

**PENGURANGAN KEBISINGAN DI PABRIK TEKSTIL MENGGUNAKAN  
U-NET CNN UNTUK MENINGKATKAN KEJELASAN KOMUNIKASI  
VERBAL**

Usulan Penelitian Tesis S-2

Untuk memenuhi persyaratan memperoleh gelar Magister Teknik  
Program Studi Magister Teknik Elektro



Diajukan oleh  
Ferry Irmansyah  
MTE20602400016

**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG  
NOVEMBER, 2024**

## ABSTRAK

Kebisingan dengan intensitas tinggi di lingkungan pabrik tekstil merupakan masalah serius yang tidak hanya berdampak pada kesehatan pendengaran pekerja tetapi juga secara signifikan menurunkan kejelasan komunikasi verbal, yang krusial untuk keselamatan dan efisiensi operasional. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem reduksi kebisingan cerdas berbasis deep learning untuk mengatasi masalah tersebut. Metode yang diusulkan menggunakan arsitektur U-Net Convolutional Neural Network (U-Net CNN) untuk memisahkan sinyal ucapan dari kebisingan latar yang kompleks. Sinyal audio yang terkontaminasi oleh derau mesin diubah menjadi representasi spektrogram 2D, yang kemudian diproses oleh model U-Net untuk mengestimasi sebuah Ideal Ratio Mask (IRM). Masker ini secara selektif menekan komponen kebisingan sambil mempertahankan karakteristik esensial dari sinyal ucapan. Hasil pengujian kuantitatif pada dataset uji menunjukkan bahwa sistem yang dikembangkan mampu mencapai peningkatan Signal-to-Noise Ratio (SNR) rata-rata sebesar 63.28 dB dan tingkat pemeliharaan spektral (spectral preservation) rata-rata 99.97%. Hasil ini membuktikan bahwa pendekatan berbasis U-Net CNN sangat efektif dalam mereduksi kebisingan industrial secara signifikan sambil menjaga kualitas dan kealamian suara asli, sehingga berpotensi besar untuk diimplementasikan sebagai alat bantu fungsional guna meningkatkan kejelasan komunikasi verbal di lingkungan pabrik.

Kata Kunci: Reduksi Kebisingan Audio, U-Net CNN, Deep Learning, Kebisingan Industri, Spektrogram.

## ABSTRACT

*High-intensity noise in textile factory environments is a critical issue that not only impacts workers' auditory health but also significantly degrades the clarity of verbal communication, which is crucial for operational safety and efficiency. This research aims to develop and evaluate an intelligent deep learning-based noise reduction system to address this problem. The proposed method utilizes a U-Net Convolutional Neural Network (U-Net CNN) architecture to separate speech signals from complex background noise. Audio signals contaminated by machinery noise are transformed into 2D spectrogram representations, which are then processed by the U-Net model to estimate an Ideal Ratio Mask (IRM). This mask selectively suppresses noise components while preserving the essential characteristics of the speech signal. Quantitative evaluation on a dedicated test dataset demonstrates that the developed system achieves an average Signal-to-Noise Ratio (SNR) improvement of 63.28 dB and an average spectral preservation rate of 99.97%. These results prove that the U-Net CNN-based approach is highly effective in significantly reducing industrial noise while maintaining the quality and naturalness of the original voice, thus showing great potential for implementation as a functional tool to enhance verbal communication clarity in factory environments.*

*Keywords: Audio Noise Reduction, U-Net CNN, Deep Learning, Industrial Noise, Spectrogram.*

## LEMBAR PENGESAHAN TESIS

### PENGURANGAN KEBISINGAN DI PABRIK TEKSTIL MENGGUNAKAN U-NET CNN UNTUK MENINGKATKAN KEJELASAN KOMUNIKASI VERBAL

Yang dipersiapkan dan disusun oleh

**Ferry Irmansyah**

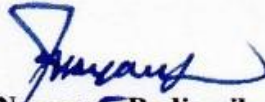
NIM : 20602400016

Telah dipertahankan di depan Dewan Penguji

Pada Tanggal : 11 November 2025

Susunan Dewan Penguji dan Pembimbing

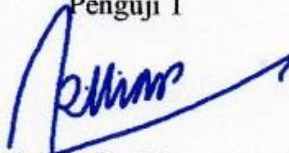
Pembimbing Utama



**Dr. Eka Nuryanto Budisusila, ST., MT**

NIDN : 0619107301

Penguji 1



**Prof. Dr. Sri Artini Dwi Prasetyowati, M. Si**

NIDN : 0620026501

Penguji 2



**Dr. Bustanul Arifin, ST, MT**

NIDN : 0614117701

Tesis ini telah disetujui sebagai salah satu persyaratan untuk memperoleh gelar

Magister Teknik

Tanggal : 11 November 2025

Ketua Program Studi Magister Teknik Elektro



**Prof. Dr. Sri Artini Dwi Prasetyowati, M. Si.**

NIDN : 0620026501

## PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan di bawah ini :

Nama : Ferry Irmansyah  
NIM : 20602400016  
Program Studi : Magister Teknik Elektro  
Fakultas : Teknologi Industri

Dengan ini saya menyatakan bahwa Tesis yang diajukan kepada program studi Magister Teknik Elektro dengan judul :

### **“Pengurangan Kebisingan Di Pabrik Tekstil Menggunakan U-Net Cnn Untuk Meningkatkan Kejelasan Komunikasi Verbal”**

Adalah hasil karya saya sendiri dengan judul tersebut belum pernah diajukan untuk memperoleh gelar Magister Teknik Elektro pada Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung (UNISSULA) ataupun Universitas lain serta belum pernah ditulis maupun diterbitkan orang lain kecuali secara tertulis diacu, disitasi, dan ditunjukkan dalam daftar Pustaka. Tesis ini adalah tanggung jawab saya.

Semarang, 11 November 2025

Penulis



FERRY IRMANSYAH

NIM : 20602400016

## PERNYATAAN PERSETUJUAN UNGGAH KARYA ILMIAH

Saya yang bertanda tangan di bawah ini :

Nama	: FERRY IRMANSYAH
NIM	: 20602400016
Program Studi	: Magister Teknik Elektro
Fakultas	: Teknologi Industri

Dengan ini menyerahkan karya ilmiah berupa ~~Tugas Akhir/Skripsi/Tesis/Disertasi\*~~ dengan judul :

PENGURANGAN KEBISINGAN DI PABRIK TEKSTIL MENGGUNAKAN U-NET  
CNN UNTUK MENINGKATKAN KEJELASAN KOMUNIKASI VERBAL

dan menyetujuinya menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak Bebas Royalti Non-eksklusif untuk disimpan, dialihmediakan, dikelola dalam pangkalan data, dan dipublikasikannya di internet atau media lain untuk kepentingan akademis selama tetap mencantumkan 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 pihak Universitas Islam Sultan Agung.

Semarang, 11 November 2025

Yang menyatakan,



(FERRY IRMANSYAH)

\*Coret yang tidak perlu

## KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah Subhānahu wa Ta‘ālā atas limpahan rahmat, taufik, dan kesehatan sehingga tesis yang berjudul *“Pengurangan Kebisingan di Pabrik Tekstil Menggunakan U-Net CNN untuk Meningkatkan Kejelasan Komunikasi Verbal”* dapat diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad ﷺ ‘Alaihi wa Sallam, yang telah memberikan teladan dalam menuntut ilmu dan memperjuangkan kemaslahatan umat.

Tesis ini disusun sebagai pemenuhan salah satu syarat untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro, Universitas Islam Sultan Agung (UNISSULA). Penelitian ini dilaksanakan sebagai bentuk kontribusi terhadap upaya peningkatan kualitas komunikasi verbal di lingkungan pabrik tekstil, khususnya di area dengan tingkat kebisingan tinggi yang dapat menghambat koordinasi kerja, keselamatan, dan efisiensi operasional.

Selama proses penyusunan tesis ini, penulis menerima berbagai dukungan, bimbingan, dan arahan yang sangat berarti dari berbagai pihak. Penulis menyampaikan apresiasi yang setinggi-tingginya atas bantuan tersebut. Penulis juga menyadari bahwa tesis ini masih memiliki keterbatasan, sehingga kritik dan saran yang membangun sangat dinantikan untuk penyempurnaan penelitian di masa mendatang.

Penulis berharap hasil penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang pemrosesan sinyal audio dan teknologi pengurangan kebisingan, serta dapat diterapkan sebagai salah satu solusi praktis di lingkungan industri tekstil.

Semarang, 11 November 2025

Penulis



Ferry Irmansyah

NIM : 20602400016

## UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kehadirat Allah Subhānahu wa Ta‘ālā atas limpahan rahmat, karunia, dan hidayah-Nya sehingga Tesis berjudul “Pengurangan Kebisingan di Pabrik Tekstil Menggunakan U-Net CNN untuk Meningkatkan Kejelasan Komunikasi Verbal” dapat diselesaikan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro, Universitas Islam Sultan Agung (UNISSULA).

Penyusunan tesis ini tidak akan terwujud tanpa bantuan, dukungan, bimbingan, serta doa dari berbagai pihak. Pada kesempatan ini, dengan segala kerendahan hati, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Dr. Eka Nuryanto Budisusila, ST., MT., selaku Dosen Pembimbing, atas bimbingan, arahan, kesabaran, serta wawasan ilmiah yang sangat berharga. Saran-saran beliau telah memberikan kontribusi mendalam dalam penyempurnaan penelitian ini.
2. Dewan Penguji Tesis, atas kritik, masukan, dan evaluasi konstruktif yang sangat membantu penyempurnaan naskah ini:
  - a. Prof. Dr. Sri Artini Dwi Prasetyowati, M.Si., selaku Penguji 1, atas pertanyaan yang tajam, kritik ilmiah yang membangun, serta pandangan akademik yang memperkaya isi penelitian ini.
  - b. Dr. Bustanul Arifin, ST., MT., selaku Penguji 2, atas saran teknis dan masukan praktis yang sangat bermanfaat bagi penguatan metodologi dan implementasi penelitian.
3. Prof. Dr. Sri Artini Dwi Prasetyowati, M.Si., selaku Ketua Program Studi Magister Teknik Elektro FTI UNISSULA, atas dukungan akademik, administrasi, dan kesempatan yang diberikan kepada penulis untuk menempuh pendidikan di program studi ini.
4. Dekan Fakultas Teknologi Industri UNISSULA beserta seluruh jajarannya, atas fasilitas, layanan, dan lingkungan akademik yang kondusif selama masa studi penulis.
5. Seluruh Dosen, tenaga kependidikan, dan staf Program Studi Magister Teknik Elektro, yang telah memberikan ilmu, bantuan, dan pelayanan terbaik sepanjang proses pembelajaran.
6. Pimpinan dan rekan kerja di PT. Embee Plumbon Tekstile, Unit 2—khususnya pihak yang telah memberikan izin, akses data, serta dukungan dalam pengambilan data penelitian.
7. Kedua orang tua tercinta, atas doa, kasih sayang, pengorbanan, dan dorongan moral yang tiada henti. Segala capaian penulis tidak lepas dari restu dan dukungan mereka.
8. Keluarga dan sahabat-sahabat terdekat, yang selalu memberikan motivasi, semangat, dan dukungan emosional sepanjang proses penyusunan tesis.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk penyempurnaan di masa mendatang. Semoga penelitian ini dapat memberikan manfaat bagi dunia akademik maupun praktis, khususnya dalam pengembangan teknologi pengurangan kebisingan di lingkungan industri.



## DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	ii
LEMBAR PENGESAHAN TESIS .....	iii
PERNYATAAN KEASLIAN TESIS .....	iv
PERNYATAAN PERSETUJUAN UNGGAH KARYA ILMIAH .....	v
KATA PENGANTAR .....	vi
UCAPAN TERIMA KASIH .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR RUMUS .....	xii
DAFTAR TABEL .....	xiii
BAB I	
PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	3
1.4. Manfaat Penelitian .....	3
1.5. Batasan Penelitian .....	3
1.6. Keaslian Penelitian .....	3
1.7. Sistematika Penulisan .....	5
BAB II	
TINJAUAN PUSTAKA DAN DASAR TEORI .....	6
2.1. Tinjauan Pustaka .....	6
2.1.1. Metode Pengurangan Kebisingan .....	6
2.1.2. Artificial Neural Network dalam Pemrosesan Sinyal .....	6
2.1.3. Penerapan ANN untuk Pengurangan Kebisingan di Pabrik Tekstil .....	6
2.2. Dasar Teori .....	7
2.2.1. Definisi Kebisingan .....	8
2.2.2. Sumber Kebisingan di Pabrik Tekstil .....	8
2.2.3. Dampak Kebisingan pada Pekerja .....	9
2.2.3.1. Pentingnya komunikasi verbal .....	9
2.2.3.2. Tantangan Komunikasi di Lingkungan Bising .....	10
2.2.4. Pendekatan Tradisional terhadap Pengurangan Kebisingan .....	10
2.2.5. Pendekatan inovatif dalam pengurangan kebisingan .....	11
2.2.6. Arsitektur Convolutional Neural Network sebagai Fondasi Model .....	11
2.2.6.1. Prinsip <i>Supervised Learning</i> pada CNN .....	12
2.2.6.2. Rumus-Rumus dalam Supervised Learning pada ANN .....	12
2.2.6.3. Proses Pelatihan Supervised Learning pada (ANN) .....	15
2.2.6.4. Keunggulan ANN dalam Pengolahan Sinyal .....	19

2.2.6.5. Aplikasi ANN dalam Pengurangan Kebisingan .....	19
2.2.7. Dari Audio ke Rekonstruksi Ucapan Menggunakan U-Net CNN .....	20
2.2.7.1. Transformasi Audio menjadi Citra melalui STFT & Spektrogram ...	20
2.2.7.2. Ekstraksi Fitur Kontekstual: Jalur Encoder pada U-Net.....	22
2.2.7.3. Jalur Decoder dan Kekuatan Skip Connections .....	23
2.2.7.4. Generasi Output dan Konversi Kembali ke Audio.....	24
2.2.8. Keuntungan dan Tantangan .....	25
2.2.9. Metrik Evaluasi Kinerja .....	25
2.2.10 Skenario Pengujian.....	27
<b>BAB III</b>	
<b>METODOLOGI PENELITIAN .....</b>	<b>28</b>
3.1. Desain Penelitian.....	28
3.2. Lokasi dan Waktu Penelitian .....	28
3.3. Populasi dan Sampel .....	28
3.4. Variabel Penelitian .....	28
3.5. Instrumen Penelitian.....	29
3.6. Teknik Pengumpulan Data .....	29
3.7. Teknik Analisis Data .....	29
3.8. Prosedur Penelitian.....	29
3.9. Etika Penelitian .....	30
3.10. Validitas dan Reliabilitas .....	30
3.11. Keterbatasan Penelitian .....	30
3.12. Diagram Alir Perancangan .....	31
3.13. Digitalisasi Suara ( <i>Analog ke Digital</i> ) .....	34
3.14. Blok Diagram Sistem .....	35
3.15. Rancangan Jaringan Saraf Tiruan (ANN) .....	39
3.16. Proses dan Konfigurasi Pelatihan Model .....	47
<b>BAB IV</b>	
<b>HASIL DAN PEMBAHASAN .....</b>	<b>48</b>
4.1. Arsitektur Proyek&Fondasi Lingkungan Penelitian yang Reproducible ...	48
4.1.1. Elaborasi Struktur Direktori .....	48
4.1.2. Justifikasi Lingkungan Penelitian yang Terkontainerisasi .....	49
4.2. Arsitektur Alur Kerja Pemrosesan Sinyal dan Inferensi Model.....	50
4.2.1. Fase 1 Transformasi dan Dekomposisi Sinyal .....	50
4.2.2. Fase 2 Inferensi Model dan Estimasi Masker Spektral.....	52
4.2.3. Fase 3 Rekonstruksi Sinyal Audio .....	53
4.3. Analisis Visual Komponen Sinyal Dataset.....	54
4.4. Analisis Komparatif Sinyal Input dan Output Model .....	56
4.4.1. Karakteristik Sinyal Input Model.....	57
4.4.2. Karakteristik Sinyal Output Hasil Pemrosesan Model.....	58
4.5. Studi Kasus: Simulasi Skenario Komunikasi Verbal .....	59
4.5.1. Analisis Skenario Komunikasi Bersih (Target).....	60

4.5.2. Analisis Skenario Komunikasi Bising (Input Model) .....	60
4.5.3. Analisis Skenario Komunikasi Hasil Proses (Output Model) .....	61
4.6. Fungsionalitas Aplikasi dan Skenario Pengujian .....	62
4.7. Proses Pengolahan Audio .....	63
4.8. Analisis Kinerja Berdasarkan Slider Strength .....	64
4.9. Hasil Pengujian Kuantitatif .....	64
4.10. Pembahasan .....	65
BAB V	
KESIMPULAN DAN SARAN .....	67
5.1. Kesimpulan .....	67
5.2. Saran .....	68
DAFTAR PUSTAKA .....	69
Lampiran 1 .....	73
Lampiran 2 .....	75

## DAFTAR GAMBAR

Gambar 2. 1. Alur Proses Pelatihan model .....	16
Gambar 3. 1. Diagram alir alur kerja pengolahan data suara.....	33
Gambar 3. 2. Blok Diagram Sistem .....	36
Gambar 3.3. Blok Diagram Jaringan Saraf Tiruan - Model U-Net.....	39
Gambar 3.4. Diagram Alir Jaringan Syaraf Tiruan - Model U-Net CNN .....	45
Gambar 4. 1. Struktur Direktori Project .....	48
Gambar 4. 2. Alur Proses Teknis Program Reduksi Kebisingan.....	51
Gambar 4. 3. Waveform dan Spektogram Sinyal Suara Asli (Clean Speech).....	55
Gambar 4.4. Waveform dan Spektogram Sinyal Bising Asli (Factory Noise).....	56
Gambar 4.5 Waveform dan Spektogram Sinyal Input Model (Bising + Suara) ..	57
Gambar 4.6. Waveform dan Spektogram Sinyal Output Model (Hasil Proses)....	58
Gambar 4.7. Komunikasi Bersih Target .....	60
Gambar 4.8. Komunikasi didalam area pabrik.....	60
Gambar 4.9. Output setelah proses .....	61
Gambar 4.10. Tampilan Antar Muka Aplikasi .....	62

## DAFTAR RUMUS

(2.1) ReLU ( <i>Rectified Linear Unit</i> ).....	12
(2.2) Tanh ( <i>Hyperbolic Tangent</i> ).....	12
(2.3) Perhitungan Output Neuron.....	13
(2.4) Fungsi Loss ( <i>Error Function</i> ).....	13
(2.5) Turunan Parsial Bobot .....	14
(2.6) Update Bobot.....	15
(2.7) <i>Short-Time Fourier Transform (STFT)</i> .....	21
(2.8) <i>Sigmoid</i> .....	22
(2.9) <i>SNR Improvement</i> .....	26
(2.10) <i>Spectral Preservation</i> .....	26
(2.11) <i>Processing Speed</i> .....	27

## DAFTAR TABEL

Tabel 4. 1. Slider Strength.....	64
Tabel 4. 2. Hasil Evaluasi Metrik Kinerja.....	65

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Industri tekstil merupakan salah satu sektor manufaktur yang paling dinamis dan penting dalam perekonomian global. Peningkatan permintaan akan produk tekstil mengakibatkan peningkatan produksi pabrik tekstil. Volume kebisingan yang besar dihasilkan di pabrik-pabrik tekstil karena adanya mesin-mesin berkecepatan tinggi, seperti mesin berputar dan tenun, atau mesin pencelupan [1]. Bekerja di area pabrik ini dapat sulit akibat tingkat kebisingan yang tinggi, menimbulkan kekhawatiran tentang kesehatan dan keselamatan pekerja, serta komunikasi yang efektif di antara karyawan.

Lingkungan kerja berisiko karena suara berlebih yang dapat menyebabkan gangguan pendengaran, peningkatan tingkat stres, kelelahan, dan kemungkinan lebih besar terjadinya masalah kardiovaskular [1]. Selain berdampak pada kesehatan, tingkat kebisingan yang tinggi juga menghambat interaksi verbal di antara karyawan. Demi keselamatan, efisiensi, dan produktivitas dalam lingkungan produksi, komunikasi yang jelas dan efektif sangat penting. Komunikasi yang tepat membantu mencegah sejumlah besar kesalahan operasional dan insiden di tempat kerja, yang dapat memengaruhi produktivitas dan efektivitas fungsi pabrik[2].

Peralatan yang ditujukan untuk keselamatan pribadi, seperti penutup telinga dan sumbat telinga, serta penggunaan dinding dan sekat peredam suara, adalah metode yang banyak diterapkan untuk mengurangi kebisingan di lingkungan manufaktur tekstil [3]. Namun, strategi ini seringkali menunjukkan fleksibilitas terbatas dan hanya menghasilkan sedikit pengurangan tingkat kebisingan. Selain itu, penggunaan alat pelindung diri dalam jangka panjang dapat menyebabkan ketidaknyamanan bagi pekerja, yang pada gilirannya dapat memengaruhi kemampuan kerja mereka. Di sisi lain, alat pelindung diri seperti sumbat telinga dan penutup telinga juga dapat mengisolasi pekerja dari lingkungan sekitar dan mempersulit komunikasi verbal.

Dalam kasus di mana kapasitas pendengaran seseorang sangat berkurang, dapat dipahami bahwa pekerja mungkin akan mengalami kesulitan besar dalam memahami dan menafsirkan instruksi atau pesan penting dari rekan atau atasan mereka dengan efektif di dalam perusahaan. Hal ini dapat menyebabkan kerusakan komunikasi serius yang dapat membahayakan keselamatan dan kesejahteraan semua orang di lingkungan kerja[4]. Selain itu, sering kali dapat diamati bahwa hambatan suara dan perangkat untuk mengurangi kebisingan sering kali tidak memadai di berbagai bagian fasilitas, terutama di daerah yang membutuhkan mobilitas dan akses bebas yang substansial, sehingga meningkatkan kerumitan dalam melaksanakan strategi pengendalian kebisingan yang efektif di lingkungan aktif seperti pabrik tekstil [5].

Dengan kemajuan teknologi, munculnya kecerdasan buatan (AI), khususnya Jaringan Saraf Buatan (ANN), menyajikan cara baru dan inovatif untuk mengatasi

tantangan terkait kebisingan [4]. ANN berfungsi sebagai sistem komputasi yang dimodelkan berdasarkan jaringan saraf biologis, yang memungkinkannya untuk belajar dari pola data yang kompleks dan membuat penyesuaian adaptif [6]. Dalam hal mengurangi kebisingan, jaringan saraf buatan (ANN) dapat dilatih untuk mengenali pola kebisingan yang berbeda dan menghasilkan output yang membantu mengurangi suara yang tidak diinginkan secara instan.

Menggunakan ANN untuk pengurangan kebisingan di manufaktur tekstil membawa beberapa manfaat. Pertama, ANN memiliki kemampuan untuk terus memproses dan mengevaluasi data kebisingan, memungkinkan sistem menyesuaikan diri sesuai dengan kondisi lingkungan yang berubah-ubah. Selanjutnya, ANN dapat dengan mudah diintegrasikan dengan teknologi pemrosesan sinyal yang ada untuk meningkatkan efektivitas pengurangan kebisingan secara keseluruhan [7]. Terakhir, karena kemampuannya untuk belajar, sistem berbasis jaringan saraf buatan (ANN) dapat terus meningkatkan dirinya sendiri dari waktu ke waktu, menghasilkan solusi yang lebih efektif dalam jangka panjang.

Tujuan penelitian ini adalah untuk menciptakan model jaringan saraf buatan (ANN) yang efektif untuk meminimalkan polusi suara di lingkungan manufaktur tekstil, serta menilai dampaknya terhadap kejelasan komunikasi verbal di antara karyawan. Inisiatif ini tidak hanya bertujuan untuk memupuk lingkungan kerja yang lebih aman dan lebih menyenangkan, tetapi juga untuk meningkatkan interaksi di antara staf - sehingga meningkatkan produktivitas dan langkah-langkah keselamatan dalam lingkungan produksi tekstil.

Selain itu, penelitian ini berupaya memberikan kontribusi signifikan pada diskusi akademis tentang penerapan ANN untuk pengurangan kebisingan, khususnya dalam sektor tekstil, sambil memperdalam pemahaman tentang bagaimana teknologi canggih dapat mengatasi tantangan akustik dan komunikasi yang dihadapi di lingkungan pabrik [8]. Dengan demikian, investigasi ini memiliki implikasi praktis yang melampaui pengetahuan teoretis dengan menawarkan kemajuan potensial dalam standar keselamatan, tingkat kenyamanan pekerja, dan efektivitas operasional di tengah masalah kebisingan yang sering dijumpai selama berbagai proses produksi tekstil.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana pengembangan model Artificial Neural Network (ANN) yang efektif dapat mendukung pengurangan kebisingan di lingkungan industri tekstil?
2. Bagaimana pengaruh penerapan ANN terhadap peningkatan kejelasan komunikasi verbal, keselamatan, dan kenyamanan kerja di lingkungan bising?



### **1.3. Tujuan Penelitian**

Penelitian ini bertujuan untuk:

1. Mengembangkan model Artificial Neural Network (ANN) yang efektif untuk mengurangi kebisingan di lingkungan industri tekstil.
2. Mengevaluasi pengaruh penerapan ANN terhadap peningkatan kejelasan komunikasi verbal, keselamatan, dan kenyamanan kerja di lingkungan bising.

### **1.4. Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Teoritis: Menambah wawasan dan literatur akademik mengenai penggunaan ANN untuk pengurangan kebisingan di industri tekstil, serta memperkaya pemahaman tentang penerapan teknologi canggih dalam mengatasi tantangan kebisingan industri.
2. Manfaat Praktis: Memberikan solusi yang inovatif dan efektif untuk mengurangi kebisingan di pabrik tekstil, yang dapat meningkatkan keselamatan, kenyamanan kerja, dan produktivitas pekerja.

### **1.5. Batasan Penelitian**

Agar penelitian ini lebih fokus dan terarah, maka ditetapkan beberapa batasan penelitian sebagai berikut:

1. Penelitian ini hanya akan mengkaji pengurangan kebisingan di pabrik tekstil dengan menggunakan model ANN.
2. Evaluasi dampak penerapan ANN akan difokuskan pada aspek kejelasan komunikasi verbal dan keselamatan kerja di lingkungan pabrik tekstil, melalui data rekaman suara.
3. Data yang digunakan dalam penelitian ini akan diperoleh dari pabrik tekstil yang bersedia bekerja sama dalam penyediaan data kebisingan dan informasi terkait lainnya, yaitu PT. EMBEE PLUMBON TEKSTILE Unit 2.

### **1.6. Keaslian Penelitian**

Pada penelitian yang dilakukan oleh Wang, dkk. [9], dikembangkan sebuah sistem diagnosis kerusakan mesin (EFD) yang menggabungkan analisis intensitas suara, *Wavelet Packet Analysis* (WPA), dan *Artificial Neural Network* (ANN). Penelitian ini menggunakan sinyal kebisingan dari mesin untuk mengekstrak fitur-fitur kerusakan spesifik melalui WPA, yang efektif untuk memproses sinyal non-stasioner. Fitur-fitur tersebut kemudian dimasukkan ke dalam model ANN yang telah dilatih untuk melakukan klasifikasi dan mengenali berbagai pola kerusakan mesin. Hasil penelitian menunjukkan bahwa model WPA-ANN yang diusulkan sangat efektif dan akurat dalam mendiagnosis kerusakan mesin berdasarkan suara, membuktikan bahwa kombinasi teknik pemrosesan sinyal canggih dengan metode pembelajaran mesin merupakan pendekatan yang kuat untuk diagnosis di bidang permesinan.

Dalam penelitian yang dilakukan P. Zannin [10], dibahas pemanfaatan Jaringan Syaraf Tiruan (ANN) dalam simulasi penghalang suara yang efektif untuk mengurangi kebisingan hasil rekonstruksi jalan raya. Koefisien absorpsi material penghalang terbukti memiliki pengaruh signifikan terhadap redaman suara, sementara tinggi penghalang berkorelasi dengan pembentukan area bayangan akustik. Hasil simulasi menunjukkan bahwa tingkat kebisingan dapat dikurangi hingga mencapai batas yang dapat diterima secara hukum, sehingga menegaskan pentingnya desain penghalang suara yang optimal dalam mitigasi dampak kebisingan di lingkungan perkotaan.

Penelitian H. Zhang [11] mengkaji penerapan Deep Adaptive Noise Cancellation (Deep ANC), yang terbukti efektif dalam mengurangi kebisingan lebar pita serta mampu beradaptasi dengan baik terhadap kebisingan yang tidak terlatih. Metode ini mampu mempertahankan sinyal target dalam kondisi bising, seperti sinyal bicara, dan menunjukkan kinerja yang lebih baik dibandingkan metode ANC tradisional, terutama dalam situasi dengan distorsi nonlinier.

Selain itu, penelitian oleh S. A. D. Prasetyowati dkk. [12] mengeksplorasi pembatalan kebisingan generator yang bersifat monoton menggunakan algoritma *Least Mean Square* (LMS) adaptif. Studi ini menganalisis karakteristik kebisingan melalui statistik, FFT/IFFT, dan distribusi frekuensi sisa kebisingan. Hasilnya menunjukkan bahwa pembatalan kebisingan dengan LMS adaptif dapat terwujud dengan baik, meskipun ada sisa kebisingan yang terdistribusi normal dan tidak mengganggu.

Terinspirasi oleh penelitian-penelitian menjanjikan tersebut, penelitian ini berfokus pada pengembangan model ANN yang spesifik untuk industri tekstil. Berbeda dengan penelitian sebelumnya, penelitian ini membahas masalah kebisingan di pabrik tekstil dengan pendekatan baru. Fokusnya yang spesifik pada industri tekstil menjadi nilai unik yang membedakannya dari penelitian pengurangan kebisingan menggunakan ANN sebelumnya.

Tujuan utama penelitian ini—yaitu meningkatkan kejelasan komunikasi verbal di antara pekerja—merupakan aspek penting yang belum banyak diteliti sebelumnya. Penggunaan *supervised learning neural network* dan pengembangan model ANN yang dirancang khusus untuk mengurangi kebisingan dan meningkatkan komunikasi verbal di pabrik tekstil menjadi kontribusi orisinal dari penelitian ini. Selain itu, penelitian ini akan diterapkan langsung di lingkungan pabrik tekstil yang nyata, memungkinkan evaluasi efektivitas model ANN dalam situasi sebenarnya. Pengembangan lebih lanjut dari penelitian ini dapat meliputi analisis jenis kebisingan spesifik di pabrik tekstil, evaluasi performa model ANN dengan berbagai metrik, serta pertimbangan implikasi etika dari penerapan model ANN tersebut.

