

**IDENTIFIKASI PENYAKIT PARU-PARU BERDASARKAN  
SUARA DAN RIWAYAT PASIEN MENGGUNAKAN MODEL  
*CROSS-ATTENTION VIDEO VISION TRANSFORMER***

**TUGAS AKHIR**

**Husni Na'fa Mubarok**

**121450078**



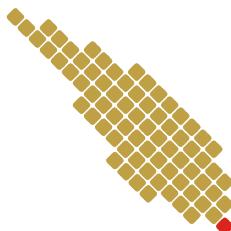
**PROGRAM STUDI SAINS DATA  
FAKULTAS SAINS  
INSTITUT TEKNOLOGI SUMATERA  
2025**

**IDENTIFIKASI PENYAKIT PARU-PARU BERDASARKAN  
SUARA DAN RIWAYAT PASIEN MENGGUNAKAN MODEL  
*CROSS-ATTENTION VIDEO VISION TRANSFORMER***

**TUGAS AKHIR**

**Diajukan sebagai syarat maju sidang tugas akhir**

**Husni Na'fa Mubarok  
121450078**



**ITERA**

**PROGRAM STUDI SAINS DATA  
FAKULTAS SAINS  
INSTITUT TEKNOLOGI SUMATERA  
2025**

## **HALAMAN PENGESAHAN**

Naskah Skripsi untuk Sidang Akhir dengan judul "**Identifikasi penyakit paru-paru berdasarkan suara dan riwayat pasien menggunakan model *Cross-Attention Video Vision Transformer***" adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan, 18 Juli 2025

Penulis,



**Husni Na'fa Mubarok**

**NIM. 121450078**

Diperiksa dan disetujui oleh,

Pembimbing I

Pembimbing II

**Christyan Tamaro Nadeak, M.Si**

**NRK. 1993120420211415**

**Luluk Muthoharoh, M.Si**

**NIP. 199504112022032014**

Disahkan oleh,

Koordinator Program Studi Sains Data

Fakultas Sains

Institut Teknologi Sumatera

**Tirta Setiawan, S.Pd., M.Si**

**NIP. 199008222022031003**

Sidang Tugas Akhir:

Penguji I : Mika Alvionita S, M.Si

Penguji II : Tirta Setiawan, S.Pd., M.Si

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah karya saya sendiri dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan benar.**

**Nama : Husni Na'fa Mubarok**

**NIM : 121450078**

**Tanda tangan :**

**Tanggal : 18 Juli 2025**

## **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Husni Na'fa Mubarok  
NIM : 121450078  
Program Studi : Sains Data  
Fakultas : Sains  
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Noneksklusif (*Non-Exclusive Royalty Free Right*) kepada Institut Teknologi Sumatera atas karya ilmiah saya yang berjudul:

**Identifikasi penyakit paru-paru berdasarkan suara dan riwayat pasien menggunakan model *Cross-Attention Video Vision Transformer***

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Lampung Selatan  
Pada tanggal : 18 Juli 2025  
Yang menyatakan

Husni Na'fa Mubarok

## **ABSTRAK**

### **Identifikasi penyakit paru-paru berdasarkan suara dan riwayat pasien menggunakan model *Cross-Attention Video Vision Transformer***

Husni Na'fa Mubarok (121450078)

Pembimbing I: Christyan Tamara Nadeak, M.Si

Pembimbing II: Luluk Muthoharoh, M.Si

Identifikasi penyakit paru-paru dapat dilakukan melalui analisis suara batuk dan riwayat pasien. Penelitian ini mengembangkan model berbasis *Cross-Attention Video Vision Transformer* yang memanfaatkan representasi fitur audio dan data riwayat pasien untuk klasifikasi penyakit paru-paru. *Mel Frequency Cepstral Coefficient* (MFCC) digunakan untuk ekstraksi fitur audio, sementara fitur riwayat pasien memperkaya informasi tanpa menambah beban komputasi. Dataset yang digunakan terdiri dari suara batuk, data riwayat pasien, dan label penyakit dengan tiga kelas: asma, COPD, dan sehat. Model *Video Vision Transformer* diterapkan dengan *embedding* spasial dan temporal, dioptimalkan melalui mekanisme *Cross-Attention*. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score, serta analisis kurva ROC-AUC untuk memeriksa performa klasifikasi antar kelas. Hasil penelitian menunjukkan bahwa model mampu mencapai akurasi terbaik sebesar 82.57% dengan F1-score 81.95% menggunakan konfigurasi Big Model 1. Model ini diharapkan dapat dikembangkan dan diimplementasikan dalam klasifikasi penyakit paru-paru berdasarkan data *multimodal*, meskipun memerlukan pengembangan lebih lanjut untuk meningkatkan kemampuan dalam membedakan kondisi penyakit yang serupa.

