DETEKSI SUARA ASLI DAN PALSU MANUSIA MENGGUNAKAN ARSITEKTUR *HYBRID CONVOLUTIONAL NEURAL NETWORK* (CNN) DAN *LONG SHORT TERM MEMORY* (LSTM)

LAPORAN TUGAS AKHIR

Laporan ini disusun guna memenuhi salah satu syarat untuk menyelesaikan Gelar Sarjana Strata 1 (S1) program studi Teknik Informatika pada Fakultas Teknologi Industri Universitas Islam Sultan Agung

Disusun Oleh :

Nama	: Alim Jatmika
Nim	: 32602100023
Program Studi	: Teknik informatika

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

FINAL PROJECT

DETECTING REAL AND FAKE HUMAN VOICES USING A HYBRID CONVOLUTIONAL NEURAL NETWORK (CNN) AND LONG SHORT TERM MEMORY (LSTM) ARCHITECTURE

Proposed to complete the requirement to obtain a bachelor's degree (S1) at Informatics Engineering Departement of Industrial Technology Faculty Sultan Agung Islamic University

Arranged By :

Alim Jatmika

32602100023

MAJORING OF INFORMATICS ENGINEERING

INDUSTRIAL TECHNOLOGY FACULTY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

2025

LEMBAR PENGESAHAN

TUGAS AKHIR

DETEKSI SUARA ASLI DAN PALSU MANUSIA MENGGUNAKAN ARSITEKTUR *HYBRID CONVOLUTIONAL NEURAL NETWORK* (CNN) DAN *LONG SHORT TERM MEMORY* (LSTM)

ALIM JATMIKA
NIM 32602100023

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : Kamis, 27 November 2025

TIM PENGUJI UJIAN SARJANA:

Ghufron, ST., M.Kom. NIK. 210622056 (Ketua Penguji)	.....	01 - 12 - 2025
Imam Much Ibnu Subroto, ST., M.Sc., Ph.D NIK. 210600017 (Anggota Penguji)	.....	01-12-2025
Sam Farisa Chaerul Haviana, ST., M.Kom. NIK. 210615046 (Pembimbing)	.....	03 - 12 - 2025

Semarang, Rabu - 03 - 12 - 2025

Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT.
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Alim Jatmika

NIM : 32602100023

Judul Tugas Akhir : Deteksi suara asli dan palsu manusia menggunakan Arsitektur Hybrid Convolutional Neural Network dan Long Short Term Memory.

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apbila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 12 / 2025

Yang Menyatakan

Alim Ja

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Alim Jatmika

NIM : 32602100023

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Demak, Jawa tengah

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **DETEKSI SUARA ASLI DAN PALSU MANUSIA MENGGUNAKAN ARSITEKTUR *HYBRID CONVOLUTIONAL NEURAL NETWORK* DAN *LONG SHORT TERM MEMORY***. Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 12 - 2025

Yang menyatakan,

Alim Jatmika

KATA PENGANTAR

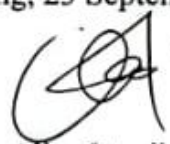
Dengan mengucap syukur allhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "deteksi suara asli dan palsu manusia menggunakan arsitektur *hybrid convolutional neural network* (CNN) dan *long short term memory* (LSTM)" dengan baik.

Tugas akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak berupa materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor Universitas Islam Sultan Agung Semarang Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN Eng
3. Dosen pembimbing penulis Bapak Sam Farisa Chaerul Haviana, ST, M.Kom yang telah meluangkan waktu dan memberi ilmu dalam penyusunan tugas akhir.
4. Orang tua penulis Alm. Bapak Ansori Kasidin dan Ibu Mardianah yang telah memberikan doa serta dukungan baik moral maupun material.
5. Teman seperjuangan yang telah memberikan dukungan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun ilmu pengetahuan dalam penyusunan laporan sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa depan.

Semarang, 23 September 2025



Alim Jatmika

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Kemajuan teknologi kecerdasan buatan, khususnya pada sintesis ucapan seperti *Text-to-Speech* (TTS) dan *voice cloning*, memunculkan ancaman serius berupa *deepfake voice* atau suara palsu yang dapat disalahgunakan untuk penipuan, manipulasi, maupun penyebaran hoaks. Untuk menjawab tantangan tersebut, penelitian ini mengembangkan sistem deteksi suara berbasis arsitektur *hybrid deep learning* dengan mengombinasikan *Convolutional Neural Network* untuk analisis spasial dan *Long Short-Term Memory* untuk pola temporal. Sistem dilengkapi fitur MFCC untuk menangkap ciri fonetik serta *Spectral Contrast* untuk mendeteksi anomali harmonik khas suara sintetis. Model dilatih menggunakan optimizer Adam dan diuji secara menyeluruh, menghasilkan akurasi 86,49%. Untuk kelas asli (0) nilai presisi sebesar 0.81 serta *recall* 0.95 dan *F1-score* sebesar 0.88, kemudian untuk kelas palsu (1) memiliki nilai presisi sebesar 0.94 serta *recall* 0.78 dan *F1-score* sebesar 0.85. Sebagai penerapan praktis, model diintegrasikan ke dalam aplikasi *web* interaktif berbasis *Streamlite* sebagai solusi deteksi suara palsu dalam menghadapi ancaman digital.

Kata kunci : Deteksi Suara Palsu, *Deepfake Audio*, *deep learning*, *Convolutional Neural Network* (CNN), *Long Short-Term Memory* (LSTM), MFCC, *Spectral Contrast*

ABSTRACT

The advancement of artificial intelligence technology, particularly in speech synthesis such as Text-to-Speech (TTS) and voice cloning, has given rise to serious threats in the form of deepfake voices or fake voices that can be misused for fraud, manipulation, or the spread of hoaxes. To address these challenges, this study developed a voice detection system based on a hybrid deep learning architecture by combining Convolutional Neural Network for spatial analysis and Long Short-Term Memory for temporal patterns. The system is equipped with MFCC features to capture phonetic characteristics and Spectral Contrast to detect harmonic anomalies typical of synthetic voices. The model, drilled using the Adam optimizer and thoroughly tested, produced an accuracy of 86.49%. For the original class (0), the precision value was 0.81, recall 0.95, and F1-score 0.88. Then for the fake class (1), the precision value was 0.94, recall 0.78, and F1-score 0.85. As a practical application, the model was integrated into a Streamlit-based interactive web application as a fake voice detection solution in dealing with digital threats.

Keywords : *Fake Voice Detection, Deepfake Audio, Hybrid Deep Learning, Convolutional Neural Network* (CNN*) Long Short-Term Memory* (LSTM)*,* MFCC*, Spectral Contrast*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi kecerdasan buatan saat ini, terutama di bidang pengolahan suara dan pembuatan suara, sudah berkembang sangat pesat. Teknologi seperti Text-to-Speech (TTS), kloning suara, dan konversi suara (VC) kini bisa menciptakan suara buatan yang sangat mirip dengan suara manusia. Salah satu dampak yang muncul karena perkembangan ini adalah munculnya *deepfake voice* atau suara palsu rekaman suara digital yang dibuat untuk menyerupai suara seseorang dengan tingkat kesamaan yang sangat tinggi. (Qazi dan Kaushik, 2020).

Walaupun memiliki potensi dalam bidang positif seperti media, hiburan, dan aksesibilitas, teknologi ini juga membawa risiko besar. *Deepfake voice* dapat digunakan untuk penipuan berbasis suara (*vishing*), manipulasi percakapan, penyebaran hoaks, hingga pencemaran nama baik dan pelanggaran privasi. Bahkan, rekaman suara yang telah direkayasa ini dapat digunakan sebagai bukti palsu dalam investigasi hukum atau kejahatan dunia maya. Oleh karena itu, kebutuhan akan sistem yang dapat mendeteksi suara palsu menjadi sangat penting.

Convolutional Neural Network (CNN) telah banyak dimanfaatkan karena kemampuannya dalam mengekstraksi pola spasial dari spektrum suara seperti MFCC (N.A.Bhaskaran dkk, 2024). Meskipun metode CNN efektif mengekstraksi fitur spasial, data suara juga memiliki dimensi temporal yang penting untuk diperhatikan. Untuk menangkap pola berurutan dalam sinyal suara, digunakanlah *Long Short-Term Memory* (LSTM) yang dirancang untuk mengenali hubungan jangka panjang dalam data sekuensial. Kombinasi CNN dan LSTM membentuk arsitektur *hybrid* CNN-LSTM yang memungkinkan sistem memanfaatkan kekuatan keduanya secara sinergis. CNN menangkap fitur spasial, sementara LSTM mempelajari dinamika temporal dari suara (Qazi dan Kaushik, 2020).

Salah satu karakteristik yang paling banyak dimanfaatkan dalam analisis suara adalah *Mel-Frequency Cepstral Coefficients* (MFCC), yang bekerja dengan mengekstraksi informasi spektral dari sinyal audio berdasarkan skala mel, yaitu

skala frekuensi non-linear yang meniru cara manusia memersepsikan suara. Fitur ini efektif dalam menangkap ciri fonetik dan vokal dari suara, sehingga banyak digunakan dalam pengenalan ucapan dan verifikasi *speaker*. Namun, dalam konteks deteksi suara palsu, MFCC memiliki keterbatasan karena hanya merepresentasikan bentuk spektral secara global dan kurang sensitif terhadap pola harmonik kompleks serta ketidakteraturan energi yang sering muncul dalam suara sintetis (Logan dan Engineers, 2020). Untuk mengatasi hal ini, fitur tambahan seperti *Spectral Contrast* digunakan, yaitu fitur yang mengukur selisih energi antara puncak dan lembah dalam spektrum suara pada beberapa *band* frekuensi untuk menggambarkan tekstur spektral secara lebih rinci. Dengan demikian, *Spectral Contrast* menjadi indikator kuat dalam mengidentifikasi kejanggalan tersebut. Ketika MFCC dan *Spectral Contrast* digabungkan sebagai input pada model *deep learning* seperti CNN-LSTM, sistem mampu menghasilkan representasi yang lebih kaya untuk mendeteksi suara palsu.

Dengan mempertimbangkan latar belakang yang telah diuraikan, penelitian ini memiliki tujuan untuk mengembangkan sistem deteksi suara palsu dengan menggunakan *model Hybrid* CNN-LSTM, serta memanfaatkan kombinasi fitur MFCC dan *Spectral Contrast* sebagai *input* utama. Diharapkan pendekatan ini mampu meningkatkan akurasi dalam mendeteksi suara yang telah dimanipulasi secara digital.

1.2 Rumusan Masalah

Bagaimana merancang dan mengimplementasikan *model deep learning* berbasis kombinasi *Convolutional Neural Network* (CNN) dan *Long Short Term Memoryi* (LSTM) untuk mendeteksi suara palsu manusia secara efektif dan akurat, serta mengoptimalkan kinerja *model* melalui teknik pelatihan dan evaluasi yang tepat.

1.3 Pembatasan Masalah

Batasan dalam penelitian ini ditentukan sebagai berikut:

1. Fokus penelitian ini hanya pada pengembangan *model deep learning* yang menggabungkan CNN untuk mengekstrak fitur spasial dan LSTM untuk menganalisis data waktu dalam mendeteksi suara palsu.
2. Teknik yang digunakan untuk mengekstrak fitur audio adalah MFCC dan *Spectral Contrast*.
3. Data set yang digunakan hanya terdiri dari dua kategori, yaitu suara asli dan suara palsu, serta semua suara menggunakan bahasa *Inggris*.
4. Hasil evaluasi model dinilai berdasarkan beberapa metrik, yaitu akurasi, *presisi*, *recall*, dan *F1-score*.

1.4 Tujuan Penelitian

Membuat Aplikasi antarmuka dan menguji model pembelajaran mendalam yang menggabungkan *Convolution Neural Network* (CNN) dan *Long Short Term Memory* (LSTM) untuk mendeteksi suara asli dan palsu dari manusia.

1.5 Manfaat

Ada beberapa manfaat yang diharapkan dalam penelitian ini sebagai berikut:

1. Memberikan solusi untuk mendeteksi suara palsu yang dapat digunakan pada bidang keamanan digital.
2. Menjadi referensi bagi peneliti atau pengembang lain dalam topik deteksi *deepfake voice*.
3. Menunjukkan efektivitas kombinasi CNN dan LSTM dengan fitur MFCC dan *Spectral Contrast* pada pengolahan suara.
4. Memudahkan pengguna umum dalam mengenali suara asli dan palsu melalui aplikasi berbasis *web*.

1.6 Sistematika Penulisan

BAB I : PENDAHULUAN

Bab ini berisi uraian mengenai latar belakang yang mendasari pemilihan judul penelitian, pembahasan masalah yang menjelaskan pokok permasalahan, batasan masalah sebagai ruang lingkup penelitian, Tujuan yang hendak diraih, keuntungan dari hasil studi, dan juga urutan penulisan tugas akhir.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian penelitian terdahulu serta landasan teori yang digunakan penulis untuk mendukung analisis permasalahan terkait deteksi suara palsu dengan CNN dan LSTM, sekaligus menjadi referensi dalam penyusunan tugas akhir.

BAB III : METODE PENELITIAN

Bab ini menjelaskan tahapan penelitian yang dilakukan, mulai dari proses pengumpulan data hingga tahap pengolahan data.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Dalam bab ini dipaparkan hasil penelitian yang mencakup tahapan akuisisi data, pengolahan data, hingga visualisasi sistem deteksi yang telah dibangun

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian serta saran yang diberikan penulis untuk pengembangan lebih lanjut.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Perkembangan teknologi kecerdasan buatan (AI) pembelajaran mendalam telah mendorong lahirnya *deepfake voice*, yaitu suara buatan yang dihasilkan secara digital dengan tingkat kemiripan sangat tinggi terhadap suara manusia. Teknologi ini mampu menirukan cara berbicara seseorang, mulai dari nada, intonasi, hingga ekspresi emosional, sehingga sering kali sulit dibedakan hanya dengan pendengaran. Keberadaan suara palsu semacam ini menimbulkan risiko serius, karena dapat dimanfaatkan untuk penipuan, manipulasi percakapan, bahkan sampai pada penyebaran informasi yang tidak benar. Dengan demikian, dibutuhkan metode deteksi yang memiliki tingkat efektivitas, akurasi, dan efisiensi tinggi untuk mengidentifikasi perbedaan antara suara asli dan suara hasil rekayasa.

Penelitian berjudul *Detecting Deep Fake Voice using Machine Learning* dengan memanfaatkan *dataset* yang terdiri dari suara asli dan palsu, yang dipotong menjadi segmen *audio* berdurasi dua detik dan diekstraksi fiturnya menggunakan *Mel-Frequency Cepstral Coefficients* (MFCC). Penelitian ini menguji delapan algoritma klasifikasi, termasuk MLP, SVM, *Random Forest*, dan *Gradient Boosting*, dengan hasil menunjukkan bahwa *MLPClassifier* memberikan performa terbaik dengan akurasi hingga 88%. MFCC terbukti efektif dalam membedakan suara asli dan palsu berdasarkan pola akustik, dan peneliti merekomendasikan pengembangan lebih lanjut untuk mengatasi gangguan suara serta mengimplementasikan sistem deteksi di perangkat *mobile*. Studi ini memberikan kontribusi penting dalam pengembangan sistem deteksi *deepfake audio* berbasis pembelajaran mesin (N.A.Bhaskaran dkk, 2024).