### **1.7. Sistematika Penulisan**

Sistematika penulisan tesis ini adalah sebagai berikut:

#### **Bab 1**

Pendahuluan – Membahas latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.

#### **Bab 2**

Tinjauan Pustaka – Mengulas teori-teori yang relevan dengan penelitian ini, termasuk konsep dasar kebisingan, komunikasi verbal di lingkungan kerja, dan aplikasi ANN.

#### **Bab 3**

Metodologi Penelitian – Menjelaskan metode penelitian yang digunakan, termasuk desain penelitian, teknik pengumpulan data, dan metode analisis data.

#### **Bab 4**

Hasil dan Pembahasan – Menyajikan hasil penelitian dan analisis data, serta pembahasan mengenai temuan-temuan penelitian.

#### **Bab 5**

Kesimpulan dan Saran – Menyimpulkan hasil penelitian dan memberikan saran-saran yang relevan berdasarkan temuan penelitian.

Dengan sistematika tersebut, diharapkan penelitian ini dapat tersusun secara sistematis dan memberikan kontribusi yang signifikan dalam bidang pengurangan kebisingan industri, khususnya di sektor tekstil.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1. Tinjauan Pustaka**

Kebisingan merupakan salah satu masalah utama di industri tekstil yang dapat berdampak negatif pada kesehatan pekerja dan efektivitas komunikasi verbal. Menurut penelitian oleh Nada et al. (2014)[13], tingkat kebisingan di pabrik tekstil seringkali melebihi batas aman yang ditetapkan oleh organisasi kesehatan, yang dapat menyebabkan gangguan pendengaran dan stres bagi pekerja. Abraham et al. (2019)[14] menambahkan bahwa kebisingan tinggi juga menghambat komunikasi verbal yang efektif, yang penting untuk koordinasi dan keselamatan di lingkungan kerja.

##### **2.1.1. Metode Pengurangan Kebisingan**

Berbagai metode telah dikembangkan untuk mengurangi kebisingan di lingkungan industri. Setyaningrum et al. (2019)[15] membahas analisis isolasi penggunaan bahan penyerap suara sebagai solusi untuk meredam kebisingan. Namun, metode ini memiliki keterbatasan dalam hal biaya dan efektivitas. Penelitian oleh Subandrio et al. (2023)[16] menunjukkan bahwa penggunaan perangkat lunak pemrosesan sinyal digital dapat memberikan hasil yang lebih baik dalam mengurangi kebisingan dibandingkan dengan metode fisik tradisional.

##### **2.1.2. Artificial Neural Network dalam Pemrosesan Sinyal**

Artificial Neural Network (ANN) telah digunakan secara luas dalam berbagai aplikasi pemrosesan sinyal, termasuk pengurangan kebisingan. Syabila et al. (2023)[17] menunjukkan bahwa ANN dapat dilatih untuk mengenali sinyal audio, sehingga meningkatkan kejelasan sinyal yang diinginkan. Krisna et al. (2018)[18] mengembangkan model ANN yang dapat mengenali data suara yang mengalami *noise* dengan menggunakan jaringan Syaraf tiruan Hebb pada Tingkat pengenalan 87,5%.

##### **2.1.3. Penerapan ANN untuk Pengurangan Kebisingan di Pabrik Tekstil**

Penerapan Artificial Neural Network (ANN) dalam pengurangan kebisingan di pabrik tekstil menunjukkan potensi besar, seperti yang ditunjukkan oleh penelitian Kwon et al. [19]. Penelitian tersebut mengimplementasikan Long Short-Term Memory (LSTM), jenis ANN, dalam algoritma kontrol kebisingan aktif (ANC) untuk memprediksi dan meminimalkan kebisingan mesin diesel secara efektif, bahkan lebih baik dari algoritma konvensional. LSTM juga menunjukkan respons cepat terhadap perubahan kebisingan tanpa memerlukan waktu adaptasi. Hal ini membuktikan bahwa teknologi LSTM, serupa dengan ANN, dapat meningkatkan kejelasan komunikasi di lingkungan pabrik tekstil dengan memisahkan suara manusia dari kebisingan mesin, sehingga meningkatkan kenyamanan dan produktivitas.

Pemilihan algoritma *supervised learning neural network* sangat tepat untuk pengurangan kebisingan di pabrik tekstil karena beberapa alasan utama:

1. Pembelajaran dari Data Berlabel:

Jaringan saraf memerlukan dataset yang telah dilabeli (input dengan output yang diketahui). Dalam pengurangan kebisingan, model belajar membedakan antara kebisingan dan suara yang relevan dari rekaman suara bising dan suara yang diinginkan.

2. Kemampuan Generalisasi

Setelah dilatih, model dapat mengenali pola kebisingan dalam situasi baru dan diterapkan pada kondisi operasional pabrik yang bervariasi.

3. Pengolahan Data Non-Linear

Jaringan saraf mampu menangani hubungan non-linear yang kompleks dan tidak teratur pada karakteristik kebisingan pabrik, memberikan solusi yang lebih efektif daripada metode linear tradisional.

4. Adaptasi terhadap Perubahan

Algoritma dapat diperbarui dengan data baru secara berkala, memungkinkan model beradaptasi dengan perubahan kondisi kebisingan pabrik dan tetap efektif.

5. Prediksi Real-Time

Jaringan saraf dapat diimplementasikan untuk respons *real-time* dalam mengidentifikasi dan mengurangi kebisingan yang muncul, menciptakan lingkungan kerja yang lebih nyaman.

6. Fleksibilitas Arsitektur

*Supervised learning neural network* fleksibel dalam desain arsitektur, memungkinkan penggunaan Convolutional Neural Networks (CNN) untuk analisis spektrum frekuensi atau Recurrent Neural Networks (RNN) untuk data urutan waktu, disesuaikan dengan kebutuhan spesifik pengurangan kebisingan.

Dengan mempertimbangkan semua karakteristik ini, algoritma *supervised learning neural network* menjadi pilihan yang sangat efektif untuk mengatasi masalah kebisingan di pabrik tekstil, memberikan solusi berbasis data yang adaptif terhadap kondisi yang berubah.

## 2.2. Dasar Teori

Pekerja di sektor tekstil jauh lebih rentan terhadap paparan kebisingan, risiko umum yang terkait dengan bidang ini [20]. Sumber utama kebisingan selama pemintalan benang berasal dari peralatan yang digunakan dalam produksi, termasuk mesin pemintalan (TFO, atau two-for-one twist) dan mesin lain yang beroperasi terus-menerus [21]. Jika dibandingkan dengan industri lain, individu yang bekerja di tekstil sering kali menghadapi tingkat kebisingan yang lebih tinggi yang berasal dari karakteristik metode produksi mereka dan ketergantungan yang besar pada mesin[22]. Paparan kebisingan tingkat tinggi yang berkepanjangan dapat

berdampak buruk pada kesehatan pekerja, yang mengakibatkan peningkatan tingkat stres, potensi kehilangan pendengaran, dan penurunan efisiensi [23].

### **2.2.1. Definisi Kebisingan**

Kebisingan adalah suara yang tidak diinginkan dan mengganggu yang dapat menyebabkan berbagai efek buruk pada kesehatan manusia dan mengganggu aktivitas sehari-hari [22]. Kebisingan diukur dalam desibel (dB), dengan tingkat kebisingan yang melebihi 85 dB dianggap berbahaya jika terpapar terus-menerus. Di sektor tekstil, tingkat kebisingan yang tinggi dapat mengakibatkan stres kronis di antara pekerja, mengganggu konsentrasi, dan menurunkan produktivitas secara keseluruhan [24]. Lebih jauh lagi, paparan kebisingan yang berkepanjangan juga dapat memicu masalah emosional seperti mudah tersinggung dan depresi. Organisasi Kesehatan Dunia (WHO) menyatakan bahwa paparan kebisingan yang berlebihan dalam jangka panjang dapat menyebabkan gangguan pendengaran bersama dengan masalah kesehatan lainnya seperti stres, gangguan tidur, dan masalah kardiovaskular.

### **2.2.2. Sumber Kebisingan di Pabrik Tekstil**

Di pabrik tekstil yang mengkhususkan diri dalam pemintalan benang katun, kebisingan merupakan masalah utama yang dihasilkan oleh berbagai peralatan dengan karakteristik pengoperasian yang berbeda. Sumber-sumber utama kebisingan berasal dari mesin-mesin berkecepatan tinggi seperti Twisting Mill (TFO) dan Mesin Pemintal Cincin, yang menghasilkan suara kontinu berintensitas tinggi [21]. Mesin lain seperti Carding Machine dan Mesin Sisir juga berkontribusi pada tingkat kebisingan keseluruhan melalui gerakan cepat dan operasional yang berat, meskipun karakteristik frekuensinya mungkin berbeda. Sementara itu, peralatan seperti

Mesin Penarik, Mesin Roving, dan Winder juga menambahkan kebisingan signifikan ke lingkungan pabrik. Secara keseluruhan, aktivitas mekanis yang intens dari berbagai mesin ini menciptakan lingkungan yang sangat bising, yang memerlukan strategi pengendalian komprehensif, termasuk penerapan teknologi pengurangan kebisingan canggih.

Karena aktivitas mekanis yang intens, pabrik tekstil menghasilkan banyak kebisingan selama pengoperasian. Secara keseluruhan, pengendalian kebisingan di pabrik benang katun memerlukan strategi menyeluruh, termasuk memilih mesin yang lebih senyap, mengatur ulang tata letak pabrik, dan memastikan bahwa karyawan mengenakan alat pelindung diri [22]. Penelitian dan pengembangan lebih lanjut tentang perangkat pengurangan kebisingan yang lebih canggih juga diperlukan untuk membuat tempat kerja lebih aman dan nyaman.

### **2.2.3. Dampak Kebisingan pada Pekerja**

Paparan kebisingan tinggi di lingkungan pabrik tekstil dapat menyebabkan berbagai dampak negatif pada kesehatan fisik, mental, dan kesejahteraan umum pekerja. Dampak-dampak ini meliputi:

#### **1. Kehilangan Pendengaran**

Paparan kebisingan berkepanjangan dapat merusak sel-sel rambut di telinga bagian dalam, menyebabkan kehilangan pendengaran permanen dan tinitus (suara berdengung di telinga), yang berdampak pada kualitas hidup dan komunikasi efektif [25].

#### **2. Stres Mental**

Kebisingan berlebihan menyebabkan stres psikologis parah, mengurangi konsentrasi, produktivitas, dan meningkatkan kemungkinan kesalahan di tempat kerja, serta memicu kelelahan mental berlebihan [26].

#### **3. Kelelahan Fisik**

Kebisingan konstan memaksa tubuh beradaptasi secara biologis, menguras energi dan menyebabkan kelelahan kronis [26].

#### **4. Dampak pada Kesehatan Mental**

Stres akibat kebisingan dapat menimbulkan gejala seperti kecemasan, kesedihan, dan insomnia [26].

#### **5. Masalah Kardiovaskular**

Kebisingan keras dapat meningkatkan tekanan darah (melalui pelepasan hormon stres seperti kortisol) dan risiko penyakit jantung koroner, serangan jantung, serta aritmia jantung [27].

### **2.2.3.1. Pentingnya komunikasi verbal**

Komunikasi verbal yang efektif sangat penting untuk koordinasi tim yang baik, terutama dalam situasi yang membutuhkan reaksi cepat dan kerja sama tim yang kuat. Komunikasi yang jelas dan akurat dapat mencegah kesalahan operasional serta menjamin keselamatan dan efisiensi tempat kerja. Hal ini esensial untuk:

#### **1. Koordinasi Tim**

Memastikan semua pekerja di industri tekstil memahami peran dan tanggung jawab mereka, menghindari kesalahpahaman yang mengganggu alur kerja.

#### **2. Petunjuk dan Perintah**

Memungkinkan pemberian instruksi yang akurat dan mudah dipahami, mengurangi risiko kesalahan operasional yang memengaruhi kualitas dan efisiensi produksi.

#### **3. Peringatan Keselamatan**

Memastikan peringatan bahaya atau darurat dikomunikasikan dan dipahami secara efektif oleh semua pekerja untuk melindungi diri mereka.

#### **2.2.3.2. Tantangan Komunikasi di Lingkungan Bising**

Tingkat kebisingan tinggi di pabrik tekstil menghambat komunikasi verbal, menyebabkan kesalahpahaman, kecelakaan, dan mengurangi efisiensi operasional. Karyawan kesulitan memahami instruksi dan peringatan penting, yang meningkatkan risiko keselamatan dan berdampak pada koordinasi tim. Tantangan komunikasi meliputi:

##### **1. Gangguan Akustik**

Kebisingan tinggi menyebabkan suara lisan sulit didengar, berpotensi mengakibatkan kesalahan operasional dan kecelakaan, terutama saat respons cepat diperlukan.

##### **2. Penggunaan Alat Pelindung Diri (APD)**

APD efektif melindungi pendengaran, namun menghalangi instruksi dan peringatan penting, mempersulit komunikasi verbal dan meningkatkan risiko keselamatan.

##### **3. Risiko Keselamatan**

Kegagalan mendengar instruksi dan peringatan keselamatan meningkatkan risiko kecelakaan kerja, seperti tidak menanggapi kerusakan mesin atau bahaya bahan.

##### **4. Efisiensi Operasional**

Miskomunikasi mengganggu alur kerja, menyebabkan penundaan, dan mengurangi efisiensi. Solusi seperti interkom atau *headset* peredam bising diperlukan untuk komunikasi efektif di lingkungan bising, yang pada akhirnya meningkatkan keselamatan dan efisiensi.

#### **2.2.4. Pendekatan Tradisional terhadap Pengurangan Kebisingan**

Pendekatan tradisional untuk pengurangan kebisingan di pabrik tekstil melibatkan penggunaan alat pelindung diri (APD) dan insulasi suara. Tujuannya adalah melindungi pekerja dari dampak negatif kebisingan.

##### **1. Alat Pelindung Diri (APD)**

Penggunaan penyumbat telinga dan penutup telinga umum dilakukan untuk membatasi tingkat suara yang mencapai telinga, sehingga melindungi pendengaran pekerja dari kerusakan. APD secara signifikan mengurangi tingkat kebisingan dan sangat penting di lingkungan kerja yang bising untuk mencegah gangguan pendengaran permanen. Penggunaan jangka panjang dapat menyebabkan ketidaknyamanan (misalnya, iritasi kulit). APD juga dapat mengisolasi pekerja dari suara penting seperti alarm atau peringatan darurat, meningkatkan risiko kecelakaan, dan mempersulit komunikasi verbal, yang memengaruhi koordinasi.

##### **2. Kedap Suara dan Insulasi**

Metode ini menggunakan dinding dan insulasi suara untuk mengurangi penyebaran suara dan menciptakan lingkungan kerja yang lebih tenang. Insulasi suara bekerja dengan menyerap dan memblokir gelombang suara, efektif terutama di area pabrik dengan konsentrasi kebisingan tinggi seperti dekat mesin besar.



Solusi ini tidak selalu praktis atau efektif di semua area. Pemasangan penghalang kebisingan tidak cocok untuk area yang memerlukan mobilitas tinggi dan akses terbuka. Biaya pemasangan dan pemeliharaan juga bisa mahal, dan hambatan kebisingan dapat menghambat alur kerja serta mengurangi fleksibilitas tata letak pabrik.

#### **2.2.5. Pendekatan inovatif dalam pengurangan kebisingan**

Selain metode tradisional, pendekatan inovatif dalam pengurangan kebisingan di pabrik tekstil meliputi teknologi peredam kebisingan aktif dan penggunaan material baru. Teknologi Peredam Kebisingan Aktif (Active Noise Cancellation - ANC), memanfaatkan mikrofon dan *speaker* untuk mendeteksi dan membatalkan kebisingan dengan menghasilkan gelombang suara yang saling berlawanan. Teknologi ini dapat diterapkan di berbagai area pabrik tanpa perubahan signifikan pada struktur fisik. Namun, penerapannya masih dalam tahap pengembangan dan memerlukan investasi awal yang signifikan.

Bahan Inovatif untuk Pengurangan Kebisingan Penggunaan material inovatif dengan kapasitas penyerapan kebisingan tinggi menawarkan solusi yang layak. Bahan seperti busa akustik, papan serat, dan komposit dapat digunakan di dinding dan langit-langit pabrik untuk mengurangi pantulan suara dan menurunkan tingkat kebisingan. Material yang mudah dipasang dan dilepas juga memungkinkan perencanaan peralatan yang fleksibel. Dengan mengadopsi metode yang sudah mapan dan baru, pabrik tekstil dapat mempertahankan efisiensi operasional tinggi serta meningkatkan keselamatan dan kenyamanan tenaga kerjanya.

#### **2.2.6. Arsitektur Convolutional Neural Network sebagai Fondasi Model**

Penelitian ini secara spesifik mengadopsi arsitektur *Convolutional Neural Network* (CNN), sebuah kelas khusus dari Jaringan Saraf Tiruan (ANN) yang fundamentalnya merupakan kerangka komputasi yang meniru cara kerja jaringan saraf biologis di otak manusia [28]. Berbeda dengan ANN konvensional, CNN dirancang secara khusus untuk memproses data yang memiliki topologi seperti grid, misalnya citra. Dalam konteks penelitian ini, sinyal audio yang telah ditransformasikan menjadi representasi spektrogram diperlakukan sebagai 'citra' 2D. Pendekatan ini memungkinkan model untuk mengeksplorasi keunggulan utama CNN, yaitu kemampuannya untuk mengenali pola-pola spasial hierarkis—seperti struktur harmonik dan formant pada spektrogram—yang sulit dideteksi oleh metode analisis sinyal tradisional.

Kemampuan CNN untuk belajar dan membuat keputusan berakar pada prinsip ANN, di mana jaringan belajar mengenali pola dari data input yang kompleks, berkembang berdasarkan pengalaman, dan membuat prediksi [29]. Arsitektur CNN yang mendalam biasanya terdiri dari tiga jenis lapisan utama: lapisan input, lapisan tersembunyi (*hidden layers*), dan lapisan output [30]. Lapisan tersembunyi pada CNN didominasi oleh lapisan konvolusional yang menerapkan serangkaian filter untuk mengekstraksi fitur, dan lapisan *pooling* yang mereduksi

dimensi spasial untuk efisiensi komputasi [31]. Melalui serangkaian lapisan ini, CNN mampu membangun representasi fitur dari level rendah (misalnya, tepi atau tekstur pada spektrogram) hingga level tinggi (misalnya, pola vokal yang kompleks). Proses pembelajaran ini dimungkinkan melalui mekanisme penyempurnaan bobot filter secara iteratif berdasarkan kesalahan prediksi, yang umumnya menggunakan algoritma *backpropagation* [28].

#### 2.2.6.1. Prinsip *Supervised Learning* pada CNN

*Supervised learning* merupakan paradigma pembelajaran mesin yang digunakan dalam penelitian ini. Pendekatan ini melibatkan pelatihan model menggunakan *dataset* yang telah dilabeli, di mana setiap data input (spektrogram bising) memiliki pasangan output target yang diketahui (spektrogram bersih) [32]. Dengan memberikan pasangan data ini, model CNN secara iteratif belajar untuk memetakan input ke output yang diinginkan. Dalam konteks reduksi kebisingan, model dilatih untuk mengestimasi sebuah masker spektral yang, ketika diaplikasikan pada spektrogram bising, akan menghasilkan spektrogram yang semirip mungkin dengan spektrogram bersih target. Proses pelatihan dan prediksi ini didasari oleh prinsip matematis yang kuat untuk meminimalkan fungsi kesalahan (*loss function*) antara prediksi dan target [33].

#### 2.2.6.2. Rumus-Rumus dalam *Supervised Learning* pada ANN

##### 1. ReLU (Rectified Linear Unit)

ReLU populer karena kemampuannya untuk menghindari masalah vanishing gradient dan sering digunakan di hidden layer.

$$f(x) = \begin{cases} x & \text{jika } x > 0 \\ 0 & \text{jika } x \leq 0 \end{cases} \quad (2.1)$$

Penjelasan:

- ReLU hanya mengembalikan nilai input jika positif, dan nol jika negatif.
- Sangat populer karena komputasi cepat dan kemampuan untuk menghindari masalah vanishing gradient dibandingkan sigmoid dan tanh.
- Umumnya digunakan di hidden layer jaringan dalam pemrosesan audio, termasuk pengurangan kebisingan.
- Dalam penelitian ini, ReLU bisa digunakan untuk menangkap pola frekuensi yang relevan dari sinyal suara.

##### 2. Tanh (Hyperbolic Tangent)

Tanh mirip dengan sigmoid, tetapi menghasilkan nilai antara -1 hingga 1, membantu dalam normalisasi data internal jaringan.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

Penjelasan:

- a. Mirip dengan sigmoid, tapi menghasilkan nilai antara -1 hingga 1 , sehingga lebih baik untuk normalisasi data internal.
- b. Digunakan ketika kita ingin neuron dapat merepresentasikan polaritas (misalnya: frekuensi tinggi vs rendah).
- c. Dalam konteks penelitian ini, tanh bisa digunakan untuk membedakan komponen frekuensi yang dominan dalam kebisingan mesin tekstil .

### 3. Perhitungan Output Neuron

Output dari suatu neuron dihitung sebagai:

$$y_j = f \sum_{i=1}^n w_{ij}x_i + b_j \quad (2.3)$$

Penjelasan:

- a.  $y_j$  = output dari neuron ke- $j$ ,
- b.  $w_{ij}$  = bobot koneksi antara neuron input  $i$  dan neuron  $j$ ,
- c.  $x_i$  = nilai input dari neuron  $i$ ,
- d.  $b_j$  = bias pada neuron  $j$ ,
- e.  $f$  = fungsi aktivasi.
- f. Setiap neuron menerima beberapa input ( $x_i$ ), yang masing-masing dikalikan dengan bobot ( $w_{ij}$ ).
- g. Hasilnya dijumlahkan lalu ditambah bias ( $j$ ), dan hasil akhirnya dilewatkan ke fungsi aktivasi .
- h. Ini adalah proses dasar bagaimana neuron memproses informasi dan menghasilkan prediksi.
- i. Dalam penelitian ini, setiap neuron memproses fitur audio frekuensi dan amplitudo dari rekaman suara.

### 4. Fungsi Loss (Error Function)

Untuk mengukur kesalahan prediksi selama pelatihan, digunakan fungsi loss seperti Mean Squared Error (MSE):

$$E = \frac{1}{m} \sum_{k=1}^m (t_k - y_k)^2 \quad (2.4)$$

Penjelasan:

- a.  $E$ : error total,
- b.  $t_k$ : target aktual (suara verbal bersih),
- c.  $y_k$ : output prediksi (suara hasil pengolahan ANN),
- d.  $m$ : jumlah data pelatihan.
- e. MSE mengukur rata-rata kuadrat selisih antara prediksi model ( $y_k$ ) dan nilai sebenarnya ( $t_k$ ).

- f. Semakin kecil nilai MSE, semakin baik model ANN dalam memprediksi sinyal verbal yang bersih dari kebisingan.

Penggunaan *Mean Squared Error* (MSE) sebagai fungsi loss sangat relevan dalam tugas ini karena tujuan utama model adalah meregresi atau mengestimasi nilai magnitudo dari spektrogram ucapan yang bersih. Dengan meminimalkan rata-rata kuadrat selisih antara magnitudo spektrogram yang diprediksi oleh model ( $y_k$ ) dan magnitudo spektrogram target ( $t_k$ ), proses pelatihan secara langsung mendorong bobot jaringan untuk menghasilkan sebuah output yang secara struktural dan numerik semirip mungkin dengan representasi spektral dari sinyal ucapan yang ideal.

## 5. Backpropagation dan Update Bobot

Backpropagation adalah mekanisme pembelajaran paling penting dalam ANN, yaitu cara sistem "belajar dari kesalahan" dan menyesuaikan diri. Bayangkan ketika sedang melatih sebuah sistem untuk mengenali pola suara. Pertama, sistem mencoba memprediksi (melalui *Forward Propagation*). Kemudian, ia menghitung seberapa besar kesalahannya (menggunakan Fungsi *Loss* seperti MSE). Setelah itu, Backpropagation mengambil kesalahan ini dan "menyebarkannya" kembali ke seluruh jaringan, dari lapisan *output* hingga lapisan *input*. Tujuannya adalah untuk mencari tahu "bobot" mana di setiap koneksi yang paling berkontribusi terhadap kesalahan tersebut.

Selama proses ini, bobot diperbarui menggunakan algoritma backpropagation. Perubahan bobot dihitung dengan:

### A. Turunan Parsial Bobot

$$\Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (2.5)$$

Penjelasan:

1.  $\eta$  = *Learning rate* menentukan seberapa besar perubahan bobot dalam satu iterasi pelatihan. Ini adalah faktor penting yang mengontrol seberapa "agresif" model dalam mengubah bobotnya. *Learning rate* yang terlalu besar bisa membuat model "melompat" melewati solusi terbaik, sementara yang terlalu kecil bisa membuat proses belajar sangat lambat.
2.  $\frac{\partial E}{\partial w_{ij}}$  = Turunan parsial dari *error* terhadap bobot menunjukkan kontribusi tiap bobot terhadap kesalahan. Nilai ini memberi tahu arah dan besaran perubahan yang ideal untuk bobot tertentu agar *error* berkurang.
3.  $w_{ij}^{baru}$  = Bobot baru setelah diperbarui. Ini adalah nilai bobot yang telah disesuaikan berdasarkan perhitungan *error* dan *learning rate*, siap digunakan untuk iterasi pelatihan berikutnya.

4.  $w_{ij}^{lama}$  = Bobot sebelum diperbarui. Ini adalah nilai bobot dari iterasi pelatihan sebelumnya.

Untuk menghitung  $\frac{\partial E}{\partial w_{ij}}$  (yaitu, seberapa besar perubahan bobot  $w_{ij}$  harus dilakukan agar *error*  $E$  berkurang), kita perlu tahu bagaimana perubahan kecil pada bobot itu memengaruhi *output* neuron, yang pada gilirannya memengaruhi *output* lapisan berikutnya, dan akhirnya memengaruhi *error* total. Karena efek ini berantai (seperti efek domino), kita menggunakan aturan matematika yang disebut "**Chain Rule**". Aturan ini memungkinkan kita menghitung turunan (tingkat perubahan) sebuah fungsi yang tersusun dari beberapa fungsi lainnya, satu per satu dari belakang ke depan. Jadi, kita mulai menghitung turunan *error* terhadap *output* terakhir, lalu *output* terakhir terhadap *input* sebelumnya, dan seterusnya, sampai ke bobot yang ingin kita perbarui. Ini memastikan bahwa penyesuaian bobot dilakukan secara akurat, berdasarkan kontribusi spesifiknya terhadap kesalahan total jaringan.

#### B. Update Bobot

$$w_{ij}^{baru} = w_{ij}^{lama} + \Delta w_{ij} \quad (2.6)$$

Proses Update Bobot (Integrasi): Formula (2.7) adalah intinya. Bobot lama ( $w_{ij}^{lama}$ ) ditambahkan dengan nilai perubahan bobot ( $\Delta w_{ij}$ ) yang dihitung berdasarkan turunan *error* dan *learning rate*. Dengan kata lain, model mengambil bobot yang ada, menghitung seberapa banyak dan ke arah mana bobot itu harus berubah untuk mengurangi kesalahan, dan kemudian menerapkan perubahan tersebut untuk mendapatkan bobot baru. Proses ini dilakukan berulang kali selama pelatihan, secara bertahap "mengukir" bobot jaringan agar semakin akurat dalam memprediksi sinyal ucapan yang bersih dari kebisingan. Dalam konteks penelitian ini, *backpropagation* digunakan untuk menyesuaikan bobot neuron agar ANN lebih tepat dalam memisahkan suara bising dari suara manusia.

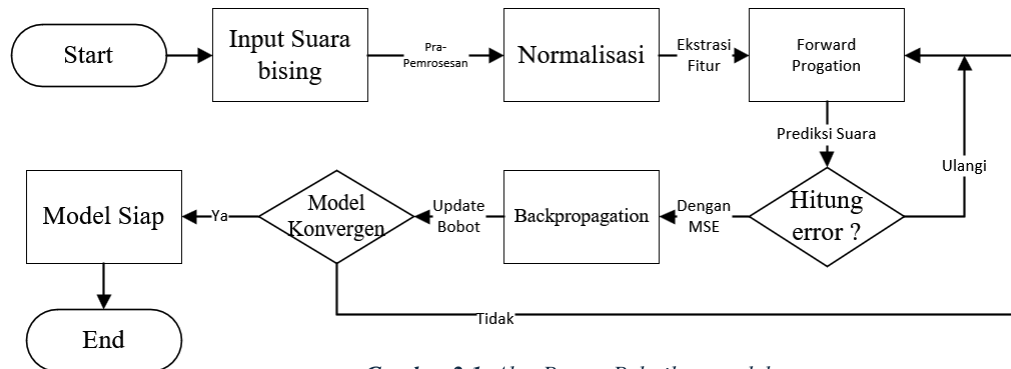
#### 2.2.6.3. Proses Pelatihan Supervised Learning pada (ANN)

Proses pelatihan *Supervised Learning* dalam pengenalan suara bising melibatkan beberapa tahapan penting yang berlangsung secara berurutan dan berulang[34]. Dimulai dari pra-pemrosesan data, dilanjutkan dengan forward propagation, perhitungan error, hingga proses backpropagation untuk memperbaiki bobot, seluruh langkah ini bertujuan untuk melatih model agar semakin akurat dalam menghasilkan prediksi hingga mencapai konvergensi atau batas maksimum epoch. Tahapan-tahapan tersebut dapat dilihat secara lebih jelas pada Gambar 2.1, yang menggambarkan alur proses pelatihan model *Supervised Learning* dari awal hingga akhir.

Untuk memperoleh pemahaman yang lebih jelas mengenai tahapan-tahapan proses pelatihan *Supervised Learning* pada *Artificial Neural Network*, silakan merujuk pada Gambar 2.1.

Alur Proses Pelatihan model yang mengilustrasikan urutan langkah-langkah yang akan dijelaskan secara terperinci. Bayangkan proses ini seperti melatih seorang siswa untuk membedakan antara suara musik dan suara kebisingan.

Gambar 2.1 mengilustrasikan alur kerja fundamental dari *supervised learning* yang digunakan untuk melatih model ANN. Berikut adalah penjelasan rinci untuk setiap tahapan kunci dalam konteks penelitian reduksi kebisingan ini:



Gambar 2.1. Alur Proses Pelatihan model

### 1. Mulai (Start)

Ini adalah titik awal proses pelatihan model *Supervised Learning*. Model siap untuk menerima data dan memulai pembelajarannya.