**Kata kunci:** *Cross-Attention*, Identifikasi penyakit paru-paru, klasifikasi suara, MFCC, *Video Vision Transformer*.

## ***ABSTRACT***

### ***Identify lung disease based on sound and patient history using the Cross-Attention Video Vision Transformer model***

Husni Na'fa Mubarok (121450078)

*Advisor I:* Christyan Tamaro Nadeak, M.Si

*Advisor II:* Luluk Muthoharoh, M.Si

*Lung disease identification can be conducted through the analysis of cough sounds and patient history. This study developed a model based on the Cross-Attention Video Vision Transformer, leveraging audio feature representations and patient history data for lung disease classification. Mel Frequency Cepstral Coefficient (MFCC) was used for audio feature extraction, while patient history features enriched the information without adding computational overhead. The dataset comprised cough audio, patient history data, and disease labels with three classes: asthma, COPD, and healthy. The Video Vision Transformer model was implemented with spatial and temporal embeddings, optimized through the Cross-Attention mechanism. Evaluation metrics included accuracy, precision, recall, F1-score, and ROC-AUC curve analysis to assess inter-class classification performance. The results showed that the model achieved the best accuracy of 82.57% and an F1-score of 81.95% using the Big Model 1 configuration. This model is expected to be developed and implemented in the classification of lung diseases based on multimodal data, though further improvements are needed to enhance its ability to distinguish between similar disease conditions.*

***Keywords :*** *Cross-Attention, Lung disease identification, MFCC, sound classification, Video Vision Transformer.*

## **MOTTO**

*Good Data, Good Decision. Bad Data, Bad Decision. No Data, No Decision.*

## **HALAMAN PERSEMBAHAN**

*Untuk Umi dan Abi  
di kampung*

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Allah SWT atas berkah dan rahmat-Nya sehingga skripsi ini dapat terselesaikan dengan baik. Skripsi ini dibuat untuk menyelesaikan pendidikan jenjang sarjana pada Institut Teknologi Sumatera. Penyusunan skripsi ini banyak mendapat bantuan dan dukungan dari berbagai pihak sehingga dalam kesempatan ini, dengan penuh kerendahan hati, penulis mengucapkan terima kasih kepada:

1. Prof. Dr. I Nyoman Pugeg Aryantha selaku Rektor Institut Teknologi Sumatera,
2. Dr. Ikah Ning Prasetyowati Permanasari, S.Si., M.Si. selaku Dekan Fakultas Sains Institut Teknologi Sumatera,
3. Tirta Setiawan, S.Pd., M.Si selaku Koordinator Program Studi Sains Data
4. Christyan Tamara Nadeak, M.Si selaku dosen pembimbing pertama yang telah membimbing dan memberikan arahan dalam penyusunan skripsi ini
5. Luluk Muthoharoh, M.Si selaku dosen pembimbing kedua yang selalu membantu dan memberikan masukan dalam penyusunan skripsi ini
6. Himanshu Kaushik dan Kaggle selaku penyedia data dalam penelitian ini.
7. Miftahul Huda dan Abdurrahman Al-Atsary selaku rekan TA dan teman diskusi
8. Pimpinan, Pengurus dan seluruh Muzakki YBM BRILiaN yang merupakan Lembaga Amil Zakat Nasional (LAZNAS) yang memberikan kesempatan beasiswa Bright Scholarship kepada saya sehingga saya dapat menyelesaikan studi S1.

Penulis menyadari bahwa penyusunan Skripsi ini jauh dari sempurna. Akhir kata penulis mohon maaf yang sebesar-besarnya apabila ada kekeliruan di dalam penulisan skripsi ini.

Lampung Selatan, 18 Juli 2025

**Husni Na'fa Mubarok**

## DAFTAR ISI

<b>HALAMAN JUDUL . . . . .</b>	<b>i</b>
<b>HALAMAN PENGESAHAN . . . . .</b>	<b>ii</b>
<b>HALAMAN PERNYATAAN ORISINALITAS . . . . .</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI</b>	<b>iv</b>
<b>ABSTRAK . . . . .</b>	<b>v</b>
<b>ABSTRACT . . . . .</b>	<b>vi</b>
<b>MOTTO . . . . .</b>	<b>vii</b>
<b>HALAMAN PERSEMBAHAN . . . . .</b>	<b>viii</b>
<b>KATA PENGANTAR . . . . .</b>	<b>ix</b>
<b>DAFTAR ISI . . . . .</b>	<b>xi</b>
<b>DAFTAR GAMBAR . . . . .</b>	<b>xiv</b>
<b>DAFTAR TABEL . . . . .</b>	<b>xv</b>
<b>I PENDAHULUAN . . . . .</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Tujuan Penelitian . . . . .	3
1.4 Batasan Masalah . . . . .	3
<b>II TINJAUAN PUSTAKA . . . . .</b>	<b>4</b>
2.1 Penelitian Terdahulu . . . . .	4
2.2 Penyakit Paru-paru . . . . .	5
2.2.1 <i>Chronic obstructive pulmonary disease (COPD)</i>	5
2.2.2 Asma . . . . .	5
2.3 <i>Mel Frequency Cepstral Coefficient (MFCC)</i> . . . . .	5
2.4 Transformer . . . . .	6