Berdasarkan penelitian yang berjudul *Hybrid Feature Optimization for Voice Spoof Detection Using CNN-LSTM* Dalam Penelitian mengembangkan sistem verifikasi pembicara otomatis (ASV) yang lebih akurat. Mereka mengusulkan model *ensemble* berbasis Pendekatan *hybrid* yang mengombinasikan CNN dan LSTM dengan memanfaatkan ekstraksi fitur MFCC dan CQCC berhasil diuji pada

dataset ASVspoof 2019. Dari hasil pengujian tersebut, model mencapai akurasi maksimal sebesar 80,48%. mengungguli metode berbasis fitur tunggal. Kombinasi CNN untuk ekstraksi pola spasial dan LSTM untuk pola temporal terbukti efektif dalam meningkatkan ketahanan sistem terhadap berbagai jenis serangan seperti seperti *voice Conversion*, *speech synthesis*, dan *replay attack*. Temuan ini menegaskan bahwa pendekatan *hybrid* dalam ekstraksi fitur dan arsitektur CNN dan LSTM dapat memperkuat sistem ASV dalam menghadapi tantangan *spoofing* yang semakin kompleks (Neelima & Prabha, 2024).

Penelitian yang dilakukan oleh (Aini dkk, 2021) dengan judul Studi tentang pemodelan CNN pada deteksi emosi dalam bahasa Indonesia yang berfokus pada pengembangan sistem *Speech Emotion Recognition* (SER) dengan menggunakan CNN, yang akan menerima input dari fitur suara berupa MFCC, *pitch*, dan RMSE.Dataset diambil dari serial TV "*Imperfect*" dengan empat label emosi: marah, senang, netral, dan sedih. Hasil pengujian menunjukkan bahwa kombinasi MFCC dan *pitch* memberikan akurasi tertinggi sebesar 98%, sedangkan penggunaan MFCC saja memberikan akurasi 83%, menunjukkan bahwa MFCC tidak cukup kuat sebagai fitur tunggal. Sebaliknya, penambahan fitur RMSE justru menurunkan akurasi hingga 72%, karena energi suara yang diukur oleh RMSE tidak selalu mencerminkan perbedaan emosi. Penelitian ini membuktikan bahwa MFCC adalah fitur dominan yang efektif dalam mendeteksi emosi dalam suara bahasa Indonesia, baik digunakan sendiri maupun dikombinasikan dengan *pitch* (Aini dkk, 2021).

Penelitian yang berjudul Penerapan Metode MFCC dan LSTM untuk *Speech Emotion Recognition* yang dilakukan oleh (Dewa Agung Adwitya Prawangsa dan Eka Karyawati, 2024) mengembangkan sistem *Speech Emotion Recognition* menggunakan fitur MFCC dan model LSTM. Data set yang digunakan adalah TESS dengan 2800 *file audio* dari tujuh kategori emosi. *Audio* diproses selama 3 detik dan diekstraksi menjadi 40 koefisien MFCC sebagai *input*. Arsitektur LSTM terdiri dari lapisan *Dense*, *Dropout*, serta aktivasi *ReLU* dan *Softmax*. Hasil terbaik diperoleh pada konfigurasi *learning rate* 0.1, *batch size* 64, dan 50 *epoch* dengan akurasi validasi 72,32%. Penggunaan *learning rate* lebih kecil justru menurunkan

akurasi. Meski cukup efektif, model masih menghadapi tantangan *overfitting* dan ketidakseimbangan data (Dewa Agung Adwitya Prawangsa dan Eka Karyawati, 2024).

Penelitian yang dilakukan oleh (Galih Ajinurseto dkk, 2023) yang membahas tentang penggunaan *Mel-Frequency Cepstral Coefficients* (MFCC) untuk aplikasi verifikasi suara. Penelitian ini membangun sistem pengenalan arti bayi menangis menggunakan fitur MFCC dan model CNN berdasarkan *dataset* publik dari *Kaggle* yang terdiri dari lima kelas emosi suara bayi. Untuk menyeimbangkan jumlah data, dilakukan augmentasi sebelum pelatihan model CNN selama 50 *epoch* dengan skema Pembagian data dilakukan 80% untuk latih dan 20% untuk uji. Akurasi pelatihan tertinggi adalah 93,84%, rata-rata 88,04%, dan akurasi uji 86%. Kelas lapa*r* mencatat *F1-score* terbaik sebesar 0,92. sementara *burping* dan *tired* menunjukkan performa rendah akibat ketidakseimbangan data. Hasil menunjukkan bahwa kombinasi MFCC dan CNN efektif dalam mengklasifikasikan tangisan bayi, namun peningkatan kualitas dan keseimbangan data diperlukan (Galih Ajinurseto dkk, 2023).

Penelitian yang dilakukan oleh (Aljufri dan Prasetio, 2022) dengan judul Sistem Deteksi Tingkat Stres Menggunakan Suara dengan Metode Jaringan Saraf Tiruan kemudian Ekstraksi Fitur menggunakan MFCC berbasis *Raspberry Pi*. mengembangkan sistem deteksi stres melalui analisis suara dengan ekstraksi fitur MFCC serta klasifikasi menggunakan *Artificial Neural Network* (ANN) pada perangkat *Raspberry Pi* 4. Sistem ini mampu mengklasifikasikan suara ke dalam tiga tingkat stres: tinggi, rendah, dan netral, menggunakan dataset SUSAS yang terdiri dari 1860 sampel suara. Model ANN dengan 7 lapisan dan *input* 40 koefisien MFCC dilatih di laptop dan menghasilkan akurasi pelatihan 88%, validasi 72%, dan pengujian 76%. Saat diuji dengan 30 data acak, sistem mencatat akurasi hingga 90%, dengan waktu komputasi 2–4 detik. Hasilnya menunjukkan bahwa MFCC dan ANN efektif untuk deteksi stres berbasis suara yang dapat diterapkan secara *portabel* (Aljufri dan Prasetio, 2022).

Pada penelitian berjudul "Identifikasi Fitur Suara Menggunakan Model *Convolutional Neural Network* (CNN) pada *Speech-to-Text* (STT)", pada Penelitian

(Susetianingtias & Patriya, 2024) yang ini mengimplementasikan metode CNN dan LSTM digunakan dalam mendeteksi berita palsu yang ditulis dalam bahasa Indonesia menggunakan dataset *dari TurnBackHoax.id* yang berisi 1786 berita (802 fakta dan 984 palsu). Setelah melalui *preprocessing* dan *word embedding* dengan *Word2Vec*, data dilatih dengan CNN dan LSTM. Hasil evaluasi menunjukkan bahwa CNN memiliki akurasi pengujian tertinggi sebesar 88%, *precision* dan *recall* 0.88, sedangkan LSTM mencapai akurasi 84%. CNN juga berhasil memprediksi semua data uji baru dengan benar, sementara LSTM *mel*akukan satu kesalahan. Dengan demikian, CNN dinilai lebih unggul dalam klasifikasi berita palsu, dan pengembangan lebih lanjut disarankan menggunakan kombinasi CNN-LSTM integrasi sistem ke platform berbasis *web* (Susetianingtias & Patriya, 2024).

Penelitian yang dilakukan oleh (Swastika dkk, 2023) dengan judul Studi perbandingan tingkat akurasi dalam mendeteksi emosi berdasarkan suara dengan memanfaatkan algoritma *Multilayer Perceptron, Random Forest, Decision Tree*, dan *K-Nearest Neighbor*. melakukan perbandingan tingkat akurasi dari empat metode klasifikasi yang terdiri dari *Multilayer Perceptron* (MLP), *Decision Tree, Random Forest,* dan *K-Nearest Neighbors*. (K-NN) dalam mendeteksi emosi suara menggunakan data set *RAVDESS* yang terdiri dari 672 *file audio* dengan empat kelas emosi: senang, sedih, netral, dan marah. Fitur suara diekstraksi menggunakan MFCC, *Chroma*, dan *Mel Spectrogram,* lalu diuji pada tiga skenario pembagian data (85:15, 80:20, dan 75:25). Hasil studi mengungkapkan bahwa algoritma Random Forest menunjukkan kinerja paling optimal dengan tingkat akurasi maksimum yang mencapai 79% pada skenario 80:20 dan akurasi rata-rata 75,5% (Swastika dkk, 2023).

Penelitian yang dilakukan oleh (Lim & Chae, 2022) dengan judul Mendeteksi Suara Palsu Menggunakan Teknik Pembelajaran Mendalam yang Terjelaskan mengusulkan pendekatan untuk mendeteksi suara *deepfake* dengan menggabungkan beberapa model pembelajaran mendalam, yaitu CNN, CNN-LSTM, dan CNN-LSTM permutasi. Penelitian ini menggunakan data set *ASVspoof* 2019 *Logical Access* dan *LJSpeech.*dengan *input* berupa *mel-spectrogram* dari suara asli dan suara palsu hasil sintesis dari berbagai model *Text-to-Speech* (TTS)

dan *Voice Conversion* (VC). Model CNN dan LSTM permutasi menunjukkan performa terbaik dengan akurasi mencapai 99,97% pada data set *LJSpeech*. Selain deteksi yang sangat akurat, metode XAI memungkinkan interpretasi *visual* dan *audio* terhadap keputusan model, seperti mengungkap perbedaan nada, ritme, dan format antara suara asli dan palsu. Dengan mengonversi skor interpretasi kembali ke bentuk suara menggunakan algoritma *Griffin–Lim*, penelitian ini memberikan kontribusi penting dalam menciptakan sistem deteksi suara *deepfake* yang akurat, transparan, dan dapat dipercaya (Lim & Chae, 2022).

Pada penelitian berjudul *Deepfake Audio Detection via MFCC Features Using Machine Learning* yang dilakukan oleh (Hamza dan Javed, 2022) mengusulkan metode deteksi *deepfake audio* menggunakan fitur MFCC dan berbagai algoritma *machine learning*. Menggunakan dataset *Fake-or-Real* yang terdiri dari lebih dari 195.000 sampel suara asli dan palsu, fitur MFCC diekstraksi dan disederhanakan dengan PCA sebelum diuji pada beberapa model, termasuk SVM, *Random Forest*, MLP, *XGBoost*, dan VGG-16. Hasilnya menunjukkan bahwa SVM menghasilkan akurasi tertinggi sebesar 90,83% pada *subset for-rerec,* sedangkan VGG-16 unggul pada *subset for-original* dengan akurasi 83%, melampaui performa LSTM. *Model* juga menunjukkan ketahanan terhadap *noise* dengan tetap mempertahankan akurasi tinggi. Studi ini membuktikan bahwa MFCC merupakan fitur yang sangat efektif dalam membedakan suara asli dan palsu, serta bahwa kombinasi dengan model seperti SVM dan VGG-16 memberikan performa deteksi yang optimal pada berbagai kondisi Data (Hamza dan Javed, 2022).

2.2 Dasar Teori

2.2.1 Suara Asli

Suara palsu adalah sinyal ujaran yang dihasilkan atau dimodifikasi menggunakan teknologi suara sintetis atau teknik manipulasi audio dengan tujuan menyerupai suara manusia asli. Dalam konteks keamanan biometrik, suara palsu dikenal sebagai serangan *spoofing* , yang bertujuan mengecoh sistem pengenalan suara agar menerima suara tiruan sebagai suara asli. Terdapat berbagai bentuk suara palsu, seperti *text-to-speech* (TTS) yang mengubah teks menjadi suara palsu, *voice*

Conversion (VC) yang mengubah suara seorang pembicara menjadi suara pembicara lain, *replay Attack* yang menggunakan rekaman suara untuk disiarkan ulang, hingga audio *deepfake* yang memanfaatkan model berbasis GAN atau difusi untuk menciptakan suara buatan berkualitas tinggi. Meskipun teknologi modern dapat menghasilkan suara yang sangat mirip dengan suara manusia, sinyal palsu tetap meninggalkan pola akustik yang tidak alami seperti harmonik yang terlalu stabil, *noise* digital, transisi fonem yang tidak halus, dan struktur energi spektral yang tidak sepenuhnya mencerminkan pola fisiologis manusia. Penelitian *ASVspoof* menunjukkan bahwa sinyal dari serangan *spoofing* memiliki karakteristik spektral dan temporal yang berbeda dari suara asli, sehingga dapat dideteksi melalui fitur akustik seperti MFCC, CQCC, kemudian *Spectral Contrast* dan diklasifikasikan menggunakan model pembelajaran mesin (Todisco dkk, 2017).

2.2.2 Suara palsu

Suara palsu adalah rekaman suara yang dihasilkan oleh kecerdasan buatan (AI) untuk meniru suara manusia secara meyakinkan. Teknologi ini menggunakan teknik seperti *Text-to-Speech* (TTS) dan *voice Conversion* untuk menghasilkan ucapan yang menyerupai suara individu tertentu, meskipun tidak pernah diucapkan oleh orang tersebut. Awalnya dikembangkan untuk aplikasi positif seperti pembuatan *audiobook* dan asisten digital, teknologi ini kini juga digunakan dalam konteks negatif, seperti penipuan *voice phishing* dan penyebaran informasi palsu (Hamza & Javed, 2022). Suara asli merupakan sinyal ujaran yang dihasilkan secara langsung melalui mekanisme fisiologis manusia tanpa campur tangan proses buatan atau manipulasi digital. Produksi suara dimulai dari aliran udara yang dikeluarkan melalui sistem respirasi, kemudian menghasilkan getaran pada pita suara dan dilanjutkan melalui struktur *artikulatoris* seperti lidah, bibir, dan rongga mulut. Proses ini menciptakan pola akustik alami yang mengandung harmoni, resonansi, serta variabilitas prosodik yang konsisten dengan karakteristik individu. Secara akustik, suara asli memiliki distribusi energi spektral yang mengikuti pola resonansi biologis, variabilitas *pitch* yang terjadi secara spontan, dan mikro-modulasi yang menunjukkan dinamika fisiologis yang sulit ditiru oleh sistem sintesis. Selain itu, suara asli membawa kebisingan alami seperti suara pernapasan dan turbulensi udara

yang memperkuat keotentikannya. Berbagai penelitian menjelaskan bahwa struktur sinyal suara asli menunjukkan stabilitas pola statistik sehingga cocok digunakan sebagai identitas biometrik dalam sistem pengenalan suara (Yi dkk, 2023)

2.2.3 *Deepfake*

Deepfake adalah teknik manipulasi media digital yang menggunakan algoritma pembelajaran mendalam untuk menghasilkan konten palsu tampak nyata, terutama pada suara manusia. Teknologi ini memanfaatkan model seperti *Text to Spech* dan *voice cloning* untuk menciptakan suara sintetis yang sulit dibedakan dari suara asli. Meskipun memiliki aplikasi positif dalam bidang hiburan dan pendidikan, *deepfake* juga menimbulkan kekhawatiran terkait penyebaran informasi palsu dan pelanggaran privasi (Deressa dkk, 2024).

Deepfake merupakan teknologi manipulasi media berbasis kecerdasan buatan yang menggunakan model *deep learning* untuk menghasilkan atau mengubah konten visual maupun audio sehingga menyerupai data asli dengan tingkat realisme yang sangat tinggi. Istilah *deepfake* merupakan kombinasi dari *deep learning* dan pemalsuan, yang merujuk pada teknik penipuan yang dilakukan menggunakan jaringan syaraf dalam jaringan saraf terdalam. Teknologi ini memanfaatkan model *generatif* seperti *Generative Adversarial Networks* (*GANs*) dan *autoencoder* , yang bekerja dengan cara mempelajari pola kompleks dari data asli dan kemudian merekonstruksi konten tiruan berdasarkan pola tersebut.

