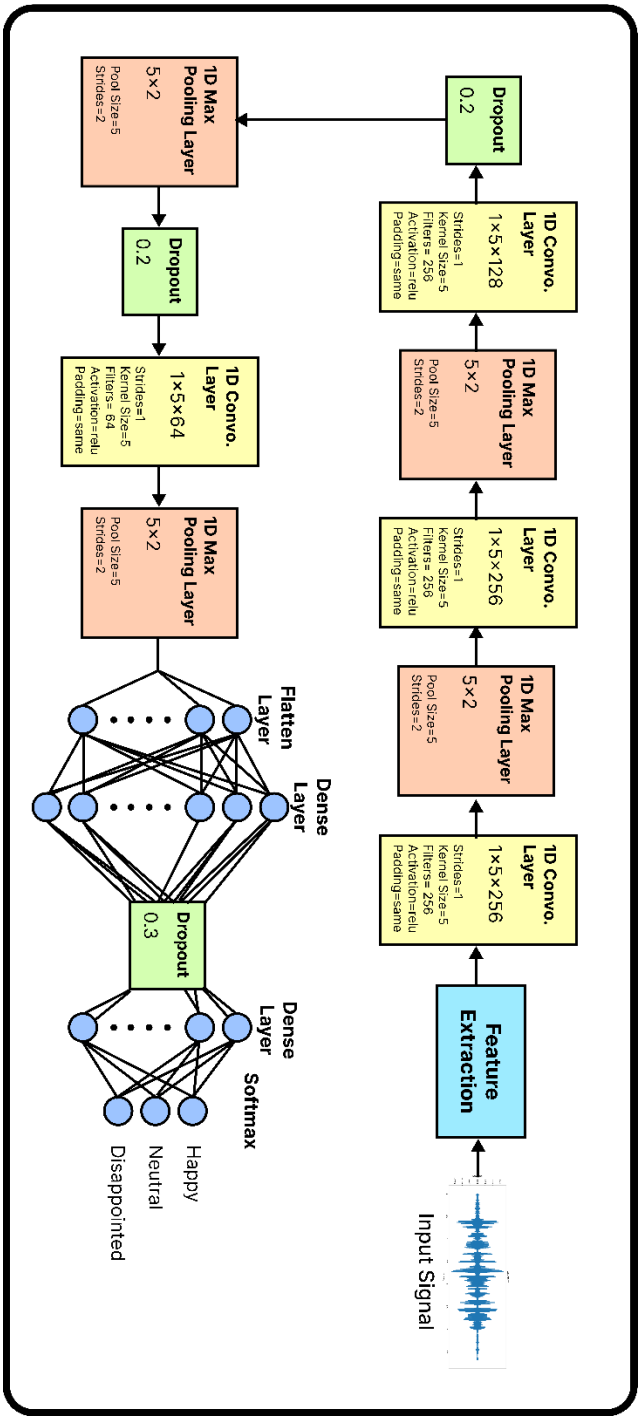
Pada skenario pertama, data audio dibagi menjadi dua *subset* utama: data pelatihan dan data validasi. Pembagian ini bertujuan untuk mengevaluasi performa model menggunakan kombinasi data awal tanpa melibatkan variasi amplitudo yang kompleks. Dalam skenario ini, *dataset* primer, sekunder, serta *noisy* dengan AF 0.7 dibagi dengan proporsi 80% untuk pelatihan dan 20% untuk validasi. Proporsi ini dipilih untuk memaksimalkan jumlah data yang tersedia untuk pelatihan, sambil memastikan bahwa model diuji pada data yang tidak terlihat selama pelatihan untuk mengukur kemampuan generalisasi.

b. Skenario 2: Pembagian Data dengan *Noise Injection*

Skenario kedua melibatkan pembagian data yang lebih kompleks, di mana data gabungan (primer dan sekunder) diinjeksi dengan tiap-tiap *noise* menggunakan berbagai *Amplitude Factor* (AF) (0,1; 0,2; hingga 1,0) mengikuti skenario *noise injection*. Setiap *dataset* hasil injeksi *noise* dipisahkan 15% untuk pengujian.