### 2. Input Suara Bising (Input)

Pada tahap ini, model menerima data suara yang masih mengandung kebisingan. Bayangkan ini seperti rekaman suara di pabrik yang penuh dengan suara mesin dan suara orang berbicara. Ini adalah "tugas" atau "soal" yang diberikan kepada siswa.

### 3. Normalisasi (Pra-pemrosesan)

Normalisasi adalah langkah pra-pemrosesan data yang krusial sebelum data dimasukkan ke dalam model deep learning. Dalam pemrosesan sinyal audio, tujuan utamanya ada dua:

- Stabilitas Pelatihan, Sinyal audio mentah memiliki rentang amplitudo yang dapat berbeda secara signifikan antar sampel. Perbedaan ini dapat menyebabkan satu sinyal dengan amplitudo tinggi mendominasi proses pembelajaran, sehingga pembaruan bobot menjadi tidak stabil. Dengan melakukan normalisasi, rentang nilai amplitudo diseragamkan sehingga model dapat mempelajari pola secara lebih konsisten.
- Efisiensi Konvergensi, Normalisasi yang mengubah data ke rentang tertentu, seperti  $[-1,1]$ , memastikan bahwa nilai input berada dalam rentang kerja optimal dari fungsi aktivasi, misalnya Sigmoid atau Tanh. Kondisi ini membantu mencegah permasalahan seperti *vanishing gradient* dan *exploding gradient*, sehingga proses pembelajaran dapat berlangsung lebih efisien dan model mencapai konvergensi lebih cepat.

Dalam penelitian ini, normalisasi dilakukan pada sinyal audio berbentuk *waveform* (sinyal satu dimensi) setelah melalui tahap *resampling*. Proses ini mencakup penskalaan amplitudo setiap sinyal sehingga nilai puncaknya mencapai 1.0 (*peak normalization*), atau menggunakan fungsi `librosa.util.normalize` yang melakukan penyesuaian berdasarkan nilai *root mean square* (RMS) dari sinyal. Langkah ini bertujuan untuk memastikan bahwa seluruh sampel audio memiliki tingkat amplitudo yang setara, sehingga tidak ada sinyal yang secara tidak proporsional lebih kuat dan berpotensi memengaruhi proses pembelajaran jaringan secara bias.

#### 4. Ekstraksi Fitur (Ekstraksi Fitur)

Setelah dinormalisasi, sinyal suara diolah untuk "mengekstrak" fitur-fitur penting darinya. Ini seperti mengambil ciri-ciri khas dari suara tersebut (misalnya pola frekuensi dan amplitudo dari waktu ke waktu), yang kemudian akan menjadi "gambar" atau representasi visual (seperti spektrogram) yang bisa dimengerti oleh jaringan saraf.

#### 5. Forward Propagation

Fitur-fitur yang sudah diekstrak ini kemudian "dimasukkan" ke dalam jaringan saraf. Ini adalah proses di mana informasi bergerak maju melalui setiap lapisan neuron dalam jaringan, dari lapisan input, melalui lapisan tersembunyi, hingga mencapai lapisan output. Di sini, jaringan saraf membuat "prediksi suara", yaitu mencoba menghasilkan suara yang menurutnya bersih dari kebisingan.

#### 6. Hitung Error? (Dengan MSE)

Tahap "Hitung Error" merupakan bagian inti dari *supervised learning*. Setelah model menghasilkan prediksi melalui *forward propagation*, langkah ini digunakan untuk mengukur selisih antara hasil prediksi dan *ground truth*. Nilai selisih tersebut menjadi indikator seberapa besar kesalahan model dan menentukan arah pembelajaran selanjutnya.

Dalam penelitian ini digunakan *Mean Squared Error* (MSE) sebagaimana dirumuskan pada Rumus (2.4). Metrik ini menghitung rata-rata selisih kuadrat antara nilai target dan nilai prediksi.

Konteks dalam Penelitian Ini

Pada pemrosesan spektrogram:

- a.  $t_k$  : merepresentasikan magnitudo spektrogram dari sinyal ucapan bersih.
- b.  $y_k$  : merupakan magnitudo spektrogram hasil prediksi model U-Net.
- c. MSE : menghitung perbedaan tersebut pada setiap elemen frekuensi-waktu, kemudian merata-ratakannya untuk mendapatkan satu nilai error keseluruhan.

Semakin kecil nilai MSE, semakin baik kemampuan model dalam merekonstruksi sinyal yang mendekati sinyal bersih.

### 7. *Backpropagation* (Update Bobot)

Backpropagation digunakan untuk memperbaiki kesalahan yang diukur oleh MSE dengan menyesuaikan bobot-bobot jaringan agar error dapat berkurang pada iterasi berikutnya.

Backpropagation berlangsung melalui langkah-langkah berikut:

#### a. Propagasi Mundur

Nilai error dari langkah sebelumnya disebarkan kembali melalui seluruh lapisan jaringan, dari lapisan output menuju lapisan sebelumnya.

#### b. Perhitungan Gradien

Kontribusi setiap bobot terhadap error dihitung melalui turunan parsial sebagaimana dijelaskan pada *Rumus (2.5)*. Nilai gradien ini menunjukkan arah perubahan bobot yang diperlukan untuk menurunkan error.

#### c. Pembaruan Bobot

Setelah gradien diperoleh, pembaruan bobot dilakukan mengikuti aturan pada *Rumus (2.6)*, yang memanfaatkan learning rate untuk menentukan besar perubahan bobot.

### 8. Ulangi (*Loop*)

Rangkaian *forward propagation* → *hitung error* → *backpropagation* diulang selama sejumlah *epoch* hingga nilai error (MSE) mencapai tingkat yang stabil dan model dianggap konvergen.

### 9. Model Konvergen (Evaluasi Konvergensi)

Selama pengulangan ini, sistem akan terus mengecek apakah model sudah "konvergen". Ini berarti apakah model sudah belajar dengan cukup baik dan kesalahannya sudah sangat kecil atau tidak ada lagi perbaikan yang signifikan.

a. Tidak: Jika model belum konvergen (masih ada kesalahan yang signifikan), proses akan terus berulang (*loop* "Tidak" kembali ke Backpropagation dan Forward Propagation).

b. Ya: Jika model sudah konvergen (sudah pintar dan akurat), maka proses pelatihan akan berhenti.

### 10. Model Siap (*Output*)

Setelah model dianggap konvergen, itu berarti model sudah "Siap" untuk digunakan. Model ini telah dilatih secara efektif untuk memisahkan suara bicara dari kebisingan.

### 11. Selesai (*End*)

Ini adalah akhir dari seluruh proses pelatihan *Supervised Learning*. Model yang telah terlatih kini dapat digunakan untuk mengurangi kebisingan pada rekaman suara baru. Secara keseluruhan, *supervised learning* ini seperti melatih seorang anak dengan memberinya banyak contoh soal dan jawabannya, lalu mengoreksi kesalahannya berulang kali sampai dia benar-benar mengerti bagaimana cara menyelesaikan soal serupa di masa depan.



#### 2.2.6.4. Keunggulan ANN dalam Pengolahan Sinyal

Jaringan Saraf Tiruan (ANN) sangat efektif dalam pemrosesan sinyal berkat beberapa keunggulan utamanya. ANN memiliki kemampuan adaptasi yang hebat [30], memungkinkannya belajar dan beradaptasi dengan berbagai pola kebisingan, termasuk kondisi yang berubah-ubah [35]. Ini memungkinkan ANN melakukan penyesuaian *real-time* untuk mengurangi kebisingan dan mengoptimalkan kualitas suara di lingkungan akustik yang dinamis. Selain itu, ANN mampu memproses data dalam jumlah besar secara paralel [36], sangat meningkatkan kecepatan dan efisiensi pemrosesan sinyal. Struktur paralel ini memungkinkan ANN memberikan respons cepat dan akurat, ideal untuk aplikasi seperti sistem komunikasi suara dan teknologi pengenalan suara. ANN juga memiliki kemampuan generalisasi tinggi, memungkinkannya mendeteksi dan menangani pola kebisingan tak terduga berdasarkan pengalaman sebelumnya, bahkan dalam situasi yang belum pernah ditemui selama pelatihan. Kemampuan ini menjadikan ANN andal dan fleksibel dalam berbagai situasi nyata.

#### 2.2.6.5. Aplikasi ANN dalam Pengurangan Kebisingan

Jaringan Saraf Tiruan (ANN), terutama Convolutional Neural Network (CNN) seperti U-Net yang akan digunakan dalam penelitian ini, telah menunjukkan penerapan yang sangat efektif dalam bidang pengurangan kebisingan (*noise reduction*). Kemampuannya untuk belajar dari pola data yang kompleks, beradaptasi dengan kondisi dinamis, dan melakukan generalisasi menjadikannya alat yang sangat kuat untuk memisahkan sinyal yang diinginkan dari gangguan. Berikut adalah beberapa aplikasi utama ANN dalam pengurangan kebisingan yang relevan dengan pemrosesan sinyal:

##### 1. Pembatalan Kebisingan Aktif (ANC)

ANN digunakan untuk memodelkan dan memprediksi pola kebisingan (mekanis, lalu lintas, latar belakang) dan menghasilkan sinyal "anti-noise" yang berlawanan untuk menekan kebisingan secara efektif [37]. Kemampuan adaptifnya memungkinkan penyesuaian parameter filter *real-time* untuk kinerja optimal dalam kondisi akustik dinamis. Pengenalan dan Peningkatan Kualitas Suara (*Speech Enhancement*).

##### 2. Pengenalan dan Peningkatan Kualitas Suara (*Speech Enhancement*)

ANN berperan penting dalam meningkatkan kualitas suara yang terdistorsi oleh kebisingan latar belakang [38]. Dengan dilatih pada data suara dan kebisingan, ANN memisahkan komponen suara yang relevan dari kebisingan, menghasilkan suara yang lebih jernih dan mudah dikenali. Pemrosesan Sinyal Audio di Perangkat Seluler.

### 3. Pemrosesan Sinyal Audio di Perangkat Seluler

ANN meningkatkan kualitas suara di lingkungan bising pada perangkat seluler dan *headset* dengan mengidentifikasi kebisingan latar belakang dan menyesuaikan pengaturan audio secara adaptif [39].

### 4. Pengurangan Kebisingan dalam Sistem Komunikasi

ANN digunakan dalam sistem komunikasi (suara atau data nirkabel) untuk mengidentifikasi dan mengurangi kebisingan pada sinyal yang diterima, memastikan transmisi informasi lebih bersih [40].

### 5. Pemrosesan Gambar untuk Pengurangan *Noise* (*Image Denoising*)

ANN, khususnya CNN, efektif dalam mendeteksi dan mengurangi *noise* pada gambar digital yang disebabkan oleh gangguan perekaman [41]. Jaringan dilatih untuk memisahkan pola *noise* dari detail visual, menghasilkan gambar yang lebih bersih dan tajam.

Secara keseluruhan, kemampuan adaptif, paralelisme, dan generalisasi ANN menjadikannya alat yang sangat efektif dan serbaguna dalam berbagai aplikasi pengurangan kebisingan, baik pada sinyal audio, gambar, maupun komunikasi, membuka peluang untuk inovasi lebih lanjut.

## 2.2.7. Dari Sinyal Audio ke Rekonstruksi Ucapan Menggunakan U-Net CNN

Pengurangan kebisingan di lingkungan yang kompleks seperti pabrik tekstil menggunakan *deep learning* bukanlah sekadar aplikasi jaringan saraf biasa. Proses ini memerlukan serangkaian transformasi data yang canggih dan arsitektur jaringan yang dirancang khusus untuk tugas separasi sinyal. Alur kerja lengkapnya dapat dipahami melalui empat tahap utama yang saling terkait erat: (1) Representasi Sinyal, (2) Ekstraksi Fitur Kontekstual, (3) Rekonstruksi Sinyal Presisi, dan (4) Generasi Output dan Konversi Kembali ke Audio.

### 2.2.7.1. Transformasi Audio menjadi Citra melalui STFT dan Spektrogram

Secara alami, manusia dapat membedakan berbagai jenis suara dengan mudah. Namun, bagi komputer, sinyal audio direpresentasikan sebagai deretan nilai satu dimensi berupa amplitudo terhadap waktu. Agar model *deep learning* berbasis visi seperti Convolutional Neural Network (CNN) dapat mengolah dan memahami pola-pola pada sinyal suara, diperlukan transformasi dari sinyal 1D menjadi representasi 2D yang lebih informatif, yaitu spektrogram.

#### 1. STFT : Analisis Frekuensi yang Berubah terhadap Waktu

STFT adalah teknik fundamental untuk menganalisis bagaimana konten frekuensi dari sebuah sinyal berubah seiring waktu. Karena sinyal seperti ucapan dan kebisingan bersifat non-stasioner (karakteristiknya berubah), STFT memecahnya menjadi segmen-segmen kecil di mana sinyal dapat diasumsikan stasioner untuk sementara. Prosesnya adalah sebagai berikut:

a. Segmentasi

Sinyal audio  $x[n]$  dibagi menjadi segmen-segmen pendek yang saling tumpang tindih (*overlapping*). Segmentasi ini diperlukan agar analisis frekuensi dapat mengikuti dinamika perubahan sinyal.

b. Windowing

Setiap segmen dikalikan dengan fungsi jendela  $w[n]$  (misalnya jendela Hann). Fungsi jendela digunakan untuk mereduksi amplitudo pada bagian tepi segmen sehingga mengurangi *spectral leakage*, yaitu penyebaran energi frekuensi ke bin tetangga dalam domain frekuensi.

c. Transformasi Fourier

Fast Fourier Transform (FFT) diterapkan pada setiap segmen yang telah diberi jendela untuk memperoleh representasi spektral dari segmen tersebut.

Secara matematis, STFT dari sinyal diskrit  $x[n]$  didefinisikan sebagai:

$$S[m, k] = \sum_{n=0}^{N-1} x[n] \cdot w[n - mH] \cdot e^{-j2\pi kn/N} \quad (2.7)$$

Penjelasan komponen rumus (2.7) adalah sebagai berikut:

- $S[m, k]$  adalah nilai STFT pada frame waktu ke- $m$  dan bin frekuensi ke- $k$ .
- $x[n]$  adalah sinyal input dalam domain waktu.
- $w[n - mH]$  adalah fungsi jendela yang digeser berdasarkan hop length.
- $N$  adalah ukuran FFT.
- $H$  adalah *hop length*, yaitu jumlah pergeseran antara dua jendela berturut-turut.

## 2. Spektrogram: "Citra" dari Suara

Meskipun STFT menghasilkan nilai kompleks, untuk input ke CNN umumnya hanya digunakan magnitudo-nya. Spektrogram magnitudo diperoleh dengan mengambil nilai absolut dari setiap elemen dalam spektrogram kompleks,  $|X(m, \omega)|$ . Untuk meningkatkan kualitas representasi, penelitian ini secara spesifik menggunakan Log-Mel Spektrogram dengan dimensi input  $128 \times N$ .

a. Skala Mel

Sumbu frekuensi pada spektrogram dikonversi ke skala Mel, yaitu skala frekuensi yang lebih mendekati sensitivitas pendengaran manusia. Skala ini memberikan resolusi lebih tinggi pada frekuensi rendah, tempat banyak informasi penting dari sinyal ucapan berada.

b. Skala Logaritmik

Nilai amplitudo dikonversi ke skala logaritmik untuk mengompresi rentang dinamis sinyal suara. Transformasi ini membuat variasi energi yang kecil tetap terlihat, sambil mengurangi dominasi komponen berenergi tinggi.

Hasil akhirnya berupa representasi visual menyerupai citra dua dimensi, di mana pola akustik dapat dianalisis. Struktur harmonik pada sinyal ucapan biasanya

terlihat sebagai garis-garis horizontal, sedangkan kebisingan mesin yang bersifat konstan dapat tampak sebagai pita energi yang berkelanjutan pada spektrogram.

### 3. Fungsi Aktivasi *Sigmoid* (untuk Masking)

Pada tahap akhir pemrosesan, model U-Net menghasilkan nilai untuk setiap elemen pada spektrogram guna menentukan apakah bagian tersebut merupakan komponen sinyal ucapan atau kebisingan. Untuk mengubah keluaran jaringan menjadi nilai yang dapat digunakan sebagai *masking*, fungsi aktivasi Sigmoid digunakan karena mampu memetakan setiap nilai input ke rentang  $[0,1]$ , sehingga dapat ditafsirkan sebagai derajat atau probabilitas.

**Rumusnya:**

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.8)$$

Penjelasan komponen rumus (2.8) adalah sebagai berikut:

#### 1. $x$ : Input ke Neuron Terakhir Ini

Merupakan nilai sebelum aktivasi (*pre-activation*) yang dihasilkan jaringan. Nilai ini dapat positif, negatif, atau nol dan mencerminkan tingkat keyakinan awal model terhadap suatu elemen spektrogram.

#### 2. $e^{-x}$ : Bagian Eksponensial

Komponen ini menentukan bentuk kurva Sigmoid.

- Jika  $x$  bernilai positif besar,  $e^{-x}$  menjadi sangat kecil sehingga keluaran Sigmoid mendekati 1.
- Jika  $x$  bernilai negatif besar,  $e^{-x}$  menjadi sangat besar sehingga keluaran Sigmoid mendekati 0.
- Jika  $x \approx 0$ , maka  $e^{-x} = 1$ , menghasilkan keluaran di sekitar 0,5.

#### 3. Hasil Akhir $f(x)$ : Nilai Masker (antara 0 dan 1)

Nilai Sigmoid  $f(x)$  berada pada rentang  $[0,1]$  dan berfungsi sebagai nilai *masking*:

- Mendekati 1 = bagian spektrogram cenderung dianggap sebagai sinyal ucapan.
- Mendekati 0 = bagian tersebut lebih terkait dengan kebisingan.
- Nilai di tengah ( $\pm 0,5$ ) = menunjukkan ketidakpastian, sehingga bagian tersebut dipertahankan sebagian.

Penggunaan fungsi Sigmoid memungkinkan model menghasilkan masker dengan nilai kontinu, bukan keputusan biner, sehingga proses pemisahan antara komponen ucapan dan kebisingan dapat dilakukan secara lebih halus dan adaptif.

### 2.2.7.2. Ekstraksi Fitur Kontekstual: Jalur Encoder pada U-Net

Setelah audio diubah menjadi spektrogram, arsitektur U-Net—seperti yang dirinci pada Gambar 3.3 dalam penelitian ini—mulai bekerja. Bagian pertama adalah Jalur Encoder, yang bertujuan untuk "memahami" konten atau konteks dari citra spektrogram melalui ekstraksi fitur hierarkis.

a. Operasi Konvolusi dan Aktivasi

Setiap blok encoder menerapkan serangkaian filter 2D (*kernel*) pada peta fitur inputnya. Sebagai contoh, Encoder Block 1 menggunakan lapisan konvolusi dengan 32 filter. Filter-filter ini belajar mengenali pola-pola lokal. Setelah konvolusi, fungsi aktivasi non-linear seperti ReLU (Rectified Linear Unit) diterapkan untuk memungkinkan model mempelajari hubungan yang kompleks.

b. Normalisasi dan Downsampling

Batch Normalization digunakan di setiap blok untuk menstabilkan dan mempercepat proses pelatihan. Setelah itu, MaxPooling mengurangi ukuran peta fitur (misalnya, dari  $128 \times N$  menjadi  $64 \times (N/2)$ ), yang bertujuan mengurangi beban komputasi dan memperluas "bidang reseptif" (*receptive field*). Ini memungkinkan neuron di lapisan berikutnya untuk memahami konteks global dari suara.

c. Representasi Abstrak

Proses ini diulang dengan jumlah filter yang meningkat (32, 64, 128) seiring data bergerak lebih dalam di sepanjang jalur encoder. Di lapisan Bottleneck, yang menggunakan 256 filter, jaringan memiliki pemahaman yang paling terkonsentrasi dan abstrak tentang "apa" yang ada di dalam sinyal, tetapi informasi spasial "di mana" (lokasi waktu dan frekuensi yang tepat) sebagian besar telah hilang.

### 2.2.7.3. Rekonstruksi Sinyal: Jalur Decoder dan Kekuatan Skip Connections

Tujuan utama dari pengurangan kebisingan adalah merekonstruksi sinyal ucapan dengan fidelitas tinggi. Ini memerlukan informasi lokasi yang presisi, yang hilang selama encoding. Di sinilah kejeniusan Jalur Decoder dan Skip Connections pada U-Net bersinar.

1. Operasi Upsampling (*Transposed Convolution*)

Jalur decoder bekerja secara terbalik dari encoder. Ia mengambil peta fitur yang abstrak dan terkompresi, lalu secara bertahap memperbesar ukurannya (*upsampling*) menggunakan *transposed convolutions* untuk merekonstruksi citra spektrogram kembali ke resolusi aslinya.

2. Kekuatan *Skip Connections*

Operasi *upsampling* sendiri akan menghasilkan output yang kabur. *Skip Connections* mengatasi masalah ini dengan menghubungkan peta fitur dari jalur encoder langsung ke lapisan yang sesuai di jalur decoder. Proses ini secara krusial "menyuntikkan" kembali informasi spasial beresolusi tinggi yang mengandung detail lokasi yang presisi. Lapisan konvolusi di decoder kemudian menggunakan informasi gabungan ini (konteks abstrak dari *upsampling* dan detail presisi dari *skip connection*) untuk merekonstruksi sinyal dengan jauh lebih akurat. Mekanisme ini memungkinkan U-Net untuk menjawab pertanyaan "di mana" harus menempatkan fitur "apa" yang telah dipelajarinya.

#### 2.2.7.4. Generasi Output dan Konversi Kembali ke Audio

Lapisan terakhir dari U-Net menghasilkan spektrogram yang telah diproses. Dalam penelitian ini, strategi yang digunakan adalah masking, yang didukung oleh penggunaan fungsi aktivasi Sigmoid pada lapisan output.

##### 1. Masking

Jaringan tidak langsung mengeluarkan spektrogram bersih; ia hanya memproduksi sebuah *masker* berupa matriks yang setiap elemennya berada di antara 0 dan 1. Agar nilai tetap dalam rentang tersebut, lapisan output dipasang fungsi aktivasi sigmoid:  $f(x) = \frac{1}{1+e^{-x}}$ . Masker ini kemudian disapkan ke spektrogram bising dengan cara mengalikan kedua matriks secara elemen-demi-elemen (perkalian Hadamard, dilambangkan seperti perkalian biasa tetapi diberi catatan “elemen-demi-elemen”):

$$\text{Spectrogram}_{\text{clean}} = \text{Spectrogram}_{\text{noisy}} \times M_{\text{predicted}}$$

Dimana “x” bukan perkalian matriks penuh, melainkan setiap elemen setelah dikalikan langsung). Nilai 1 pada masker berarti “pertahankan frekuensi-waktu ini” (dianggap ucapan), nilai 0 berarti “redam sepenuhnya” (dianggap noise). Masker yang dipelajari berbentuk *Ideal Ratio Mask* (IRM). IRM bukan masker biner, melainkan memberikan bobot kontinu 0–1 yang menyatakan rasio energi ucapan terhadap total energi (ucapan + noise) pada setiap titik. Dengan IRM, komponen yang sedikit tercampur noise hanya dilemahkan sedikit, sedangkan komponen yang kuat noise-nya dilemahkan besar, sehingga hasil audio mengalami distorsi lebih rendah dan terdengar lebih natural.

##### 2. Fungsi Loss dan Pelatihan

Model dilatih dengan meminimalkan kesalahan antara output yang dihasilkan (setelah masking) dan target sebenarnya (spektrogram ucapan bersih). Fungsi kesalahan seperti Mean Squared Error (MSE) digunakan untuk mengukur perbedaan ini dan memperbarui bobot jaringan melalui proses *backpropagation*.

##### 3. Konversi Kembali ke Audio

Setelah spektrogram bersih diperoleh, langkah terakhir adalah mengubahnya kembali menjadi sinyal audio 1D yang dapat didengar menggunakan algoritma Inverse Short-Time Fourier Transform (ISTFT). Proses ini juga memerlukan estimasi fasa dari sinyal untuk merekonstruksi gelombang suara secara utuh.

Secara keseluruhan, arsitektur U-Net yang diterapkan pada spektrogram menyediakan kerangka kerja *end-to-end* yang sangat kuat dan sesuai dengan rancangan penelitian, mampu belajar memisahkan pola ucapan yang kompleks dari kebisingan industri yang menantang dengan mempertahankan detail dan kejernihan sinyal yang diinginkan.

### 2.2.8. Keuntungan dan Tantangan

Penerapan ANN untuk pengurangan kebisingan menawarkan serangkaian keuntungan signifikan yang didasari oleh kemampuannya untuk belajar dan beradaptasi. Keunggulan utamanya adalah kemampuan adaptasi terhadap pola data yang kompleks, yang memungkinkan sistem untuk menyesuaikan diri secara *real-time* dengan kondisi lingkungan akustik yang bervariasi. Hal ini didukung oleh kemampuan pemrosesan paralel yang dapat mengolah data dalam jumlah besar secara efisien, sehingga sistem mampu memberikan respons yang cepat dan akurat. Selain itu, ANN memiliki daya generalisasi yang tinggi untuk mengenali pola-pola baru yang tidak terduga berdasarkan pengalaman sebelumnya, yang pada akhirnya meningkatkan kualitas suara dengan memisahkan sinyal relevan dari kebisingan dan membuka peluang untuk solusi inovatif di berbagai bidang seperti manufaktur dan komunikasi.

Meskipun demikian, implementasi ANN juga dihadapkan pada sejumlah tantangan yang perlu diatasi. Tantangan mendasar adalah kebutuhan akan data pelatihan yang besar dan beragam, di mana proses pengumpulan dan pengolahannya bisa menjadi sangat kompleks, terutama di lingkungan industri. Pembangunan model yang efektif juga memerlukan pemahaman mendalam tentang arsitektur jaringan dan parameter pelatihan, yang sering kali menuntut keterampilan khusus serta sumber daya yang signifikan. Selama pelatihan, terdapat pula risiko *overfitting*, di mana model gagal melakukan generalisasi pada data baru di dunia nyata. Dari sisi praktis, tantangan lainnya meliputi biaya implementasi yang dapat menjadi penghalang bagi perusahaan kecil, kesulitan integrasi dengan sistem industri yang sudah ada, serta ketergantungan yang tinggi pada teknologi yang menuntut adanya rencana pemeliharaan dan cadangan yang baik.

Pada akhirnya, meskipun terdapat berbagai tantangan, keuntungan yang ditawarkan oleh ANN dalam pengurangan kebisingan sangatlah signifikan. Dengan kemampuan adaptasi, pemrosesan paralel, dan generalisasi yang baik, ANN dapat memberikan solusi inovatif untuk masalah kebisingan di berbagai sektor, termasuk industri tekstil. Namun, perhatian penuh harus diberikan pada tantangan-tantangan yang ada untuk memastikan implementasi dapat berjalan dengan efektif dan efisien.

### 2.2.9. Metrik Evaluasi Kinerja

Untuk mengukur efektivitas dan performa dari sistem pengurangan kebisingan yang dikembangkan, serangkaian metrik kuantitatif dan kualitatif digunakan. Metrik-metrik ini diadopsi dari dokumentasi teknis proyek untuk memastikan penilaian yang komprehensif terhadap kualitas audio dan karakteristik sistem.

#### 1. SNR Improvement (Peningkatan SNR)

*Signal-to-Noise Ratio* (SNR) adalah metrik fundamental yang mengukur rasio antara daya sinyal yang diinginkan (ucapan bersih) dengan daya sisa kebisingan (*residual noise*) setelah diproses. Peningkatan SNR, yang diukur dalam desibel

(dB), menjadi indikator utama keberhasilan sistem dalam meningkatkan kejelasan audio. Peningkatan sebesar 6-15 dB menandakan reduksi noise yang sangat efektif.

Rumus: SNR dihitung menggunakan rumus berikut:

$$SNR = 10 \log_{10} \left( \frac{\sum_{n=0}^{N-1} s(n)^2}{\sum_{n=0}^{N-1} e(n)^2} \right) \quad (2.9)$$

Di mana:

- $s(n)$  adalah sampel dari sinyal ucapan bersih (target).
- $e(n)$  adalah sampel dari sinyal error atau sisa kebisingan (hasil proses - sinyal bersih).
- $N$  adalah jumlah total sampel.

Peningkatan *SNR* (*SNR Improvement*) dihitung sebagai selisih antara SNR sinyal hasil pemrosesan dan SNR sinyal asli sebelum proses reduksi kebisingan. Peningkatan antara 6–15 dB umumnya menunjukkan reduksi kebisingan yang efektif.

## 2. *Spectral Preservation* (Pemeliharaan Spektral)

Metrik ini bertujuan untuk mengukur seberapa baik sistem mempertahankan konten frekuensi dari sinyal ucapan asli setelah proses *denoising*. Target pemeliharaan di atas 95% menunjukkan bahwa sistem mampu menghilangkan noise tanpa merusak atau menghilangkan detail penting dari suara asli, sehingga menjaga kealamian dan karakter vokal.

Rumus untuk Pemeliharaan spektral dapat dikuantifikasi menggunakan *Spectral Preservation Index* (SPI) yang secara konseptual membandingkan magnitudo spektrogram bersih dengan hasil proses:

$$SPI = 1 = \left( \frac{\sum_{k,m} |S_{clean}(k,m) - S_{proc}(k,m)|}{\sum_{k,m} |S_{clean}(k,m)|} \right) \times 100\% \quad (2.10)$$

Di mana:

- $S_{clean}(k,m)$  adalah magnitudo pada bin frekuensi  $k$  dan frame waktu  $m$  dari spektrogram ucapan bersih.
- $S_{proc}(k,m)$  adalah magnitudo pada bin frekuensi  $k$  dan frame waktu  $m$  dari spektrogram hasil proses.

Nilai SPI di atas 95% mengindikasikan bahwa sistem berhasil menjaga karakteristik spektral sinyal ucapan sambil tetap menekan kebisingan.

## 3. *Artifact Minimization* (Minimisasi Artefak)

Metrik ini bersifat kualitatif untuk mengevaluasi munculnya distorsi suara yang tidak diinginkan sebagai efek samping dari pemrosesan, seperti *musical noise* (artefak nada pendek acak) atau efek "robotik". Sistem yang baik harus mampu



meminimalkan artefak ini, yang dalam proyek ini dicapai melalui teknik penghalusan masker (*mask smoothing*) untuk memastikan transisi yang mulus.

#### 4. Processing Speed (Kecepatan Pemrosesan)

*Processing Speed* atau Kecepatan Pemrosesan adalah metrik yang mengukur seberapa cepat dan efisien program Anda dalam menyelesaikan tugas reduksi noise pada sebuah file audio. Metrik ini dinyatakan sebagai rasio perbandingan, yang menunjukkan berapa kali lebih cepat program Anda dibandingkan durasi asli audio tersebut (*real-time*).