1. *Deepfake video*

   Dalam ranah *video*, *deepfake* banyak digunakan untuk penempatan wajah (*face swapping*) dan manipulasi ekspresi wajah (*face reenactment*). Model seperti *Autoencoder, StyleGAN*, dan GAN varian terbaru memungkinkan pembuatan wajah sintetis dengan detail yang hampir tidak dapat dibedakan dari wajah aslinya. Di sisi lain,

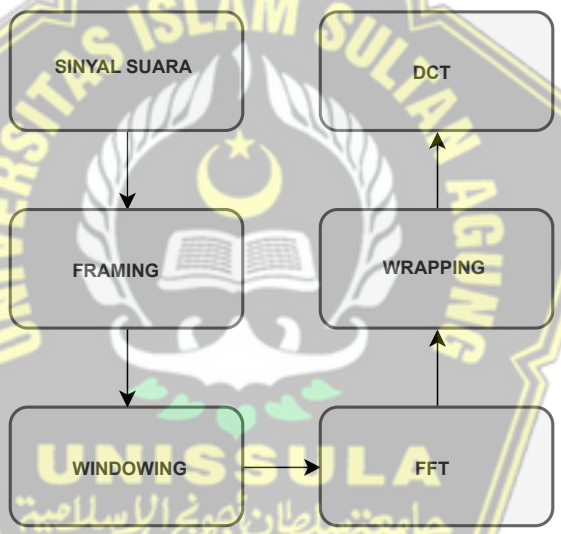
2. *Deepfake Audio*

   pada ranah *audio*, *deepfake* diwujudkan melalui sistem sintesis suara dan konversi suara yang mampu meniru karakter suara seseorang dengan sangat presisi, termasuk intonasi, ritme, dan kualitas vokal. *Model* kemajuan seperti

WaveNet, Tacotron 2, *VITS*, *HiFi-GAN* telah mendorong peningkatan kualitas audio sintetis sehingga semakin sulit dibedakan dari suara manusia asli.

2.2.4 *Mel-Frequency Cepstral Coefficients* (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) merupakah representasi numerik dari spektrum suara yang dirancang untuk meniru persepsi pendengaran manusia. MFCC banyak digunakan dalam pengenalan ucapan, identifikasi pembicara karena kemampuannya dalam mengekstraksi fitur-fitur penting dari sinyal *audio* (Ali dkk. 2020). Dari berbagai metode ekstraksi fitur yang tersedia, MFCC menjadi sangat terkenal untuk aplikasi indentifikasi jenis suara. Ketenaran fitur ini juga disebabkan oleh adanya metode penghitungan yang efektif yang menjadikannya tangguh terhadap perbedaan suara. (Sahidullah dan Saha, 2012).



Gambar 2. 1 diagram alur proses ekstraksi MFCC (Sahidullah dan Saha, 2012)

Pada gambar 2.1 di atas adalah diagram alur proses ekstraksi fitur suara menggunakan *Mel-Frequency Cepstral Coefficients* (MFCC). MFCC sering digunakan dalam pemrosesan sinyal suara karena kemampuannya meniru cara manusia mendengar dan mengambil informasi fonetik dari suara (Ali dkk, 2020). Proses ini terdiri dari beberapa tahap utama, yang secara berurutan dijelaskan sebagai berikut:

1. *Framing*

   Framing merupakan tahap awal dalam perhitungan MFCC, di mana sinyal suara dibagi ke dalam potongan durasi suara serta nilai sampel yang

setara. Tahap tersebut akan berulang-ulang hingga semua sinyal selesai dianalisis. Umumnya, setiap *frame* dibuat saling tumpang tindih atau (*overlapping*) dengan panjang *overlap* sekitar 30%–50% dari ukuran *frame*. Teknik ini bertujuan untuk menjaga agar ciri atau karakteristik suara pada bagian perbatasan antar *frame* tidak hilang.

2. *Windowing*

Setelah proses *framing*, selanjutnya dilakukan langkah *windowing* dengan mengalikan masing-masing *frame* menggunakan jendela Hamming. Proses ini bertujuan untuk menjaga kelancaran sinyal pada sisi awal dan akhir *frame* agar tidak terjadi diskontinuitas. Prosedur ini bertujuan untuk mengurangi ketidaklancaran sinyal saat dimulainya dan diakhiri setiap bingkai. Jendela *Hamming*, w(n) yang diterapkan dalam MFCC (Yusdiantoro dan Sasongko, 2023).

3. *Fast Fourier Transform* (FFT)

Fast Fourier Transform (FFT) digunakan untuk mengubah *frame* yang telah diproses *windowing* dari bentuk waktu ke bentuk frekuensi. Proses transformasi ini sangat penting karena karakteristik sinyal menjadi lebih terlihat jelas di domain frekuensi.

4. Skala *mel*

Dalam sistem pendengaran manusia, suara tidak dipersepsikan secara linier. Oleh karena itu, Pada frekuensi nada untuk beberapa karakteristik akan di analisis menggunakan satuan khusus yaitu skala *mel* . Skala ini bersifat linier di bawah frekuensi 1000 Hz, kemudian untuk pola atas yaitu 1000 berubah menjadi satuan *mel logaritmik* (Yusdiantoro dan Sasongko, 2023).

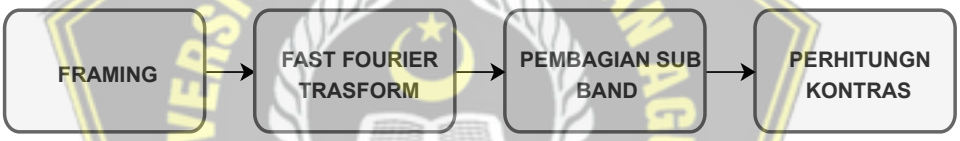
5. *Discrete Cosine Transform* DCT

Koefisien MFCC hanya dapat diamati dalam format vektor setelah menerapkan perhitungan *Discrete Cosine Transform* (DCT). DCT adalah tahap terakhir dalam proses utama pengambilan ciri MFCC. Prinsip fundamental DCT adalah untuk menghilangkan korelasi antara mel spektrum kemudian akan menghasilkan bentuk numerik yang konsisten dari spektrum suara. Secara dasar, mekanisme DCT mirip dengan TFI. Akan tetapi, hasil

DCT mirip dengan hasil dari PCA (*Principal Component Analysis*). Itulah sebabnya DCT sering kali menggantikan transformasi *Fourier invers* dalam proses ekstraksi ciri MFCC (Yusdiantoro & Sasongko, 2023).

2.2.5 Spectral Contrast

Spectral Contrast merupakan suatu metode fitur suara yang berasal dari sinyal suara berfungsi untuk mengukur perbedaan logaritmik antara puncak (*peak*) dan lembah (*valley*) dalam spektrum frekuensi yang dibagi ke dalam beberapa *sub-band*. Berbeda dengan MFCC yang merangkum energi rata-rata dalam band frekuensi, *Spectral Contrast* mempertimbangkan penyebaran energi, sehingga memberikan informasi yang lebih kaya mengenai struktur harmonik dari suara. Fitur ini banyak digunakan dalam klasifikasi suara, pengenalan emosi, hingga deteksi suara palsu karena kemampuannya dalam membedakan tekstur spektral dari sinyal asli dan sinyal hasil rekayasa(Kumar dan Thiruvenkadam, 2021)



Gambar 2. 2 Diagram blok (Kumar dan Thiruvenkadam, 2021)

Gambar 2.2 menunjukkan proses ekstraksi *Spectral Contrast* dari sinyal *audio*. Proses ini terdiri dari tahapan-tahapan utama seperti *framing*, *transformasi Fourier*, pembagian *sub-band* frekuensi, dan perhitungan kontras spektral antara puncak dan lembah dalam masing-masing *band*. Penelitian oleh (Kumar dan Thiruvenkadam, 2021) menunjukkan bahwa *Spectral Contrast* secara signifikan dapat meningkatkan performa sistem deteksi emosi suara ketika digabungkan dengan MFCC, karena fitur ini menangkap variasi frekuensi lokal yang tidak terlihat oleh MFCC saja.

1. *Framing*

   Sinyal suara dibagi menjadi segmen-segmen kecil yang disebut *frame*, dengan durasi sekitar 20–30 milidetik dan tumpang tindih (*overlap*) sekitar 50% antar *frame*. Tujuan dari proses ini adalah untuk menjadikan sinyal *audio* sebagai sinyal stasioner secara lokal, sehingga analisis frekuensi dapat dilakukan lebih akurat pada setiap *frame* (Kumar & Thiruvenkadam, 2021).

2. *Fast Fourier Transform*

   Setelah proses *framing*, setiap *frame* diubah mulai bentuk waktu menuju bentuk pola frekuensi dengan proses *Fast Fourier Transform*. *Transformasi* ini menghasilkan spektrum magnitudo, yang merupakan representasi dari sebaran energi frekuensi dalam satu *frame*. Spektrum ini kemudian digunakan untuk tahap pembagian *sub-band* dan penghitungan kontras spektral (Kumar & Thiruvenkadam, 2021).

3. Pembagian *Sub-band* Frekuensi

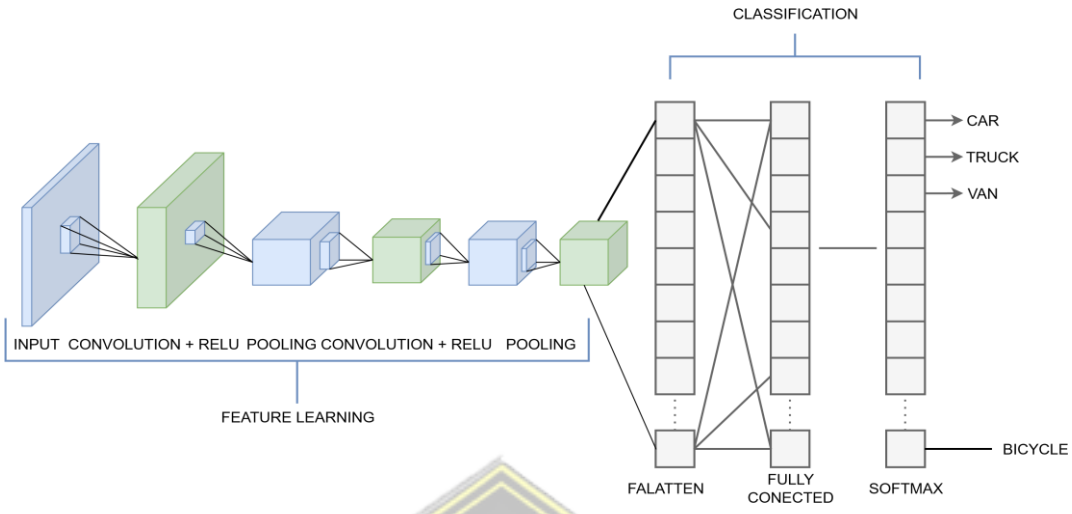
   Spektrum frekuensi dibagi ke dalam beberapa *sub-band*, biasanya antara 6 hingga 7 *band*, menggunakan skala logaritmik seperti skala *Mel* atau skala oktaf. Masing-masing *sub-band* merepresentasikan rentang frekuensi tertentu yang mencerminkan karakteristik harmonik atau *noise* dari sinyal *audio* (Kumar & Thiruvenkadam, 2021).

4. Perhitungan *Spectral Contrats*

   Dalam setiap *sub-band*, dihitung nilai puncak (*peak*) dan nilai lembah (*valley*), yang biasanya diambil berdasarkan kuantil atas dan bawah (misalnya kuantil 90% dan 10%). Kontras spektral untuk setiap *sub-band* kemudian dihitung menggunakan rumus (Kumar & Thiruvenkadam, 2021).

2.2.6 *Convolutional Neural Network*

Convolutional Neural Network adalah suatu jenis jaringan saraf buatan di gunakan untuk menangani data berdimensi dua seperti citra atau representasi *visual* dari sinyal *audio*. Dalam konteks deteksi suara palsu, CNN digunakan untuk menganalisis spektrum suara, seperti MFCC yang diperoleh dari transformasi sinyal suara satu dimensi menjadi bentuk *visual* dua dimensi



Gambar 2. 3 Struktur CNN (Neelima & Prabha, 2024)

Berdasarkan gambar 2.3 di atas, Secara umum struktur CNN terbagi menjadi dua bagian utama *feature extraction* dan *classification*. CNN bekerja dengan cara melakukan proses *input* data dengan sejumlah *kernel* yang digunakan dengan filter, yang memungkinkan jaringan mengenali pola-pola spesifik pada data. Dalam deteksi suara palsu, pola-pola ini dapat berupa ketidakwajaran atau distorsi frekuensi yang biasanya muncul pada suara sintetis atau hasil rekayasa *audio* (Albadawy dkk. 2020). Setelah melewati *layer* konvolusi, data biasanya diproses melalui *ReLU* (*Rectified Linear Unit*) sebagai fungsi aktivasi, diikuti dengan lapisan *pooling* untuk memperkecil data dalam suatu dimensi dan menjaga data inputan yang paling dominan. Bentuk Arsitektur ini umumnya di bagi menjadi beberapa layer inti yaitu:

1. Layer Konvolusi (*Convolutional Layer*)

   Layer ini merupakan komponen inti dalam arsitektur CNN yang memiliki fungsi utama untuk mengekstraksi ciri-ciri lokal dari data *input*, seperti pola frekuensi dan karakteristik harmonik pada representasi spektral *audio*. Proses ini dilakukan melalui penerapan filter (*kernel*) yang bergerak melintasi *input* untuk menghasilkan *feature map* berisi informasi signifikan dari sinyal. Dalam aplikasi deteksi suara palsu, lapisan konvolusi dapat membantu mengenali perbedaan kecil yang membedakan antara suara asli dan suara hasil sintesis (Purwins dkk, 2019).

Operasi ini melibatkan penggunaan matriks kecil yang disebut *Filter* atau *Kernel* yang digeser *slide* melintasi citra *input*. Pada setiap posisi, *filter* dikalikan dengan bagian *input* yang ditutupinya perkalian elemen demi elemen, dan hasilnya dijumlahkan menjadi satu nilai. Nilai ini kemudian membentuk Peta Fitur *Feature Map*.

$$Y(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m,\ j+n)\ K(m,n) \quad (1)$$

- X : data inputan
- K : *filter*/*kernel* (misal 3x3 atau 5x5)
- Y(i,j) : hasil konvolusi di posis (i,j)
- M,n : posisi di dalam *filter*

Pada operasi konvolusi, bagian kecil dari citra *input* diambil sesuai ukuran *filter*, misalnya input berukuran 10×10 dengan *filter* 3×3 berarti sistem mengambil *patch* 3×3 yang dimulai dari posisi (i, j). Setiap elemen dalam *patch* tersebut kemudian dikalikan dengan elemen *filter* pada posisi yang sama, misalnya $X_{11} \times K_{11}$, $X_{12} \times K_{12}$, dan seterusnya hingga semua pasang elemen selesai dikalikan. Seluruh hasil perkalian itu kemudian dijumlahkan totalnya menjadi *output* konvolusi di titik Y(i, j). Fungsi Aktivasi (*ReLU*) Fungsi ini memotong nilai negatif dan mempertahankan nilai positif, sehingga jaringan menjadi lebih stabil dan mampu belajar pola non-linear (Simonyan dan Zisserman, 2015).