2.4.1	<i>Positional Encoding</i>	7
2.4.1.1	<i>Sinusoidal Positional Encoding</i>	7
2.4.1.2	Trainable Positional Encoding	7
2.4.2	<i>Attention Mechanism</i>	8
2.4.3	<i>Position-wise Feed-Forward Networks</i>	9
2.4.4	<i>Layer Normalization</i>	9
2.5	<i>Video Vision Transformer</i>	10
2.6	<i>Sound to Noise Ratio (SNR)</i>	11
<b>III</b>	<b>METODE PENELITIAN</b>	<b>13</b>
3.1	Deskripsi Data	13
3.2	Rancangan Penelitian	16
3.2.1	Pemrosesan Suara	17
3.2.2	Pemrosesan Data Tabel Riwayat Pasien	20
3.2.3	Pipeline dan Pembagian Data	21
3.2.4	Arsitektur	23
3.2.5	Variasi Model	24
3.2.6	<i>Loss Function</i>	25
3.2.7	Pelatihan Model	25
3.3	Evaluasi Model	26
3.3.1	<i>Confusion Matrix</i>	26
3.3.2	Kurva ROC-AUC	27
<b>IV</b>	<b>HASIL DAN PEMBAHASAN</b>	<b>29</b>
4.1	<i>Data Cleaning</i> dan <i>Data Preprocessing</i>	29
4.1.1	Pemprosesan Tabel riwayat pasien	29
4.1.2	<i>Audio Processing</i>	30
4.2	Visualisasi Input Model	32
4.2.1	<i>Input Spatial Embedding</i>	32
4.2.2	<i>Input Temporal Embedding</i>	33
4.3	Evaluasi Model	34
4.3.1	Variasi Model	34

4.3.2	Akurasi dan Loss Pelatihan . . . . .	35
4.3.3	<i>Confusion Matrix</i> . . . . .	40
4.3.4	Kurva AUC-ROC . . . . .	42
4.4	Studi Ablasi . . . . .	47
4.4.1	Positional Encoding . . . . .	47
4.4.2	<i>Cross-Attention</i> . . . . .	48
4.4.3	<i>Shared-weights</i> Feed-Forward Network (FFN) .	49
4.4.4	Layer Encoder ( $N$ ) . . . . .	49
4.4.5	Dimensi Embedding ( $d_{model}$ ) . . . . .	50
4.5	Prediksi Data baru . . . . .	51
<b>V</b>	<b>KESIMPULAN DAN SARAN</b> . . . . .	<b>53</b>
5.1	Kesimpulan . . . . .	53
5.2	Saran . . . . .	53
<b>DAFTAR PUSTAKA</b>	. . . . .	<b>55</b>
<b>LAMPIRAN</b>	. . . . .	<b>62</b>
<b>A</b>	<i>Mel frequency Capstral Coefficients</i> (MFCC) . . . . .	<b>63</b>
A.1	Pre-Emphasis . . . . .	63
A.2	<i>Framing</i> dan <i>Windowing</i> . . . . .	63
A.3	<i>Discrete Fourier Transform</i> (DFT) . . . . .	64
A.4	<i>Mel-Frequency Filter Bank</i> . . . . .	65
A.5	<i>Discrete Cosine Transform</i> (DCT) . . . . .	66
<b>B</b>	<b>Perhitungan Encoder Cross-Attention</b> . . . . .	<b>67</b>
B.1	<i>Self-Attention</i> dan <i>Multi-Head Attention</i> . . . . .	68
B.2	<i>Feed-Forward Network</i> . . . . .	70
B.3	Output Encoder $Y_s$ dan $Y_t$ . . . . .	71
<b>C</b>	<b>Optimizer Adam</b> . . . . .	<b>73</b>
C.1	Optimizer Adam pada Transformers . . . . .	73
C.2	Persamaan Oprimizer Adam . . . . .	73
C.3	Contoh Perhitungan . . . . .	74

<b>D Rangkuman Model . . . . .</b>	<b>76</b>
D.1 Model Base 1 . . . . .	76
D.2 Model Base 2 . . . . .	77
D.3 Model Big 1 . . . . .	79
D.4 Model Big 2 . . . . .	82
<b>E Data Test . . . . .</b>	<b>85</b>

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur Transformer . . . . .	6
Gambar 3.1	Visualisasi gelombang suara tiap kelas . . . . .	14
Gambar 3.2	<i>Flowchart</i> rancangan penelitian . . . . .	16
Gambar 3.3	Desain rancangan arsitektur model. . . . .	23
Gambar 3.4	Kurva ROC-AUC . . . . .	28
Gambar 4.1	Visualisasi distribusi SNR . . . . .	31
Gambar 4.2	Segmentasi Suara batuk . . . . .	32
Gambar 4.3	Visualisasi MFCC . . . . .	32
Gambar 4.4	Visualisasi <i>Input Spatial Embedding</i> . . . . .	33
Gambar 4.5	Visualisasi <i>Input Temporal Embedding</i> . . . .	34
Gambar 4.6	Akurasi model per epoch . . . . .	37
Gambar 4.7	Loss model per epoch . . . . .	39
Gambar 4.8	Confusion Matrix . . . . .	40
Gambar 4.9	Kurva AUC-ROC antar kelas Base Model 1	42
Gambar 4.10	Kurva AUC-ROC antar kelas Base Model 2	43
Gambar 4.11	Kurva AUC-ROC antar kelas Big Model 1 .	45
Gambar 4.12	Kurva AUC-ROC antar kelas Big Model 2 .	46
Gambar B.1	Encoder Cross Attention . . . . .	67
Gambar B.2	<i>Self-Attention</i> atau <i>Scaled Dot-Product Attention</i> . . . . .	68
Gambar B.3	<i>Multi-Head Attention</i> . . . . .	68
Gambar E.1	Sampel audio data test . . . . .	86