Rumus untuk menghitung kecepatan pemrosesan sangat sederhana:

$$\text{Kecepatan Pemrosesan} = \frac{\text{Waktu Pemrosesan (detik)}}{\text{Durasi Audio Asli (detik)}} \quad (2.11)$$

a. Durasi Audio Asli

Panjang total file audio yang Anda proses, dalam satuan detik.

b. Waktu Pemrosesan

Waktu yang dibutuhkan oleh program Anda untuk menjalankan seluruh algoritma dari awal hingga akhir, dalam satuan detik.

Nilai rasio yang lebih besar menunjukkan kinerja pemrosesan yang lebih cepat.

Sebagai contoh apabila memiliki sebuah file audio dengan detail berikut:

- Durasi Audio Asli: 60 detik (1 menit)
- Program membutuhkan waktu 12 detik untuk membersihkan noise dari file tersebut.

Maka, perhitungannya adalah:  $\text{Kecepatan Pemrosesan} = \frac{60 \text{ (detik)}}{12 \text{ (detik)}} = 5$

Hasil 5 ini berarti kecepatan pemrosesan program adalah 5x lebih cepat dari *real-time*. Dengan kata lain, untuk setiap satu detik audio, program hanya butuh 0.2 detik untuk memprosesnya.

Nilai ini sangat penting untuk menunjukkan bahwa aplikasi tidak hanya efektif dalam mengurangi noise, tetapi juga praktis dan tidak membuat pengguna menunggu terlalu lama.

#### 2.2.10. Skenario Pengujian

Untuk mengukur performa sistem secara kuantitatif, serangkaian pengujian dilakukan dengan menjalankan skrip *evaluate\_model.py* pada dataset uji. Dataset ini terdiri dari pasangan file audio: versi asli yang bising (*noisy*) dan versi bersih sebagai *ground truth*. Skrip akan memproses file bising dan membandingkan hasilnya dengan file bersih untuk menghitung metrik performa.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Desain Penelitian**

Penelitian ini merupakan penelitian terapan dengan pendekatan simulasi dan rekayasa model. Tujuannya adalah membangun sistem pengurangan kebisingan berbasis deep learning menggunakan arsitektur U-Net Convolutional Neural Network (CNN). Sistem akan dilatih untuk membedakan sinyal bicara dari kebisingan pabrik tekstil.

Desain penelitian mencakup:

- 1) Pengumpulan data audio (clean speech dan noisy speech).
- 2) Praproses data menjadi bentuk spectrogram.
- 3) Pelatihan model menggunakan dataset tersebut.
- 4) Evaluasi hasil menggunakan metrik kinerja seperti SNR Improvement, Spectral Preservation, dan Artifact Minimization.

#### **3.2. Lokasi dan Waktu Penelitian**

Penelitian dilaksanakan di PT. Embee Plumbon Tekstile, Unit 2, yang memiliki tingkat kebisingan tinggi di area produksi. Lokasi ini dipilih karena menyediakan data nyata untuk melatih dan menguji model ANN. Waktu penelitian direncanakan berlangsung dari Januari hingga Juli 2025, mencakup tahap pengumpulan data, pelatihan model, dan evaluasi hasil.

#### **3.3. Populasi dan Sampel**

Populasi dalam penelitian ini adalah seluruh area produksi di PT. Embee Plumbon Tekstile yang menghasilkan kebisingan akibat operasional mesin. Dari populasi tersebut, sampel dipilih menggunakan metode *purposive sampling*, yang mencakup lima area produksi utama dengan tingkat kebisingan tertinggi. Di setiap area sampel, dilakukan perekaman suara selama 1-20 menit untuk mengumpulkan data kebisingan dan komunikasi verbal secara representatif.

Dari lima area produksi yang dijadikan lokasi penelitian, dilakukan proses perekaman audio untuk mengumpulkan data sampel. Secara total, berhasil dikumpulkan 9 sampel rekaman suara bersih (*clean speech*) dan 9 sampel rekaman kebisingan pabrik murni (*factory noise*). Sampel-sampel inilah yang menjadi fondasi untuk proses augmentasi data, di mana keduanya digabungkan secara sintesis dalam berbagai rasio untuk menciptakan dataset pelatihan yang bervariasi dan robust.

#### **3.4. Variabel Penelitian**

Untuk memastikan objektivitas dan kejelasan, penelitian ini mendefinisikan variabel secara spesifik untuk menganalisis hubungan sebab-akibat. Variabel bebas (*independent variable*) dalam penelitian ini adalah Model *Artificial Neural Network* (ANN) berbasis U-Net CNN yang dikembangkan, yang bertindak sebagai

intervensi untuk mengurangi kebisingan. Efek dari intervensi ini diukur melalui variabel terikat (*dependent variables*), yaitu tingkat kebisingan yang berhasil direduksi, yang dianalisis secara kuantitatif dalam desibel (dB). Guna memastikan bahwa perubahan yang diamati disebabkan oleh variabel bebas, beberapa variabel kendali (*control variables*) turut diperhatikan, seperti parameter mesin produksi dan kondisi lingkungan pabrik saat pengambilan data.

### 3.5. Instrumen Penelitian

Penelitian ini menggunakan beberapa instrumen utama untuk mendukung proses pengumpulan dan analisis data. Instrumen tersebut meliputi mikrofon kondensor untuk merekam data suara dari area produksi, perangkat lunak pengolahan suara berbasis Python untuk melakukan *preprocessing* data, serta *platform* TensorFlow dan Python untuk pengembangan dan pelatihan model ANN. Selain itu, sebuah aplikasi *decibel meter* yang diukur melalui *handphone* digunakan untuk mengukur tingkat kebisingan sebelum dan sesudah penerapan sistem ANN.

### 3.6. Teknik Pengumpulan Data

Teknik pengumpulan data diawali dengan perekaman suara langsung di lokasi produksi menggunakan *handphone* dengan metode perekaman bergerak untuk menangkap variasi suara dari berbagai sudut, yang mencakup suara mesin dan komunikasi verbal pekerja. Data mentah ini kemudian diperbanyak melalui proses augmentasi data, di mana variasi *noise* ditambahkan untuk melatih ANN agar lebih adaptif terhadap berbagai kondisi kebisingan. Tahap terakhir adalah *preprocessing*, di mana data suara diolah melalui normalisasi, penghapusan *noise* yang tidak relevan, dan ekstraksi fitur-fitur penting seperti frekuensi dan amplitudo.

### 3.7. Teknik Analisis Data

Pengumpulan data dilakukan dengan merekam suara langsung di lima area produksi PT. Embee Plumbon Tekstile Unit 2 menggunakan mikrofon kondensor via *handphone*, mencakup suara mesin dan komunikasi verbal. Data kemudian diaugmentasi, yaitu diperbanyak dengan menambahkan variasi kebisingan ke rekaman suara bicara yang bersih agar model ANN lebih adaptif. Selanjutnya, data dipraproses melalui normalisasi, pemisahan sinyal, dan ekstraksi fitur spektral (misalnya Log-Mel Spectrogram) agar siap menjadi input model.

### 3.8. Prosedur Penelitian

Prosedur penelitian ini dilaksanakan melalui tiga tahap utama: persiapan, pelaksanaan, dan evaluasi. Tahap persiapan meliputi identifikasi lokasi dan alat, pengumpulan data suara, serta penyusunan data pelatihan untuk ANN. Selanjutnya, pada tahap pelaksanaan, model ANN dilatih dengan data yang telah di-*preprocess* dan diimplementasikan di lokasi uji coba. Proses evaluasi kemudian dilakukan dengan mengukur tingkat kebisingan menggunakan *decibel meter*, mengumpulkan tanggapan pekerja mengenai kejelasan komunikasi, dan membandingkan hasil dengan kondisi awal.

Seluruh penelitian ini dijalankan dengan mematuhi pedoman etika, yang mencakup perolehan persetujuan tertulis dari manajemen pabrik, menjaga anonimitas data suara, dan membatasi penggunaan data hanya untuk keperluan akademis. Aspek validitas penelitian dipastikan melalui pengukuran kebisingan berulang, sementara reliabilitas diuji dengan mengevaluasi model pada dataset yang berbeda untuk memastikan konsistensi dan kemampuan generalisasi. Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, antara lain ketergantungan pada kualitas rekaman suara, kompleksitas desain ANN yang memerlukan waktu pelatihan lama, dan tidak mencakup analisis dampak jangka panjang terhadap produktivitas kerja.

### **3.9. Etika Penelitian**

Penelitian ini dilaksanakan dengan mengikuti pedoman etika penelitian yang ketat. Prinsip utama yang dipegang adalah mendapatkan persetujuan tertulis dari pihak manajemen pabrik sebelum proses pengumpulan data dimulai. Selain itu, anonimitas data dijaga dengan memastikan bahwa data suara yang direkam tidak mengidentifikasi individu tertentu. Data yang terkumpul juga hanya digunakan untuk keperluan akademis dan tidak akan dibagikan kepada pihak ketiga tanpa izin resmi.

### **3.10. Validitas dan Reliabilitas**

Untuk memastikan kualitas hasil penelitian, aspek validitas dan reliabilitas menjadi perhatian utama. Validitas data diuji untuk memastikan representasi yang akurat dari kondisi lapangan, yang dilakukan melalui pengukuran kebisingan secara berulang. Sementara itu, reliabilitas model ANN diuji dengan mengaplikasikannya pada dataset yang berbeda untuk memastikan konsistensi hasil serta kemampuan generalisasi model terhadap data baru.

### **3.11. Keterbatasan Penelitian**

Dalam pelaksanaannya, penelitian ini menghadapi beberapa keterbatasan yang perlu diakui. Keterbatasan tersebut antara lain adalah ketergantungan pada kualitas data rekaman suara, di mana suara yang tidak jelas atau terdistorsi berpotensi memengaruhi efektivitas pelatihan model. Selain itu, terdapat tantangan dalam kompleksitas desain ANN yang membutuhkan waktu pelatihan yang relatif lama. Keterbatasan lainnya adalah cakupan penelitian yang tidak mencakup analisis jangka panjang mengenai dampak penerapan sistem ANN terhadap produktivitas kerja.

### 3.12. Diagram Alir Perancangan

Dalam pengembangan sistem pemrosesan suara yang efektif, diperlukan sebuah alur kerja yang sistematis dan terstruktur. Untuk memberikan gambaran yang jelas mengenai tahapan yang dilakukan, keseluruhan proses pengolahan data suara mulai dari augmentasi hingga evaluasi model dirangkum dalam diagram alir pada Gambar 3.1. Setiap kotak dan panah dalam diagram ini merepresentasikan langkah atau keputusan spesifik yang dirancang untuk memastikan kualitas hasil yang optimal.

Untuk memperoleh pemahaman yang lebih jelas mengenai tahapan-tahapan proses pengembangan model dalam proyek ini, silakan merujuk pada Gambar 3.1. Diagram alir alur kerja pengolahan data suara yang akan dijelaskan secara terperinci sebagai berikut:

#### 1. Persiapan Dataset (*Data Acquisition & Augmentation*)

Gambar 3.1 mengilustrasikan alur kerja komprehensif pengolahan data suara yang dimulai dari tahap Persiapan Dataset. Pada tahap ini, data sumber diperoleh melalui perekaman langsung di lingkungan PT. Embee Plumbon Tekstile Unit 2. Sesuai dengan metodologi yang dijelaskan dalam Sub-bab 3.3 (Populasi dan Sampel), data dikumpulkan secara terpisah dalam dua kategori utama:

- a. Sinyal Ucapan Bersih (Clean Speech)  
Direkam dalam kondisi akustik terkendali, berisi percakapan instruksional yang akan menjadi target output sistem.
- b. Sinyal Kebisingan Pabrik (Factory Noise)
- c. Direkam dari operasional mesin tekstil (TFO, Carding, dll.) tanpa kontaminasi sinyal ucapan.

Pendekatan rekaman terpisah ini dipilih untuk memberikan fleksibilitas dalam sintesis data dan kontrol yang ketat terhadap variabel tingkat kebisingan selama proses augmentasi.

#### 2. Pemisahan Data (*Data Separation for Supervised Learning*)

Tahap Augmentasi Data merupakan proses kritis dalam penelitian ini. Mengingat keterbatasan memperoleh data percakapan dalam kondisi bersih dan bising secara simultan di lingkungan industri, diterapkan teknik pencampuran sintetis (*synthetic mixing*) untuk menghasilkan dataset pelatihan yang komprehensif.

Secara teknis, proses augmentasi dilakukan dengan:

- a. Mengambil sampel sinyal ucapan bersih  $s(t)$ .
- b. Mencampurkannya dengan sampel kebisingan pabrik  $n(t)$ .
- c. Menggunakan variasi *Signal-to-Noise Ratio* (SNR) yang berbeda
- d. Menghasilkan sinyal input:  $x(t) = s(t) + n(t)$

Hasil proses ini menghasilkan pasangan data pelatihan yang terdiri dari:

- a. Input: Sinyal "Suara Bising + Bicara"
- b. Target: Sinyal "Suara Tanpa Bising" asli

Proses ini diulang secara sistematis dengan berbagai kombinasi segmen suara dan kebisingan untuk menciptakan dataset yang robust dan beragam, yang essential untuk melatih model U-Net agar mampu beradaptasi dengan berbagai kondisi akustik.

### 3. Pra-pemrosesan (*Preprocessing*)

Secara teknis, tahap prapemrosesan ini diawali dengan *Resampling* seluruh file audio ke frekuensi sampel standar 16.000 Hz untuk memastikan konsistensi. Selanjutnya, dilakukan Normalisasi amplitudo untuk mencegah variasi volume yang ekstrem. Normalisasi merupakan tahap krusial dalam prapemrosesan data audio sebelum dimasukkan ke dalam model jaringan saraf. Tujuan utama dari normalisasi adalah untuk menyeragamkan skala nilai amplitudo dari seluruh sinyal audio ke rentang yang konsisten, biasanya antara -1 dan 1. Proses ini, yang dalam penelitian ini dilakukan menggunakan fungsi *librosa.util.normalize*, sangat penting karena beberapa alasan.

Pertama, ia mencegah sampel audio dengan volume yang sangat tinggi mendominasi proses pembelajaran secara tidak proporsional. Kedua, normalisasi memastikan bahwa model menerima input dengan distribusi yang lebih stabil, yang dapat mempercepat proses konvergensi selama pelatihan dan meningkatkan kinerja generalisasi model secara keseluruhan. Tanpa normalisasi, model mungkin akan kesulitan untuk belajar secara efektif karena harus beradaptasi dengan rentang data input yang sangat bervariasi.

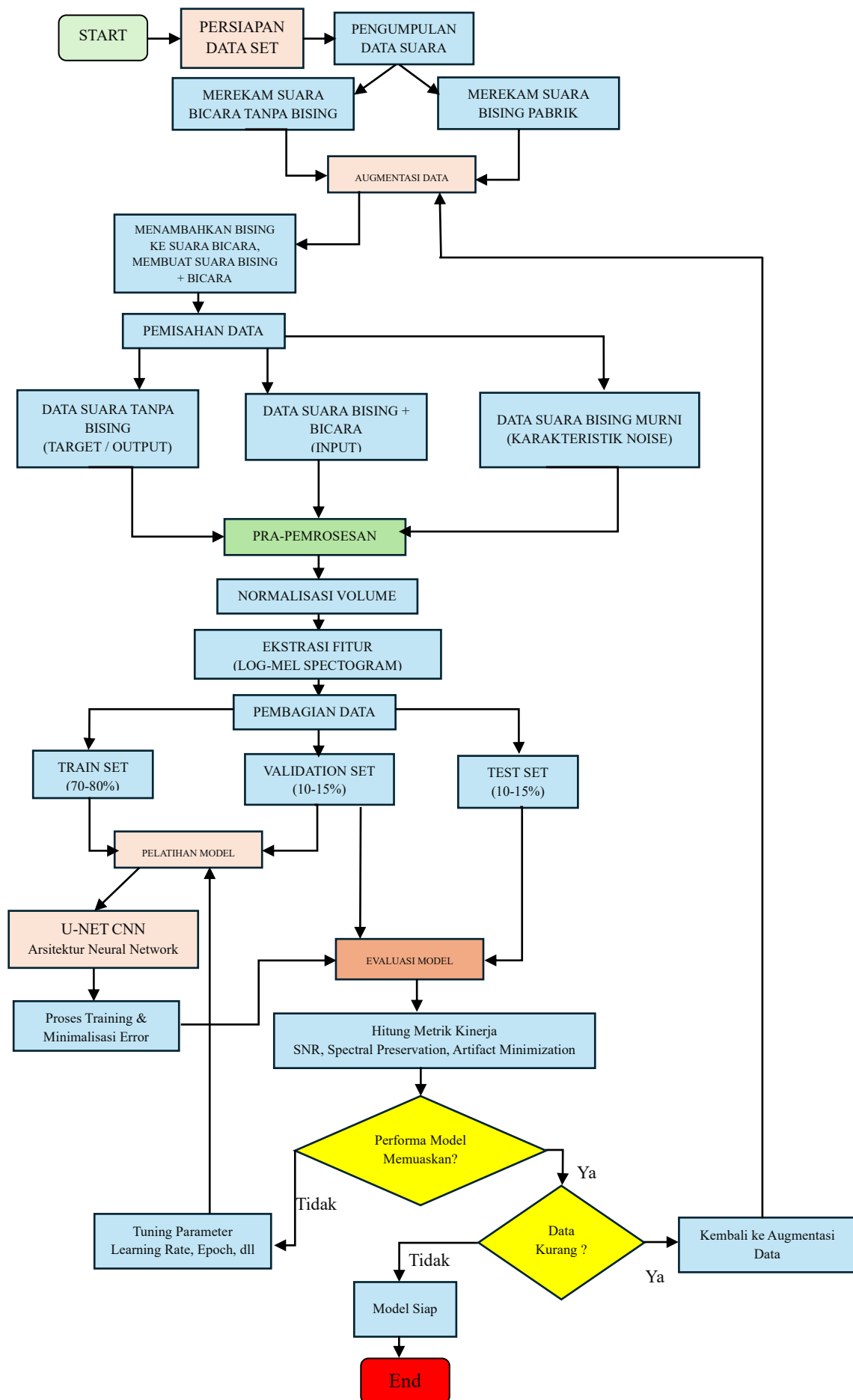
Proses augmentasi data (*Noise Addition*) kemudian dilakukan untuk menciptakan dataset pelatihan yang robust. Untuk ekstraksi fitur, setelah melalui proses STFT, representasi akhir yang menjadi input bagi model adalah *Log-Mel Spectrogram* yang dikonversi menggunakan 128 filter Mel (*n\_mels*) untuk meniru persepsi pendengaran manusia.

### 4. Pembagian Data untuk Pelatihan (*Dataset Splitting*)

Setelah melalui proses augmentasi, dataset yang terbentuk kemudian dibagi secara proporsional menjadi tiga subset yang tidak saling tumpang-tindih (*non-overlapping datasets*):

- a. Training Set (70-80%)
- b. Validation Set (10-15%), Digunakan untuk memantau proses pelatihan dan mencegah overfitting
- c. Test Set (10-15%), Digunakan untuk evaluasi final kinerja model

Pembagian ini memastikan bahwa model dapat dievaluasi secara objektif pada data yang belum pernah dilihat selama proses pelatihan, sebagaimana tercermin dalam hasil kinerja yang dilaporkan pada Tabel 4.2 di Bab IV.



*Gambar 3.1. Diagram alir alur kerja pengolahan data suara*

### 5. Pelatihan Model (*Model Training*)

Ini adalah inti dari pengembangan proyek, di mana model ANN benar-benar "belajar". Model dengan arsitektur U-Net CNN dilatih menggunakan data dari *train set*. Selama pelatihan, model akan memprediksi suara bersih dari *input* bising dan terus-menerus menyesuaikan pengaturan internalnya (bobot) untuk meminimalkan kesalahan prediksi, sebuah proses yang berulang hingga model mencapai tingkat keahlian yang diinginkan.

### 6. Evaluasi Model (*Model Evaluation*)

Proses evaluasi kinerja model dilaksanakan secara kuantitatif dan sistematis untuk mengukur efektivitas sistem dari berbagai aspek. Mengacu pada prosedur yang diilustrasikan pada Lampiran 2, pengujian ini dijalankan menggunakan skrip *evaluate\_model.py* pada *dataset* uji yang telah dipisahkan secara khusus .

*Dataset* ini diorganisir ke dalam struktur direktori yang terdiri dari folder *clean* untuk data referensi (*ground truth*) dan *noisy* untuk data masukan . Skrip evaluasi secara otomatis memproses setiap file audio dari folder *noisy*, menyimpan hasilnya ke folder *processed\_output*, dan kemudian membandingkan hasil tersebut dengan data referensi yang bersesuaian untuk menghitung serangkaian metrik kinerja secara objektif, yang meliputi Peningkatan SNR, Pemeliharaan Spektral, dan Kecepatan Proses.

### 7. Siklus Iteratif & Pengambilan Keputusan

Tahap ini menunjukkan sifat iteratif dari pengembangan model melalui sebuah siklus pengambilan keputusan. Setelah evaluasi, ditentukan apakah performa model telah mencapai kinerja yang diinginkan. Jika ya, model dianggap siap dan proses berlanjut ke tahap akhir. Namun, jika performa belum memuaskan, ada dua jalur perbaikan yang dapat ditempuh: melakukan *tuning parameter* (seperti *learning rate* atau *epoch*) dan melatih kembali model, atau jika data diduga kurang, maka alur akan kembali ke tahap augmentasi untuk menambah variasi data sebelum seluruh proses diulang. Setelah semua siklus perbaikan selesai dan performa model dianggap memuaskan serta konvergen, model dinyatakan siap.

### 8. Akhir (*End*)

Tahap ini adalah titik akhir dari seluruh proses pengembangan model. Model yang telah terlatih kini dapat digunakan untuk aplikasi sebenarnya dalam mengurangi kebisingan, yang sekaligus menandai selesainya proyek.

## 3.13. Digitalisasi Suara (*Analog ke Digital*)

Proses awal yang fundamental dan esensial dalam setiap sistem pengolahan suara digital adalah digitalisasi suara, yaitu konversi sinyal audio analog menjadi representasi digital. Sinyal audio analog, seperti gelombang suara kontinu yang ditangkap oleh Mikrofon Kondensor (yang berfungsi sebagai sensor input dalam penelitian ini ), secara alami bervariasi secara terus-menerus dalam amplitudo dan waktu. Agar sinyal ini dapat diproses, dianalisis, dan dimanipulasi oleh komputer



serta jaringan saraf tiruan (seperti CNN), ia harus diubah menjadi serangkaian nilai numerik diskrit.

Konversi ini secara fisik dilakukan oleh perangkat keras yang disebut *Analog-to-Digital Converter (ADC)*. Proses digitalisasi ini melibatkan dua parameter kunci yang sangat memengaruhi kualitas dan akurasi representasi digital:

#### 1. Frekuensi Sampling (*Sampling Rate*)

Ini menentukan berapa kali per detik sinyal analog diukur atau "disampel" untuk diubah menjadi sampel digital. Menurut teorema *Nyquist-Shannon*, frekuensi sampling harus setidaknya dua kali lipat dari frekuensi tertinggi yang ingin direproduksi. Semakin tinggi frekuensi sampling, semakin akurat representasi komponen frekuensi tinggi dari sinyal analog asli, yang krusial untuk menjaga fidelitas audio.

#### 2. Kedalaman Bit (*Bit Depth/Quantization*)

Parameter ini mengacu pada jumlah bit yang digunakan untuk merepresentasikan amplitudo setiap sampel digital. Kedalaman bit menentukan jumlah kemungkinan nilai diskrit yang dapat digunakan untuk menyimpan setiap sampel. Semakin tinggi kedalaman bit (misalnya, 16-bit, 24-bit), semakin besar rentang dinamis (perbedaan antara suara paling keras dan paling lembut) dan resolusi amplitudo sinyal yang dapat direproduksi, sehingga mengurangi *quantization noise* dan meningkatkan fidelitas suara secara keseluruhan.

Sinyal digital yang dihasilkan dari proses ini, yang terdiri dari serangkaian sampel diskrit dengan nilai amplitudo tertentu, inilah yang kemudian menjadi input yang valid dan dapat diolah lebih lanjut oleh tahapan berikutnya dalam alur pemrosesan, khususnya transformasi ke spektrogram.

### 3.14. Blok Diagram Sistem

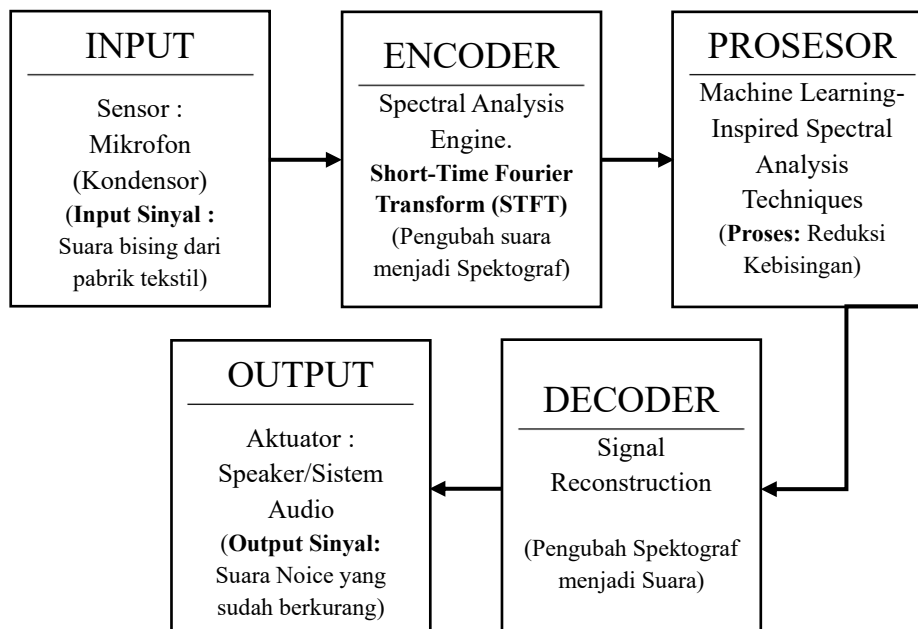
Penelitian ini mengadopsi pendekatan terapan dengan fokus pada simulasi dan rekayasa model untuk mengatasi tantangan kebisingan di pabrik tekstil. Inti dari sistem yang dikembangkan adalah arsitektur U-Net Convolutional Neural Network (CNN), sebuah model *deep learning* yang dirancang khusus untuk membedakan sinyal bicara dari kebisingan latar belakang yang kompleks. Desain penelitian secara holistik mencakup pengumpulan data audio, pra-pemrosesan data menjadi representasi spektral, pelatihan model CNN dengan dataset yang disiapkan, dan evaluasi hasil menggunakan metrik objektif SNR. Untuk pemahaman alur kerja yang lebih detail, mari kita tinjau diagram alir perancangan dan blok diagram sistem yang menjadi fondasi implementasi ini.

Proses pengurangan kebisingan dirancang untuk mengubah sinyal audio mentah yang terkontaminasi kebisingan menjadi output suara yang lebih jernih. Alur ini mencakup serangkaian tahapan yang saling terkait, dimulai dari akuisisi suara hingga rekonstruksi akhir, dengan fokus pada pemrosesan cerdas di domain frekuensi untuk isolasi dan eliminasi kebisingan, seperti yang diilustrasikan pada Gambar 3.2 di bawah ini.

Untuk memperoleh pemahaman yang lebih mendalam mengenai arsitektur dan alur kerja Audio Noise Reduction Tool secara visual, kita dapat merujuk pada Gambar 3.2. Blok Diagram Sistem. Diagram ini secara jelas mengilustrasikan setiap tahapan kunci dan interaksi antar komponen, yang akan kita bahas lebih lanjut sebagai berikut:

### 1. Akuisisi Suara Mentah (Microphone - Suara Raw)

Sebagai titik awal dalam rantai pemrosesan, tahap Microphone (Suara Raw) bertanggung jawab atas akuisisi sinyal audio awal dari lingkungan sekitar. Pada fase ini, suara asli, yang merupakan campuran dari sinyal yang diinginkan (misalnya, ucapan manusia) dan berbagai jenis kebisingan latar belakang (seperti suara kipas, lalu lintas, atau desisan elektronik), ditangkap oleh mikrofon. Hasilnya adalah data audio mentah yang belum diproses, yang kemudian akan menjadi input bagi algoritma pengurangan kebisingan selanjutnya. Kualitas input pada tahap ini sangat memengaruhi kinerja keseluruhan sistem.



Gambar 3.2. Blok Diagram Sistem

### 2. Transformasi ke Spektrogram (Encoder)

Setelah sinyal audio berhasil didigitalisasi, tahap selanjutnya yang krusial adalah Transformasi Suara Digital ke Representasi Citra (Encoder). Dalam konteks arsitektur internal alat ini, fungsi ini secara spesifik diimplementasikan oleh *Spectral Analysis Engine* yang memanfaatkan teknik *Short-Time Fourier Transform (STFT)*. STFT adalah metode fundamental dalam pemrosesan sinyal yang memecah sinyal audio digital yang bervariasi seiring waktu menjadi segmen-segmen pendek yang tumpang tindih. Pada setiap segmen tersebut, dilakukan Transformasi Fourier untuk menganalisis konten frekuensinya, menghasilkan spektrum frekuensi lokal. Hasil dari STFT adalah spektrogram, sebuah representasi visual dua dimensi yang kaya informasi. Pada spektrogram, sumbu horizontal secara

tipikal merepresentasikan waktu, sumbu vertikal merepresentasikan frekuensi, dan intensitas warna (atau kecerahan piksel) pada setiap titik menunjukkan amplitudo atau energi suara pada frekuensi spesifik di titik waktu yang bersangkutan. Representasi visual ini secara efektif mengonversi sinyal audio yang kompleks dan berbasis waktu menjadi format "citra" yang terstruktur dan berbasis frekuensi.

Lebih lanjut, dalam banyak aplikasi *deep learning* untuk audio, termasuk penelitian ini, spektrogram ini sering kali dikonversi menjadi Log-Mel Spectrogram. Konversi ke skala Mel melibatkan pemetaan frekuensi linier ke skala non-linier yang lebih mirip dengan cara telinga manusia memproses frekuensi, sehingga menekankan area frekuensi yang paling relevan untuk persepsi suara manusia. Sementara itu, penerapan skala logaritmik membantu menekan rentang dinamis data, menjadikannya lebih stabil dan cocok sebagai input untuk jaringan saraf. Representasi

Log-Mel Spectrogram inilah yang secara efektif bertindak sebagai "citra" 2D dari sinyal audio. Ia berfungsi sebagai "encoder" karena secara efektif mengonversi representasi audio yang kompleks menjadi format visual yang dapat diolah langsung oleh jaringan saraf konvolusional (CNN) untuk mengekstraksi fitur spasial dan temporal layaknya pada sebuah citra, sehingga memungkinkan analisis dan manipulasi kebisingan yang lebih presisi.