2. Lapisan *Pooling* (*Pooling Layer*)

Pooling berfungsi untuk menyederhanakan data hasil konvolusi dengan cara mereduksi dimensi tidak menghapus data penting. Proses ini bertujuan untuk meminimalisir kompleksitas komputasi serta membantu model agar tidak mengalami *overfitting*. Selain itu, *pooling* memperkuat kemampuan model dalam mengenali fitur yang dominan dan membuat sistem lebih stabil terhadap variasi kecil dalam sinyal suara. *Layer* ini biasanya diletakkan setelah *Layer* Konvolusi dan *ReLU*. Tujuannya adalah untuk mengurangi dimensi spasial *downsampling* dari peta fitur, mengurangi jumlah parameter, dan

membuat *model* lebih resisten terhadap pergeseran kecil *translation invariance*. Jenis *pooling* yang paling populer adalah *MaxPooling*. Ini melibatkan pengambilan nilai tertinggi pada area kecil (misalnya, 2x2 *piksel*) dan membuang nilai lainnya(Hershey dkk, 2017).

Rumus *max polling*

$$\text{Output Ukuran } O_p = \left\lfloor \frac{W - K}{S} \right\rfloor + 1 \quad (1)$$

- W : Ukuran *input*.
- K : Ukuran *pool* (*pool size*).
- S : *Stride*.

3. Lapisan *Flatten* dan *Fully Connected*

Setelah tahap ekstraksi dan reduksi fitur, CNN mengubah struktur data dua dimensi menjadi vektor satu dimensi melalui proses *flattening*. Vektor ini kemudian dikirim ke lapisan *fully connected*, yaitu jaringan yang menghubungkan seluruh *neuron* secara langsung. Di sini, sistem memproses informasi yang telah dikumpulkan dan memulai tahap klasifikasi berdasarkan fitur yang telah dipelajari(Purwins dkk, 2019).

Proses *flattening* mengubah peta fitur tiga dimensi yang terdiri atas tinggi, lebar, dan jumlah *channel* menjadi sebuah vektor satu dimensi sehingga dapat diproses oleh lapisan berikutnya. Vektor hasil *flattening* kemudian dimasukkan ke dalam lapisan *fully connected*, yaitu jaringan saraf tiruan di yang mana semua *neuron* terkoneksi dengan seluruh neuron pada lapisan sebelumnya untuk mempelajari hubungan *antarfitur* secara menyeluruh. Pada tahap akhir, layer *output* berfungsi fungsi untuk aktivasi *Softmax* yang memperoleh probabilitas pada semua kelas, sehingga model dapat menentukan kelas mana yang paling sesuai dengan pola yang telah dipelajari.

Rumus *shofmax*

$$P(\text{kelas } k) = \frac{e^{z_k}}{\sum_{i=1}^{N} e^{z_i}} \quad (2)$$

- N : Jumlah total kelas.
- e : Basis logaritma natural.

Fungsi ini mengubah nilai-nilai bebas (*logits*) menjadi distribusi probabilitas, memastikan bahwa probabilitas untuk semua kelas adalah 1(He dkk, 2016).

4. Lapisan *Output* (*Output Layer*)

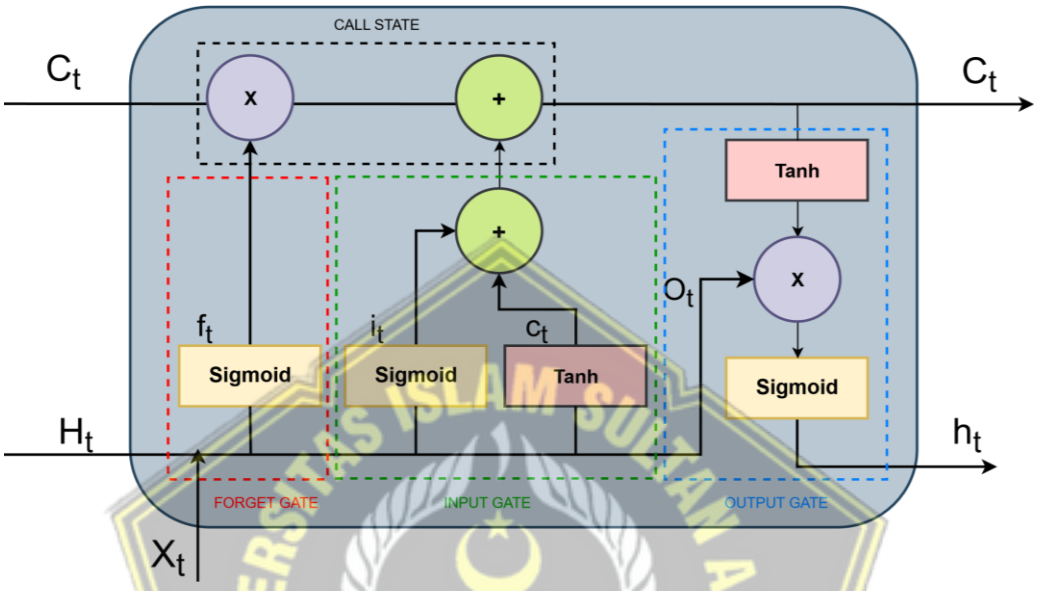
   Lapisan akhir dalam struktur CNN adalah *output layer*, yang berfungsi untuk menghasilkan nilai prediksi berupa probabilitas. Dalam kasus klasifikasi biner seperti deteksi suara asli dan palsu, biasanya digunakan fungsi aktivasi *sigmoid* untuk menentukan seberapa besar kemungkinan suatu *input* termasuk ke dalam satu kelas. Hasil dari lapisan ini menjadi dasar pengambilan keputusan akhir oleh model (Zhang dkk, 2021).

Dalam konteks deteksi suara palsu (*fake audio detection*), CNN digunakan untuk menganalisis representasi *visual* dari sinyal *audio*, seperti MFCC mengenali pola-pola unik dari suara asli maupun suara yang dimanipulasi seperti dari *Text-to-Speech* atau *voice cloning*. *Output* dari CNN kemudian dapat dihubungkan ke *layer* lain seperti LSTM atau *Dense* (*Fully Connected Layer*) untuk proses klasifikasi akhir. CNN tidak hanya efektif dalam menangkap ciri-ciri lokal dari sinyal, tetapi juga mampu menyesuaikan terhadap variasi *input* tanpa memerlukan proses ekstraksi fitur manual (Mais dkk, 2015).

2.2.7 *Long Short-Term Memory*

LSTM promosikan (Hochreiter dan Schimidhuber, 2016) dalam makalah mereka yang berjudul *"Long Short-Term Memory"* yang dipublikasikan di jurnal *Neural Computation* . Arsitektur LSTM dirancang untuk memungkinkan jaringan saraf mengingat informasi dalam jangka waktu yang panjang, menjadikannya sangat efektif dalam menangani data sekuensial seperti teks, *audio*, dan deret waktu. Secara struktural, LSTM terdiri dari sel memori *memory cell* yang memiliki mekanisme internal berupa beberapa gerbang yang terdiri dari *forget gate , input* gate, kemudian *output gate*. Ketiga gerbang ini berfungsi mengatur aliran informasi masuk dan keluar dalam unit LSTM, sehingga jaringan dapat mempertahankan informasi penting dari langkah-langkah sebelumnya sekaligus membuang informasi yang tidak relevan (Waqas dan Humphries, 2024). *Input* gate mengatur

seberapa banyak informasi yang akan disimpan menuju ke penyimpanan, *forget gate* memutuskan bagian mana dari informasi lama yang harus dilupakan, dan *output* gate menentukan informasi mana yang akan diteruskan ke langkah berikutnya dalam urutan(Liu, 2024).



Gambar 2. 4 Struktur LSTM (Liu, 2024)

Pada Gambar 2.4 menampilkan Struktur Arsitektur internal unit LSTM, yang terdiri dari beberapa kotak utama yang mewakili *cell state*, serta beberapa pintu inti yang mengelilinginya: *Input Gate* (hijau), *Output Gate* (biru) serta *Forget Gate* (warna merah). Setiap gerbang terdiri dari *layer sigmoid* untuk menghasilkan filter [0,1], serta operasi perkalian (x) atau penjumlahan (+) pada jalur *cell state.*

1. *Memori cell*

   Sel memori adalah "pita konveyor" yang berjalan di sepanjang rantai unit LSTM. Ia membawa informasi yang relevan dari awal urutan ke langkah waktu saat ini (τ). Informasi yang mengalir di sel ini hanya mengalami interaksi linear dan non-linear minimal, sehingga mudah bagi informasi penting untuk tetap utuh.

2. Gerbang (*Gates*)

   Gerbang adalah inti dari kecerdasan LSTM. Setiap gerbang adalah lapisan Jaringan Saraf Tiruan dengan fungsi aktivasi *Sigmoid* (σ) dan operasi perkalian elemen demi elemen. Karena *Sigmoid* menghasilkan *output* antara 0

dan 1, gerbang ini berfungsi sebagai *filter* atau *regulator* aliran data: Nilai 1 berarti biarkan semua informasi ini lewat dan Nilai 0 berarti blokir semua informasi ini.

a. *Forget Gate*

Tugas pertama LSTM adalah memutuskan informasi apa yang harus dibuang atau dilupakan dari *Cell State* sebelumnya *Input Gate. Gate* ini menentukan seberapa banyak informasi baru dari yang ditambahkan ke memori (Greff dkk, 2017).

Rumus Forget Gate :

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

- W_f : Matriks bobot untuk Gerbang Lupa.
- b_f : Vektor bias untuk Gerbang Lupa.
- $[h_t - 1, x_t]$: Gabungan *concatenation* saat ini.
- σ : Fungsi *Sigmoid*. *Output* f_t adalah vektor nilai antara 0 dan 1.

b. *Input gate*

Langkah berikutnya akan memilih data terbaru apa yang perlu dimasukan kedalam *cellstate* C_t . Proses ini melibatkan dua komponen utama. Pertama, gerbang *input* berfungsi mengatur bagian mana dari informasi baru yang akan diperbarui, dengan nilai yang berkisar antara 0 hingga menjadi 1 untuk menunjukkan seberapa besar informasi tersebut diperbolehkan masuk. Kedua, kandidat *cell state* dibentuk sebagai vektor baru yang mewakili informasi potensial yang dapat ditambahkan ke memori, di mana fungsi aktivasi tanh digunakan untuk menghasilkan nilai antara –1 dan 1. Kedua komponen ini bekerja bersama memilih konten terbaru apa yang perlu disimpan dalam *cell state* pada langkah waktu tersebut(Greff dkk, 2017)

Rumus *input gate* :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

Rumus kandidat *cellstate* :

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

- W_i, b_i, W_C, b_C : Bobot dan bias yang akan dipelajari untuk komponen-komponen ini.
- Tanh : Fungsi tangen hiperbolik.

c. *Output gate*

Langkah terakhir adalah memutuskan *output* yang akan dikeluarkan oleh unit LSTM pada langkah waktu ini, yaitu *Hidden State*. *Hidden* State ini didasarkan pada *Cell State* yang baru diperbarui.

Rumus Gerbang *Output* :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

Rumus *Hidden State* Baru :

$$h_t = o_t \odot \tanh(C_t) \quad (5)$$

- W_o, b_o : Bobot dan bias untuk Gerbang Output.

Hidden State $h_t$ini kemudian digunakan sebagai *output model* untuk langkah waktu (misalnya, untuk membuat prediksi kata berikutnya) dan juga diteruskan ke unit LSTM berikutnya pada langkah waktu $t + 1$. Kombinasi dari *forget* dan *input gate* ini memperbarui memori sel bagian lama diingat, dan bagian baru ditambahkan(Greff dkk, 2017).

Keunggulan utama dari LSTM terletak pada kemampuannya untuk mengingat konteks jangka panjang dalam urutan data. Misalnya, dalam konteks pemrosesan suara manusia, LSTM dapat mengenali perubahan intonasi, durasi, atau struktur ritmik dari suatu ujaran yang mungkin tidak dapat di deteksi dengan model konvensional. Karena itu, LSTM sangat cocok digunakan dalam berbagai tugas seperti pengenalan suara (*speech recognition*), deteksi emosi dari *audio*, pemrosesan bahasa alami (*natural language processing*), dan dalam konteks penelitian ini deteksi suara palsu. Secara keseluruhan, LSTM merupakan salah satu arsitektur *deep learning* yang paling banyak digunakan dan terbukti efektif dalam mendeteksi pola jangka panjang dalam data sekuensial. Dalam konteks deteksi

suara palsu, LSTM memberikan kontribusi signifikan dalam mengenali keanehan temporal atau ketidakwajaran sekuensial pada *audio* sintetis yang tidak mudah terdeteksi oleh manusia maupun sistem sederhana (Liu, 2024).

BAB III
METODELOGI PENENLITIAN

3.1 Deskrisi Sistem

Penelitian ini akan menerapkan metode *deep learning* dengan pendekatan Pendekatan gabungan antara *Convolutional Neural Network* dan Long *Short Term Memory*. dalam merancang aplikasi deteksi suara asli dan suara palsu. Sistem dikembangkan menggunakan bahasa pemrograman *Python* serta didukung oleh beberapa perpustakaan populer di bidang pemrosesan audio dan pembelajaran mesin, seperti *Librosa*, *TensorFlow* dan *Keras*. Data yang digunakan terdiri dari rekaman suara asli dan suara hasil manipulasi (sintetik), yang terlebih dahulu diproses melalui ekstraksi *Mel-Frequency Cepstral Coefisien* (MFCC) dan *Spectral Contrast* agar ciri-ciri penting dari sinyal suara dapat diperoleh secara maksimal.

Penggunaan *Hybrid* CNN dan LSTM dipilih karena CNN efektif dalam untuk mengenali pola lokal pada data audio, sedangkan LSTM berperan dalam menangkap hubungan dalam urutan sinyal suara. Kombinasi kedua metode ini menjadikan sistem lebih akurat dalam membedakan suara asli dan suara palsu. Hasil akhir dari sistem berupa klasifikasi audio ke dalam kategori asli atau palsu, yang diharapkan dapat dimanfaatkan untuk mendeteksi teknologi Suara asli maupun palsu.

3.2 Studi Literatur

Studi literatur pada penelitian ini dilakukan dengan mengkaji berbagai referensi dari jurnal, tesis, artikel ilmiah, dan *e-book* yang fokus pada pemrosesan sinyal suara, deteksi audio deepfake, serta penerapan metode *deep learning* dalam klasifikasi audio. Kajian tersebut membahas teknik ekstraksi ciri seperti MFCC dan *Spectral Contrast* yang efektif dalam merepresentasikan karakteristik akustik, serta penggunaan CNN untuk menangkap pola spasial dan LSTM untuk menganalisis hubungan temporal. Studi literatur ini menjadi dasar teoritis dan metodologis dalam membangun sistem deteksi suara asli dan palsu dengan arsitektur *Hybrid* CNN dan LSTM .

3.3 Rancangan Alur Model

Pada tahap desain alur kerja model yang dikembangkan, proses tersebut disajikan dalam bentuk bagan alir yang ditampilkan pada Gambar.

Gambar 3. 1 *Flowchart* Rancangan Alur Model

pada Gambar 3.1 menggambarkan tahapan perancangan sistem deteksi suara asli dan palsu yang memanfaatkan kombinasi CNN dan LSTM sebagai model.