5. CNN *training process*

Dalam arsitektur jaringan saraf tiruan yang dikenal sebagai *Convolutional Neural Network* (CNN), terdapat serangkaian lapisan yang berbeda-beda yang dikombinasikan dalam keseluruhan arsitektur. Pengaturan CNN memerlukan sejumlah keputusan penting yang berkaitan dengan pola arsitektur seperti jumlah lapisan konvolusi dan *pooling*, format data input, dimensi filter, dan lainnya. Selain itu, bagian terkait *hyperparameter* seperti *learning rate*, probabilitas *dropout*, jumlah *epoch*, ukuran *batch*, dan lain-lain juga sangat krusial untuk pelatihan yang efektif (Das et al., 2020). Dalam penelitian kali arsitektur model yang akan digunakan dalam proses *training* ditunjukkan pada Gambar 45.



Gambar 45. Arsitektur model CNN untuk pengenalan emosi dari sinyal suara.

a. Lapisan Konvolusi

Lapisan konvolusi merupakan inti dari CNN, di mana filter konvolusi diterpakan pada data masukan untuk mengekstraksi fitur-fitur penting. Lapisan ini menggunakan filter yang memiliki ukuran *kernel* dan jumlah yang spesifik untuk memproses data.

- Memilih ukuran *kernel size* 5x1 memungkinkan jaringan untuk menangkap dependensi waktu yang penting dalam sinyal audio, dengan mengambil lima titik data secara *sequential* dalam satu dimensi frekuensi. Ini membantu dalam mendeteksi pola temporal seperti perubahan *pitch* atau ritme dalam suara.
- Jumlah filter (256, 128, 64) yang dimulai dengan 256 filter memungkinkan jaringan untuk menangkap berbagai fitur dari data audio. Penurunan jumlah filter pada lapisan berikutnya membantu dalam mengkondensasi informasi menjadi representasi yang lebih abstrak, yang penting untuk klasifikasi emosi.
- Dengan menggunakan dua lapisan dengan filter yang sama (256), model memperoleh keuntungan dari kapasitas tambahan untuk menangkap fitur kompleks, tetapi tetap menjaga jumlah parameter yang relatif efisien. Menambah lapisan lebih banyak (misalnya dengan mengurangi jumlah filter) bisa membuat model lebih lambat dan lebih rentan terhadap overfitting..
- *Padding* '*Same*' dipilih untuk menjamin bahwa dimensi *output* sama dengan input, memastikan tidak ada informasi yang hilang selama operasi konvolusi.
- Nilai *stride* berjumlah 1 dipilih untuk memastikan bahwa filter bergerak melintasi data masukan satu langkah pada satu waktu yang memungkinkan deteksi fitur yang lebih detail.

b. Fungsi Aktivasi

Fungsi aktivasi yang akan digunakan dalam penelitian ini yaitu ReLU (*Rectified Linear Unit*) yang dipilih karena efisiensinya dalam mempercepat konvergensi pelatihan dan mengurangi masalah *gradient* yang hilang. ReLU efektif karena hanya mengaktifkan neuron saat menerima input positif yang memudahkan optimasi jaringan.

c. Lapisan *Pooling*

Pemilihan ukuran *max pooling* 5x2 didasarkan pada beberapa pertimbangan khusus yang mengutamakan efisiensi dan efektivitas dalam pengolahan fitur temporal. Pertama, ukuran '5' pada dimensi temporal direncanakan untuk menangkap variasi yang lebih luas dalam sinyal suara, seperti perubahan nada atau intonasi, yang penting dalam menentukan emosi. Ukuran ini memungkinkan jaringan untuk mengidentifikasi pola atau perubahan penting sepanjang waktu,

menjadikan model lebih peka terhadap fitur-fitur temporal yang merupakan penanda emosional. Kedua, lebar '2' dalam dimensi kedua mengurangi jumlah komputasi dan dimensi *output* tanpa menghilangkan informasi penting, memberikan keseimbangan antara efisiensi komputasi dan retensi fitur penting. Hal ini membantu dalam menjaga fitur dominan sambil mengurangi risiko *overfitting*, memastikan bahwa jaringan lebih tahan terhadap variasi kecil dalam posisi fitur-fitur penting.