### 3. Pemrosesan Pengurangan Kebisingan (Proses CNN - Noise Reduction)

Inti dari kapabilitas pengurangan kebisingan pada aplikasi ini terletak pada implementasi model *deep learning* dengan arsitektur Convolutional Neural Network (CNN) U-Net. Model ini telah dilatih secara khusus pada dataset pasangan audio bising dan bersih untuk mempelajari cara memisahkan suara manusia dari kebisingan latar secara cerdas.

Prosesnya memperlakukan spektrogram audio sebagai sebuah "citra". Model U-Net menganalisis citra spektrogram yang bising ini dan belajar untuk merekonstruksi versi bersihnya melalui beberapa tahapan inti:

#### A. *Encoder Path* (Ekstraksi Fitur)

Tahap pertama adalah menganalisis spektrogram bising. Menggunakan serangkaian lapisan konvolusional (*convolutional layers*) dan *pooling*, *Encoder Path* secara bertahap mengurangi dimensi "citra" spektrogram sambil mengekstrak fitur-fitur abstrak yang esensial. Proses ini memungkinkan model untuk memahami konteks dan membedakan antara karakteristik pola suara manusia dengan pola kebisingan pabrik yang kompleks.

#### B. *Decoder Path* (Rekonstruksi Spektrogram)

Setelah fitur diekstraksi, *Decoder Path* bertugas untuk membangun kembali spektrogram menjadi versi yang bersih. Ini dilakukan melalui lapisan *upsampling* atau *transposed convolution*. Keunggulan utama arsitektur U-Net terletak pada penggunaan *skip connections*, yang menghubungkan langsung lapisan dari *Encoder*

ke *Decoder*. Fitur ini memungkinkan detail-detail penting dari sinyal asli yang mungkin hilang selama kompresi untuk dipulihkan, sehingga hasil audio menjadi jernih tanpa terdengar terdistorsi atau "mendem".

#### C. Prediksi Masker Spektral dan Aplikasi

Pada lapisan akhirnya, U-Net tidak melakukan *gating* sederhana, melainkan memprediksi sebuah masker spektral (*spectral mask*) yang canggih. Masker ini adalah sebuah matriks bernilai antara 0 dan 1 yang kemudian dikalikan dengan spektrogram bising asli. Nilai yang mendekati 1 akan mempertahankan komponen sinyal (suara), sementara nilai yang mendekati 0 akan menekan komponen sinyal (noise). Kemampuan untuk menghasilkan masker yang bernilai gradien ini (bukan hanya 0 atau 1) memungkinkan proses penyaringan yang jauh lebih halus dan alami, serta secara efektif meminimalkan artefak seperti *musical noise*.

#### D. Rekonstruksi Sinyal Audio (ISTFT)

Tahap terakhir adalah mengambil magnitudo spektrogram yang telah dibersihkan (setelah aplikasi masker) dan menggabungkannya kembali dengan informasi fase dari sinyal asli. Hasilnya kemudian diubah kembali dari domain frekuensi ke domain waktu menggunakan *Inverse Short-Time Fourier Transform* (ISTFT) untuk menghasilkan file audio akhir yang jernih.

#### 4. Rekonstruksi Sinyal Audio (*Decoder*)

Setelah spektrogram berhasil "dibersihkan" dari kebisingan melalui proses *filtering* oleh CNN, tahap selanjutnya yang krusial adalah Pengubah Spektrogram Menjadi Suara (*Decoder*), atau yang lebih dikenal sebagai rekonstruksi sinyal audio. Fungsi vital ini diemban oleh komponen Signal Reconstruction dalam inti algoritma. Pada dasarnya, tahap ini melakukan operasi invers dari *Short-Time Fourier Transform* (STFT) yaitu, ia mengubah representasi data dari domain frekuensi (spektrogram yang telah dimodifikasi dan difilter) kembali ke domain waktu.

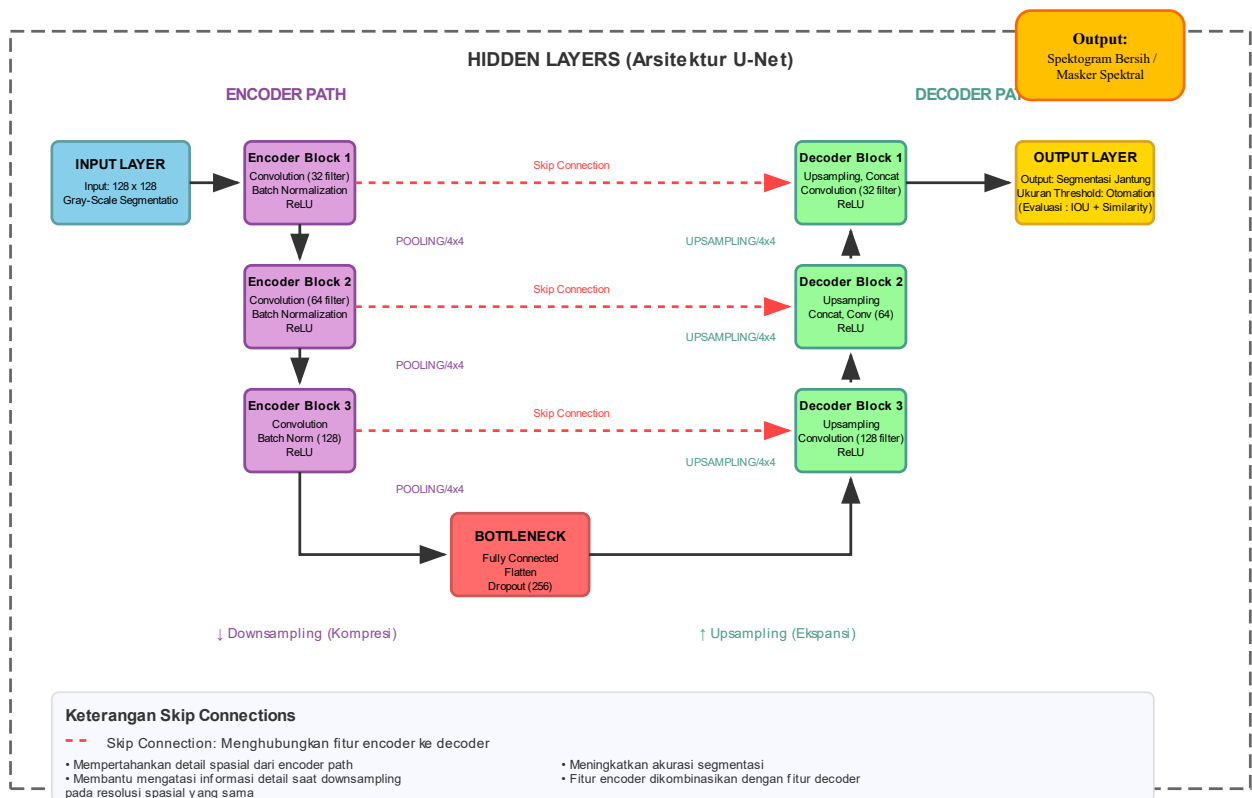
Penting untuk dicatat bahwa informasi fase asli dari sinyal yang diperoleh dari STFT awal tetap dipertahankan dan digabungkan kembali selama proses rekonstruksi ini untuk memastikan akurasi dan koherensi sinyal audio yang dihasilkan. Tanpa informasi fase yang tepat, suara yang direkonstruksi akan terdengar tidak alami atau terdistorsi, meskipun konten frekuensinya sudah benar. Hasil dari tahap ini adalah file audio yang telah dikurangi kebisingannya secara signifikan dan siap untuk didengar. Proses ini secara efektif bertindak sebagai "decoder", menerjemahkan kembali "citra" visual dan termanipulasi (spektrogram bersih) menjadi bentuk audio yang dapat dipersepsi oleh telinga manusia. Ini adalah langkah terakhir dalam mengembalikan data dari representasi visual yang diolah oleh CNN ke bentuk suara yang dapat digunakan.

#### 5. Output Audio Akhir (*Speaker*)

Sebagai tahap final dalam pipeline pemrosesan, *Speaker (Output)* merepresentasikan titik di mana sinyal audio yang telah dimurnikan disalurkan ke pengguna. Pada fase ini, suara yang telah melewati seluruh proses pengurangan kebisingan dan rekonstruksi dapat diputar melalui perangkat *speaker* untuk didengarkan secara langsung, atau disimpan sebagai file audio digital yang telah diproses dan bersih. Ini menandai penyelesaian siklus penuh pengurangan kebisingan, menyajikan hasil akhir berupa audio yang ditingkatkan dengan kejernihan yang jauh lebih baik dan pengurangan kebisingan yang signifikan dibandingkan dengan input mentahnya.

### 3.15. Rancangan Jaringan Saraf Tiruan (ANN)

Model *Artificial Neural Network (ANN)* yang dirancang untuk penelitian ini adalah *U-Net Convolutional Neural Network (CNN)*. Arsitektur ini dipilih karena performanya yang sangat baik dalam tugas pemrosesan audio seperti pengurangan kebisingan, di mana ia mampu mempertahankan detail penting dari sinyal asli (suara verbal) sambil secara efektif menghilangkan komponen kebisingan yang kompleks. Arsitektur dari model U-Net ini, yang terdiri dari *Input Layer*, *Hidden Layers* (mencakup *Encoder Path*, *Bottleneck*, dan *Decoder Path* dengan *Skip Connections*), serta *Output Layer*, diilustrasikan secara visual pada Gambar 3.3. memproses data input berupa *Log-Mel Spectrogram* untuk menghasilkan output berupa spektrogram yang telah dikurangi kebisingannya atau segmentasi fitur tertentu. Berikut adalah penjelasan untuk setiap komponen utama dalam arsitektur tersebut:



Gambar 3.3. Blok Diagram Jaringan Saraf Tiruan (ANN) Model U-Net

## 1. INPUT LAYER

Lapisan ini menerima input berupa matriks Log-Mel Spectrogram dengan dimensi  $128 \times N$  (di mana  $N$  adalah jumlah frame waktu) dan dalam format *Gray Scale Segmentatio*. Data ini kemudian diteruskan ke Encoder Path.

## 2. HIDDEN LAYERS (Arsitektur U-Net)

Bagian ini merupakan inti dari arsitektur model yang digunakan, terdiri dari serangkaian lapisan tersembunyi yang membentuk struktur U-Net. Komponen-komponen ini bekerja secara sinergis untuk memproses dan memfilter spektrogram, yang dapat dibagi menjadi tiga bagian fungsional utama: Jalur Pengekstraksi Fitur (*Encoder Path*), Leher Botol (*Bottleneck*), dan Jalur Rekonstruksi (*Decoder Path*).

### A. Jalur Pengekstraksi Fitur (*Encoder Path*)

Jalur *encoder* berfungsi sebagai ekstraktor fitur (*feature extractor*) dalam arsitektur CNN. Tujuannya adalah untuk mengekstraksi representasi fitur yang semakin kaya dan abstrak dari spektrogram input. Proses ini dicapai melalui serangkaian blok konvolusi yang secara progresif mengurangi resolusi spasial dari peta fitur melalui operasi *downsampling* (Pooling). Setiap blok dalam jalur ini mulai dari *Encoder Block 1* hingga 3 menerapkan operasi konvolusi untuk mendeteksi pola-pola lokal, diikuti oleh *Batch Normalization* untuk menstabilkan pelatihan, dan fungsi aktivasi *ReLU* untuk memperkenalkan non-linearitas.

Proses ekstraksi fitur hierarkis ini dimulai pada *Encoder Block 1*, yang menerima input langsung dari *Input Layer* dan menerapkan 32 filter konvolusi. Outputnya kemudian diteruskan ke *Encoder Block 2* yang menggunakan 64 filter, dan selanjutnya ke *Encoder Block 3* dengan 128 filter. Peningkatan jumlah filter pada setiap blok memungkinkan model untuk mempelajari fitur yang lebih kompleks pada setiap level abstraksi. Di akhir setiap blok, operasi *MaxPooling* mengurangi dimensi spasial peta fitur, yang berfungsi untuk meningkatkan bidang reseptif (*receptive field*). Hal ini memungkinkan neuron pada lapisan yang lebih dalam untuk "melihat" konteks yang lebih luas dari spektrogram, yang krusial untuk membedakan antara pola kebisingan yang tersebar dan pola ucapan yang terstruktur.

### B. Leher Botol (*Bottleneck*)

Lapisan *bottleneck* merupakan titik transisi krusial dalam arsitektur U-Net, berfungsi sebagai jembatan yang menghubungkan akhir dari jalur *encoder* dengan awal dari jalur *decoder*. Pada level ini, peta fitur telah mencapai dimensi spasial terkecil namun dengan kedalaman fitur (jumlah *channel*) terbesar, yaitu 256 filter. Lapisan ini mengkonsolidasikan informasi kontekstual paling abstrak yang telah diekstraksi dari keseluruhan spektrogram.

Dalam model yang diusulkan, sebuah pendekatan unik diterapkan dengan mengintegrasikan lapisan *Fully Connected*. Setelah peta fitur 2D dikonversi menjadi vektor 1D melalui lapisan *Flatten*, ia diproses oleh lapisan *Fully*

*Connected*. Desain ini memungkinkan model untuk menangkap dependensi global antara semua fitur yang telah diekstraksi, melampaui hubungan lokal yang ditangkap oleh lapisan konvolusional. Lapisan ini bertindak sebagai konsolidator informasi tingkat tinggi sebelum sinyal direkonstruksi oleh *decoder*. Untuk mencegah *overfitting* pada representasi fitur yang sangat terkonsentrasi ini, mekanisme regularisasi *Dropout* juga diterapkan.

### C. Jalur Rekonstruksi (*Decoder Path*)

Jalur *decoder* bertugas sebagai bagian generatif dari arsitektur, yang bertujuan untuk merekonstruksi spektrogram bersih secara bertahap kembali ke resolusi spasial aslinya. Proses ini dilakukan melalui operasi *upsampling*, yang dalam implementasi ini menggunakan lapisan *Conv2DTranspose*. Lapisan ini memperbesar dimensi peta fitur sambil mempelajari bobot konvolusional untuk mengisi detail.

Fitur kunci dari arsitektur U-Net terletak pada penggunaan *skip connections*. Pada setiap blok *decoder*, peta fitur yang telah di-*upsample* digabungkan (melalui operasi *Concatenate*) dengan peta fitur beresolusi tinggi yang bersesuaian dari jalur *encoder*. Mekanisme ini secara fundamental memungkinkan model untuk mengkombinasikan informasi semantik abstrak dari *bottleneck* ("apa" yang ada di dalam sinyal) dengan informasi spasial presisi dari *encoder* ("di mana" lokasi fitur tersebut). Penggabungan ini secara signifikan meningkatkan akurasi rekonstruksi, mencegah hasil yang kabur, dan memastikan detail penting dari sinyal ucapan tetap terjaga. Proses rekonstruksi ini berlanjut secara simetris, di mana *Decoder Block 3* menggunakan 128 filter, *Decoder Block 2* menggunakan 64 filter, dan *Decoder Block 1* menggunakan 32 filter.

### D. Mekanisme *Skip Connection*

*Skip connection* merupakan inovasi arsitektural yang menjadi kunci utama dari keunggulan U-Net, yang secara visual direpresentasikan oleh panah penghubung antara jalur *encoder* dan *decoder* pada Gambar 3.3. Mekanisme ini secara fundamental mengatasi tantangan hilangnya informasi spasial yang umum terjadi pada arsitektur *encoder-decoder* standar. Fungsinya adalah untuk mengambil peta fitur dari setiap level di jalur *encoder* dan menggabungkannya secara langsung (melalui operasi *Concatenate*) dengan peta fitur yang relevan di jalur *decoder*.

Koneksi langsung ini memberikan tiga manfaat krusial:

#### 1. Preservasi Detail Spasial

Informasi spasial beresolusi tinggi, seperti detail tepi dan tekstur halus pada spektrogram yang ditangkap oleh lapisan awal *encoder*, akan hilang selama proses *downsampling*. *Skip connection* memastikan detail-detail penting ini tidak hilang dan "disuntikkan" kembali selama proses rekonstruksi, sehingga output yang dihasilkan lebih tajam dan akurat.

## 2. Mitigasi Masalah Gradien

Pada jaringan yang sangat dalam, gradien bisa menjadi sangat kecil saat melalui proses *backpropagation* dari akhir ke awal jaringan (*vanishing gradient problem*). *Skip connection* menyediakan jalur pintas yang memungkinkan gradien mengalir lebih mudah ke lapisan-lapisan awal, sehingga proses pelatihan menjadi lebih stabil dan efektif.

## 3. Peningkatan Akurasi Rekonstruksi

Dengan mengkombinasikan informasi semantik yang abstrak dari *decoder* (konteks "apa" yang ada di dalam sinyal) dengan informasi lokasional yang presisi dari *encoder* (konteks "di mana" posisi fitur tersebut), *skip connection* secara signifikan meningkatkan akurasi dan fidelitas sinyal yang direkonstruksi.

### E. Lapisan Output (*Output Layer*)

Lapisan terakhir dari jaringan ini menerima output dari *Decoder Block 1* dan bertanggung jawab untuk menghasilkan luaran akhir model. Lapisan ini terdiri dari satu lapisan konvolusi (*Conv2D*) dengan satu filter dan fungsi aktivasi Sigmoid. Penggunaan aktivasi Sigmoid sangat krusial karena ia memetakan semua nilai output ke rentang antara 0 dan 1. Hasilnya bukanlah spektrogram bersih secara langsung, melainkan sebuah masker spektral (*spectral mask*) yang canggih. Masker ini nantinya akan dikalikan dengan spektrogram bising asli untuk menekan komponen derau dan mempertahankan komponen ucapan, sehingga menghasilkan spektrogram bersih sebagai output akhir dari keseluruhan sistem.

Model *Artificial Neural Network (ANN)* yang dirancang untuk penelitian ini adalah *U-Net Convolutional Neural Network (CNN)*. Arsitektur ini dipilih karena performanya yang sangat baik dalam tugas pemrosesan audio seperti pengurangan kebisingan, di mana ia mampu mempertahankan detail penting dari sinyal asli (suara verbal) sambil secara efektif menghilangkan komponen kebisingan yang kompleks. Struktur fundamental dari model U-Net ini, yang mencakup Input Layer, Hidden Layers dengan jalur encoder-decoder dan Skip Connections, serta Output Layer, diilustrasikan secara visual pada Gambar 3.3.

Setelah memahami gambaran umum arsitektur dan komponen-komponen utama U-Net melalui Gambar 3.3, pemahaman yang lebih mendalam mengenai mekanisme kerja internalnya menjadi krusial.

Untuk itu, Gambar 3.4 menyajikan sebuah diagram alir yang komprehensif. Diagram ini dirancang untuk memfasilitasi pemahaman yang lebih detail, tidak hanya dengan merinci setiap langkah sekuensial dalam pemrosesan data, tetapi juga dengan memvisualisasikan bagaimana fitur-fitur diekstraksi, ditransformasi, dan direkonstruksi, serta bagaimana berbagai titik keputusan logis—seperti evaluasi kualitas, signifikansi fitur, dan ambang batas performa—secara aktif mengatur dan mengoptimalkan aliran informasi melalui beragam tahapan yang kompleks di dalam keseluruhan jaringan U-Net.



Gambar 3.4. Diagram Alir Jaringan Syaraf Tiruan (ANN) Model U-Net CNN

Arsitektur Jaringan Saraf Tiruan (ANN) yang dirancang, seperti yang diilustrasikan pada Gambar 3.4, adalah sebuah model U-Net Convolutional Neural Network (CNN). Model ini secara sistematis memproses data audio, yang telah dikonversi menjadi spektrogram, untuk secara cerdas memisahkan sinyal suara verbal dari kebisingan latar belakang yang kompleks. Diagram alir ini merinci setiap tahapan fungsional, menunjukkan bagaimana fitur-fitur diekstraksi melalui konvolusi, ditransformasi, direkonstruksi, serta bagaimana berbagai titik keputusan logis dan mekanisme umpan balik mengatur aliran informasi untuk mengoptimalkan kinerja keseluruhan jaringan CNN.

#### 1. Tahap Inisiasi dan Pemrosesan Input Awal

Alur operasional model dimulai pada titik *"Start"*, menandakan kesiapan sistem untuk menerima dan memproses data input. Data yang menjadi masukan utama adalah *"Input Layer: 128 x N log-mel Spectrogram"*, sebuah representasi visual dua dimensi dari sinyal audio yang akan diolah lebih lanjut. Input mentah (*"Raw input"*) ini kemudian segera diteruskan ke *Encoder Block 1*.

Di dalam blok ini, dilakukan transformasi awal melalui penerapan dua lapisan konvolusi 2D dengan masing-masing 32 filter (*"Apply 2x Conv2D 32 filters"*), yang bertujuan untuk mengekstraksi fitur-fitur dasar dan pola-pola level rendah dari spektrogram input. Setelah proses konvolusi, hasil peta fitur dievaluasi melalui sebuah titik keputusan *"Activation Threshold?"*.

Jika kualitas peta fitur dianggap lemah (*"Feature Map Weak"*), sistem akan mengaktifkan mekanisme penyempurnaan fitur potensial (*"Potential feature Refinement"*) untuk meningkatkan representasi fitur. Sebaliknya, jika peta fitur dinilai kuat (*"Feature Map Strong"*), alur proses dapat melanjutkan ke tahap berikutnya tanpa intervensi tambahan. Output dari keseluruhan tahap *Encoder Block 1* ini kemudian mengalami operasi *MaxPooling*, sebuah proses downsampling yang penting untuk mereduksi dimensi spasial dari peta fitur sambil berupaya mempertahankan informasi yang paling signifikan atau dominan.

#### 2. Proses Encoding Lanjutan, Seleksi Fitur, dan Pembentukan Representasi Terkonsentrasi

Setelah operasi *MaxPooling* pertama, model melakukan evaluasi terhadap signifikansi fitur-fitur yang telah diekstraksi (*"Feature Signification"*). Titik keputusan ini mengarahkan alur data berdasarkan tingkat kepentingan fitur: fitur dengan signifikansi tinggi (*"High Significance"*) akan diarahkan ke tahap pemrosesan yang lebih lanjut dan mendalam, yaitu *"Advanced Encoding"* yang menuju ke *Encoder Block 3*.

Sementara itu, fitur dengan signifikansi rendah (*"Low Significance"*) dapat mengambil jalur alternatif (*"Alternative Path"*) menuju *Encoder Block 2*, yang merupakan tahap encoding berikutnya namun mungkin dengan perlakuan yang berbeda atau untuk fitur yang kurang dominan.

*Encoder Block 3* dirancang untuk menangani fitur-fitur yang lebih signifikan, melakukan pemrosesan dengan strategi retensi fitur parsial (*"Partial Feature Retention"*) untuk menjaga informasi penting, dan supresi fitur (*"Feature Suppression"*) untuk menghilangkan kontribusi dari fitur yang dianggap tidak relevan atau noise. Selanjutnya, diterapkan dua lapisan konvolusi 2D dengan 128 filter (*"Apply 2x Conv2D, 128 Filters"*) guna menghasilkan fitur-fitur yang lebih kompleks dan abstrak (*"Complex Features"*), yang memiliki daya representasi lebih tinggi. Output dari *Encoder Block 3* ini kemudian juga diakhiri dengan operasi *MaxPooling* untuk reduksi dimensi lebih lanjut.

Secara paralel, *Encoder Block 2* memproses fitur dari jalur alternatif dan juga diakhiri dengan operasi *MaxPooling*. Sebelum masuk ke lapisan *Bottleneck*, dilakukan evaluasi kompleksitas fitur (*"Feature Complexity"*) terhadap output dari *Encoder Block 2*. Fitur yang telah disederhanakan (*"Simplified Features"*) dari tahap ini kemudian digabungkan atau diintegrasikan dengan fitur kompleks yang berasal dari *Encoder Block 3* (setelah *MaxPooling*) untuk membentuk sebuah representasi fitur yang terkonsentrasi dan kaya informasi (*"Concentrated Representation"*).

### 3. Lapisan Bottleneck dan Inisiasi Jalur Ekspansi (*Decoding*)

Representasi fitur yang terkonsentrasi dari jalur encoding tersebut kemudian menjadi input bagi lapisan *Bottleneck*. Pada lapisan ini, diterapkan satu lapisan konvolusi 2D dengan 256 filter (*"1x Conv2D 256 Filters"*). Tujuan utama *Bottleneck* adalah untuk mengurangi kompleksitas lebih lanjut (*"Reduce Complexity"*) dan mengkonsolidasikan fitur-fitur (*"Feature Consolidation"*) menjadi sebuah vektor fitur yang paling esensial, padat informasi, dan memiliki dimensi yang lebih rendah dibandingkan inputnya.

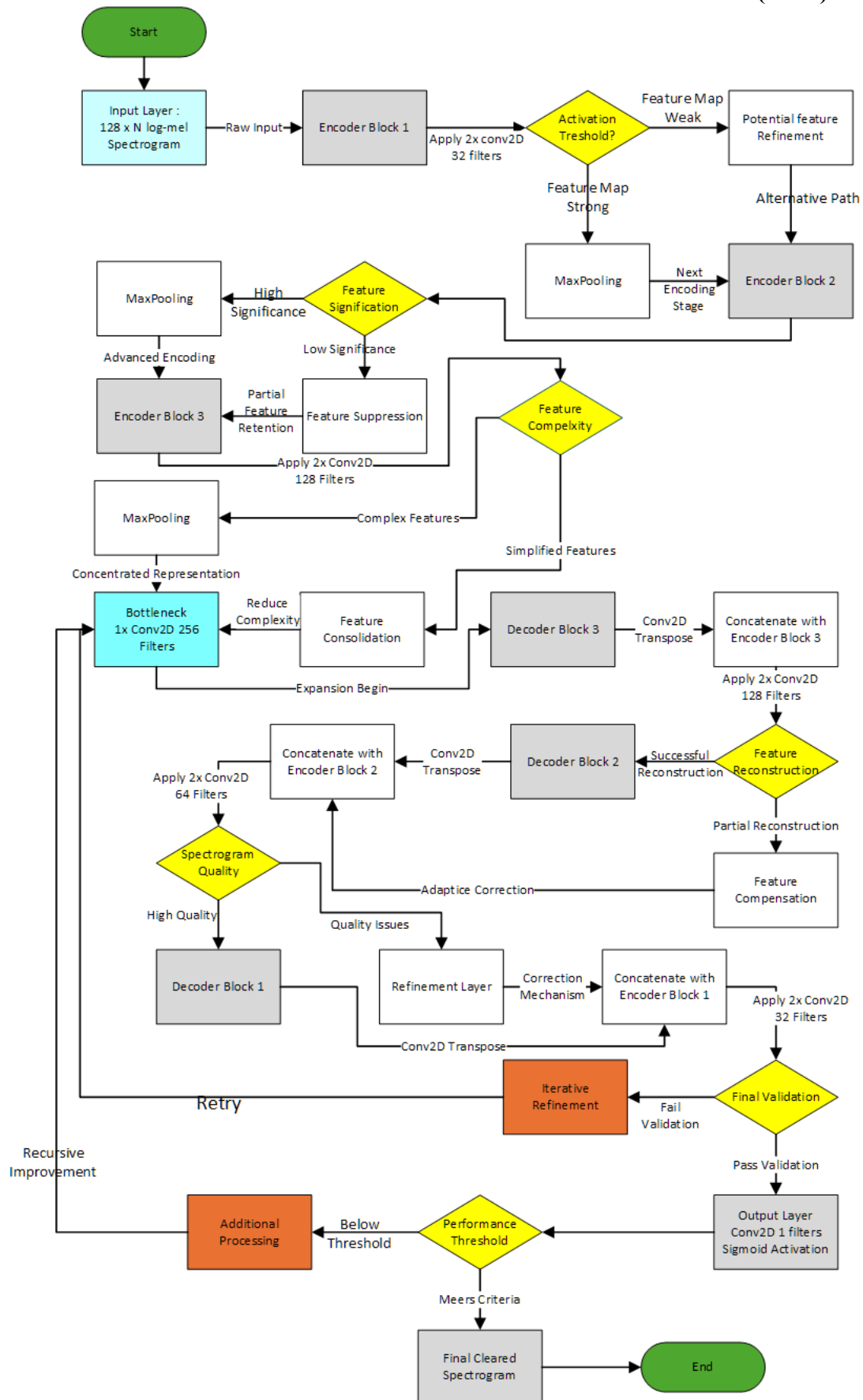
Lapisan ini berfungsi sebagai jembatan kritis antara jalur kompresi (*encoder*) dan jalur ekspansi (*decoder*). Setelah pemrosesan di *Bottleneck*, proses berbalik arah menuju ekspansi atau rekonstruksi sinyal (*"Expansion Begin"*), yang dimulai dengan mengalirkan output *Bottleneck* ke *Decoder Block 3*.

### 4. Jalur Dekoding (*Decoder Path*) dengan Pemanfaatan *Skip Connections*

Jalur dekoding bertugas untuk merekonstruksi sinyal atau peta fitur secara bertahap ke resolusi spasial aslinya, sambil memanfaatkan informasi detail yang ditangkap oleh jalur encoding melalui mekanisme *skip connections*. Proses dimulai pada *Decoder Block 3*, yang menerima fitur dari *Bottleneck* (setelah *Feature Consolidation*) dan fitur yang telah disederhanakan dari tahap encoding sebelumnya.

Di sini, operasi *Conv2D Transpose (upsampling)* dilakukan untuk memperbesar dimensi fitur. Langkah krusial berikutnya adalah konkatenasi (penggabungan) dengan peta fitur yang relevan dari *Encoder Block 3* melalui *skip connection* (*"Concatenate with Encoder Block 3"*). Penggabungan ini memungkinkan informasi spasial detail dari tahap encoding untuk diintegrasikan kembali.

## DIAGRAM ALIR RANCANGAN JARINGAN SARAF TIRUAN (ANN)



Gambar 3.4. Diagram Alir Jaringan Syaraf Tiruan (ANN) Model U-Net CNN

Setelah penggabungan, diterapkan dua lapisan konvolusi 2D dengan 128 filter ("*Apply 2x Conv2D 128 Filters*") untuk memproses fitur gabungan dan memulai rekonstruksi pada resolusi yang lebih tinggi. Alur serupa berlanjut ke *Decoder Block 2*, yang menerima output dari *Decoder Block 3*. Dilakukan operasi *Conv2D Transpose*, kemudian konkatenasi dengan fitur dari *Encoder Block 2* (melalui *skip connection*), diikuti oleh dua lapisan konvolusi 2D dengan 64 filter ("*Apply 2x Conv2D 64 Filters*").