3.4 Tahapan perancangan model

3.4.1 Pengumpulan Data

Data set yang digunakan dalam studi ini terbagi menjadi dua jenis *audio*, yakni suara otentik yang tidak dimodifikasi serta suara buatan yang dihasilkan melalui aplikasi *text-to-speech* dan *cloning* suara. Data set tersebut diperoleh dari sumber terbuka seperti *Kaggle*, yang menyediakan koleksi *audio* orisinal dan *audio* yang telah disintesis. Keseluruhan data set dipecah menjadi tiga sub set utama, yaitu data pelatihan, data validasi, dan data pengujian. Pemisahan ini dilakukan untuk memastikan bahwa proses pengembangan model berlangsung dengan cara yang terstruktur dan menyeluruh. Data set disimpan dalam folder terpisah dan diunggah ke *Google Drive* untuk memudahkan integrasi dengan *Google Colab* selama pelatihan dan evaluasi model. Proposisi pembagian data set dibuat seimbang antara suara otentik dan buatan untuk memastikan akurasi serta kemampuan generalisasi yang baik dari model.

Tabel 3. 1 pembagian data set

data	Suara asli	Suara palsu
Data Training	6400	6400
Data Validasi	1600	1600
Data uji	37	37

Distribusi *dataset* yang terdiri atas suara asli dan suara palsu pada tiga bagian utama training, validasi, dan uji menunjukkan bahwa penelitian ini menggunakan pendekatan *balanced dataset* yang sangat penting untuk menjaga objektivitas model selama proses pembelajaran. Pada tahap pelatihan, masing-masing kelas memiliki 6400 sampel sehingga total data mencapai 12.800 sampel, jumlah ini memberikan ruang yang luas bagi model CNN-LSTM untuk mempelajari berbagai variasi pola spektral dan temporal dari suara asli serta karakteristik artefak yang muncul pada suara palsu. Selanjutnya, data validasi yang berjumlah 1600 sampel per kelas digunakan untuk memonitor performa model dan memastikan tidak terjadi *overfitting*,jadi total dari jumlah data set adalah 16.000 data set yang berisi suara asli dan palsu. Sementara itu, data uji yang berjumlah 37 sampel untuk masing-

masing kelas berfungsi sebagai bahan evaluasi akhir untuk mengukur kemampuan generalisasi model dalam mengenali suara yang belum pernah dilihat sebelumnya.

3.4.2 *Preprocessing* Data

Preprocessing bertujuan untuk membersihkan dan menyeragamkan data *audio* sebelum diproses lebih lanjut, khususnya sebelum dilakukan ekstraksi fitur menggunakan MFCC. Langkah-langkah *preprocessing* dalam penelitian ini meliputi:

1. Konversi Format & *Resampling*

   Semua *file audio* dikonversi ke format *wav* dengan *sample rate* yang seragam, seperti 16 kHz, serta diubah ke mode mono. Hal ini dilakukan agar data *audio* memiliki format yang konsisten dan kompatibel dengan proses ekstraksi fitur dan pelatihan *model*.

2. Normalisasi Amplitudo

   Setiap *file audio* dinormalisasi agar memiliki tingkat amplitudo yang seragam. Tujuannya adalah untuk menghindari perbedaan *volume* yang ekstrem antar *file*, sehingga model dapat fokus pada fitur suara, bukan pada perbedaan intensitasnya.

3. *Trimming & Padding*

   Audio yang berdurasi terlalu panjang atau terlalu pendek disesuaikan dengan cara memangkas bagian yang tidak relevan (*trimming*) atau menambahkan *padding* (suara hening) agar semua *file* memiliki durasi yang seragam, misalnya antara 5 hingga 10 detik.

4. *Noise Reduction*

   Proses ini bertujuan untuk menghilangkan atau mengurangi suara latar *audio* tersebut agar fitur yang diekstraksi lebih merepresentasikan suara utama. Dengan data yang lebih bersih, proses pelatihan *model* akan menjadi lebih optimal.

3.4.3 Ekstraksi Fitur *Audio* MFCC dan *Spectral Contrats*

Ekstraksi fitur merupakan tahap penting untuk mengubah sinyal suara mentah menjadi representasi numerik yang dapat dipahami oleh model deep *learning.* Dalam penelitian ini, proses ekstraksi dilakukan menggunakan dua fitur utama, yaitu:

1. *Mel Frequency Cepstral Coefficients* MFCC

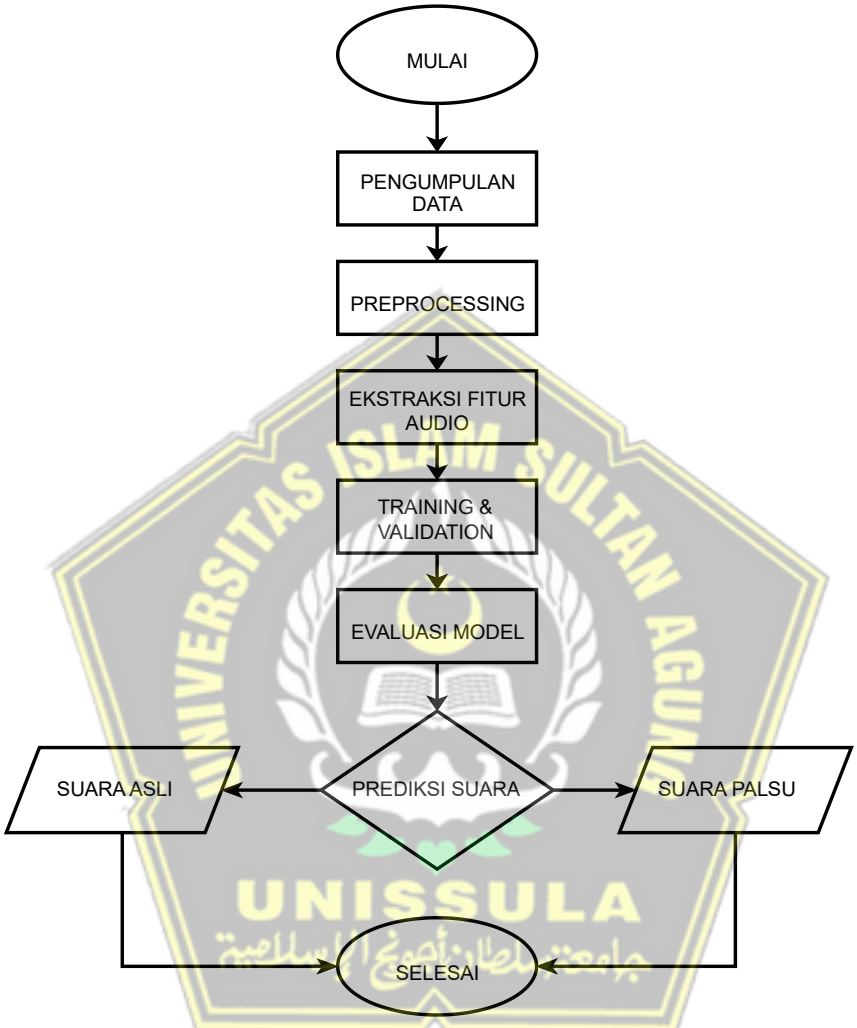
   Fitur ini dipakai sebagai proses pengubahan sinyal karakteristik spektral frekuensi berdasarkan persepsi pendengaran manusia. MFCC menghitung koefisien spektrum logaritmik setelah dipetakan ke dalam skala *Mel*, yang sesuai dengan cara manusia membedakan frekuensi suara. Biasanya diambil 13 hingga 20 koefisien MFCC per *frame*.

2. *Spectral Contrats*

   Fitur ini mengukur perbedaan logaritmik antara puncak (*peak*) dan lembah (*valley*) dalam spektrum frekuensi pada setiap *frame*. Spektrum dibagi menjadi beberapa *sub-band* (biasanya 6-7), dan dalam masing-masing *sub-band* dihitung nilai kontras energi. *Spectral Contrast* efektif untuk membedakan tekstur spektral dan pola harmonik dari suara asli dan palsu.

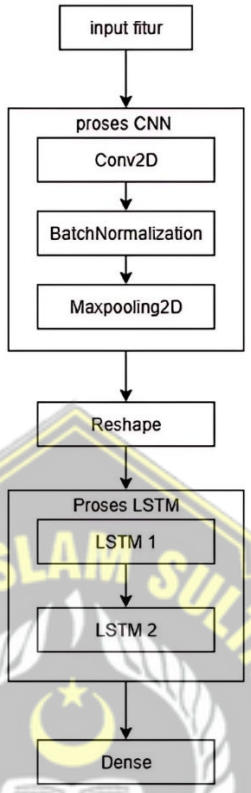
3.4.4 Pembangunan Model

Pada tahapan perancangan alur model yang akan dibuat maka berikut akan dipresentasikan dalam bentuk *flowchart* pada gambar 3.2.



Gambar 3. 2 flowchart rancangan model sistem deteksi

Pada Gambar 3.2 di atas merupakan sebuah flowchart untuk merancang sistem deteksi suara asli dan palsu manusia menggunakan Hybrid CNN dan LSTM. dibangun arsitektur *hybrid* yang menggabungkan *Convolutional Neural Network* (CNN) digunakan untuk menangkap data spasial hasil dari MFCC, dan *Long Short-Term Memory* (LSTM) untuk mengenali pola temporal dari urutan data suara. Kombinasi ini dirancang untuk meningkatkan akurasi dalam membedakan suara asli dan palsu.



Gambar 3. 3 proses *hybrid* CNN dan LSTM

Gambar 3.3 di atas merupakan diagram alur arsitektur model *hybrid* CNN-LSTM untuk deteksi suara palsu. Proses dimulai dari *input* fitur *audio* yang telah diekstraksi (misalnya MFCC atau *Spectral Contrast*). Selanjutnya, data masuk ke dalam blok proses CNN yang terdiri dari lapisan *Conv2D* untuk mengekstraksi fitur spasial, diikuti oleh *Batch Normalization* untuk menstabilkan distribusi data antar *layer*, dan *MaxPooling2D* untuk mereduksi dimensi fitur serta mempertahankan informasi penting. Setelah itu, hasil dari CNN diubah bentuknya melalui proses *Reshape* agar sesuai dengan format *input* LSTM. Data kemudian diteruskan ke blok proses LSTM yang terdiri dari dua lapisan LSTM berurutan, yaitu LSTM 1 dan LSTM 2, yang berfungsi untuk memahami urutan temporal dan pola dinamis dalam data *audio*. Akhirnya, *output* dari LSTM dikirim ke lapisan *Dense* yang melakukan klasifikasi untuk memutuskan apakah suara tersebut asli atau palsu.

3.4.5 Training Model Dan Validation

Dataset audio yang telah melewati tahap *preprocessing*, meliputi normalisasi durasi serta ekstraksi fitur MFCC dan *Spectral Contrast*, kemudian dimanfaatkan untuk melatih model *Hybrid* CNN dan LSTM. Proses pelatihan ini menggunakan beberapa *hyperparameter* penting, yaitu:

1. Bentuk *Input*

   Bentuk *input* merupakan dimensi fitur yang akan diberikan ke model. Dalam penelitian ini, *input* berbentuk matriks hasil ekstraksi MFCC dan *Spectral Contrast* dengan ukuran tertentu Menjadi 47 x 400 *frame*, yang kemudian diolah melalui lapisan CNN dan LSTM.

2. *Batch Size*

   Batch size adalah jumlah data audio yang diproses secara bersamaan sebelum bobot model diperbarui pada setiap iterasi. *Batch size* yang kecil membutuhkan memori lebih sedikit dan dapat membantu model belajar lebih stabil, namun proses pelatihan menjadi lebih lama. Sebaliknya, ukuran *batch* yang besar mempercepat latihan tetapi berisiko *overfitting* jika tidak diatur dengan benar. Oleh karena itu, pemilihan ukuran *batch* perlu menyesuaikan kapasitas CPU atau GPU yang digunakan.

3. *Epoch*

   Istilah epoch mengacu pada banyaknya siklus pelatihan penuh terhadap seluruh *dataset*. Pengaturan *epoch* yang terlalu rendah dapat menyebabkan *underfitting* karena pola data tidak ter pelajari dengan baik, sedangkan jumlah *epoch* yang berlebihan berpotensi menimbulkan *overfitting*, yakni model yang terlalu terikat pada data latih dan kurang mampu mengenali pola baru.

4. *Learning Rate*

   Learning rate adalah *hyperparameter* yang mengatur seberapa besar bobot model yang diperbarui pada setiap langkah optimasi. Jika *Learning rate* kecil akan memperlambat proses pelatihan, sedangkan *learning rate* yang terlalu besar dapat membuat pelatihan tidak stabil dan gagal konvergen. Oleh karena itu, pengaturan *learning rate* harus optimal agar model dapat belajar dengan baik.

Selanjutnya data set akan diproses melalui tahap validasi dengan membagi data menjadi 80 % pelatihan, 20% untuk validasi. Tahap validasi ini bertujuan mengetahui nilai dan sejauh mana model mampu mengenali pola pada data baru serta mengidentifikasi potensi masalah seperti *overfitting* atau *underfitting*. Hasil dari validasi ini akan digunakan sebagai acuan dalam menentukan strategi pelatihan yang optimal sebelum model diuji secara menyeluruh, dengan evaluasi kinerja dengan metrik utama yaitu akurasi, *presisi*, *recall* kemudian *F1-score*.

3.4.6 Testing Model

Tahap testing merupakan proses pengujian akhir model setelah melalui *training* dan *validation* dengan menggunakan data uji yang belum pernah digunakan sebelumnya. Pada tahap ini, data suara diproses melalui preprocessing seperti normalisasi, konversi ke mono, pemotongan durasi, serta ekstraksi fitur audio akan sama dengan input data model. Selanjutnya, CNN dan LSTM digunakan untuk memprediksi apakah suara termasuk kategori asli atau palsu, kemudian hasilnya dianalisis dengan metrik utama yaitu akurasi, presisi, *recall* kemudian *F1-score*, dan yang terakhir yaitu *confusion matrix*. Melalui tahap *testing* ini dapat diketahui kemampuan model dalam mendeteksi suara secara nyata sehingga dapat disimpulkan apakah model siap digunakan atau masih memerlukan perbaikan.

3.4.7 Evaluasi *Model*

Setelah *model* dilatih, dilakukan analisis terhadap performa model dengan menggunakan data uji. Bertujuan memastikan bahwa sistem dapat mendeteksi suara palsu secara efektif. Jika performa model belum mencapai target, proses pelatihan diulang dengan penyesuaian parameter untuk memperoleh hasil deteksi yang lebih optimal. Berikut merupakan penjelasan mengenai metrik akurasi berdasarkan rumus.

1. *True Positive* (TP): data sampel suara palsu yang betul terdeteksi sebagai palsu oleh model.
2. *True Negative* (TN): Jumlah sampel suara asli yang benar-benar terdeteksi sebagai asli oleh model.
3. *False Negative* (FN): Jumlah sampel suara asli yang gagal terdeteksi palsu.