Dalam arsitektur CNN untuk pengenalan emosi dari sinyal suara, nilai *stride*s sebesar 2 pada lapisan *pooling* dipilih untuk efektivitas dalam mengurangi dimensi *output*, mempercepat komputasi, dan mengurangi risiko *overfitting*. Penggunaan *stride*s ini memungkinkan jendela *pooling* bergerak dua unit sekaligus, memotong jumlah operasi *pooling* dan secara efisien mempertahankan hanya fitur yang paling dominan dan relevan. Hal ini juga meminimalkan detail yang berlebihan yang dapat mengganggu model dalam belajar fitur umum dan penting, menjadikan jaringan lebih cepat dan lebih *general* dalam mengenali emosi dari sinyal audio.

d. *Dropout*

Nilai *dropout* sebesar 0.2 dipilih berdasarkan keseimbangan yang ditawarkannya dalam mencegah *overfitting* sambil mempertahankan kemampuan pembelajaran yang efektif. Sebesar 20% dinonaktifkan secara acak pada setiap iterasi pelatihan, yang cukup untuk mengurangi ketergantungan model pada koneksi neuron tertentu dan memaksa model untuk belajar representasi yang lebih *robust*. Nilai ini digunakan karena dinilai optimal untuk memberikan efek regularisasi yang cukup tanpa mengganggu proses belajar dari jaringan yang terlalu banyak, memfasilitasi generalisasi yang baik tanpa kehilangan informasi penting selama pelatihan.

e. *Fully connected Layer*s

Dalam arsitektur CNN yang dirancang untuk pengenalan emosi dari suara, lapisan terhubung penuh atau *fully connected layers* berperan krusial setelah proses *flattening*, yang mengonversi *output* dari lapisan sebelumnya menjadi vektor satu dimensi. Lapisan *dense* pertama menggunakan 32 unit dengan aktivasi ReLU untuk mengintegrasikan *non-linearitas*, yang memungkinkan model menangani hubungan kompleks antar fitur. *Dropout* sebesar 0.3 akan diterapkan untuk mencegah *overfitting*, dengan secara acak menonaktifkan sebagian neuron selama pelatihan, sehingga meningkatkan kemampuan generalisasi model. Lapisan *output* dengan tiga unit (tiga emosi) menggunakan aktivasi *softmax* untuk menghasilkan probabilitas klasifikasi antara tiga kategori emosi. Model ini dikompilasi dengan *optimizer* Adam dan fungsi kerugian *categorical_crossentropy*,

mendukung optimasi yang efisien dan penanganan efektif pada tugas klasifikasi multi-kelas.

6. *Validation* process

Pada tahap pelatihan model, proses validasi akan dilakukan untuk memastikan model dapat menggeneralisasi dengan baik pada data yang tidak terlihat selama pelatihan. Validasi ini akan dilakukan dengan memonitor metrik performa model pada *validation set* di setiap *epoch*. *Validation set* merupakan bagian dari *dataset* yang telah dipisahkan sebelum pelatihan dimulai dan tidak akan digunakan dalam pembaruan bobot selama pelatihan.

Metrik yang akan dipantau meliputi *loss* dan *accuracy*. Metrik *loss* akan digunakan untuk memantau seberapa baik model dalam meminimalkan kesalahan pada data pelatihan dan validasi. *Training loss* menggambarkan seberapa baik model belajar dari data pelatihan, sementara *validation loss* memberikan indikasi kemampuan model untuk menggeneralisasi pada data yang tidak dilihat selama pelatihan. Metrik *accuracy* akan mengukur persentase prediksi yang benar baik pada data pelatihan maupun validasi, dengan *training accuracy* menunjukkan akurasi pada data pelatihan dan *validation accuracy* menunjukkan kinerja pada data validasi.