Hasil dari *Decoder Block 2* ini kemudian dievaluasi kualitas rekonstruksinya melalui titik keputusan "*Successful Reconstruction?*". Jika rekonstruksi dianggap belum optimal atau hanya parsial ("*Partial Reconstruction*"), sistem akan mengaktifkan mekanisme kompensasi fitur ("*Feature Compensation*") untuk memperbaiki kekurangan tersebut. Tahap terakhir pada jalur dekoding dilakukan oleh *Decoder Block 1*.

Blok ini menerima output dari *Decoder Block 2*, dengan kemungkinan adanya koreksi adaptif ("*Adaptive Correction*") jika pada evaluasi sebelumnya terdeteksi masalah kualitas ("*Quality Issues*") pada spektrogram. Operasi yang dilakukan meliputi *Conv2D Transpose*, kemudian konkatenasi dengan fitur dari *Encoder Block 1* (melalui *skip connection*), dan diakhiri dengan dua lapisan konvolusi 2D dengan 32 filter ("*Apply 2x Conv2D 32 Filters*").

#### 5. Penyempurnaan Output, Validasi, dan Keputusan Akhir

Output yang dihasilkan dari *Decoder Block 1* selanjutnya dievaluasi kualitas spektrogramnya secara keseluruhan pada titik keputusan "*Spectrogram Quality*". Jika kualitasnya dinilai tinggi ("*High Quality*"), proses dapat melanjutkan ke tahap akhir. Namun, apabila teridentifikasi adanya masalah kualitas ("*Quality issues*"), data akan diarahkan ke sebuah lapisan penyempurnaan ("*Refinement Layer*") yang melibatkan mekanisme koreksi ("*Correction Mechanism*") dan kemungkinan operasi *Conv2D Transpose* tambahan untuk perbaikan.

Model ini juga menyertakan mekanisme untuk penanganan kegagalan validasi. Jika hasil validasi pada tahap tertentu gagal ("*Fail Validation*"), sistem dapat dirancang untuk melakukan penyempurnaan secara iteratif ("*Iterative Refinement*") atau bahkan mengulang sebagian proses ("*Retry*") melalui jalur "Recursive Improvement" guna mengoptimalkan hasil. Setelah melalui berbagai tahap penyempurnaan potensial, dilakukan validasi akhir ("*Final Validation*") terhadap output yang dihasilkan. Jika hasil validasi ini lolos ("*Pass Validation*"), data dianggap siap dan diteruskan ke *Output Layer*.

Langkah terakhir sebelum finalisasi adalah evaluasi terhadap ambang batas performa yang telah ditetapkan ("*Performance Threshold*"). Jika performa output masih berada di bawah ambang batas ("*Below Threshold*"), akan dipertimbangkan untuk dilakukan pemrosesan tambahan ("*Additional Processing*"). Jika semua kriteria performa telah terpenuhi ("*Meets Criteria*"), maka spektrogram jernih final

("Final Cleared Spectrogram") dianggap telah berhasil dihasilkan dengan kualitas yang diinginkan.

#### 6. Generasi Output Akhir oleh Output Layer dan Terminasi Proses

Lapisan *Output Layer* bertugas untuk menghasilkan *output* final dari model. Lapisan ini menerapkan satu lapisan konvolusi 2D dengan 1 filter dan fungsi aktivasi Sigmoid ("*Output Layer Conv2D 1 filters Sigmoid Activation*").

Fungsi aktivasi Sigmoid memastikan bahwa output berada dalam rentang tertentu (biasanya antara 0 dan 1), yang sangat sesuai untuk tugas-tugas seperti pembuatan masker biner dalam segmentasi atau estimasi probabilitas, sehingga menghasilkan "*Final Cleared Spectrogram*". Setelah output akhir ini berhasil dihasilkan, alur operasional model kemudian berakhir pada titik "*End*".

### 3.16. Proses dan Konfigurasi Pelatihan Model

Proses pelatihan model dilakukan dengan konfigurasi spesifik untuk mencapai konvergensi yang optimal dan mencegah *overfitting*. Model dikompilasi menggunakan *optimizer* Adam dengan *learning rate* awal sebesar 0.001. Fungsi *loss* yang dipilih untuk mengukur kesalahan antara prediksi model dan target sebenarnya adalah 'mean\_squared\_error' (MSE), yang efektif untuk tugas regresi seperti merekonstruksi spektogram. Untuk mengelola proses pelatihan secara otomatis dan efisien, serangkaian *callbacks* dari TensorFlow diimplementasikan:

#### 1. EarlyStopping

Memantau *val\_loss* (kesalahan pada data validasi) dan akan menghentikan pelatihan jika tidak ada perbaikan setelah 5 *epoch* (*patience*=5), serta mengembalikan bobot model terbaik.

#### 2. ModelCheckpoint

Menyimpan bobot model (*best\_model.h5*) hanya pada saat *val\_loss* mencapai nilai terendah yang baru.

#### 3. ReduceLROnPlateau

Mengurangi *learning rate* sebesar 50% (*factor*=0.5) jika *val\_loss* tidak menunjukkan perbaikan selama 2 *epoch* (*patience*=2), untuk membantu model keluar dari *local minima*.

Model dilatih dengan ukuran *batch* (*batch\_size*) sebanyak 32 sampel per iterasi.

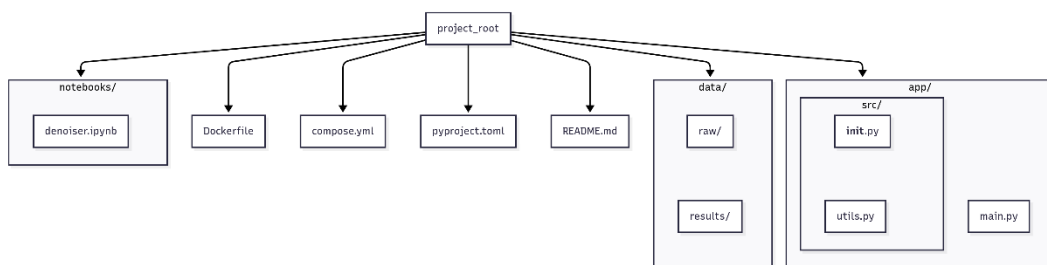
## BAB IV

### HASIL DAN PEMBAHASAN

Sistem yang dikembangkan merupakan sebuah aplikasi *web* berbasis Streamlit yang dirancang untuk menghilangkan *noise* (kebisingan) dari file audio. Aplikasi ini memungkinkan pengguna mengunggah file audio, mengatur kekuatan reduksi *noise*, serta melihat dan mengunduh hasil pembersihan suara dengan tampilan visualisasi yang interaktif berupa *waveform* dan *spectrogram*. Proses utama melibatkan konversi format audio, analisis spektral, serta penerapan model *deep learning* berbasis arsitektur U-Net CNN.

#### 4.1. Arsitektur Proyek dan Fondasi Lingkungan Penelitian yang Reproducible

Dalam penelitian yang mengandalkan metode komputasi, penyajian hasil tidak dapat dipisahkan dari arsitektur perangkat lunak dan lingkungan di mana hasil tersebut diperoleh. Oleh karena itu, sub-bab ini secara rinci menguraikan arsitektur direktori proyek dan justifikasi teknologi yang digunakan, yang berfungsi sebagai fondasi untuk memastikan validitas, modularitas, dan terutama, reproduisibilitas dari seluruh temuan penelitian. Struktur yang diimplementasikan, sebagaimana diilustrasikan pada Gambar 4.1, merupakan adopsi dari praktik terbaik dalam rekayasa perangkat lunak dan ilmu data.



Gambar 4. 1. Struktur Direktori Project

Gambar 4.1 menyajikan cetak biru dari keseluruhan ekosistem penelitian. Untuk mengapresiasi bagaimana setiap komponen saling berinteraksi untuk membentuk sebuah alur kerja yang koheren dan dapat diverifikasi secara ilmiah, analisis terperinci dari setiap direktori dan file konfigurasi akan dielaborasi di bawah ini.

##### 4.1.1. Elaborasi Struktur Direktori

###### 1. Direktori Aplikasi (*app/*):

Direktori ini menampung artefak final dari penelitian, yaitu aplikasi fungsional yang mampu melakukan reduksi kebisingan. Pemisahan internalnya menjadi *main.py* dan direktori *src/* mengimplementasikan prinsip rekayasa perangkat lunak pemisahan kepentingan (*separation of concerns*). *main.py* berfungsi sebagai lapisan presentasi (*presentation layer*) yang mengelola antarmuka pengguna (UI) dan interaksi dengan pengguna. Sementara itu, *src/* berisi logika inti (*core logic*)

dari penelitian, termasuk fungsi-fungsi untuk prapemrosesan sinyal, inferensi model, dan pascapemrosesan. Arsitektur ini memastikan bahwa kode ilmiah dapat dikembangkan dan diuji secara independen dari antarmukanya.

## 2. Direktori Data (*data/*):

Direktori ini berfungsi sebagai pusat manajemen data yang menjamin ketertelusuran data (*data provenance*). Pemisahan antara sub-direktori *raw/* dan *results/* adalah fundamental. *raw/* berisi dataset sumber yang bersifat immutable atau tidak boleh diubah, seperti rekaman audio asli dari suara bersih dan kebisingan pabrik. Hal ini memastikan bahwa sumber data primer selalu terjaga integritasnya. Sebaliknya, *results/* adalah repositori untuk semua artefak yang dihasilkan secara komputasi, termasuk model yang telah dilatih, data yang telah diproses, dan visualisasi seperti spektogram. Struktur ini memungkinkan setiap hasil dapat ditelusuri kembali ke data mentah dan proses yang menghasilkannya.

## 3. Direktori Eksplorasi (*notebooks/*):

Direktori ini diibaratkan sebagai "laboratorium" atau "buku catatan digital" peneliti. Di sinilah Jupyter Notebooks (*denoiser.ipynb*) digunakan untuk melakukan analisis data eksploratif, pengujian hipotesis awal, visualisasi data interaktif, dan iterasi cepat dalam pengembangan model. Kode dalam direktori ini bersifat eksperimental dan berfungsi sebagai dasar sebelum logika yang telah terbukti solid difaktorkan ulang dan diformalkan ke dalam modul di direktori *app/*.

### 4.1.2. Justifikasi Lingkungan Penelitian yang Terkontainerisasi

Salah satu tantangan terbesar dalam penelitian komputasi adalah krisis reproduktibilitas, di mana hasil penelitian sulit atau tidak mungkin direplikasi oleh pihak lain. Untuk mengatasi hal ini, proyek ini mengadopsi teknologi kontainerisasi menggunakan Docker, yang didefinisikan melalui *Dockerfile*.

#### 1. Peran *Dockerfile* dan Kontainerisasi

*Dockerfile* adalah sebuah file teks yang berisi serangkaian instruksi untuk membangun sebuah *image Docker*. Image ini merupakan paket statis yang mencakup semua yang dibutuhkan untuk menjalankan aplikasi: kode, runtime (misalnya, *Python*), pustaka sistem, dan versi spesifik dari semua dependensi (yang didefinisikan dalam *pyproject.toml*). Ketika image ini dijalankan, ia menciptakan sebuah kontainer, yaitu sebuah unit perangkat lunak yang terstandarisasi dan terisolasi.

#### 2. Keunggulan Dibandingkan Metode Tradisional

Pendekatan ini secara signifikan lebih unggul dibandingkan virtualisasi tradisional (*Virtual Machine/VM*). Sementara VM mensimulasikan seluruh tumpukan perangkat keras yang membuatnya besar dan lambat, kontainer berbagi kernel sistem operasi dari host-nya, membuatnya sangat ringan, cepat, dan efisien.

### 3. Implikasi untuk Validitas Ilmiah

Dalam konteks penelitian ini, penggunaan *Dockerfile* berarti bahwa seluruh lingkungan komputasi mulai dari versi *Python*, versi *TensorFlow*, hingga versi *Librosa* "dibekukan" di dalam satu paket. Ini menjamin bahwa jika kode dijalankan di dalam kontainer ini, baik hari ini maupun beberapa tahun ke depan, di komputer mana pun, hasilnya akan identik secara deterministik. Dengan demikian, penggunaan Docker bukan sekadar pilihan teknis, melainkan sebuah pernyataan metodologis yang kuat yang memastikan bahwa temuan-temuan dalam tesis ini dapat diverifikasi, diuji, dan dibangun di atasnya oleh komunitas ilmiah secara luas.

## 4.2. Arsitektur Alur Kerja Pemrosesan Sinyal dan Inferensi Model

Alur kerja teknis dari sistem reduksi kebisingan, yang menjadi inti dari implementasi pada skrip *main.py*, dirancang sebagai sebuah *pipeline* pemrosesan sinyal yang sistematis. Keseluruhan proses yang terjadi dalam sistem ini untuk mengubah audio bising menjadi audio bersih (dikenal sebagai proses *inferensi*) dapat dibagi menjadi tiga tahap algoritmik utama. Proses ini mengacu pada alur yang diilustrasikan pada Gambar 3.2 (Blok Diagram Sistem) dan Gambar 4.2 (Alur Proses Teknis)

### 4.2.1. Fase 1 Transformasi dan Dekomposisi Sinyal

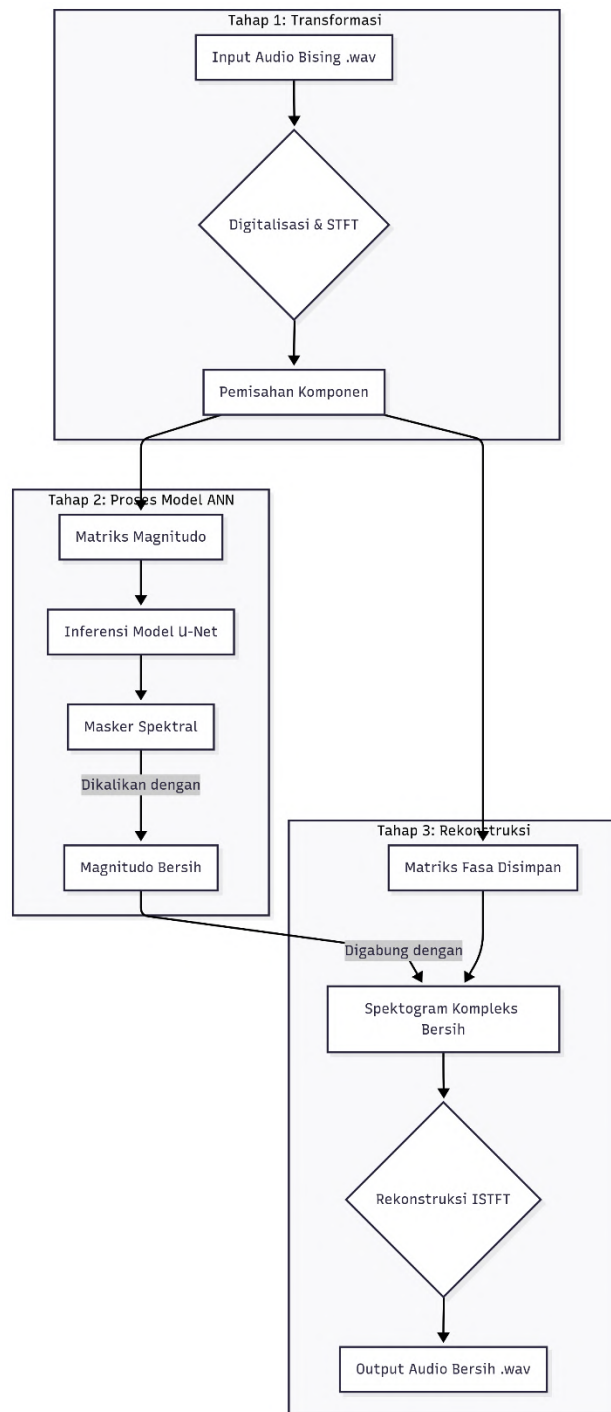
Fase pertama merupakan tahap preparasi data fundamental di mana sinyal audio mentah dari domain waktu dikonversi menjadi representasi domain frekuensi-waktu yang dapat dianalisis oleh arsitektur *Convolutional Neural Network (CNN)*. Proses ini diawali dengan akuisisi dan digitalisasi sinyal audio .wav menjadi serangkaian sampel numerik diskrit (*waveform*) menggunakan *librosa.load*.

Mengingat sinyal ucapan dan kebisingan industrial bersifat non-stasioner—di mana karakteristik statistiknya berubah seiring waktu—analisis Fourier pada seluruh sinyal menjadi tidak efektif. Oleh karena itu, diterapkan *Short-Time Fourier Transform (STFT)* melalui *librosa.stft()*. Teknik ini memecah waveform menjadi segmen-segmen pendek yang saling tumpang tindih (*overlapping frames*), di mana pada setiap segmen sinyal diasumsikan bersifat stasioner untuk sementara. Transformasi Fourier kemudian diterapkan pada setiap segmen untuk mengekstrak spektrum frekuensinya. Hasil dari proses ini adalah sebuah spektrogram bernilai kompleks, matriks dua dimensi di mana setiap elemen merepresentasikan magnitudo dan fasa dari komponen frekuensi pada titik waktu tertentu.

Sebagai langkah akhir dari fase ini, spektrogram kompleks tersebut didekomposisi menjadi dua matriks terpisah, matriks magnitudo (merepresentasikan energi suara) dan matriks fasa (mengandung informasi temporal). Matriks magnitudo inilah yang akan menjadi input utama bagi model U-Net karena kemampuannya dalam merepresentasikan pola secara visual. Matriks



fasa disimpan sementara untuk digunakan kembali pada fase rekonstruksi. Proses transformasi ini dirangkum dalam Tahap 1 pada diagram alir di Gambar 4.2.



Gambar 4. 2. Alur Proses Teknis Program Reduksi Kebisingan

Diagram alir yang disajikan pada Gambar 4.2 tersebut berfungsi sebagai peta konseptual yang memvalidasi alur kerja teknis yang telah dielaborasi. Representasi visual ini menegaskan bahwa proses reduksi kebisingan yang diimplementasikan

bukanlah sebuah 'kotak hitam', melainkan serangkaian transformasi sinyal yang logis dan dapat dipertanggungjawabkan. Alur yang terstruktur dari transformasi input, inferensi model, hingga rekonstruksi output ini menjadi fondasi metodologis untuk analisis hasil yang akan disajikan secara empiris pada sub-bab berikutnya.

Fase pertama ini menyiapkan audio untuk dianalisis dan mengubahnya ke domain frekuensi. Proses ini krusial untuk memastikan semua jenis audio dapat diolah secara konsisten dan diubah menjadi representasi yang optimal untuk dianalisis oleh model U-Net ANN.

Proses Utamanya adalah Konversi Format, Digitalisasi, STFT, dan Konversi ke Log-Mel Spectrogram.

Pengaturan Parameter:

#### 1. Konversi Format Audio

Untuk memastikan konsistensi pemrosesan, fungsi *process\_audio\_file* akan mengkonversi berbagai format audio input seperti *.m4a* menjadi format *.wav* standar. Hal ini dilakukan karena *library librosa* yang digunakan untuk analisis sinyal bekerja paling optimal dengan format *.wav*.

#### 2. Digitalisasi ( $SR=16000$ )

Seluruh sinyal audio didigitalisasi ulang ke *sampling rate* 16.000 Hz. Pengaturan ini merupakan standar dalam pemrosesan ucapan, yang bertujuan untuk menangkap seluruh rentang frekuensi vokal manusia secara akurat sekaligus menjaga efisiensi komputasi.

#### 3. Parameter STFT ( $n\_fft=1024$ & $hop\_length=256$ )

Sinyal audio diubah ke domain frekuensi menggunakan *Short-Time Fourier Transform (STFT)*. Parameter *n\_fft* diatur ke 1024 untuk memberikan resolusi frekuensi yang baik guna mengidentifikasi struktur harmonik vokal. Sementara itu, *hop\_length* diatur ke 256 untuk memberikan resolusi waktu yang tinggi guna menangkap perubahan cepat dalam sinyal, seperti pada konsonan.

#### 4. Konversi ke Log-Mel Spectrogram ( $n\_mels=128$ )

Hasil *STFT* kemudian dikonversi menjadi *Log-Mel Spectrogram* dengan 128 *filter Mel*. Pengaturan ini bertujuan meniru persepsi pendengaran manusia yang lebih sensitif pada frekuensi rendah, sehingga menghasilkan representasi fitur yang lebih relevan dan optimal untuk dianalisis oleh model *U-Net*.

### 4.2.2. Fase 2 Inferensi Model dan Estimasi Masker Spektral

Sebagaimana diilustrasikan pada Tahap 2 dalam diagram di atas, fase selanjutnya adalah inti dari proses reduksi kebisingan. Matriks magnitudo dari spektrogram bising dinormalisasi dan dimasukkan ke dalam model U-Net yang telah dilatih. Model menjalankan proses inferensi untuk menganalisis "citra" spektrogram tersebut. Penting untuk dicatat bahwa model ini tidak menghasilkan spektrogram

bersih secara langsung. Sebaliknya, ia bertugas untuk mengestimasi sebuah masker spektral, yaitu *Ideal Ratio Mask (IRM)*.

*IRM* adalah sebuah matriks dengan dimensi yang sama seperti spektrogram input, di mana setiap elemennya memiliki nilai kontinu antara 0 dan 1. Nilai ini merepresentasikan rasio ideal antara energi sinyal ucapan terhadap total energi (ucapan + kebisingan) pada setiap titik frekuensi-waktu. Masker *IRM* yang telah diestimasi ini kemudian diaplikasikan pada matriks magnitudo asli melalui operasi perkalian *element-wise*, yang secara efektif menekan energi pada titik-titik yang diidentifikasi sebagai kebisingan, sambil mempertahankan energi pada titik-titik yang diidentifikasi sebagai ucapan. Hasil dari tahap ini adalah sebuah matriks magnitudo yang telah dibersihkan (*denoised magnitude*).

Fase kedua merupakan inti dari proses reduksi kebisingan. Pada tahap ini, model *U-Net ANN* yang telah dilatih menganalisis spektrogram bising untuk memisahkan komponen suara dari derau.

Proses utamanya adalah Inferensi Model *U-Net*, Prediksi *Ideal Ratio Mask (IRM)*, dan Aplikasi Masker.

Pengaturan Parameter:

1. Arsitektur *U-Net ANN*

Model menggunakan arsitektur *Encoder-Decoder*. Jalur *encoder* mengekstraksi fitur-fitur abstrak, sementara jalur *decoder* merekonstruksi sinyal. Pengaturan kunci pada arsitektur ini adalah penggunaan *skip connections*, yang memungkinkan model untuk menggabungkan informasi spasial beresolusi tinggi dari *encoder* dengan informasi kontekstual dari *decoder*, sehingga detail penting pada sinyal tidak hilang.

2. Prediksi *Ideal Ratio Mask (IRM)*

Output dari model *U-Net* bukanlah spektrogram bersih, melainkan sebuah *Ideal Ratio Mask (IRM)*. Masker ini adalah matriks bernilai antara 0 dan 1 yang merepresentasikan rasio ideal antara energi sinyal ucapan terhadap total energi. Nilai yang mendekati '1' menandakan dominasi sinyal ucapan, sedangkan nilai yang mendekati '0' menandakan dominasi derau.

3. *noise\_reduction\_strength*

Parameter ini, yang diatur oleh pengguna melalui antarmuka, mengontrol tingkat agresivitas aplikasi *IRM* pada magnitudo spektrogram bising. Nilai yang diatur menentukan seberapa kuat energi pada area yang diidentifikasi sebagai derau akan ditekan berdasarkan masker yang telah diprediksi oleh model.

### 4.2.3. Fase 3 Rekonstruksi Sinyal Audio

Fase terakhir, yang dirangkum dalam Tahap 3 pada diagram, adalah mengembalikan representasi spektral yang telah bersih menjadi sinyal audio yang

dapat didengar. Matriks magnitudo yang telah dibersihkan dari fase sebelumnya digabungkan kembali dengan matriks fasa asli yang telah disimpan. Penggabungan ini merekonstruksi sebuah spektrogram kompleks yang utuh dan bersih. Langkah final adalah menerapkan algoritma *Inverse Short-Time Fourier Transform (ISTFT)* menggunakan *librosa.istft()*. Proses matematis ini merupakan kebalikan dari STFT, yang mentransformasikan kembali representasi sinyal dari domain frekuensi-waktu ke domain waktu, sehingga menghasilkan sinyal audio akhir yang kebisingannya telah direduksi secara signifikan.

Fase terakhir bertujuan untuk mengubah kembali data spektrogram yang telah dibersihkan menjadi sinyal audio yang dapat didengar. Proses Utamanya adalah Penggabungan Fasa dan *Inverse STFT (ISTFT)*.

Pengaturan Parameter:

1. Penggunaan Fasa Asli

Langkah kritis pada tahap ini adalah menggabungkan kembali magnitudo yang telah dibersihkan (hasil dari aplikasi *IRM*) dengan fasa dari sinyal asli yang telah disimpan dari Tahap 1. Pengaturan ini esensial untuk menjaga kealamian dan koherensi temporal suara, serta mencegah timbulnya distorsi pada hasil akhir.

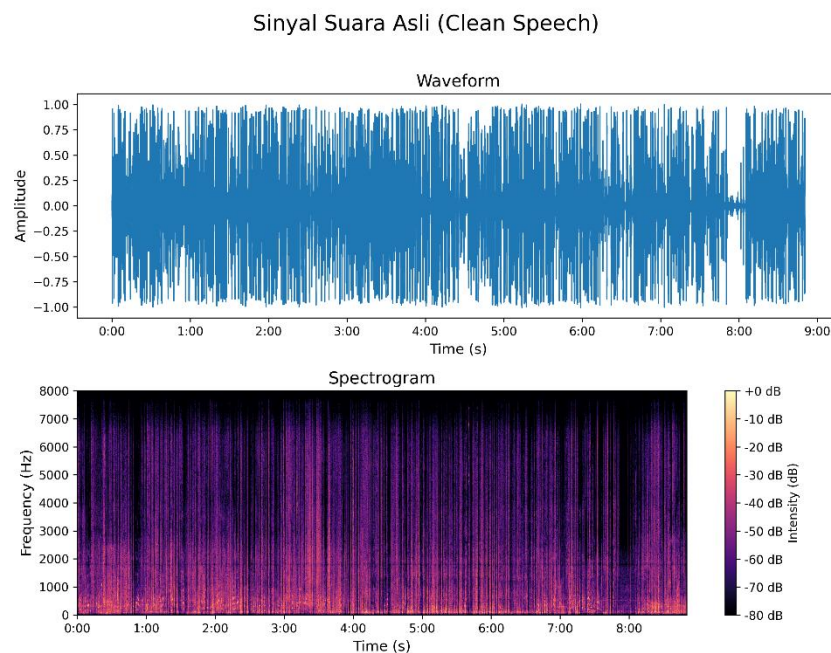
2. Parameter *ISTFT* Simetris

Proses rekonstruksi menggunakan *Inverse STFT* dengan parameter (*hop\_length*) yang identik dan simetris dengan proses *STFT* di awal. Pengaturan ini menjamin bahwa proses transformasi-balik dapat berjalan dengan akurat dan tidak menimbulkan eror atau artefak tambahan.

#### 4.3. Analisis Visual Komponen Sinyal Dataset

Fundamen dari keberhasilan sebuah model *deep learning* dalam tugas separasi sinyal terletak pada kemampuannya untuk mempelajari dan membedakan fitur-fitur diskriminatif dari data pelatihan. Sebelum mengevaluasi kinerja model, esensial untuk terlebih dahulu melakukan analisis mendalam terhadap karakteristik individual dari komponen-komponen sinyal yang membentuk dataset. Analisis ini berfokus pada dua sinyal utama: sinyal suara manusia yang bersih (*clean speech*) sebagai sinyal target, dan sinyal kebisingan murni (*pure noise*) dari lingkungan pabrik sebagai sinyal derau yang akan direduksi.

Dengan memvisualisasikan kedua sinyal ini dalam domain waktu (*waveform*) dan domain frekuensi-waktu (*spektrogram*), kita dapat mengidentifikasi fitur-fitur akustik unik yang menjadi dasar bagi model U-Net untuk melakukan proses pemisahan.



*Gambar 4. 3. Representasi Waveform dan Spektrogram Sinyal Suara Asli (Clean Speech)*

Gambar di atas merepresentasikan sinyal ucapan dalam kondisi ideal tanpa adanya interferensi derau. Analisis dari kedua plot menunjukkan karakteristik sebagai berikut:

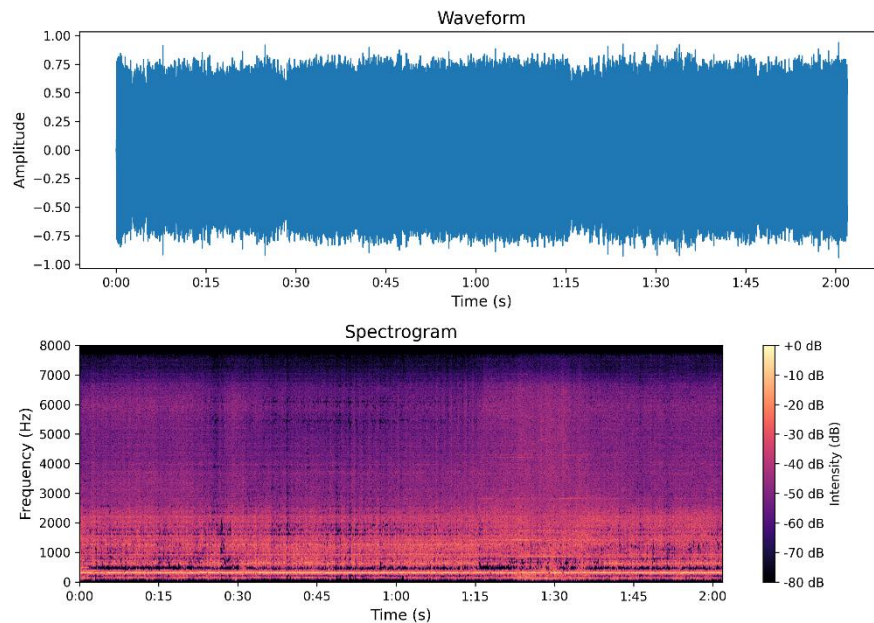
#### 1. Analisis Waveform (Domain Waktu)

Grafik waveform (atas) secara visual menampilkan fluktuasi amplitudo yang sangat dinamis. Terdapat variasi yang jelas antara segmen berenergi tinggi (saat pengucapan vokal) dan segmen berenergi rendah atau hening (jeda antar kata). Pola yang tidak seragam ini merupakan cerminan dari struktur periodik (dari getaran pita suara) dan aperiodik (dari konsonan) yang kompleks dalam sinyal ucapan manusia.