4. *False Positive* (FP): Data sampel suara palsu yang gagal terdeteksi suara asli.
   a. Akurasi: merupakan menghitung berapa banyak data prediksi benar kemudian dibandingkan kepada seluruh data uji. Untuk Akurasi dapat dihitung rumus berikut:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \quad (1)$$

   b. *Precision*: Mengukur seberapa tepat model dalam memprediksi suara palsu. *Precision* tinggi berarti sebagian besar prediksi palsu memang benar-benar palsu. *Precision* dapat dihitung dengan rumus berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

   c. *Recall* : Mengukur seberapa baik model dalam menemukan semua suara palsu dari keseluruhan data palsu yang ada. *Recall* dapat dihitung dengan rumus berikut

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

   d. *F1 Score* : Merupakan nilai rata-rata harmonik dari *precision* dan *recall*, berguna untuk menilai performa model saat terjadi ketidakseimbangan data. *F1 score* dapat dihitung dengan rumus berikut :

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

3.4.8 *Software* penunjang penelitian

Dalam proses pembuatan sistem akan digunakan beberapa jenis *software* yang akan digunakan untuk mengolah data, pembuatan model serta evaluasi hasil. Berikut daftar *software* yang akan digunakan:

1. *Python*

   Python berfungsi untuk pembuatan coding utama yang mempunyai ekosistem pustaka yang sangat komplit, mudah digunakan, serta mendukung pengembangan sistem berbasis *machine learning* dan *deep learning.*

2. *Library*

   library ini dipakai untuk pengolahan sinyal audio, khususnya pada tahap *preprocessing* dan ekstraksi fitur. Dengan *Librosa*, fitur seperti MFCC dan *Spectral Contrast* dapat diekstrak dengan mudah untuk dijadikan *input* model.

3. *NumPy* dan *Pandas*

   NumPy digunakan untuk pengolahan data numerik, sedangkan *Pandas* berperan dalam manajemen data set serta penyusunan data dalam bentuk tabel sehingga lebih mudah diproses sebelum masuk ke tahap pelatihan model.

4. *Matplotlib*

   Digunakan untuk visualisasi data, seperti menampilkan bentuk gelombang suara, *spectrogram*, MFCC, serta hasil evaluasi model seperti *confusion matrix* atau grafik akurasi dan *loss*.

5. *TensorFlow* dan *Keras*

   Framework deep learning yang berfungsi membuat model serta melatih model. *TensorFlow* berfungsi sebagai mesin komputasi, sedangkan Keras memudahkan dalam perancangan arsitektur jaringan karena memiliki API yang sederhana.

6. *Jupyter Notebook* / *Google Colaboratory*

   Lingkungan pengembangan interaktif yang dipakai untuk menulis kode, menjalankan eksperimen, serta melakukan analisis. *Google Colab* juga menyediakan dukungan GPU yang mempercepat proses pelatihan model.

7. *Streamlit*

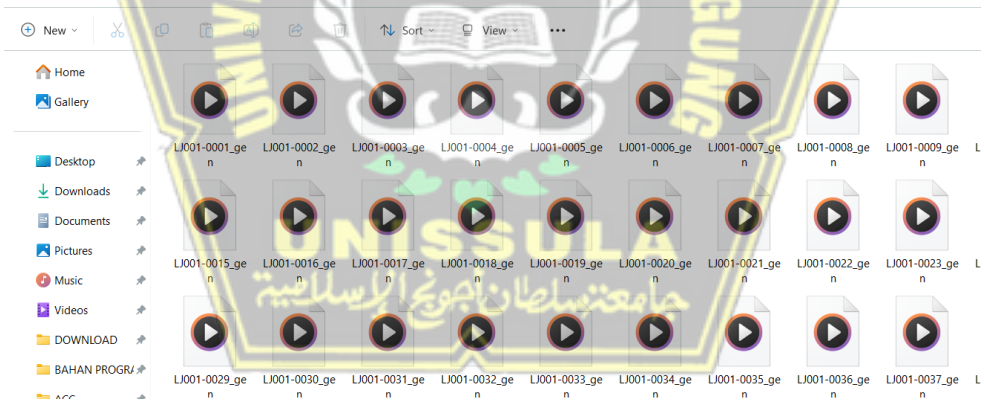
Digunakan untuk membangun antarmuka sistem sehingga pengguna dapat melakukan pengujian suara asli dan palsu secara langsung melalui aplikasi berbasis web. Dengan *Streamlit*, sistem dapat diakses lebih mudah tanpa perlu menjalankan kode secara manual.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil

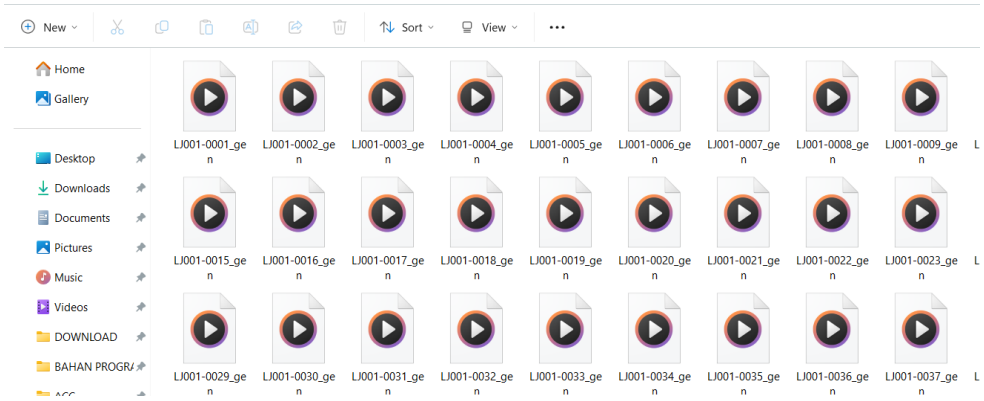
4.1.1 Pengumpulan data

Pada tahap pengumpulan data set, data dikumpulkan dari berbagai sumber publik, seperti *Kaggle* dan *Mozilla Fox Voice*, untuk memperoleh koleksi suara asli maupun suara palsu. Data set yang diperoleh kemudian di pecah menjadi data pelatihan dan data validasi, kemudian untuk data uji akan di buat secara manual menggunakan web bernama *Elevenlabs*. pemisahan ini berfungsi supaya proses pengembangan sistem dapat dilakukan secara sistematis dan terstruktur. Seluruh data disimpan dalam folder terpisah sesuai penyebabnya dan diunggah ke *Google Drive* untuk mempermudah integrasi dengan *Google Colab* selama proses pelatihan dan evaluasi. Proporsi pembagian data set dibuat seimbang untuk menjaga akurasi model agar dapat sekaligus meningkatkan kemampuan generalisasinya.



Gambar 4. 1 Data suara Palsu

Gambar 4.1 diatas adalah contoh data Suara palsu untuk proses *training* data yang akan dilakukan tahapan *preprocessing* nanti. sebelum digunakan untuk melatih model agar bisa mendeteksi Suara asli atau palsu.



Gambar 4. 2 Data suara Asli

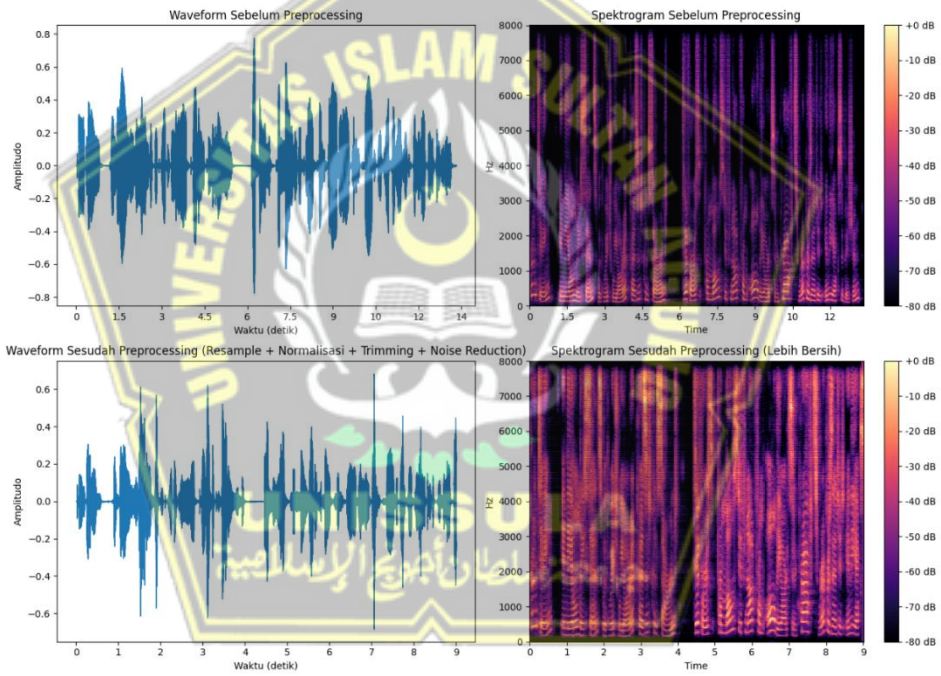
Gambar 4.2 diatas adalah contoh data suara asli dan palsu yang berfungsi sebagai data training dan validasi yang sebelumnya akan dilakukan tahapan *preprocessing* sebelum digunakan untuk melatih model agar bisa mendeteksi suara asli atau palsu.

4.1.2 Hasil *Preprocessing* Audio

Tabel 4. 1 *Preprocessing Audio*

Tahap	proses	hasil
Load Audio	Akan Membaca file dan audio asli menggunakan `librosa.load()` dengan sample rate bawaan.	Audio berhasil dibaca dari file audio asli dalam bentuk sinyal mentah (*raw signal*)
Resampling	`librosa.resample()` mengubah sample rate ke 16 kHz.	Sample rate audio diubah menjadi 16 kHz sehingga lebih konsisten dari pada kebutuhan model.
Normalisasi	`librosa.util.normalize()` menyesuaikan amplitudo sinyal.	Amplitudo sinyal berada pada rentang yang seimbang dan lebih stabil.
Trimming/Padding	Jika durasi lebih dari 9 detik dipotong; jika kurang ditambah *padding*.	Panjang data suara audio disesuaikan menjadi 9 detik; audio dipotong jika lebih panjang dan diberi *padding* jika lebih pendek.
Noise Reduction	*librosa.effects.preemphasis()* untuk penekanan derau.	Derau berkurang dan suara utama lebih jelas setelah proses *pre-emphasis*.

preprocessing pada data set audio menunjukkan bahwa seluruh file telah berhasil dikonversi ke format .wav dengan *sample rate* 16 kHz dan mode mono sehingga format data menjadi seragam dan sesuai dengan kebutuhan ekstraksi fitur. Proses normalisasi amplitudo membuat setiap file memiliki tingkat amplitudo yang konsisten dalam rentang -1 hingga 1, sehingga model tidak terpengaruh oleh perbedaan volume antar *file*. Tahap *trimming* dan *padding* juga berhasil menyeragamkan durasi audio, di mana *file* yang terlalu panjang dipangkas pada bagian yang tidak relevan, sedangkan *file* yang terlalu pendek diberi tambahan suara hening hingga mencapai durasi standar sekitar 9 detik. Selain itu, proses *noise reduction* mampu mengurangi suara latar sehingga sinyal utama.



Gambar 4. 3 visualisais Audio sebelum dan sesudah *preprocessing*

Pada Gambar 4.3 menunjukkan perbandingan sinyal audio sebelum dan sesudah ketika melalui tahap *preprocessing*. Pada bagian *waveform* ini sebelum *preprocessing*, terlihat bahwa sinyal audio masih memiliki amplitudo yang bervariasi dengan durasi yang tidak seragam serta adanya kemungkinan *noise* dari lingkungan. Hal ini juga tampak pada *spektrogram* sebelum *preprocessing*, di mana distribusi energi frekuensi masih tidak teratur dan terdapat banyak area gelap yang menunjukkan adanya gangguan *noise*. Setelah dilakukan *preprocessing* berupa

resampling, *normalisasi*, *trimming*, dan *noise reduction*, *waveform* menjadi lebih terstruktur dengan durasi yang seragam serta *amplitudo* yang seimbang tanpa lonjakan ekstrem. Perbaikan ini juga terlihat pada *spektrogram* sesudah *preprocessing* yang tampak lebih bersih, dengan pola energi frekuensi yang lebih jelas dan konsisten. Hasil ini menunjukkan bahwa proses *preprocessing* berhasil meningkatkan kualitas data audio

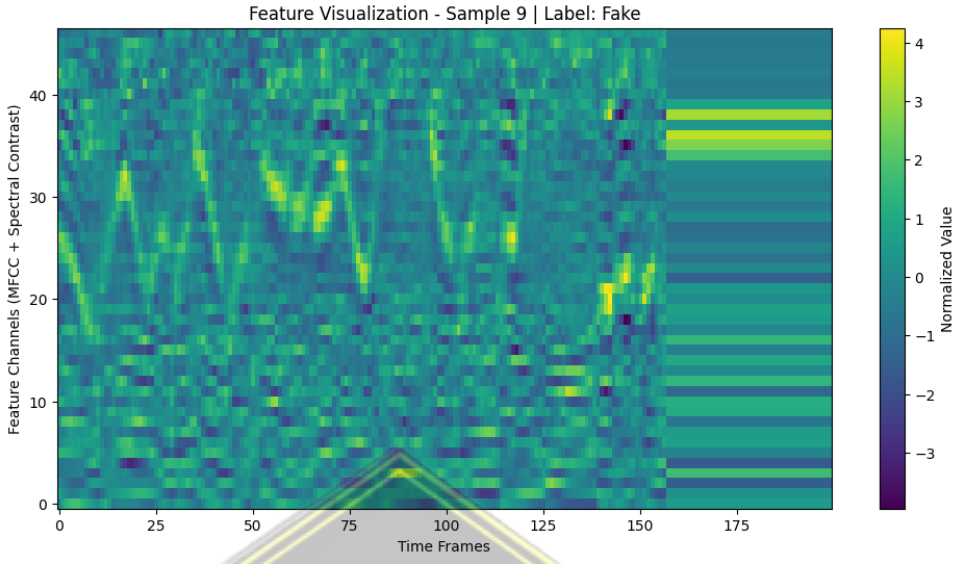
4.1.3 Hasil Ekstraksi fitur Awal

Tahap ekstraksi fitur ini bertujuan untuk mengambil ciri-ciri utama dari suara yang membedakan antara audio asli dan palsu, sehingga model dapat belajar secara efektif.

Tabel 4.2 tahap ekstraski mfcc dan *spectral contras*

Tahap proses	output	keterangan
Muat & Resample (16 kHz)	(T,)	Sinyal audio dengan format seragam
Ekstraksi MFCC	(40, T)	Representasi ciri akustik suara
Ekstraksi Spectral contrast	(7, T)	Distribusi energi antar frekuensi
Pad/Trim ke 200 frames	(40, 400)	Menyamakan panjang durasi audio
Penggabungan fitur	(47, 400)	40 MFCC + 7 Spectral Contrast
Dataset Akhir	X = (N, 47, 400) y = (N,)	N = jumlah file audio; Label 0 = Asli, 1 = Palsu

Pada tahap ekstraksi fitur, setiap *file* audio pertama-tama dikonversi ke format mono dengan *sample rate* 16 kHz untuk menyeragamkan kualitas data. Dari sinyal audio tersebut diekstraksi 40 koefisien MFCC (40, T) yang merepresentasikan pola akustik berdasarkan skala Mel, serta 7 fitur *Spectral Contrast* (7, T) yang menggambarkan distribusi energi antar frekuensi. Agar durasi data seragam, dilakukan *Pad/Trim* ke 400 *frame*, sehingga menghasilkan ukuran (40, 400) untuk MFCC dan (7, 400) untuk *Spectral Contrast*. Kedua fitur ini kemudian digabungkan menjadi matriks (47, 400) per file audio. Selanjutnya seluruh data dikompilasi menjadi data set akhir berukuran (N, 47, 400) dengan label 0 untuk suara asli dan 1 untuk suara palsu, yang digunakan sebagai *input* model CNN dan LSTM.