Selain itu, akan digunakan *callback* ReduceLROnPlateau untuk menyesuaikan *learning rate* secara dinamis berdasarkan performa model. *Callback* ini akan memonitor metrik *loss*, dan jika tidak ada perbaikan dalam nilai *loss* pada *validation set* setelah dua *epoch* berturut-turut, *learning rate* akan dikurangi dengan faktor 0.5. Hal ini bertujuan untuk membantu model menemukan solusi optimal dengan lebih baik saat mendekati konvergensi.

Untuk memahami bagaimana model belajar selama pelatihan, hasil pelatihan akan divisualisasikan melalui grafik *training* & *validation loss* serta *training* & *validation accuracy*. Grafik ini akan membantu dalam mengevaluasi apakah model mengalami *overfitting* atau underfitting.

Setelah proses pelatihan selesai, model akan dievaluasi pada *validation set* untuk menghitung akurasi akhir. Nilai akurasi ini akan memberikan gambaran seberapa baik model telah dilatih untuk mengenali pola-pola dari data yang tidak terlihat selama pelatihan. Grafik yang menunjukkan *training loss* dan *validation loss* serta *training accuracy* dan *validation accuracy* akan dihasilkan untuk setiap *epoch*, yang akan digunakan untuk menganalisis kinerja model selama pelatihan dan untuk memastikan tidak ada tanda-tanda *overfitting*. Selain itu, model akan disimpan dalam format .keras untuk digunakan dalam proses *testing*.

A. Skenario Pelatihan

Pelatihan dilakukan dengan beberapa skenario untuk menguji ketahanan dan konsistensi model dalam berbagai kondisi perlakuan data. Data yang akan digunakan pada skenario ini merupakan data skenario 1 (data dasar) yaitu dengan persentase 80% data untuk proses *training* dan

20% untuk proses validasi. Skenario pelatihan ini dibagi menjadi beberapa kategori untuk melihat performa model dalam berbagai kondisi, yaitu kondisi *clean* atau *dataset* yang bersih dan bebas dari *noise* audio, sedangkan kondisi *noisy* merupakan *dataset* yang telah dilakukan injeksi berbagai jenis *noise* dengan melakukan normalisasi amplitudo dan pengaturan *amplitudo factor* pada *noise* sebesar 0,7 serta penerapan proses *augmentation* pada data atau tidak. Berikut adalah skenario pelatihan awal yang akan digunakan.

A. Skenario Pelatihan Data Primer
   - Data Primer *Clean*
   - Data Primer *Clean with Augmentation*
   - Data Primer *Noisy*
   - Data Primer *Noisy with Augmentation*
   - Data Primer *Clean* & *Noisy*
   - Data Primer *Clean* & *Noisy with Augmentation*

B. Skenario Pelatihan Data Sekunder
   - Data Sekunder *Clean*
   - Data Sekunder *Clean with Augmentation*
   - Data Sekunder *Noisy*
   - Data Sekunder *Noisy with Augmentation*
   - Data Sekunder *Clean* & *Noisy*
   - Data Sekunder *Clean* & *Noisy with Augmentation*

C. Skenario Pelatihan Data Gabungan (Primer dan Sekunder)
   - Data Sekunder *Clean*
   - Data Sekunder *Clean with Augmentation*
   - Data Sekunder *Noisy*
   - Data Sekunder *Noisy with Augmentation*
   - Data Sekunder *Clean* & *Noisy*
   - Data Sekunder *Clean* & *Noisy with Augmentation*

2.5.4 Data *testing*

Setelah model *Convolutional Neural Network* (CNN) selesai dilatih dan divalidasi, langkah berikutnya adalah melakukan pengujian untuk mengevaluasi performa akhir model. Data *testing* yang digunakan terdiri dari 15% dari total *dataset* yang tidak pernah dilihat oleh model selama proses pelatihan dan validasi. Tujuan utama dari pengujian ini adalah untuk memberikan gambaran yang akurat mengenai kemampuan model dalam menggeneralisasi pengetahuan yang telah dipelajari ke data baru yang belum pernah dihadapi sebelumnya. Tahapan yang dilakukan pada proses ini adalah sebagai berikut:

1. Input Data

   Dataset testing mencakup dua jenis data: data bersih (*clean*) dan data berisik (*noisy*) yang telah diinjeksi dengan berbagai tingkat *Amplitude Factor*

(AF). Setiap *dataset* berfungsi untuk menguji kemampuan model dalam mengenali emosi suara di berbagai kondisi lingkungan, termasuk situasi yang realistis di mana sinyal audio mungkin terganggu oleh gangguan akustik.