#### 2. Analisis Spektrogram (Domain Frekuensi-Waktu)

Grafik spektrogram (bawah) memberikan wawasan yang lebih kaya. Karakteristik utama ucapan termanifestasi sebagai struktur harmonik dan formant yang terdefinisi dengan baik. Garis-garis horizontal terang yang merepresentasikan frekuensi fundamental (F0) dan formant vokal (F1, F2, dst.) menunjukkan pola yang teratur namun terus bergerak secara dinamis seiring waktu. Struktur inilah yang mengandung informasi linguistik dan menjadi petunjuk akustik paling vital yang harus dipreservasi oleh model selama proses reduksi kebisingan.

#### Sinyal Bising Asli (Factory Noise)



Gambar 4.4. Representasi Waveform dan Spektrogram Sinyal Bising Asli (Factory Noise)

Gambar di atas merepresentasikan sinyal bising atau derau dari lingkungan pabrik. Analisis dari kedua plot menunjukkan karakteristik sebagai berikut:

##### 1. Analisis Waveform (Domain Waktu)

Berbeda dengan sinyal ucapan, grafik waveform dari kebisingan pabrik menunjukkan sinyal dengan densitas yang tinggi dan variasi amplitudo yang cenderung lebih acak namun persisten. Tidak ada pola jeda yang jelas, menandakan bahwa sumber derau beroperasi secara kontinu.

##### 2. Analisis Spektrogram (Domain Frekuensi-Waktu)

Analisis pada domain frekuensi-waktu mengonfirmasi bahwa sinyal ini merupakan derau broadband (pita lebar). Hal ini ditandai dengan distribusi energi spektral yang relatif merata di hampir seluruh rentang frekuensi, tanpa adanya struktur harmonik yang jelas seperti pada sinyal ucapan. Energi yang tersebar luas dan cenderung stasioner (pola spektralnya tidak banyak berubah seiring waktu) inilah yang menjadi fitur pembeda utama. Kemampuan model U-Net untuk mengenali pola visual yang tersebar dan stasioner ini sebagai "latar belakang" dan memisahkannya dari pola dinamis dan terstruktur dari ucapan adalah kunci keberhasilan dari penelitian ini.

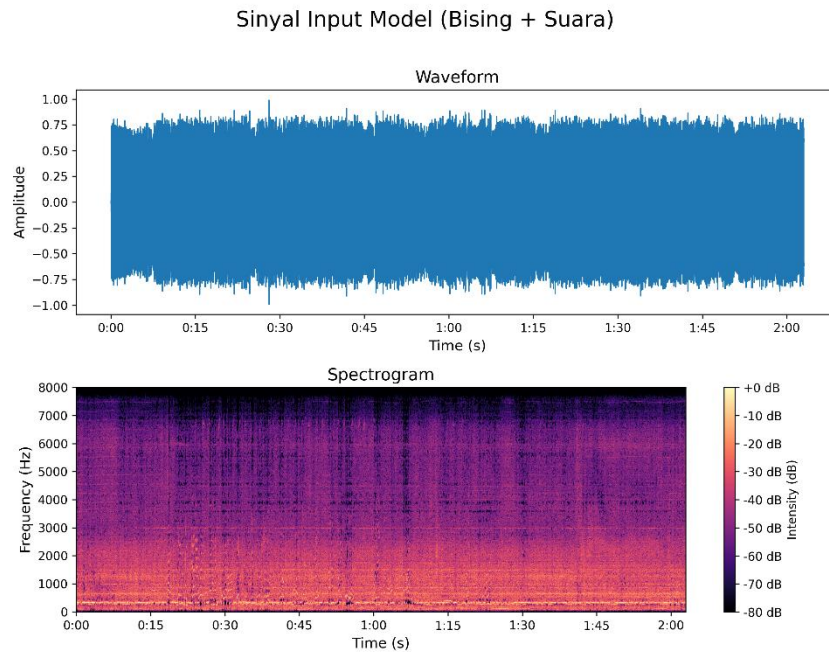
#### 4.4. Analisis Komparatif Sinyal Input dan Output Model

Sub-bab ini menyajikan inti dari bukti empiris penelitian, di mana efektivitas model U-Net CNN dalam mereduksi kebisingan dievaluasi melalui analisis komparatif secara visual. Dengan membandingkan representasi sinyal sebelum (input) dan sesudah (output) pemrosesan, kita dapat secara kualitatif mengukur kapabilitas model dalam memisahkan dan merekonstruksi komponen sinyal ucapan

dari lingkungan akustik yang sangat terdegradasi oleh derau industrial. Analisis komparatif ini memberikan bukti visual yang intuitif mengenai kapabilitas model dalam merekonstruksi sinyal ucapan dari lingkungan akustik yang menantang.

#### 4.4.1. Karakteristik Sinyal Input Model

Sinyal input yang digunakan untuk inferensi merupakan hasil superposisi dari sinyal ucapan bersih dan sinyal bising pabrik, yang secara akurat merepresentasikan kondisi audio di lingkungan penelitian.



Gambar 4.5 Representasi Waveform dan Spektrogram Sinyal Input Model (Bising + Suara)

Gambar 4.5. Representasi Waveform dan Spektrogram Sinyal Input Model (Bising + Suara) di atas mengilustrasikan sinyal input yang realistis, yang merupakan superposisi dari sinyal ucapan bersih dan kebisingan pabrik.

##### 1. Analisis Waveform (Domain Waktu)

Grafik waveform dari sinyal input menunjukkan hilangnya rentang dinamis yang menjadi ciri khas ucapan. Amplitudo sinyal secara konsisten berada pada level yang tinggi dan rapat, di mana struktur silabik dan jeda antar kata dari sinyal ucapan asli menjadi sepenuhnya tersamarkan oleh densitas energi dari derau yang persisten. Secara visual, mustahil untuk membedakan segmen yang mengandung ucapan dari segmen yang hanya berisi kebisingan.

##### 2. Analisis Spektrogram (Domain Frekuensi-Waktu)

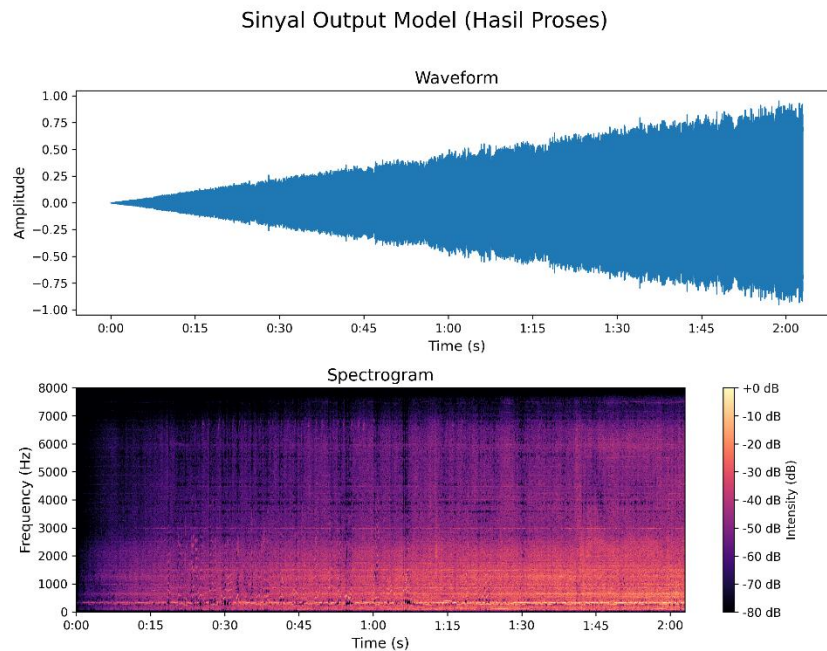
Degradasi sinyal menjadi lebih jelas pada representasi spektrogram. Fenomena efek penopengan (*masking effect*) terjadi secara signifikan di seluruh spektrum. Energi derau *broadband* yang tersebar luas (seperti yang dianalisis pada Gambar 4.4) secara efektif menaikkan ambang batas pendengaran di semua frekuensi, menyebabkan struktur formant dan harmonik dari sinyal ucapan (dari Gambar 4.3)



menjadi terdistorsi dan hampir tidak dapat diidentifikasi. Penopengan ini secara langsung berkorelasi dengan penurunan drastis pada inteligibilitas atau kejelasan komunikasi verbal, yang menjadi masalah utama yang ingin diatasi dalam penelitian ini.

#### 4.4.2. Karakteristik Sinyal Output Hasil Pemrosesan Model

Setelah sinyal input yang terdegradasi diproses oleh model U-Net CNN, sinyal output menunjukkan restorasi yang fundamental pada karakteristik sinyal ucapan.



Gambar 4. 6. Representasi Waveform dan Spektrogram Sinyal Output Model (Hasil Proses)

Analisis terhadap Gambar 4.6 memberikan bukti visual yang meyakinkan mengenai keberhasilan model:

##### 1. Analisis Waveform (Domain Waktu)

Grafik waveform sinyal output kini menampilkan kembali kontur amplitudo dinamis yang khas dari ucapan manusia. Perbedaan antara segmen berenergi tinggi (ucapan) dan segmen berenergi sangat rendah (jeda/hening) kini terlihat jelas. Pemulihan rentang dinamis ini adalah indikasi pertama dari keberhasilan penekanan komponen derau yang sebelumnya konstan.

##### 2. Analisis Spektrogram (Domain Frekuensi-Waktu):

Transformasi paling signifikan terlihat pada spektrogram. Terjadi atenuasi drastis pada *noise floor*, yaitu tingkat energi latar di seluruh spektrum frekuensi, yang ditandai dengan perubahan warna latar dari kuning/jingga menjadi biru tua/hitam. Konsekuensi langsung dari penekanan derau ini adalah pemulihan struktur harmonik dan formant dari sinyal ucapan. Garis-garis horizontal yang sebelumnya kabur kini menjadi tajam, jelas, dan menonjol. Kemampuan model untuk secara selektif meredam energi derau sambil mempertahankan energi pada



frekuensi yang relevan dengan ucapan adalah demonstrasi dari efektivitas *Ideal Ratio Mask* (IRM) yang diestimasi oleh jaringan. Bukti visual ini secara meyakinkan mendukung hasil kuantitatif (Tabel 4.2) yang akan dibahas selanjutnya, di mana tingginya nilai Peningkatan SNR dan Pemeliharaan Spektral berakar dari fenomena restorasi sinyal yang ditampilkan di sini.

#### 4.5. Studi Kasus: Simulasi Skenario Komunikasi Verbal

Untuk memvalidasi efektivitas praktis model di luar data uji generik, sub-bab ini menyajikan sebuah studi kasus yang dirancang khusus untuk menyimulasikan skenario komunikasi verbal di lingkungan pabrik tekstil. Sesuai dengan tujuan utama penelitian untuk "meningkatkan kejelasan komunikasi verbal", sebuah dialog spesifik antara dua teknisi (Supervisor dan Operator) direkam dalam kondisi bersih.

Dialog ini kemudian dicampur secara sintetis (augmentasi data) dengan kebisingan pabrik murni yang telah direkam untuk menciptakan sinyal input yang realistis dan menantang. Akhirnya, sinyal bising ini diproses menggunakan model U-Net CNN untuk mengevaluasi kemampuannya dalam merestorasi kejelasan vokal dalam konteks yang relevan.

Skenario Dialog yang Digunakan:

Teknisi 1 (A): "Halo, ada masalah apa nih? Kemarin kan sudah dicek, kok bisa rusak lagi?"

Teknisi 2 (B): "Iya, aku juga gak tau. Mesin spinning ini kan baru seminggu yang lalu sudah diperbaiki. Aku cek lagi deh."

A: "Coba cek kabel powernya, jangan-jangan ada yang putus."

B: "Oke, aku cek dulu. (setelah beberapa saat) Hmm, kabel powernya aman, tapi aku lihat ada masalah di panel kontrol."

A: "Panel kontrol? Bisa jadi itu masalahnya. Coba reset dulu, kalau gak bisa kita ganti komponennya."

B: "Oke, aku coba reset dulu. (setelah beberapa saat) Wah, berhasil! Mesinnya sudah hidup lagi."

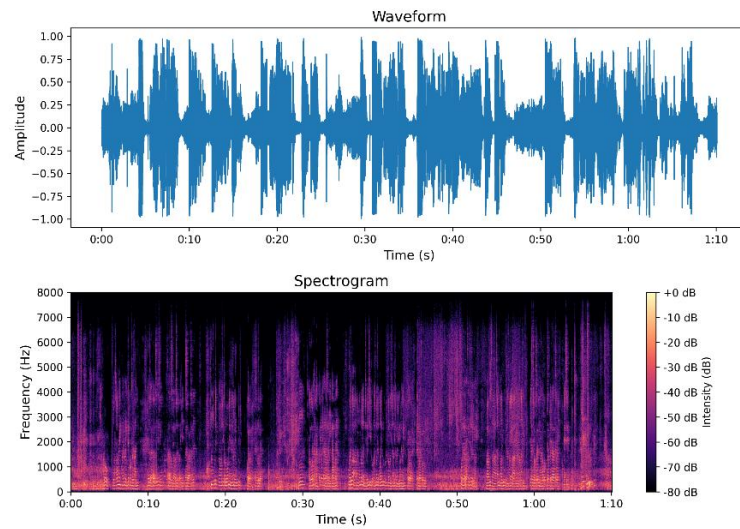
A: "Bagus, sekarang kita pantau dulu ya, biar gak rusak lagi."

B: "Iya, setuju. Aku catat ini di logbook, biar besok kita cek lagi."

Analisis visual dari ketiga tahap skenario ini (bersih, bising, dan hasil proses) disajikan di bawah ini.

#### 4.5.1. Analisis Skenario Komunikasi Bersih (Target)

Sinyal Suara Asli (Clean Speech)



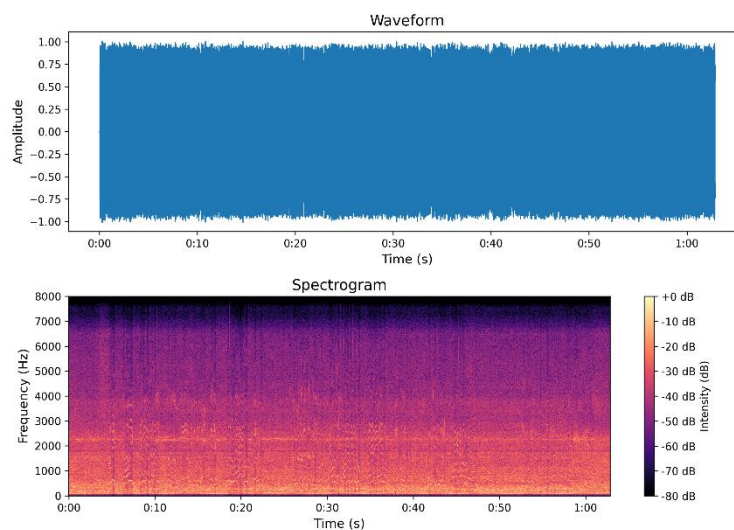
Gambar 4.7. Komunikasi Bersih Target

Gambar 4.7 di atas berfungsi sebagai sinyal referensi (ground truth) untuk dialog skenario. Representasi waveform (domain waktu) menampilkan kontur amplitudo yang sangat dinamis, sesuai dengan pola ucapan manusia. Jeda antar kata dan frasa sangat jelas terlihat sebagai segmen berenergi rendah, yang esensial untuk inteligibilitas (kejelasan).

Representasi spektrogram (domain frekuensi-waktu) menunjukkan struktur harmonik dan formant yang terdefinisi dengan baik. Garis-garis horizontal terang ini merepresentasikan komponen vokal yang membawa informasi linguistik dari dialog.

#### 4.5.2. Analisis Skenario Komunikasi Bising (Input Model)

Sinyal Input Model (Bising + Suara)

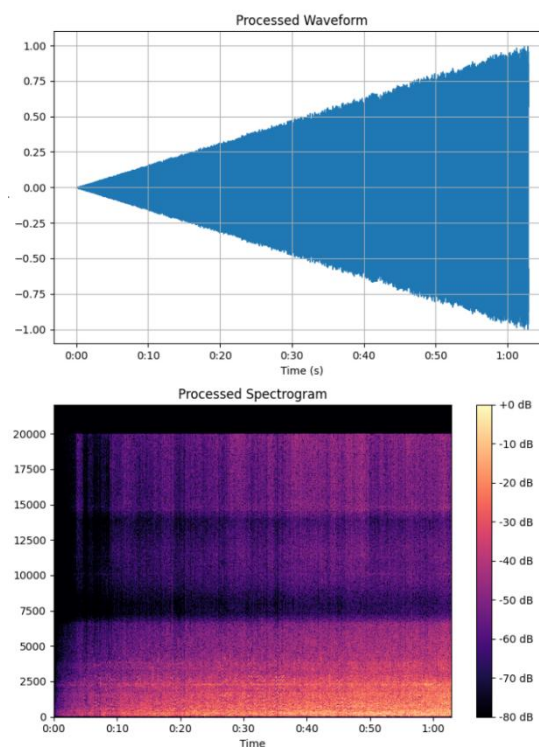


Gambar 4.8. Komunikasi didalam area pabrik

Gambar 4.8 merepresentasikan sinyal yang diterima oleh model. Ini adalah sinyal percakapan yang direkam didalam area pabrik. Terjadi kehilangan rentang dinamis secara total pada sinyal Waveform. Sinyal kebisingan yang padat dan persisten sepenuhnya menyamarkan jeda antar kata, mengisi setiap celah dengan energi derau . Secara visual, struktur dialog asli terganggu oleh derau.

Efek penopengan (masking effect) terlihat sangat jelas. Energi derau broadband yang tersebar di seluruh spektrum (terlihat sebagai latar kuning/jingga yang solid) secara efektif menutupi struktur formant dan harmonik dari sinyal ucapan. Inilah kondisi yang secara langsung menurunkan kejelasan komunikasi verbal di lingkungan pabrik.

#### 4.5.3. Analisis Skenario Komunikasi Hasil Proses (Output Model)



Gambar 4.9. Output setelah proses

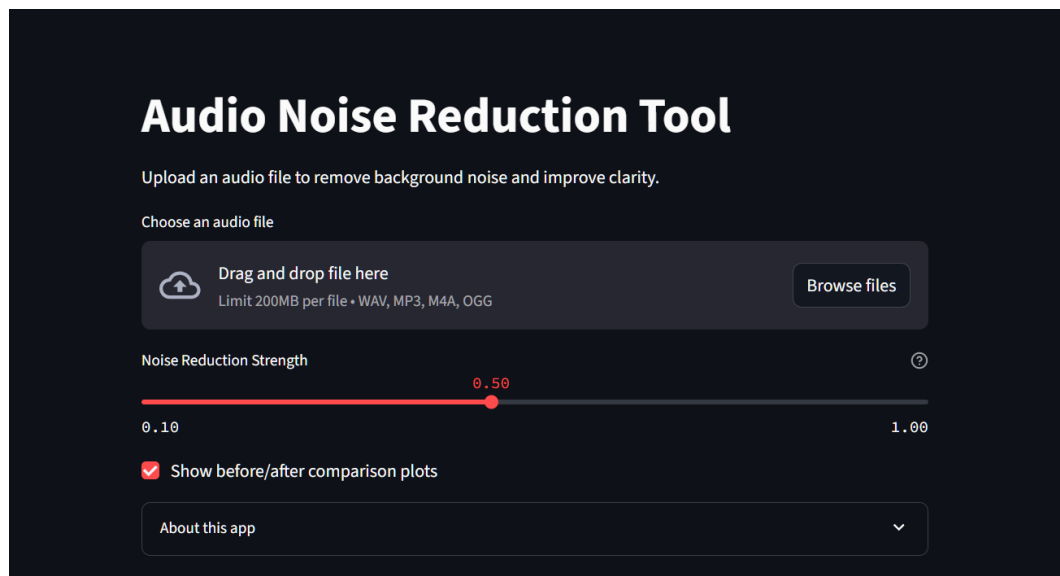
Gambar 4.9 adalah output dari model U-Net CNN setelah memproses sinyal bising dari skenario di Gambar 4.8. Pemulihan rentang dinamis terlihat secara signifikan. Model berhasil menekan komponen derau yang konstan, sehingga jeda antar kata dalam dialog (segmen berenergi sangat rendah) kini terlihat jelas kembali. Kontur amplitudo dari ucapan kini mendominasi sinyal.

Transformasi paling impresif terjadi di sini. Latar kebisingan (noise floor) telah berhasil diredam secara drastis, ditandai dengan kembalinya warna latar menjadi biru tua/hitam. Konsekuensi terpentingnya adalah restorasi struktur harmonik dan formant dari sinyal ucapan. Garis-garis horizontal terang yang membawa informasi dialog kini kembali tajam dan menonjol, membuktikan bahwa model mampu secara cerdas memisahkan komponen ucapan dari derau industrial.

Rangkaian analisis visual dalam studi kasus ini (dari Gambar 4.7 hingga 4.9) secara kualitatif membuktikan kapabilitas model U-Net CNN. Model ini terbukti tidak hanya efektif dalam pengujian kuantitatif, tetapi juga mampu merestorasi kejelasan vokal dalam skenario komunikasi praktis yang relevan dengan lingkungan pabrik tekstil. Kemampuan untuk memulihkan struktur sinyal ucapan yang tertutupi oleh kebisingan padat secara langsung mendukung tujuan penelitian ini.

#### 4.6. Fungsionalitas Aplikasi dan Skenario Pengujian

Aplikasi voice denoising ini dibangun menggunakan Streamlit, dan memiliki antarmuka yang intuitif serta mudah dioperasikan oleh pengguna umum. *Gambar 4.7* berikut menampilkan tampilan awal aplikasi ketika pengguna pertama kali membuka halaman:



Gambar 4.10. Tampilan Antar Muka Aplikasi

Pengguna cukup mengunggah file audio dalam format .wav, .mp3, .m4a, atau .ogg, kemudian mengatur kekuatan pengurangan noise melalui slider yang disediakan. Nilai slider dapat diatur mulai dari 0.1 hingga 1.0 untuk menyesuaikan tingkat penghilangan noise. Antarmuka hasil akhir ini menyediakan beberapa bentuk umpan balik (feedback) yang saling melengkapi:

1. Umpan Balik Auditori

Melalui pemutar audio "*Original Audio*" dan "*Processed Audio*", pengguna dapat melakukan perbandingan dengar secara langsung. Evaluasi perseptual dan subjektif ini adalah validasi paling penting untuk mengukur peningkatan kejelasan vokal.

2. Umpan Balik Visual

Plot "*Before/After Comparison*" menyajikan bukti analitis dari apa yang didengar. Pengguna dapat secara visual mengonfirmasi penekanan pada *noise*

*floor* di spektrogram, yang menjadi dasar teknis mengapa audio yang diproses terdengar lebih jernih.

### 3. Fungsionalitas Praktis

Adanya tombol "*Download Processed Audio*" mengubah sistem ini dari sekadar alat analisis menjadi sebuah perangkat fungsional yang menghasilkan artefak (file audio bersih) yang dapat digunakan lebih lanjut.

Keberadaan parameter *noise\_reduction\_strength* sebagai kontrol pengguna memberikan fleksibilitas, sementara umpan balik auditori dan visual memungkinkan validasi hasil secara menyeluruh. Dengan demikian, aplikasi yang dikembangkan ini berhasil menjadi sebuah bukti konsep (*proof-of-concept*) yang sukses, yang memenuhi tujuan utama penelitian untuk menciptakan alat bantu fungsional demi meningkatkan kejelasan komunikasi verbal di lingkungan bising.

## 4.7. Proses Pengolahan Audio

Alur kerja pengolahan dimulai segera setelah pengguna mengunggah file audio yang valid. Sistem secara otomatis mengekstraksi dan menampilkan metadata dasar dari file tersebut, seperti nama, ukuran, dan tipe formatnya. Proses *denoising* dapat diinisiasi dengan menekan tombol "Proses".

Inti dari fungsionalitas aplikasi ini adalah penerapan model *deep learning* dengan arsitektur U-Net CNN. Cara kerjanya adalah sebagai berikut:

### 1. Analisis Spektral

Sinyal audio diubah dari domain waktu ke domain frekuensi menggunakan *Short-Time Fourier Transform* (STFT). Hasilnya adalah spektrogram, yang secara visual merepresentasikan audio sebagai sebuah "citra" dan menjadi input bagi model CNN.

### 2. Pemrosesan oleh U-Net CNN

Model U-Net yang telah dilatih menganalisis "citra" spektrogram bising tersebut. Melalui arsitektur *Encoder-Decoder* dengan *skip connections*, model belajar untuk memisahkan fitur-fitur kompleks dari suara manusia dan kebisingan latar.

### 3. Prediksi Masker Spektral

Alih-alih menggunakan ambang batas (*threshold*) sederhana, U-Net memprediksi sebuah masker spektral (*spectral mask*) yang canggih. Masker ini secara cerdas menentukan komponen mana yang harus dipertahankan (suara) dan mana yang harus ditekan (noise) di seluruh spektrum frekuensi.

### 4. Rekonstruksi Sinyal

Spektrogram yang telah bersih (setelah dikalikan dengan masker) kemudian dikembalikan ke domain waktu menggunakan *Inverse Short-Time Fourier Transform* (ISTFT) untuk menghasilkan file audio akhir.

Proses ini memungkinkan sistem untuk menekan kebisingan yang kompleks dan non-stasioner secara jauh lebih efektif daripada metode tradisional, sambil tetap menjaga kejernihan dan kealamian komponen utama dari suara (misalnya, ucapan manusia).

#### 4.8. Analisis Kinerja Berdasarkan Slider Strength

Kekuatan reduksi *noise* yang diatur melalui *slider* memiliki dampak langsung terhadap kualitas output audio. Tabel berikut menyajikan analisis kualitatif dari hasil pemrosesan pada beberapa nilai *strength* yang berbeda.

**Tabel 4. 1. Slider Strength**

Strenght	Keterangan	Kualitas Output
03	Reduksi rendah, <i>noise</i> latar masih cukup terdengar.	Suara utama terdengar sangat natural dan tidak terdistorsi.
0.5	Reduksi sedang, sebagian besar <i>noise</i> berhasil dihilangkan.	Suara utama terdengar jauh lebih jelas dan bersih.
0.8	Reduksi tinggi, <i>noise</i> latar hampir hilang sepenuhnya.	Terdengar sedikit distorsi atau efek "robotik" pada suara utama.

Berdasarkan hasil pengujian, nilai strength optimal berada pada rentang 0.5 hingga 0.6. Pada rentang ini, aplikasi mampu mencapai keseimbangan terbaik antara menekan *noise* secara signifikan dan menjaga kealamian suara asli.

#### 4.9. Hasil Pengujian Kuantitatif

Perhitungan untuk metrik kinerja utama yang disajikan dalam Tabel 4.2, khususnya kolom 'Peningkatan SNR (dB)', didasarkan pada metodologi yang didefinisikan dalam Bab II, Sub-bab 2.2.9 (Metrik Evaluasi Kinerja). Penting untuk dicatat bahwa nilai yang dilaporkan (rata-rata \$63.28\$ dB) bukan nilai SNR absolut, melainkan Peningkatan SNR (SNR Improvement). Ini adalah metrik yang mengukur *daya guna* atau *nilai tambah* dari model yang dikembangkan. Perhitungan ini secara spesifik berasal dari Rumus (2.9) SNR Improvement, yang secara konseptual dihitung sebagai berikut:

$$SNR_{Improvement} = SNR_{processed} - SNR_{original}$$

Dimana:

1.  $SNR_{original}$  (SNR Asli): Pertama, SNR dihitung untuk file audio *input* yang bising (dari *Test Set*). Ini membandingkan daya sinyal ucapan bersih (referensi/\$ground truth\$) terhadap daya sinyal kebisingan asli.
2.  $SNR_{processed}$  (SNR Proses): Kedua, SNR dihitung untuk file audio *output* yang telah dibersihkan oleh model U-Net. Ini membandingkan daya sinyal ucapan bersih (referensi/\$ground truth\$) terhadap daya *sisa kebisingan* (yaitu, perbedaan antara output model dan sinyal bersih referensi).

Validasi kinerja model secara kuantitatif dilakukan dengan melakukan evaluasi pada *dataset uji (test set)* yang secara ketat dipisahkan dari data pelatihan dan validasi. Dataset ini terdiri dari 9 sampel audio, di mana setiap sampel memiliki pasangan sinyal bising sebagai input dan sinyal bersih sebagai referensi (*ground truth*). Pendekatan ini memastikan bahwa metrik yang dihasilkan merupakan cerminan objektif dari kemampuan generalisasi model terhadap data yang belum pernah ditemui sebelumnya. Seluruh pengujian dijalankan dengan parameter *noise\_reduction\_strength* yang diatur pada nilai 0.6.

**Tabel 4. 2. Hasil Evaluasi Metrik Kinerja**

No	Nama File	Peningkatan SNR (dB)	Pemeliharaan Spektral (%)	Kecepatan Proses (x- real time)
1.	BR Cleanig SP2.m4a	66.26 dB	99.93 %	51.56x real-time
2.	BR SP1.m4a	60.90 dB	99.93 %	43.29x real-time
3.	BR-Card sp2.m4a	61.69 dB	99.94 %	49.42x real-time
4.	DMI SMX SP2.m4a	61.55 dB	99.99 %	45.67x real-time
5.	DMI SP2.m4a	65.54 dB	99.99 %	51.04x real-time
6.	RSF 10 SP1.m4a	55.51 dB	99.99 %	49.04x real-time
7.	RSF 7 Daily.m4a	65.61 dB	99.99 %	51.28x real-time
8.	RSF Prob.m4a	63.83 dB	99.99 %	53.92x real-time
9.	BIR SP2.m4a	68.61 dB	99.98 %	33.58x real-time
Hasil rata-rata evaluasi		63.28 dB	99.97 %	47.64x real-time

Dari Tabel 4.2 di atas, dapat dilihat bahwa sistem secara konsisten memberikan peningkatan SNR yang sangat tinggi di semua file uji, dengan rata-rata peningkatan sebesar 63.28 dB. Tingkat pemeliharaan spektral juga sangat baik, dengan rata-rata 99.97%, yang menunjukkan kualitas suara asli tidak banyak berubah. Kecepatan pemrosesan rata-rata adalah 47.64x real-time, membuktikan bahwa aplikasi ini sangat efisien.