Gambar 4. 4 visualisasi mfcc dan *Spectral contras*

Pada Gambar 4.4 di atas merupakan visualisasi hasil ekstraksi fitur audio yang menggabungkan 40 koefisien MFCC dan 7 fitur *Spectral Contrast*, sehingga membentuk representasi berukuran (47×400). Sumbu vertikal menunjukkan dimensi fitur, sedangkan sumbu horizontal merepresentasikan *frame* waktu yang telah diseragamkan melalui proses *pad/trim*. Skala warna pada gambar menunjukkan nilai intensitas fitur, di mana variasi warna menggambarkan perbedaan karakteristik akustik pada sinyal suara. Representasi ini penting karena memungkinkan model CNN dan LSTM untuk menangkap pola perbedaan antara suara asli dan suara palsu secara lebih efektif dalam proses klasifikasi.

4.1.4 Hasil pelatihan Model

Model ini memanfaatkan arsitektur *hybrid* CNN dan LSTM dirancang untuk mendeteksi suara asli dan palsu.

Tabel 4.3 Arsitektur Model dan Layer

Arsitektur Model dan Layer	Keterangan
Input Shape	(47, 400, 1)
Conv2D + Pooling	32, 64, 128 filter (ReLU + BatchNorm + MaxPooling)
LSTM	2 lapisan, 128 unit, dropout 0.3
Dense	128 neuron (ReLU, L2), Dropout 0.5
Output	1 neuron (Sigmoid)

Berdasarkan Pada tabel 4.3 Data *audio* dengan ukuran (47, 400) terlebih dahulu diubah menjadi format 2D (47, 400, 1) agar dapat diproses CNN. CNN dengan filter 32, 64, dan 128 mengekstraksi pola lokal, dilanjutkan *Batch Normalization* dan *MaxPooling* untuk menstabilkan serta mereduksi dimensi. Hasil CNN kemudian diproses LSTM dengan 128-unit untuk menangkap pola temporal dua arah. Selanjutnya, lapisan Dense 128 neuron dengan *ReLU* dan *dropout* 0.5 mencegah *overfitting*, lalu lapisan *Dense sigmoid* menghasilkan probabilitas klasifikasi biner (asli = 0, palsu = 1). Model dikompilasi dengan *optimizer Adam, loss binary crossentropy*, dan akurasi sebagai metrik evaluasi utama.

Tabel 4.4 Model dikompilasi

Parameter	Keterangan
Optimizer	Adam
Loss Function	Binary Crossentropy
Metrics	Accuracy

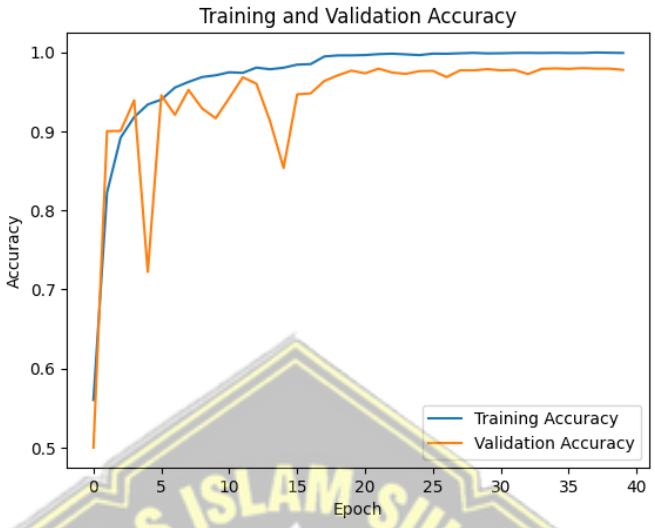
Pada Tabel 4.4 diatas menjelaskan parameter utama yang digunakan dalam proses pelatihan model. *Optimizer Adam* dipilih karena mampu menyesuaikan laju belajar secara otomatis sehingga proses *training* menjadi lebih cepat dan stabil. *Binary Crossentropy* digunakan sebagai *loss function* karena tugas penelitian ini adalah klasifikasi biner, yaitu membedakan suara asli dan suara palsu. Sementara itu**,** *Accuracy* digunakan sebagai metrik untuk mengukur seberapa banyak prediksi model yang benar selama proses pelatihan dan evaluasi. Dengan kombinasi tiga parameter ini, *model* dapat dilatih secara efektif dan performanya dapat dipantau dengan jelas.

Tabel 4.5 Training

Parameter	Keterangan
Batch size	64
Epoch	50
Regularisasi	L2(0.001), Dropout 0.5
EarlyStopping	10

Berdasarkan tabel 4.5 diatas Model dilatih selama 50 *epoch* dengan ukuran *batch* 64. Selama proses pelatihan, performa model pada data latih (*training*) dan

data validasi (*validation*) dipantau untuk memastikan sistem generalisasi dengan baik serta tidak terjadi bias data yang signifikan.



Gambar 4. 5 diagram evaluasi model

Pada gambar 4.5 di atas merupakan Hasil dari pemantauan menunjukkan bahwa model menunjukkan kinerja yang sangat baik sepanjang proses pelatihan. Nilai akurasi meningkat secara konsisten, sementara nilai *loss* mengalami penurunan berkelanjutan di setiap *epoch*. Tren ini menunjukkan bahwa model tersebut berhasil mempelajari data pelatihan dan mampu mengenali serta mengklasifikasikan pola dengan tepat. Perilaku yang stabil ini menandakan bahwa model arsitektur dan pengaturan parameter yang digunakan sudah efektif. Pada akhir pelatihan di *epoch* ke-40, *model* mencapai akurasi pelatihan sekitar 98,7% dan akurasi validasi sebesar 96,2%. kemudian Akurasi validasi menjadi metrik penting karena mencerminkan kemampuan model dalam menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Selisih yang relatif kecil antara akurasi pelatihan dan validasi, yaitu 2,5%, membuktikan efektivitas mekanisme regularisasi, termasuk penggunaan L2 dan *Dropout* 0,5.

4.1.5 Hasil Evaluasi Model

Hasil evaluasi dari sebuah model klasifikasi yang dilatih untuk membedakan antara suara asli dan palsu. model ini mencapai akurasi keseluruhan 86,49%. Akurasi ini dihitung dari total 74 sampel yang digunakan untuk pengujian, yang terdiri dari 37 suara asli dan 37 suara palsu. Secara lebih rinci, performa model dapat dianalisis melalui metrik presisi, *recall*, dan F1-score untuk masing-masing kelas. Untuk kelas palsu, model menunjukkan presisi yang sangat tinggi, yaitu 0,94, yang berarti 94% dari suara yang diprediksi sebagai palsu memang benar-benar palsu. Sebaliknya, *recall* untuk kelas ini sedikit lebih rendah, yaitu 0,78, menunjukkan model berhasil menemukan 78% dari total suara palsu.

Gambar 4. 6 evaluasi model dengan data uji

Berdasarkan pada Gambar 4.6 bisa di simpulkan bahwa Di sisi lain, untuk kelas asli, model memiliki *recall* yang sangat baik, yaitu 0,95, yang menandakan bahwa model berhasil mengidentifikasi 95% dari seluruh suara asli. Namun, presisinya sedikit lebih rendah, yaitu 0,81. Keseimbangan antara presisi dan *recall* dapat dilihat dari *F1-score*, di mana kelas asli memiliki nilai 0,88 dan kelas palsu

0,85. Analisis mendalam dari *confusion matrix* juga memperkuat temuan ini, menunjukkan bahwa dari 37 suara asli, model berhasil mengklasifikasikan 35 di antaranya dengan benar dan hanya 2 yang salah. Sebaliknya, dari 37 suara palsu, model berhasil mengklasifikasikan 29 dengan benar, tetapi 8 di antaranya salah diklasifikasikan sebagai suara asli. Secara keseluruhan, meskipun model memiliki performa yang baik, ia cenderung sedikit lebih sering keliru saat mengidentifikasi suara palsu sebagai suara asli.

Tabel 4.6 Tabel Evaluasi

Matrix	Asli	Palsu
Akurasi	$\frac{35 + 29}{35 + 29 + 8 + 2} = \frac{64}{74} = 0.8649$	
Presisi	$\frac{35}{35 + 8} = \frac{35}{43} = 0.81$	$\frac{29}{29 + 2} = \frac{29}{31} = 0.94$
Recall	$\frac{35}{35 + 2} = \frac{35}{37} = 0.95$	$\frac{29}{29 + 8} = \frac{29}{37} = 0.78$
F1-Score	$2 \times \frac{0{,}81 \times 0.95}{0{,}81 \times 0{,}95} = 0.88$	$2 \times \frac{0.94 \times 0.78}{0.94 \times 0.78} = 0.85$

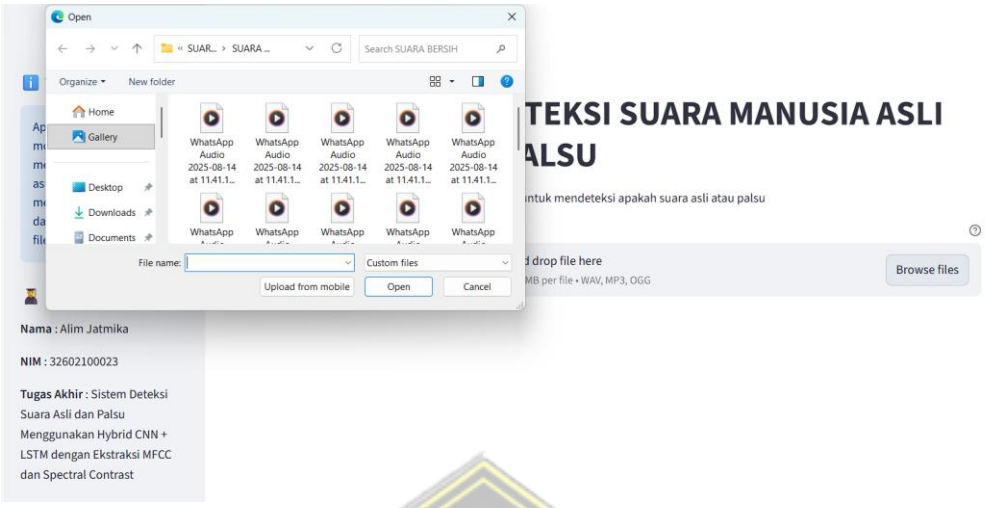
Tabel 4.6 menampilkan hasil evaluasi kinerja model dalam membedakan suara asli dan suara palsu menggunakan empat metrik utama, yaitu akurasi, *presisi*, *recall*, dan *F1-score*. Nilai rata-rata yang diperoleh sebesar 0,8649, menunjukkan bahwa 64 dari 74 data uji berhasil diklasifikasikan dengan benar oleh model. Pada kelas asli , nilai *presisi* mencapai 0,81, yang berarti sebagian besar prediksi suara asli sesuai dengan kondisi sebenarnya. Nilai *recall* pada kelas ini berada pada angka 0,95, menunjukkan bahwa hampir seluruh sampel suara asli dapat dikenal dengan baik oleh *model*. Untuk kelas palsu , presisi yang dihasilkan sebesar 0,94 menunjukkan bahwa prediksi suara palsu memiliki tingkat presisi yang tinggi, meskipun nilai *recall* sebesar 0,78 menunjukkan masih adanya sampel palsu yang tidak berhasil terdeteksi. Kombinasi kedua metrik tersebut menghasilkan *F1-score* sebesar 0,88 untuk kelas asli dan 0,85 untuk kelas palsu. Secara keseluruhan, hasil ini menunjukkan bahwa model memiliki performa yang stabil dan mampu memberikan deteksi yang cukup efektif pada kedua kelas.

4.2 Implementasi Model Deteksi suara Asli dan palsu

Pada tahap ini, sistem deteksi suara asli dan palsu manusia yang telah dibangun dengan arsitektur *Hybrid Convolutional Neural Network* (CNN) dan *Long Short Term Memory* (LSTM) akan diimplementasikan ke dalam sebuah aplikasi berbasis web menggunakan *Streamlit*. Proses implementasi ini bertujuan agar model yang telah dilatih dapat diakses dan digunakan secara langsung oleh pengguna dengan antarmuka yang sederhana dan interaktif.

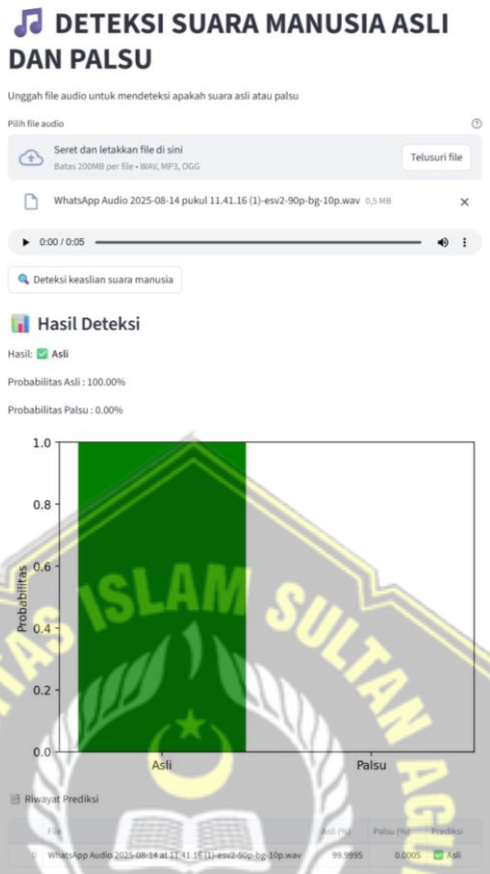
Gambar 4. 7 Tampilan antarmuka sistem

Pada Gambar 4.7 merupakan tampilan antarmuka sistem deteksi suara yang dibangun menggunakan *Streamlit*, pengguna dapat melakukan pengujian secara langsung terhadap model yang telah dibor. Aplikasi ini menyediakan fasilitas untuk mengunggah file audio dengan format .wav, .mp3, maupun .ogg, yang kemudian akan diproses secara otomatis oleh sistem. Setelah *file* audio berhasil diunggah, sistem akan melakukan ekstraksi ciri (ekstraksi fitur) menggunakan metode *Mel-Frequency Cepstral Coefisien* (MFCC) sebagai fitur utama. Selain itu, dalam kode program juga telah disiapkan proses ekstraksi *Spectral Contrast* sebagai fitur tambahan, meskipun dalam model implementasi yang digunakan saat ini fitur tersebut tidak diikutsertakan dalam proses prediksi, melainkan hanya sebagai pelengkap agar sistem lebih fleksibel untuk pengembangan di masa mendatang. Setelah fitur berhasil diekstrak, data kemudian diproses melalui arsitektur Gabungan arsitektur untuk melakukan proses klasifikas apakah suara yang diuji termasuk kategori asli atau palsu.