## DAFTAR TABEL

Tabel 2.1	Penelitian Terdahulu . . . . .	4
Tabel 2.2	Rentang Signal-to-Noise Ratio (SNR) . . . . .	12
Tabel 3.1	Tabel riwayat pasien . . . . .	15
Tabel 3.2	<i>Confusion Matrix</i> untuk 3 Kelas (a), Rumus metrik evaluasi (b) . . . . .	27
Tabel 4.1	Jumlah Baris Kosong per Kolom . . . . .	29
Tabel 4.2	Jumlah Baris Kosong per Kolom setelah menghapus . . . . .	30
Tabel 4.3	Variasi model . . . . .	35
Tabel 4.4	Akurasi, Presisi, Recall, dan F1-Score dari 4 Model . . . . .	41
Tabel 4.5	Perbandingan Sinusoidal Positional Encoding dengan Trainable Positional Encoding. . . . .	47
Tabel 4.6	Perbandingan model dengan Cross-Attention dan tanpa Cross-Attention .	48
Tabel 4.7	Perbandingan model dengan Shared-Weights Feed-Forward Network .	49
Tabel 4.8	Perbandingan Ukuran Layer Encoder ( $N$ ) .	50
Tabel 4.9	Perbandingan Ukuran Embedding ( $d_{model}$ ) .	51
Tabel 4.10	Prediksi data baru . . . . .	52
Tabel B.1	Tabel Variabel untuk Encoder Transformer Model Big 1 . . . . .	69
Tabel D.1	Rangkuman Encoder Block Model Base 1 .	76
Tabel D.2	Rangkuman Classifier Model Base 1 . . . . .	77
Tabel D.3	Rangkuman Model Base 1 . . . . .	77

Tabel D.4	Rangkuman Encoder Block Model Base 2 . . . . .	78
Tabel D.5	Rangkuman Model Classifier . . . . .	79
Tabel D.6	Rangkuman Model Base 2 . . . . .	79
Tabel D.7	Rangkuman Encoder Block Model Big 1 . . . . .	79
Tabel D.8	Rangkuman Classifier Model Big 1 . . . . .	81
Tabel D.9	Rangkuman Model Big 1 . . . . .	81
Tabel D.10	Rangkuman Encoder Block Model Big 2 . . . . .	82
Tabel D.11	Rangkuman Classifier Model Big 2 . . . . .	84
Tabel D.12	Rangkuman Model Big 2 . . . . .	84
Tabel E.1	Tabel riwayat pasien data test . . . . .	87

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Paru-paru merupakan organ vital manusia yang berfungsi untuk pertukaran oksigen dan karbon dioksida pada darah[1]. Gangguan pada paru-paru dapat berakibat buruk pada sistem pernapasan sehingga dapat menimbulkan penyakit. Menurut *Lung Foundation Australia*, penyakit paru-paru dapat disebabkan oleh berbagai faktor seperti umur, perokok aktif atau pasif, lingkungan yang terpapar debu, gas, uap dan zat kimia[2]. Salah satu penyakit paru yang berisiko yaitu *Chronic obstructive pulmonary disease* (COPD) dan asma.

*Chronic obstructive pulmonary disease* (COPD) adalah gangguan pernapasan yang umum dan progresif yang ditandai dengan keterbatasan aliran udara yang terus-menerus dan sering dikaitkan dengan penyakit paru-paru lainnya seperti bronkitis kronis dan asma [3]. Menurut data *World Health Organization* (WHO) *Chronic obstructive pulmonary disease* (COPD) merupakan penyebab kematian keempat di seluruh dunia, menyebabkan 3.5 juta kematian pada tahun 2021 [4], dan diperkirakan COPD akan menjadi penyebab kematian ketiga di dunia pada tahun 2030 [5]. Berdasarkan survei data BPJS tahun 2024, hampir 19 juta jumlah pasien COPD yang berobat ke rumah sakit dengan penyakit tersebut [6]. Oleh sebab itu, diperlukan metode yang dapat mengidentifikasi jenis penyakit paru-paru berdasarkan gejala yang diketahui, salah satunya yaitu suara.

Suara batuk atau paru-paru mengandung banyak informasi tentang

kondisi paru-paru dan dapat digunakan untuk menilai serta mendiagnosis penyakit pernapasan [7]. Penggunaan suara untuk identifikasi penyakit paru meningkatkan minat terhadap perawatan medis tanpa kontak untuk pemeriksaan paru-paru secara otomatis. Model Deep Learning banyak digunakan peneliti dalam menganalisis suara seperti LungRN+NL [8] yang menggunakan augmentasi data campuran dan arsitektur ResNet [9] untuk mengatasi ketidakseimbangan kelas data. RespireNet [10] menggunakan model *pre-trainer* pada ImageNet dengan strategi *fine-tuning device-specific*.

Model *General-Purpose* representasi audio lainnya seperti CLAP [11] menggunakan dua encoder untuk memproses input yaitu *Audio Encoder* yang memproses input suara dan *Text Encoder* yang memproses input berupa teks. Namun, penggunaan dua encoder mengakibatkan beban komputasi yang tinggi sehingga memerlukan sumber daya komputasi yang besar.

Oleh sebab itu, pada penelitian ini mengadopsi konsep dari *Video Vision Transformer* [12] yang menggunakan embedding spasial  $Z_s$  dan temporal  $Z_t$  dengan input berupa representasi MFCC dan fitur riwayat pasien pada satu encoder yang sama sehingga mengurangi beban komputasi saat proses *training*. *Cross-Attention* [13] digunakan agar suatu *sequence* dapat memperhatikan informasi dari *sequence* lainnya.