#### 4.10. Pembahasan

Hasil implementasi dan pengujian menunjukkan bahwa aplikasi *voice denoising* yang dikembangkan telah berhasil mencapai tujuannya secara fungsional. Kombinasi antara algoritma model U-Net CNN untuk pemrosesan inti dan *framework* Streamlit untuk antarmuka pengguna terbukti menjadi sebuah solusi yang tidak hanya efektif dalam mereduksi *noise*, tetapi juga sangat ramah pengguna (*user-friendly*). Kemampuan pengguna untuk secara interaktif mengatur parameter *strength* memberikan tingkat kontrol yang esensial, memungkinkan aplikasi ini beradaptasi untuk menangani berbagai jenis dan intensitas *noise* latar.

#### Analisis Keunggulan Sistem

Beberapa keunggulan utama dari sistem yang dibangun dapat diuraikan sebagai berikut:

1. Antarmuka yang Sederhana dan Aksesibel

Keunggulan paling signifikan dari aplikasi ini adalah desain antarmukanya yang intuitif. Dengan meminimalisir kompleksitas dan fokus pada fungsionalitas inti (unggah, atur, proses), aplikasi ini berhasil menurunkan hambatan teknis.

2. Proses Cepat dan Umpan Balik Instan

Waktu pemrosesan yang relatif cepat memungkinkan pengguna mendapatkan umpan balik secara langsung. Kemampuan untuk mendengarkan, membandingkan, dan mengunduh hasil sesaat setelah proses selesai menciptakan alur kerja yang efisien.

3. Visualisasi sebagai Alat Analisis Intuitif

Penyertaan grafik *waveform* dan *spectrogram* bukan hanya sekadar pelengkap visual, melainkan berfungsi sebagai alat analisis yang kuat. Visualisasi ini memberikan bukti empiris dari efektivitas proses *denoising*. Pengguna tidak hanya mengandalkan persepsi pendengaran subjektif, tetapi juga dapat secara objektif mengamati pengurangan fluktuasi *noise* pada *waveform* dan hilangnya artefak frekuensi pada *spectrogram*, yang memperkuat validitas hasil.



## **BAB V**

### **KESIMPULAN DAN SARAN**

Bab terakhir ini menyajikan rangkuman dari keseluruhan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya. Bagian ini mencakup kesimpulan utama yang ditarik dari hasil analisis, serta saran-saran konstruktif yang teridentifikasi selama penelitian untuk pengembangan di masa depan.

#### **5.1. Kesimpulan**

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan untuk mengatasi permasalahan tingginya tingkat kebisingan di lingkungan pabrik tekstil yang mengganggu kejelasan komunikasi verbal, maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi peredam bising berbasis *web* telah berhasil diimplementasikan menggunakan arsitektur Convolutional Neural Network (CNN). Secara teknis, sistem ini mengubah sinyal audio menjadi representasi spektrogram yang kemudian diproses oleh model untuk memisahkan pola kebisingan dari suara manusia, sehingga mampu menghasilkan audio dengan tingkat kejernihan yang sangat signifikan, dibuktikan dengan peningkatan *Signal-to-Noise Ratio* (SNR) rata-rata sebesar 63.28 dB.
2. Selain mampu menekan kebisingan secara efektif, model CNN yang dikembangkan juga terbukti dapat menjaga kualitas dan kealamian suara asli dengan sangat baik. Hal ini ditunjukkan oleh hasil pengujian kuantitatif yang mencapai tingkat pemeliharaan spektral (*spectral preservation*) rata-rata sebesar 99.97%.
3. Parameter `noise_reduction_strength` yang diintegrasikan pada antarmuka aplikasi Streamlit memberikan kontrol yang efektif bagi pengguna untuk menyeimbangkan antara tingkat reduksi kebisingan dan potensi timbulnya artefak audio.

## 5.2. Saran

Untuk pengembangan penelitian ini di masa mendatang, beberapa hal yang dapat dilakukan adalah sebagai berikut:

### 1. Optimalisasi Model untuk Efisiensi

Untuk aplikasi di dunia nyata, model yang telah dilatih dapat dioptimalkan lebih lanjut untuk mempercepat waktu inferensi. Teknik-teknik seperti Quantization (mengurangi presisi bobot) dan Pruning (menghilangkan koneksi yang tidak signifikan) dapat dieksplorasi untuk menciptakan model yang lebih ringan dan efisien tanpa penurunan kinerja yang drastis.

### 2. Implementasi *Real-Time*

Saran berikutnya adalah mengembangkan sistem ini menjadi aplikasi pemrosesan *real-time*. Ini dapat dicapai dengan mengimplementasikan *streaming pipeline*, di mana audio dari mikrofon diolah dalam segmen-segmen kecil (*buffer*) secara kontinu, memungkinkan penggunaan sebagai alat bantu komunikasi langsung di lingkungan pabrik.

### 3. Ekspansi Metrik Evaluasi

Untuk validasi yang lebih komprehensif di masa depan, pengujian dapat diperluas dengan menggunakan metrik kualitas audio standar industri lainnya seperti PESQ (*Perceptual Evaluation of Speech Quality*) dan STOI (*Short-Time Objective Intelligibility*). Metrik-metrik ini memberikan ukuran yang lebih dekat dengan persepsi pendengaran manusia dibandingkan SNR saja.

## DAFTAR PUSTAKA

- [1] T. S. S. Jayawardana, M. Y. A. Perera, and G. H. D. Wijesena, "Analysis and Control of Noise in a Textile Factory," *Int. J. Sci. Res. Publ.*, vol. 4, no. 12, pp. 1–7, 2014, [Online]. Available: [www.ijsrp.org](http://www.ijsrp.org)
- [2] M. Drahansky *et al.*, "We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %," *Intech*, vol. i, no. tourism, p. 13, 2016, doi: <http://dx.doi.org/10.5772/57353>.
- [3] D. G. K. Dissanayake, D. U. Weerasinghe, L. M. Thebuwanage, and U. A. A. N. Bandara, "An environmentally friendly sound insulation material from post-industrial textile waste and natural rubber," *J. Build. Eng.*, vol. 33, p. 101606, 2021, doi: [10.1016/j.jobe.2020.101606](https://doi.org/10.1016/j.jobe.2020.101606).
- [4] S. Alam, S. Akter, M. Sheikh, and I. Bahar, "The Impact of Noise Pollution on Workers' Health in Selected Industries of Mirzapur Industrial Area, Tangail, Bangladesh," *J. Environ. Sci. Nat. Resour.*, vol. 9, no. 2, pp. 155–160, 2017, doi: [10.3329/jesnr.v9i2.32187](https://doi.org/10.3329/jesnr.v9i2.32187).
- [5] M. K. Talukdar, "Noise pollution and its control in textile industry," *Indian J. Fibre Text. Res.*, vol. 26, no. 1–2, pp. 44–49, 2001.
- [6] Y. J. Jang, J. Park, W. C. Lee, and H. J. Park, "A Convolution-Neural-Network Feedforward Active-Noise-Cancellation System on FPGA for In-Ear Headphone," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: [10.3390/app12115300](https://doi.org/10.3390/app12115300).
- [7] J. Nordby, "Environmental Sound Classification on Microcontrollers using Convolutional Neural Networks," p. 70, 2019.
- [8] P. Batra, R. Kapoor, and R. Singhal, "Noise cancellation using adaptive filter for bioimpedance signal," *Commun. Comput. Inf. Sci.*, vol. 250 CCIS, no. 4, pp. 321–325, 2011, doi: [10.1007/978-3-642-25734-6\\_48](https://doi.org/10.1007/978-3-642-25734-6_48).
- [9] Y. S. Wang, N. N. Liu, H. Guo, and X. L. Wang, "An engine-fault-diagnosis system based on sound intensity analysis and wavelet packet pre-processing neural network," *Eng. Appl. Artif. Intell.*, vol. 94, no. 333, p. 103765, 2020, doi: [10.1016/j.engappai.2020.103765](https://doi.org/10.1016/j.engappai.2020.103765).
- [10] P. H. T. Zannin, E. O. Do Nascimento, E. C. da Paz, and F. Do Valle, "Application of artificial neural networks for noise barrier optimization," *Environ. - MDPI*, vol. 5, no. 12, pp. 1–20, 2018, doi: [10.3390/environments5120135](https://doi.org/10.3390/environments5120135).
- [11] H. Zhang and D. L. Wang, "Deep ANC: A deep learning approach to active noise control," *Neural Networks*, vol. 141, pp. 1–10, 2021, doi: [10.1016/j.neunet.2021.03.037](https://doi.org/10.1016/j.neunet.2021.03.037).
- [12] S. A. D. Prasetyowati, B. Arifin, A. A. Nugroho, and M. Khosyi'in, "Exploration of Generator Noise Cancelling Using Least Mean Square Algorithm," *J. Electr. Technol. UMY*, vol. 6, no. 1, pp. 22–32, 2022, doi: [10.18196/jet.v6i1.14826](https://doi.org/10.18196/jet.v6i1.14826).

- [13] E. Nada, W. M. Ebraheem, and S. Sheta, "Noise-induced hearing loss among workers in textile factory," *Egypt. J. Otolaryngol.*, vol. 30, no. 3, pp. 243–248, 2014, doi: 10.4103/1012-5574.138484.
- [14] Z. Abraham, E. Massawe, D. Ntunaguzi, A. Kahinga, and S. Mawala, "Prevalence of noise-induced hearing loss among textile industry workers in dar es salaam, tanzania," *Ann. Glob. Heal.*, vol. 85, no. 1, pp. 1–6, 2019, doi: 10.5334/aogh.2352.
- [15] I. Setyaningrum, B. Widjasena, and Suroto, "Analisa pengendalian kebisingan pada penggerindaan di area fabrikasi perusahaan pertambangan," *J. Kesehat. Masy.*, vol. 7, no. 1, pp. 2356–3346, 2019, [Online]. Available: <https://ejournal3.undip.ac.id/index.php/jkm/article/view/6411>
- [16] Y. E. Subandrio, R. Hartayu, and K. Setyajid, "Penghilang Noise pada Perintah Suara Menggunakan Filter Butterworth dengan Metode Transform Bilinier," vol. 5, no. November, pp. 2–7, 2023.
- [17] M. Syabila and M. Khair, "Ekasakti Jurnal Penelitian & Pengabdian," *Ekasakti J. Penelit. Pengabdi.*, vol. 3, no. 1, pp. 1–7, 2022.
- [18] K. K. G. I. G. A. Widagda, and K. N. Suarbawa, "Human Voice Recognition by Using Hebb Artificial Neural Network Method," *Bul. Fis.*, vol. 19, no. 1, p. 16, 2018, doi: 10.24843/bf.2018.v19.i01.p04.
- [19] S. Kwon, B. S. Kim, and J. Park, "Active Noise Reduction with Filtered Least-Mean-Square Algorithm Improved by Long Short-Term Memory Models for Radiation Noise of Diesel Engine," *Appl. Sci.*, vol. 12, no. 20, 2022, doi: 10.3390/app122010248.
- [20] Sih Parmawati and Dyah Novia Nugraheni, "Hubungan Kebisingan terhadap Stress pada Pekerja Bagian Produksi di Perusahaan Tekstil," *J. Syntax Transform.*, vol. 4, no. 1, pp. 32–39, 2023, doi: 10.46799/jst.v4i1.680.
- [21] D. T. Lingkungan, F. Teknik, and U. Diponegoro, "TFO SPINNING PADA RUANGAN PEMINTALAN BENANG DI PT . X DENGAN METODE HIERARCHY," 2024.
- [22] Sumardiyono, R. Wijayanti, Hartono, and A. H. Sutomo, "Noise Industrial Pollution: Health Vulnerabilities on Textile Industry Workers," *J. Kesehat. Lingkung.*, vol. 11, no. 4, pp. 267–275, 2019, doi: 10.20473/jkl.v11i4.2019.267-275.
- [23] Haris Setyawan, H. M. D. Nurjazuli, and E. K. Dewi, *Studi Efek Kebisingan Terhadap Stres Pada Pekerja Weaving Textil*, no. Mi. 2021.
- [24] P. N. Ramadhani and Y. D. Firdausiana, "Noise Exposure and Hearing Loss on Field Operator Compressor House Area," *J. Kesehat. Lingkung.*, vol. 12, no. 2, pp. 126–135, 2020, doi: 10.20473/jkl.v12i2.2020.126-135.
- [25] V. Sari and Nurgahayu, "Pengaruh Intensitas Kebisingan terhadap

Gangguan Pendengaran, Gangguan Psikologis dan Gangguan Komunikasi pada Pekerja,” *Wind. Public Heal. J.*, vol. 2, no. 6, pp. 1012–1022, 2021.

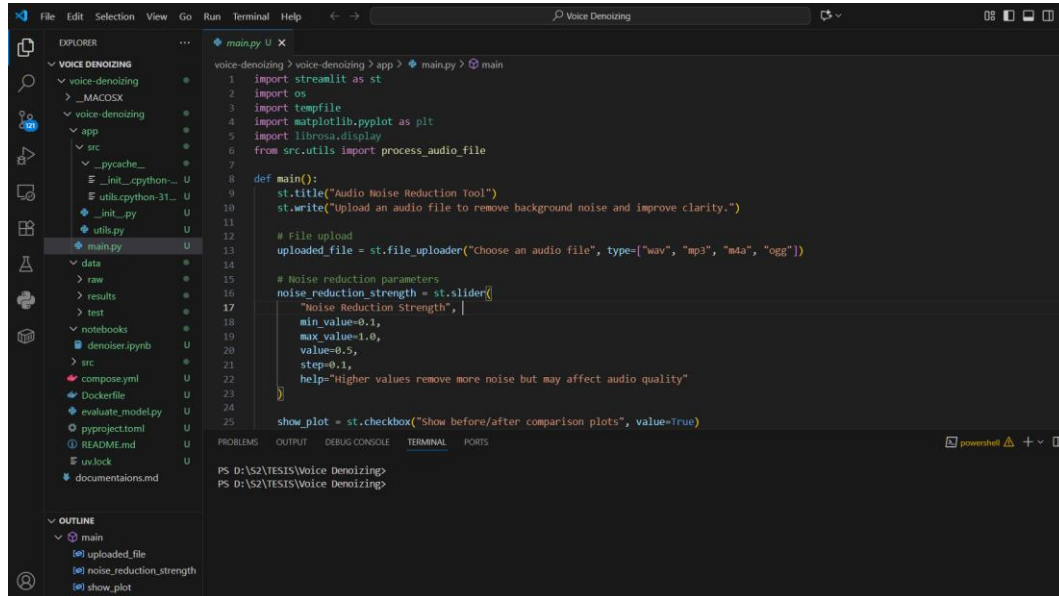
- [26] Y. R. Septio, B. Suhardi, R. D. Astuti, and I. Adiasa, “Analisis Tingkat Kebisingan, Beban Kerja dan Kelelahan Kerja Bagian Weaving di PT. Wonorejo Makmur Abadi Sebagai Dasar untuk Perbaikan Proses Produksi,” *Performa Media Ilm. Tek. Ind.*, vol. 19, no. 1, pp. 19–26, 2020, doi: 10.20961/performa.19.1.40111.
- [27] A. T. Dewi, T. Joko, and Y. H. Darundiati, “Hubungan Intensitas Kebisingan Di Lingkungan Kerja Dengan Peningkatan Tekanan Darah Pada Pekerja Pt X Semarang,” *J. Kesehat. Masy.*, vol. 9, no. 6, pp. 832–840, 2021, doi: 10.14710/jkm.v9i6.31784.
- [28] A. H. Wijaya, “Artificial Neural Network Untuk Memprediksi Beban Listrik Dengan Menggunakan Metode Backpropagation,” *J. CoreIT*, vol. 5, no. 2, pp. 61–70, 2019.
- [29] R. Ryandhi, “Penerapan Metode Artificial Neural Network (ANN) Untuk Peramalan Inflasi Di Indonesia,” *Theses*, p. 231, 2020.
- [30] R. H. Dananjaya, S. Sutrisno, and S. Fitriady, “Penerapan Artificial Neural Network (Ann) Dalam Memprediksi Kapasitas Dukung Fondasi Tiang,” *Matriks Tek. Sipil*, vol. 10, no. 4, p. 419, 2022, doi: 10.20961/mateksi.v10i4.65034.
- [31] B. Raharjo, *Deep Learning dengan Python*. 2022.
- [32] S. S. Zarin, E. Mustafa, S. K. uz Zaman, A. Namoun, and M. H. Alanazi, “An Ensemble Approach for Speaker Identification from Audio Files in Noisy Environments,” *Appl. Sci.*, vol. 14, no. 22, pp. 1–17, 2024, doi: 10.3390/app142210426.
- [33] S. Sharma, R. Malhotra, and A. Sharma, “An improved model for unsupervised voice activity detection,” *Int. J. Nanotechnol.*, vol. 20, no. 1/2/3/4, p. 235, 2023, doi: 10.1504/ijnt.2023.10056468.
- [34] V. Vysotska, V. Lytvyn, M. Nazarkevych, S. Vladov, R. Yakovliev, and A. Yurko, “Training neural network method modification for forward error propagation based on adaptive components,” *CEUR Workshop Proc.*, vol. 3711, pp. 138–168, 2024.
- [35] N. A. Purwitasari and M. Soleh, “Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing,” *J. IPTEK*, vol. 6, no. 1, pp. 14–21, 2022, doi: 10.31543/jii.v6i1.192.
- [36] K. Onggrono, T. Tulus, and E. B. Nababan, “Analisis Penggunaan Parallel Processing Multithreading Pada Resilient Backpropagation,” *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 2, no. 1, pp. 33–40, 2017, doi: 10.30743/infotekjar.v2i1.146.
- [37] A. R. Yuliani, M. F. Amri, E. Suryawati, A. Ramdan, and H. F. Pardede,

- “Speech Enhancement Using Deep Learning Methods: A Review,” *J. Elektron. dan Telekomun.*, vol. 21, no. 1, p. 19, 2021, doi: 10.14203/jet.v21.19-26.
- [38] J. Jiang and Y. Li, “Review of active noise control techniques with emphasis on sound quality enhancement,” *Appl. Acoust.*, vol. 136, no. April, pp. 139–148, 2018, doi: 10.1016/j.apacoust.2018.02.021.
- [39] A. Buteikienė, “A Review of Active Noise and Vibration Control in Road Vehicles,” *ISVR Tech. Memo. No. 981*, no. 981, pp. 61–64, 2008.
- [40] L. Peng, S. Fang, Y. Fan, M. Wang, and Z. Ma, “A Method of Noise Reduction for Radio Communication Signal Based on RaGAN,” *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010475.
- [41] A. E. Ilesanmi and T. O. Ilesanmi, “Methods for image denoising using convolutional neural network: a review,” *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2179–2198, 2021, doi: 10.1007/s40747-021-00428-4.

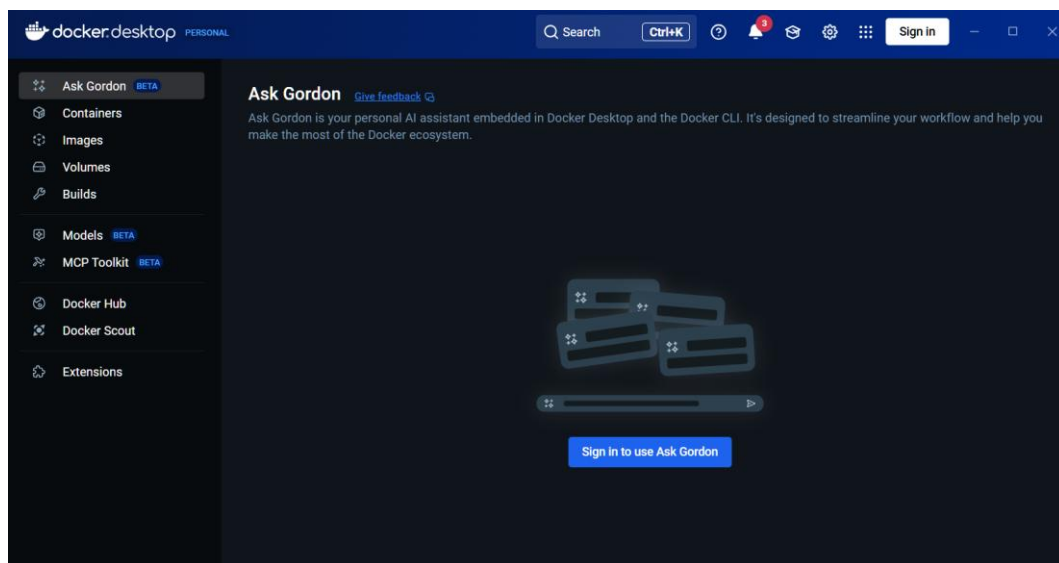
## Lampiran 1

### Proses memulai aplikasi

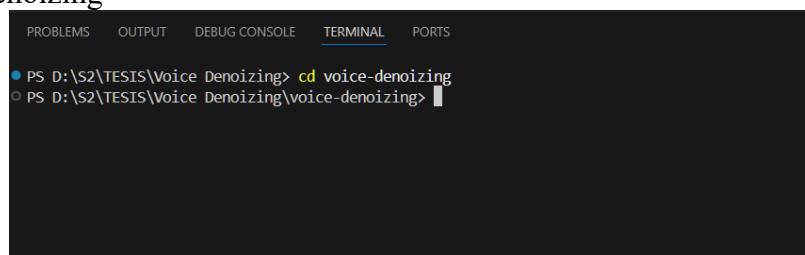
1. Buka folder file yang sudah di siapkan di Visual Studio Code



2. Buka Aplikasi Docker Desktop yang sudah di install sebelumnya



3. Kembali ke visual Studio code, buka terminal dan masuk ke folder voice-denoizing



4. Jalankan[ `docker compose up -d --build` ] di terminal, untuk membuat container di docker dan untuk sekaligus menjalankan aplikasi

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\S2\TESIS\Voice Denoizing\voice-denoizing> docker compose up -d --build

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [internal] load build context
#5 DONE 0.0s

#6 [1/9] FROM docker.io/library/python:3.12-slim-bookworm@sha256:3ad2a947749a3eb74acd9e00636ffa0def5aae0bbbd9fa4fff6253e404e2fe15
#6 resolve docker.io/library/python:3.12-slim-bookworm@sha256:3ad2a947749a3eb74acd9e00636ffa0def5aae0bbbd9fa4fff6253e404e2fe15 0.0s done
#6 DONE 0.0s

#7 [3/9] ADD https://astral.sh/uv/install.sh /uv-installer.sh
#7 DONE 1.1s

#5 [internal] load build context
#5 transferring context: 353B done
#5 DONE 0.0s

#8 [8/9] RUN mkdir -p data/raw data/results
#8 CACHED

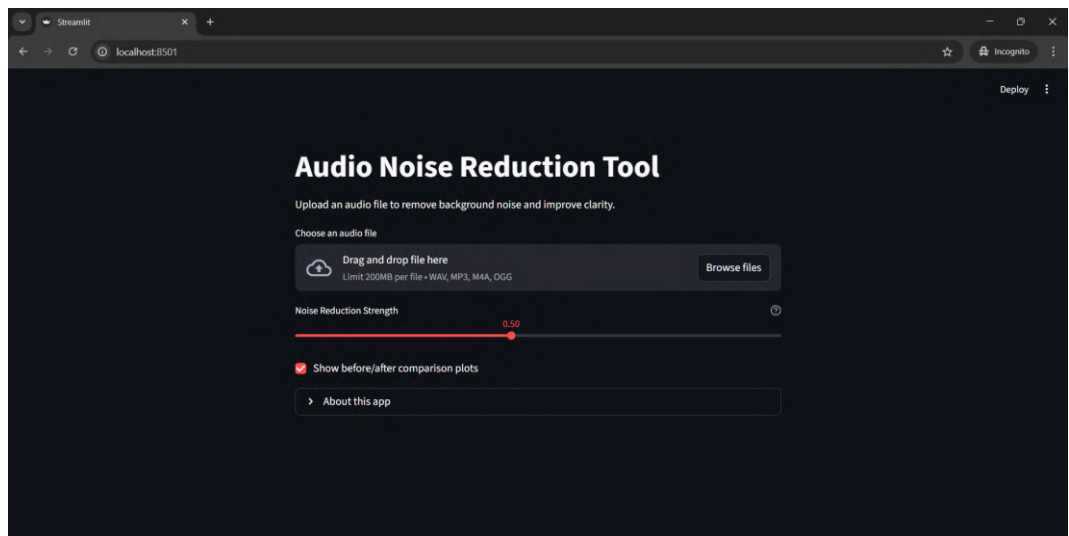
#9 [2/9] RUN apt-get update && apt-get install -y --no-install-recommends curl ca-certificates gcc build-essential ffmpeg libsndfil
e1 libportaudio2 libportaudiocpp0 portaudio19-dev && apt-get clean && rm -rf /var/lib/apt/lists/*
#9 CACHED

#10 [6/9] COPY pyproject.toml .
#10 CACHED

#11 [5/9] WORKDIR /app
#11 CACHED

#12 [3/9] ADD https://astral.sh/uv/install.sh /uv-installer.sh
#12 CACHED
```

5. Buka browser dan ketikkan[ <http://localhost:8501/> ], aplikasi siap digunakan

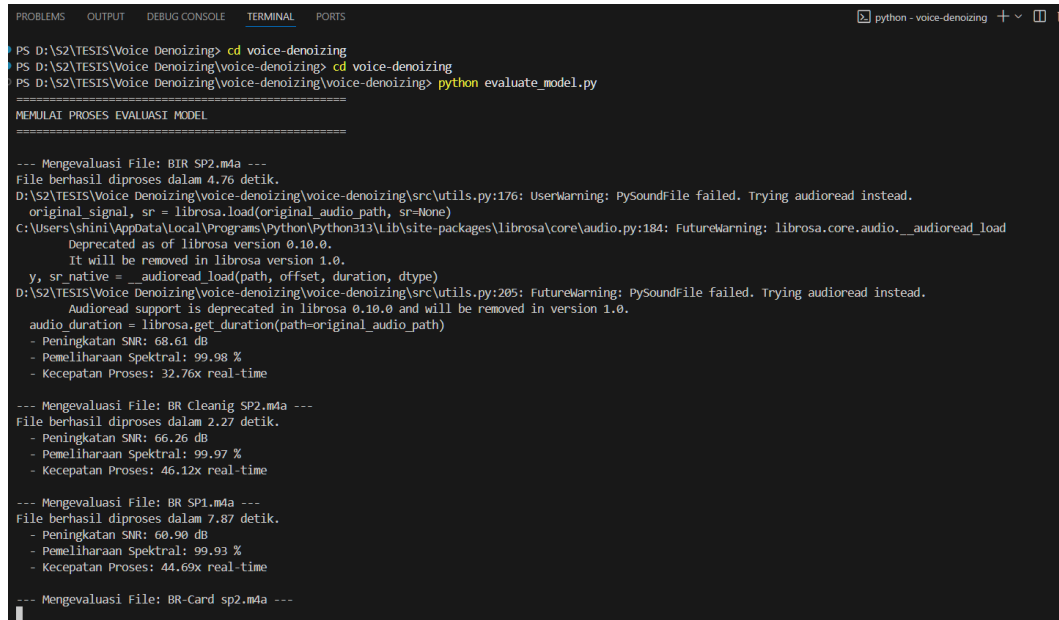




## Lampiran 2

### Proses memulai Program evaluasi

1. Buka terminal dan masuk ke folder kemudian masukkan [ python evaluate\_model.py ]



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python - voice-denoizing + - 
PS D:\S2\TESIS\Voice Denoizing> cd voice-denoizing
PS D:\S2\TESIS\Voice Denoizing\voice-denoizing> cd voice-denoizing
PS D:\S2\TESIS\Voice Denoizing\voice-denoizing\voice-denoizing> python evaluate_model.py

=====
MEMULAI PROSES EVALUASI MODEL
=====

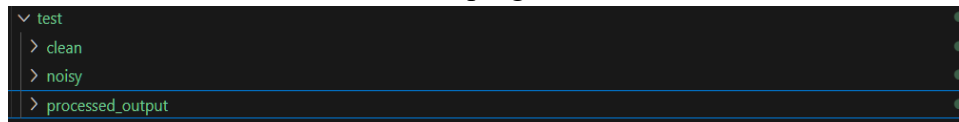
--- Mengevaluasi File: BIR SP2.m4a ---
File berhasil diproses dalam 4.76 detik.
D:\S2\TESIS\Voice Denoizing\voice-denoizing\src\utils.py:176: UserWarning: PySoundfile failed. Trying audioread instead.
  original_signal, sr = librosa.load(original_audio_path, sr=None)
C:\Users\shini\AppData\Local\Programs\Python\Python313\Lib\site-packages\librosa\core\audio.py:184: FutureWarning: librosa.core.audio._audioread_load
  deprecated as of librosa version 0.10.0.
  It will be removed in librosa version 1.0.
  y, sr_native = _audioread_load(path, offset, duration, dtype)
D:\S2\TESIS\Voice Denoizing\voice-denoizing\src\utils.py:205: FutureWarning: PySoundfile failed. Trying audioread instead.
  Audioread support is deprecated in librosa 0.10.0 and will be removed in version 1.0.
  audio_duration = librosa.get_duration(path=original_audio_path)
- Peningkatan SNR: 68.61 dB
- Pemeliharaan Spektral: 99.98 %
- Kecepatan Proses: 32.76x real-time

--- Mengevaluasi File: BR Cleanig SP2.m4a ---
File berhasil diproses dalam 2.27 detik.
- Peningkatan SNR: 66.26 dB
- Pemeliharaan Spektral: 99.97 %
- Kecepatan Proses: 46.12x real-time

--- Mengevaluasi File: BR SP1.m4a ---
File berhasil diproses dalam 7.87 detik.
- Peningkatan SNR: 60.90 dB
- Pemeliharaan Spektral: 99.93 %
- Kecepatan Proses: 44.69x real-time

--- Mengevaluasi File: BR-Card sp2.m4a ---
```

2. Berikut adalah format folder dalam program evaluasi :



- A. data/test/ (root folder) Isi Folder utama yang menampung ketiga folder di atas. Berfungsi menjadi dataset uji (test set) untuk menjalankan evaluasi model.
- B. noissy, berisi File audio yang sudah bising (noisy). Berfungsi sebagai input untuk model. Setiap file di sini akan diproses untuk "dibersihkan".
- C. Clean, berisi File audio bersih (tanpa noise) yang menjadi ground truth. Berfungsi untuk evaluasi. Program membandingkan hasil denoising (processed\_output) dengan file ini untuk menghitung metrik seperti: [Peningkatan SNR, Kualitas spektral]. Nama file di clean harus sama persis dengan nama file di noisy.
- D. processed\_output, berisi file audio hasil denoising dari model. Disimpan setelah file noisy diproses. Nama file akan berakhiran \_processed.wav