Gambar 4. 8 Tampilan *input audio* asli kedalam aplikasi

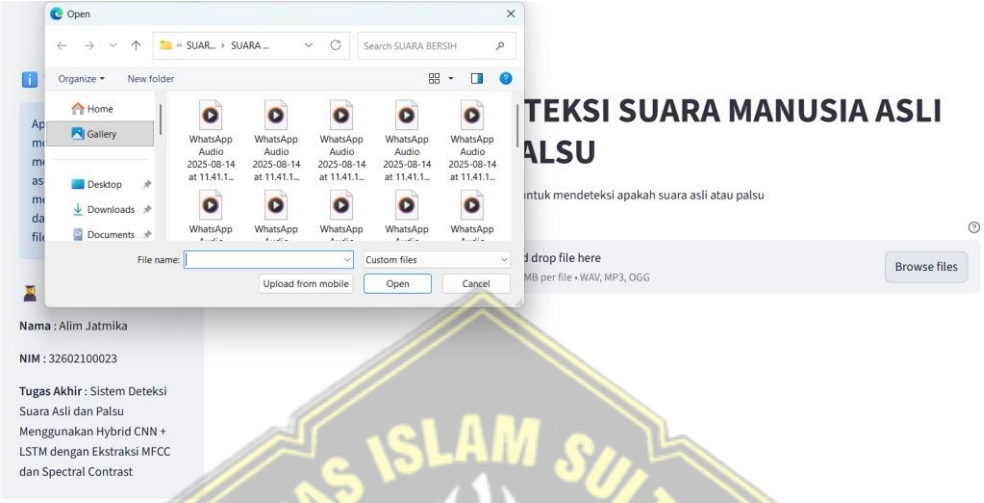
Pada gambar 4.8 diatas merupakam Tahap input *audio* asli pada aplikasi dilakukan melalui tombol Telusuri *file* pada antarmuka Streamlit. Saat tombol ditekan, jendela *File Explorer* akan terbuka untuk memilih file suara dari folder yang tersedia. Setelah file dipilih dan di *upload* , audio tersebut menjadi data uji yang akan diproses lebih lanjut melalui proses ekstraksi fitur MFCC dan *Spectral Contrast*, kemudian diklasifikasikan oleh model *Hybrid* CNN dan LSTM untuk menentukan apakah suara termasuk asli atau palsu.



Gambar 4. 9 Tampilan hasil deteksi suara asli

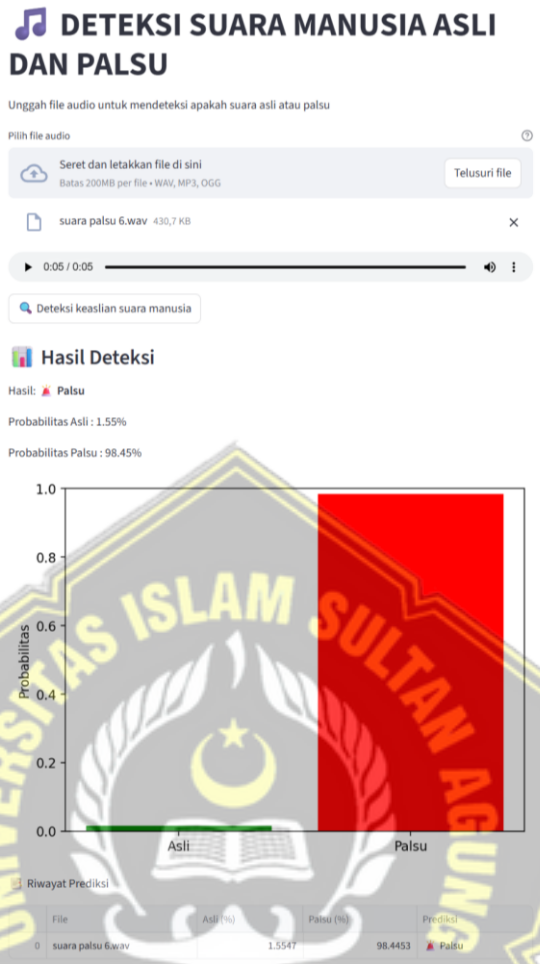
Pada gambar 4.9 diatas merupakan Hasil pengujian pada gambar tersebut menunjukkan proses klasifikasi suara menggunakan aplikasi deteksi suara asli dan palsu. Pada tahap awal, pengguna mengunggah *file* audio dengan format *wav* melalui sistem antarmuka, yang kemudian diproses menggunakan model *deep learning* berbasis arsitektur *hybrid* CNN dan LSTM dengan fitur MFCC dan *Spectral Contrast*. Setelah *file* diproses, sistem menampilkan hasil deteksi yang menunjukkan bahwa suara yang diuji terklasifikasi sebagai suara asli dengan probabilitas sebesar 100% , sedangkan probabilitas sebagai suara palsu adalah 0% . Visualisasi hasil berupa grafik batang memperkuat informasi ini, di mana kategori *Asli* ditampilkan penuh dengan warna hijau, sementara kategori *Palsu* tidak menunjukkan nilai probabilitas. Selain itu, aplikasi juga menyimpan riwayat prediksi dalam bentuk tabel yang berisi nama *file* audio, nilai probabilitas masing-masing kelas, serta hasil prediksi akhir. Berdasarkan hasil tersebut, dapat

disimpulkan bahwa sistem mampu mendeteksi suara dengan tingkat keyakinan yang sangat tinggi, dan pada kasus ini suara yang diuji dipastikan merupakan suara asli.



Gambar 4. 10 Tampilan *input* audio palsu kedalam aplikasi

Pada gambar 4.10 diatas merupakan Tahap *input audio* palsu dilakukan dengan cara yang sama seperti saat memasukkan audio asli, yaitu melalui tombol *Telusuri file* pada antarmuka *Streamlit*. Ketika tombol tersebut ditekan, jendela *File* Explorer akan muncul untuk memilih *file* suara palsu yang telah disiapkan dalam folder. Setelah *file* dipilih dan diunggah, audio tersebut langsung menjadi data uji yang akan diproses lebih lanjut melalui tahapan ekstraksi fitur MFCC dan *Spectral Contrast*. Hasil ekstraksi kemudian dikirim ke model *Hybrid* CNN-LSTM untuk dianalisis dan diklasifikasikan, sehingga sistem dapat menentukan apakah audio yang di input merupakan suara palsu sesuai dengan label dan karakteristik akustiknya.



Gambar 4. 11 Tampilan hasil deteksi palsu

Pada Gambar 4.10 diatas menunjukkan hasil pengujian sistem deteksi suara asli dan palsu menggunakan aplikasi berbasis *Streamlit*. Pada contoh ini, pengguna mengunggah *file* audio bernama suara (palsu 6.wav) untuk diuji. Setelah melalui proses ekstraksi serta klasifikasi, sistem menampilkan hasil bahwa suara tersebut terdeteksi sebagai palsu dengan probabilitas sebesar 98,45%, sedangkan probabilitas sebagai suara asli hanya 1,55%. Hasil ini divisualisasikan dalam bentuk grafik batang, di mana kelas palsu ditunjukkan dengan batang merah yang mendominasi, sedangkan kelas *asli* hanya ditunjukkan dengan batang hijau yang sangat kecil.

BAB V
KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi suara palsu yang dikembangkan dalam penelitian ini berhasil berfungsi secara efektif dan memberikan performa yang kuat dalam membedakan suara asli dari suara sintetis. Sistem ini dirancang menggunakan pendekatan *deep learning* berbasis arsitektur *hybrid* yang menggabungkan *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), di mana CNN berperan dalam mengekstraksi ciri-ciri penting pada representasi *spektrogram*, sementara LSTM menangani pola *temporal* yang muncul pada rangkaian sinyal suara. Kedua arsitektur tersebut diperkuat dengan penggunaan dua jenis fitur akustik, yaitu *Mel-Frequency Cepstral Coefficients* (MFCC) dan *Spectral Contrast*, yang terbukti mampu menyoroti perbedaan karakteristik *spektral* antara suara manusia asli dan suara palsu yang dihasilkan melalui teknik sintesis. Dari hasil pengujian, sistem menunjukkan performa yang baik dengan tingkat akurasi mencapai 86,49%. Pada kelas suara asli (label 0), model menghasilkan nilai *presisi* sebesar 0.81, *recall* 0.95, dan *F1-score* 0.88, yang menunjukkan bahwa model mampu mengenali sebagian besar suara asli dengan benar serta menjaga keseimbangan antara ketepatan dan sensitivitasnya. Sementara itu, pada kelas suara palsu (label 1), model mencatat nilai *presisi* sebesar 0.94, *recall* 0.78, dan *F1-score* 0.85, yang menggambarkan kemampuan model mendeteksi suara palsu dengan tingkat ketepatan yang sangat tinggi meskipun masih terdapat beberapa sampel yang tidak teridentifikasi secara optimal. Sebagai implementasi nyata, sistem ini juga telah berhasil diterapkan dalam bentuk aplikasi web interaktif berbasis *Streamlit* yang menyediakan antarmuka sederhana dan mudah digunakan untuk mengunggah *file audio* dan memperoleh hasil klasifikasi secara otomatis. Kehadiran aplikasi ini menunjukkan bahwa sistem yang dibangun tidak hanya *valid* secara akademis, tetapi juga memiliki nilai guna praktis untuk meningkatkan keamanan digital dan mencegah penyalahgunaan teknologi.

4.2 Saran

Untuk pengembangan di masa depan, disarankan beberapa penyempurnaan berikut:

1. Pengembangan Data set: Menambah variasi data dengan beragam bahasa, aksen, dan jenis manipulasi suara yang lebih canggih untuk meningkatkan kemampuan generalisasi model.
2. Peningkatan dalam Ketahanan Terhadap Derau: Mengintegrasikan teknik *noise reduction* yang lebih canggih untuk meningkatkan akurasi model pada kondisi lingkungan yang bising.
3. Eksplorasi Fitur Tambahan: Meneliti fitur audio lain seperti *pitch* dan *zero-crossing rate* untuk meningkatkan kemampuan deteksi anomali pada suara sintetis.
4. Optimalisasi Arsitektur: Bereksperimen dengan model *deep learning* yang lebih maju, untuk potensi peningkatan akurasi

DAFTAR PUSTAKA

Aini, Y. K., Santoso, T. B., & Dutono, T. (2021). Pemodelan CNN Untuk Deteksi Emosi Berbasis Speech Bahasa Indonesia. *Jurnal Komputer Terapan*, *7*(1), 143–152. https://doi.org/10.35143/jkt.v7i1.4623

Ali, S., Tanweer, S., Khalid, S. S., & Rao, N. (n.d.). *Mel Frequency Cepstral Coefficient : A Review*. https://doi.org/10.4108/eai.27-2-2020.2303173

Aljufri, M. N., & Prasetio, B. H. (2022). Sistem Deteksi Tingkat Stress Menggunakan Suara dengan Metode Jaringan Saraf Tiruan dan Ekstraksi Fitur MFCC berbasis Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(11), 5278–5285. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11842

Deressa, D. W., Lambert, P., Van Wallendael, G., Atnafu, S., & Mareen, H. (2024). Improved Deepfake Video Detection Using Convolutional Vision Transformer. *2024 IEEE Gaming, Entertainment, and Media Conference, GEM 2024*. https://doi.org/10.1109/GEM61861.2024.10585593

Dewa Agung Adwitya Prawangsa, I., & Eka Karyawati, A. (2024). *Penerapan Metode MFCC dan LSTM untuk Speech Emotion Recognition*. *12*(4), 2654–5101. www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess.

Galih Ajinurseto, La Ode Bakrim, N. I. (2023). Penerapan Metode Mel Frequency Cepstral Coefficients pada Sistem Pengenalan Suara Berbasis Desktop. *Infomatek*, *25*(1), 11–20. https://doi.org/10.23969/infomatek.v25i1.6109

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Hamza, A., & Javed, A. R. (2022). Deepfake Audio Detection via MFCC Features Using Machine Learning. *IEEE Access*, *10*(December), 134018–134028. https://doi.org/10.1109/ACCESS.2022.3231480

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, *2016-December*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., & Wilson, K. (2017). CNN architectures for large-scale audio classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 131–135. https://doi.org/10.1109/ICASSP.2017.7952132

Hochreiter, S. (2016). *Long Short-Term Memory*. *November 1997*. https://doi.org/10.1162/neco.1997.9.8.1735

Kumar, S., & Thiruvenkadam, S. (2021). An Analysis of the Impact of Spectral Contrast Feature in Speech Emotion Recognition. *International Journal of Recent Contributions from Engineering, Science & IT (IJES)*, *9*(2), 87. https://doi.org/10.3991/ijes.v9i2.22983

Lim, S., & Chae, D. (2022). *applied sciences Detecting Deepfake Voice Using Explainable Deep Learning Techniques*.

Liu, C. (2024). Long short-term memory (LSTM)-based news classification model. *PLoS ONE*, *19*(5 May), 1–23. https://doi.org/10.1371/journal.pone.0301835

Logan, B., & Engineers, E. (2014). *Mel Frequency Cepstral Coefficients for Music Modeling*. *November 2000*.

Ma, N., Brown, G. J., & Gonzalez, J. A. (2015). Exploiting top-down source models to improve binaural localisation of multiple sources in reverberant environments. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2015-Janua*, 160–164. https://doi.org/10.21437/interspeech.2015-76

N.A.Bhaskaran, Srinadh, M., Dhamodhar, K., & R, M. M. (2024). Detecting Deep Fake Voice using Machine Learning. *Shanlax International Journal of Economics*, *11*(S3), 53–60.

Neelima, M., & Prabha, I. S. (2024). Hybrid Feature Optimization for Voice Spoof Detection Using CNN-LSTM. *Traitement Du Signal*, *41*(2), 717–727. https://doi.org/10.18280/ts.410214

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal on Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/JSTSP.2019.2908700

Qazi, H., & Kaushik, B. N. (2020). A Hybrid Technique using CNN LSTM for Speech Emotion Recognition. *International Journal of Engineering and Advanced Technology*, *9*(5), 1126–1130. https://doi.org/10.35940/ijeat.e1027.069520

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Susetianingtias, D. T., & Patriya, E. (2024). *Identifikasi Fitur Suara Menggunakan Model Convolutional Neural Network (CNN) pada Speech-to-Text (STT)*. *4*(3), 809–820.

Swastika, W., Oepojo, A. A., & Irawan, P. L. T. (2023). *Perbandingan Akurasi Deteksi Emosi Pada Suara Menggunakan Multilayer Perceptron , Random Forest , Decision Tree dan K-NN*. *05*(01), 1–6. https://doi.org/10.52985/insyst.v5i1.264

Todisco, M., Delgado, H., & Evans, N. (2017). Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech and Language*, *45*, 516–535. https://doi.org/10.1016/j.csl.2017.01.001

Waqas, M., & Humphries, U. W. (2024). A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX*, *13*(September), 102946. https://doi.org/10.1016/j.mex.2024.102946

Yi, J., Wang, C., Tao, J., Zhang, X., Zhang, C. Y., & Zhao, Y. (2023). *Audio Deepfake Detection: A Survey*. *14*(8), 1–20. http://arxiv.org/abs/2308.14970

Yusdiantoro, S. Y., & Sasongko, T. B. (2023). Implementasi Algoritma MFCC dan CNN dalam Klasifikasi Makna Tangisan Bayi. *Indonesian Journal of Computer Science*, *12*(4), 1957–1968. https://doi.org/10.33022/ijcs.v12i4.3243