Secara keseluruhan, kontribusi penelitian ini dapat dirangkum yaitu MFCC digunakan untuk merepresentasikan suara yang membantu model memahami dan memproses suara manusia secara lebih efektif. Penelitian ini mengadopsi model *Video Vision Transformer* untuk memahami fitur temporal dan spasial dari data audio. Data riwayat pasien digunakan untuk memperkaya fitur tanpa menambah encoder fitur sehingga beban komputasi menjadi lebih ringan.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, berikut merupakan rumusan masalah pada penelitian tugas akhir ini:

1. Bagaimana penerapan konsep *Video Vision Transformer* pada data suara penyakit paru dengan penambahan fitur riwayat pasien menggunakan metode *Cross-Attention* dapat mengidentifikasi jenis penyakit paru-paru?
2. Bagaimana performa penerapan konsep *Video Vision Transformer* dalam mengklasifikasikan jenis penyakit paru-paru berdasarkan data suara dan riwayat pasien?

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini berdasarkan rumusan masalah yang juga menjadi dasar dilakukannya penelitian ini adalah sebagai berikut:

1. Membuat model *Deep Learning* untuk mengklasifikasikan jenis penyakit paru-paru menggunakan konsep model *Video Vision Transformer* dengan metode *Cross-Attention*.
2. Mengevaluasi performa model *Video Vision Transformer* dalam mengidentifikasi fitur suara dan riwayat pasien sehingga menghasilkan klasifikasi yang sesuai

## **1.4 Batasan Masalah**

1. Penelitian ini hanya menggunakan data suara batuk dan riwayat pasien tanpa menyertakan identitas atau informasi lainnya.
2. Jenis penyakit paru-paru yang diidentifikasi terbatas pada dataset yang digunakan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

**Tabel 2.1** Penelitian Terdahulu

<b>Penulis</b>	<b>Metode</b>	<b>Hasil Penelitian</b>
Hao Xue et al.[14]	Transformer-CP	Model berbasis Transformer dengan Constraintive Pre-training yang menggunakan random masking dengan fitur ekstraksi MFCC mendapatkan akurasi 87,74% pada data suara batuk pasien covid-19
Li Xiao et al.[15]	LungAdapter	Metode ini menggabungkan blok <i>trainable</i> ke dalam model AST yang telah dilatih sebelumnya, yang memungkinkan ekstraksi informasi penting tentang klasifikasi suara paruparupu dari model. Model mencapai kinerja yang baik dengan score 62,40%.
Victor Basu et al.[16]	GRU, MFCC	Lapisan GRU (gated recurrent unit) digunakan untuk memecahkan masalah gradien yang hilang dalam RNN standar. Akurasi: $95,67 \pm 0,77\%$ .

Penelitian terdahulu umumnya telah banyak menggunakan model berbasis transformer [14], [15] dan RNN [16] untuk mengklasifikasikan suara batuk. Penelitian ini akan mengadaptasi model *Video Vision Transformer* (ViViT) [12] yang memiliki embedding spasial-temporal dengan input berupa representasi MFCC dan riwayat pasien. *Cross-Attention* yang menggunakan *attention* pada *query* dan *key* yang berbeda memungkinkan model untuk menggabungkan informasi suara batuk dan riwayat kesehatan

pasien. Hal ini diharapkan akan meningkatkan akurasi klasifikasi suara batuk dengan mempertimbangkan konteks medis pasien.

## 2.2 Penyakit Paru-paru

### 2.2.1 *Chronic obstructive pulmonary disease (COPD)*

*Chronic Obstructive Pulmonary Disease* (COPD) memiliki berbagai gejala yang sering kali dapat disalahartikan sebagai kondisi pernapasan lainnya, sehingga mempersulit diagnosis dan penanganannya. Gejala utamanya meliputi batuk kronis [17], dispnea [18], dan produksi sputum [19], yang tumpang tindih dengan kondisi seperti asma dan bronkitis.

### 2.2.2 Asma

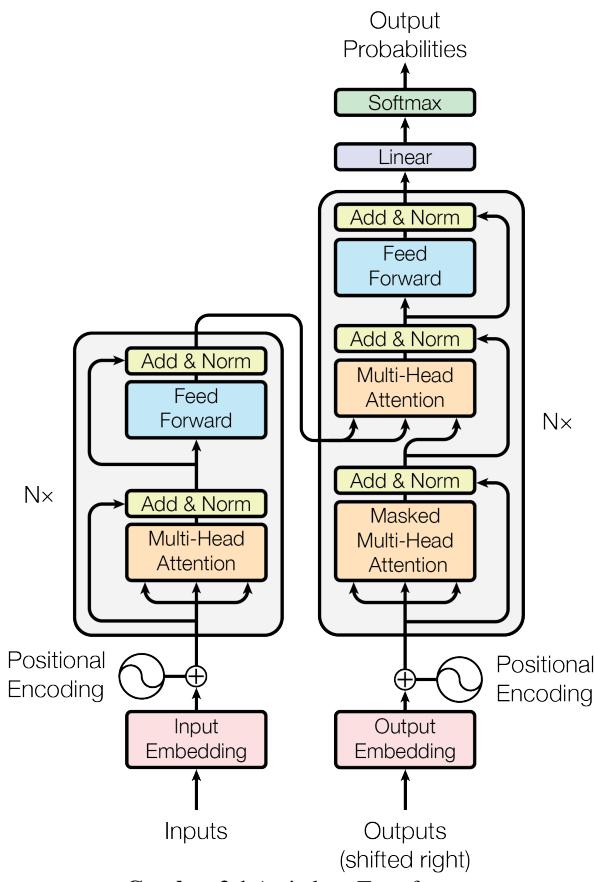
Asma adalah suatu kondisi pernapasan kronis yang ditandai dengan gejala yang bervariasi, terutama mengi, batuk, dada terasa sesak, dan dispnea yang intensitas dan frekuensinya dapat berfluktuasi [20]. Gejala-gejala ini sering kali timbul dari pemicu seperti alergen, infeksi, atau olahraga, dan mungkin terjadi secara intermiten, dengan beberapa pasien mengalami periode bebas gejala [20], [21].