2. *Feature Extraction*

   Sebelum data *testing* dapat digunakan untuk pengujian model, dilakukan proses ekstraksi fitur dengan metode yang sama seperti pada data pelatihan dan validasi. Fitur-fitur penting seperti *Mel-Frequency Cepstral Coefficients* (MFCCs), *Zero Crossing Rate* (ZCR), *mel spectrogram*, *Chroma* STFT, dan *Root Mean Square* (RMS) diekstraksi dari data *testing*. Ekstraksi fitur ini bertujuan untuk menangkap karakteristik penting dari sinyal audio yang dapat membantu model dalam mengenali emosi dari suara. Proses ini memastikan bahwa data *testing* memiliki representasi yang tepat dan konsisten untuk digunakan dalam klasifikasi oleh model CNN.

3. Klasifikasi Emosi Suara

   Setelah melalui tahap ekstraksi fitur, data *testing* digunakan untuk menguji performa model yang telah dilatih. Pengujian ini bertujuan untuk menilai seberapa baik model dalam mengklasifikasikan emosi suara berdasarkan data *testing* yang telah dipersiapkan. Hasil dari pengujian ini memberikan wawasan mengenai kemampuan model dalam mengidentifikasi berbagai emosi di bawah kondisi yang beragam. Pada proses ini, model yang akan digunakan adalah 2 model utama, yaitu model *Primary Secondary Clean Augmentation* dan *Primary Secondary Clean Noisy Augmentation*

   A. Skenario Pengujian

      Untuk mendapatkan pemahaman yang komprehensif mengenai kinerja model, pengujian dilakukan dalam tiga skenario berbeda:

      1. Performa pada *Dataset* Uji dengan Gabungan Lima Jenis *Noise*

         Pada skenario ini, model yang telah dilatih dengan berbagai jenis data akan diuji dengan *dataset* yang telah diinjeksi dengan gabungan lima jenis *noise* berbeda secara bersamaan. Tujuan dari skenario ini adalah untuk menilai kemampuan model dalam menangani kondisi ekstrem di mana sinyal audio terganggu oleh beberapa jenis *noise* secara simultan. Pengujian ini membantu dalam mengevaluasi ketahanan model dalam situasi akustik yang sangat bising.

      2. Performa pada *Dataset* Uji dengan Tiap-Tiap Jenis *Noise*

         Dalam skenario ini, model diuji dengan *dataset* yang telah diinjeksi dengan satu jenis *noise* pada satu waktu. Setiap jenis *noise* diuji secara terpisah untuk mengukur seberapa baik model dapat beradaptasi dengan gangguan spesifik. Hasil dari skenario ini memberikan wawasan tentang sensitivitas model terhadap masing-masing jenis *noise* dan apakah ada jenis *noise* tertentu yang lebih sulit ditangani oleh model.

      3. Performa pada *Dataset* Uji dengan *Noise* Tambahan

Skenario ini melibatkan pengujian model dengan *dataset* yang diinjeksi dengan *noise* tambahan yang tidak ada pada tahap pelatihan. Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan model dalam menghadapi *noise* baru yang belum pernah ditemui selama pelatihan, sehingga dapat menilai seberapa baik model dapat menggeneralisasi pada situasi yang benar-benar baru.

4. Evaluasi sistem

   Setelah pengujian dilakukan dengan menggunakan data *testing* pada berbagai skenario, hasil dari klasifikasi emosi suara akan dievaluasi. Evaluasi ini dilakukan dengan menggunakan metrik evaluasi yang telah dijelaskan sebelumnya, seperti akurasi, *Weighted Accuracy* (WA), dan *confusion matrix*. Metrik-metrik ini akan memberikan wawasan yang mendalam tentang performa model dan membantu mengidentifikasi area di mana model mungkin memerlukan perbaikan lebih lanjut.