## 2.3 *Mel Frequency Cepstral Coefficient (MFCC)*

MFCC memanfaatkan persepsi frekuensi suara telinga manusia, menggunakan skala Mel non-linier untuk mengubah sinyal audio menjadi representasi yang lebih relevan secara persepsi. Transformasi ini dicapai melalui serangkaian langkah, termasuk *pre-emphasis*, *windowing* dan *framing*, transformasi Fourier, pemrosesan *Mel filter bank*, dan analisis *cepstral*[22], [23]. Proses MFCC lebih detail dijelaskan pada lampiran A.

## 2.4 Transformer

Arsitektur Transformer pertama kali diperkenalkan oleh Vaswani et al. (2017) pada paper yang berjudul "*Attention Is All You Need*"[24] yang terdiri dari dua bagian utama yaitu encoder dan decoder. Karena fleksibilitasnya dalam menangani *sequence* dan kemampuannya dalam memahami konteks, peneliti banyak mengadopsi arsitektur ini pada berbagai media seperti suara [15], [25], gambar [26], [27] dan video [12], [28]. Desain arsitektur transformer dapat dilihat pada gambar 2.1.



Gambar 2.1 Arsitektur Transformer

## 2.4.1 Positional Encoding

Arsitektur Transformer bersifat paralel sehingga diperlukan *Positional Encoding* untuk memberikan informasi urutan pada *sequence*. *Positional Encoding* memiliki ukuran dimensi yang sama dengan *embedding* input sehingga dapat dijumlahkan. Positional encoding dijelaskan pada persamaan 2.4.1.1 dan 2.4.1.2.

### 2.4.1.1 Sinusoidal Positional Encoding

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \quad (2.1)$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \quad (2.2)$$

Keterangan:

$PE_{(pos,2i)}$  : posisi pada indeks genap

$PE_{(pos,2i+1)}$  : posisi pada indeks ganjil

$pos$  : posisi sampel/data dalam *sequence*

$i$  : dimensi data

$d_{\text{model}}$  : ukuran dimensi embedding

$pos$  merupakan posisi dan  $i$  merupakan dimensi.  $2i$  dan  $2i + 1$  adalah indeks genap dan ganjil dalam embedding. *Positional encoding sinusoidal* dipilih karena memungkinkan model untuk mengekstrapolasi *sequence* yang lebih panjang daripada yang ditemui selama pelatihan.

### 2.4.1.2 Trainable Positional Encoding

*Trainable Embedding* banyak digunakan pada *Large Language Model* (LLM) seperti BERT [29] dan GPT [30]. *Trainable Embedding* dapat memahami konteks data lebih baik karena nilainya di-update selama proses pelatihan. Pemahaman

kontekstual ini membuat *Trainable embedding* jauh lebih optimal dalam tugas-tugas Deep Learning yang kompleks karena dapat memahami pola yang mungkin terlewatkan oleh *Embedding Statis*.

#### 2.4.2 Attention Mechanism

Fungsi *Attention* dapat dideskripsikan sebagai pemetaan *Query* dan sekumpulan pasangan *Key-Value* ke suatu *Output*, di mana *Query*, *Key*, *Value*, dan *Output* semuanya merupakan vektor. *Self-Attention* menghubungkan semua posisi dengan jumlah operasi *sequence* yang konstan sehingga mempercepat operasi dibandingkan layer recurrent pada RNN. Fungsi *Attention* dijelaskan pada persamaan 2.3

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{\text{Head\_DIM}}} \right) V \quad (2.3)$$

Keterangan:

$\text{Attention}(Q, K, V)$  : mekanisme *self-attention*

$Q$  : *Query*

$K$  : *Key*

$V$  : *Value*

$\text{Head\_DIM}$  : dimensi head

*Multi-Head Attention* memungkinkan model untuk secara bersamaan memberikan *Attention* informasi dari subruang representasi yang berbeda pada posisi yang berbeda 2.4.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{dengan } \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (2.4)$$

Keterangan:

$\text{MultiHead}(Q, K, V)$  : mekanisme multi-head attention  
 $\text{head}_i$  : mekanisme attention  
 $W_i^Q$  : bobot untuk *Query*  
 $W_i^K$  : bobot untuk *Key*  
 $W_i^V$  : bobot untuk *Value*  
 $W^O$  : bobot untuk multi-head attention

### 2.4.3 Position-wise Feed-Forward Networks

Masing-masing lapisan dalam encoder dan decoder terdapat *Feed-Forward Networks* yang saling terhubung, yang diterapkan ke setiap posisi secara terpisah dan identik. Fungsi *Feed-Forward Networks* dapat dilihat pada persamaan 2.5.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.5)$$

Keterangan:

$\text{FFN}(x)$  : mekanisme Feed-Forward Networks  
 $x$  : nilai input neuron  
 $W_1$  : bobot pertama  
 $b_1$  : bias pertama  
 $W_2$  : bobot kedua  
 $b_2$  : bias kedua

Meskipun transformasi linier sama di berbagai posisi, FNN menggunakan parameter yang berbeda pada setiap lapisan.

### 2.4.4 Layer Normalization

*Layer Normalization* digunakan untuk menormalkan output dari setiap jaringan, sehingga distribusi nilainya stabil selama *training*.  
 2.6. Parameter  $\gamma$  dan  $\beta$  merupakan parameter pelatihan dan  $\epsilon$

merupakan nilai konstan  $10^{-6}$ .

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta \quad (2.6)$$

Keterangan:

$\text{LayerNorm}(x)$	: Layer Normalisasi
$x$	: nilai input layer
$\mu$	: rataan
$\sigma^2$	: standar deviasi
$\epsilon$	: nilai galat agar pembagi tidak bermilai 0
$\gamma, \beta$	: parameter pelatihan

## 2.5 Video Vision Transformer

Pada ViViT, video input  $V$  dengan dimensi  $D$  (depth),  $H$  (tinggi),  $W$  (lebar), dan  $C$  (channel) dibagi-bagi menjadi potongan-potongan kecil yang disebut *patch*[12]. Setiap *patch*  $P$  memiliki dimensi  $P_d \times P_h \times P_w \times C$ . Proses pembagian video menjadi *patch* ini dapat divisualisasikan sebagai persamaan 2.7

$$V \in \mathbb{R}^{D \times H \times W \times C} \rightarrow \text{Reshape} \rightarrow P \in \mathbb{R}^{N \times (P_d \times P_h \times P_w \times C)} \quad (2.7)$$

Keterangan:

$V$	: input embedding video
$P$	: input patch embedding
Reshape	: mengubah input video ke patch
$D$	: ukuran <i>depth</i> kedalaman/frame
$H$	: ukuran <i>height</i> atau tinggi
$W$	: ukuran <i>width</i> atau lebar
$C$	: jumlah <i>channel</i> atau kanal

di mana  $N = (D/P_d) \times (H/P_h) \times (W/P_w)$  adalah jumlah total *patch*

yang dihasilkan. Selanjutnya, setiap *patch*  $P$  diproyeksikan ke dalam ruang *embedding* dengan dimensi  $d$  melalui sebuah lapisan linear. Hasil proyeksi ini membentuk sebuah *sequence* token  $Z$  dengan dimensi  $N \times d$  sesuai persamaan 2.8

$$Z = \text{Linear}(P) \in \mathbb{R}^{N \times d} \quad (2.8)$$

Keterangan:

$Z$  : hasil proyeksi *patch embedding*

$P$  : *patch embedding*

*Sequence* token  $Z$  ditambahkan dengan *positional encoding* yang kemudian menjadi input untuk *encoder transformer*. Terdapat dua metode utama untuk menghasilkan *patch embedding* yaitu, *Uniform Frame Sampling* yang hanya memiliki informasi spasial dan *Tubelet Embedding* yang menghasilkan *patch* yang mencakup informasi spasial dan temporal.

## 2.6 Sound to Noise Ratio (SNR)

*Sound to Noise Ratio* (SNR) adalah rasio antara kekuatan sinyal yang diinginkan (suara batuk) dan kekuatan noise (suara latar belakang). SNR yang tinggi menunjukkan bahwa suara batuk lebih jelas dibandingkan dengan noise, sedangkan SNR yang rendah menunjukkan bahwa noise mengganggu suara batuk. SNR dapat dihitung dengan rumus 2.9.

$$SNR = 10 \cdot \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (2.9)$$

Keterangan:

$P_{\text{signal}}$  : daya sinyal yang diinginkan (suara batuk)

$P_{\text{noise}}$  : daya noise (suara latar belakang)

SNR yang tinggi menunjukkan bahwa suara batuk lebih jelas dibandingkan dengan noise, sedangkan SNR yang rendah menunjukkan bahwa noise mengganggu suara batuk. SNR dapat digunakan untuk mengevaluasi kualitas rekaman suara batuk dan membantu dalam proses pelatihan model untuk mengenali suara batuk dengan lebih baik. Nilai kualitas SNR diberbagai rentang dapat dilihat pada tabel 2.2.

**Tabel 2.2** Rentang Signal-to-Noise Ratio (SNR)

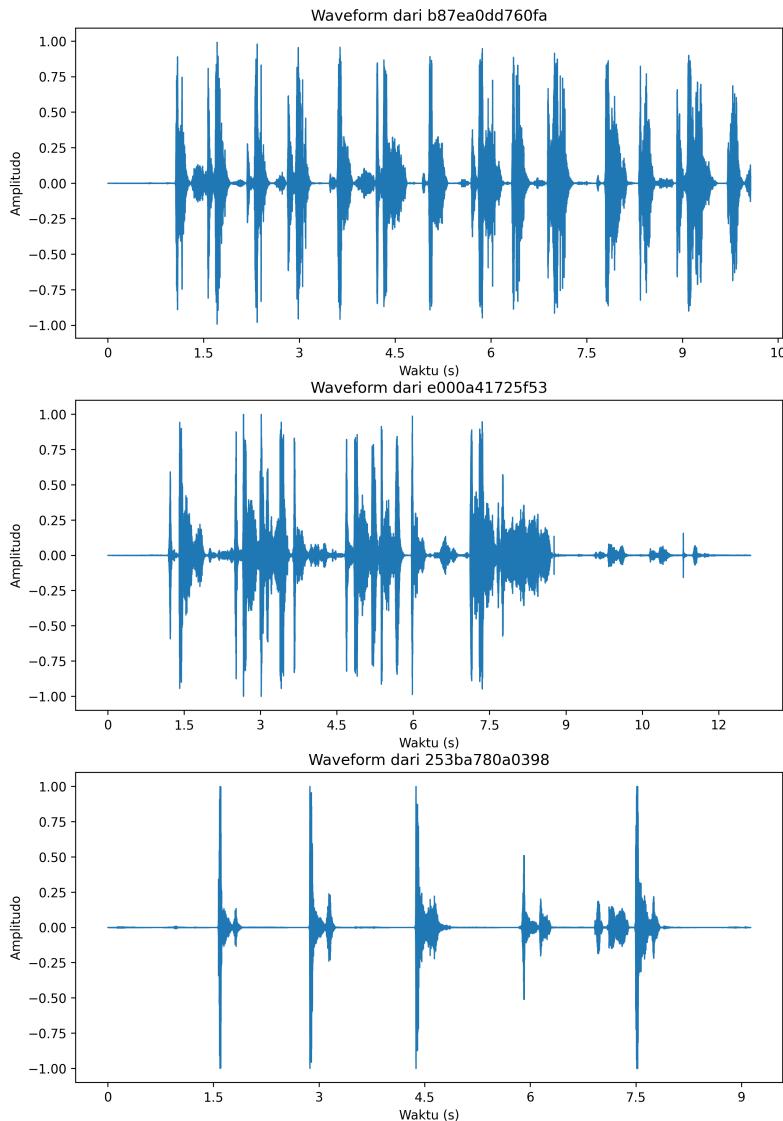
<b>Rentang SNR (dB)</b>	<b>Kualitas Audio</b>
$\geq 60$	Sangat baik
50– < 60	Baik
40– < 50	Cukup
30– < 40	Rendah
< 30	Sangat rendah

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Deskripsi Data**

Dataset yang digunakan adalah *Medical Sound Classification Challenge* yang diselenggarakan oleh Himanshu Kaushik pada platform Kaggle[31]. Dataset ini memiliki 882 jumlah data yang terdiri dari tiga jenis data untuk setiap individu yaitu audio batuk, vowel dan fitur riwayat pasien. Data diidentifikasi menggunakan candidateID yang ditetapkan untuk setiap orang. File suara dan embedding terdapat di folder candidateID yang diberikan. Dataset terbagi menjadi dua jenis yaitu 544 untuk data latih dan 338 untuk data uji yang belum memiliki label. Label penyakit terdiri dari 3 kelas yaitu asma, COPD dan pasien sehat.



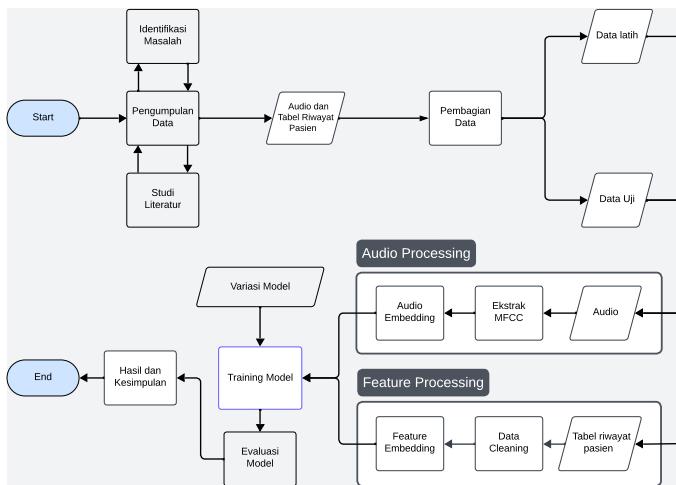
**Gambar 3.1** Visualisasi gelombang suara tiap kelas

**Tabel 3.1** Tabel riwayat pasien

<b>candidateID</b>	<b>age</b>	<b>gender</b>	<b>tbContact</b>	<b>wheezing</b>	<b>phlegm</b>	<b>Cough</b>	<b>familyAsthma</b>	<b>feverHistory</b>	<b>packYears</b>	<b>disease</b>
2bbd6c5ecf1ce	55	1	0.0	0.0	0.0	1.0	0.0	0	0	1
75fa6e335b5ca	65	0	0.0	1.0	0.0	0.0	0.0	560	2	
7dc99cfcb5aa	43	0	0.0	1.0	0.0	0.0	0.0	0	1	
59cf4a7821471	74	0	0.0	1.0	0.0	0.0	0.0	800	2	
59f9fe56c2f12	28	0	0.0	0.0	1.0	0.0	0.0	0	0	0
caee891a86d8d	56	1	0.0	1.0	0.0	0.0	0.0	0	1	
5b8578b39385f	31	0	0.0	0.0	1.0	0.0	0.0	0	0	0
f3e7d50ce7288	35	0	0.0	1.0	1.0	0.0	0.0	0	1	
ad5fa122d4efb	56	1	0.0	0.0	0.0	1.0	0.0	0	0	0
14b58d18c66c7	57	0	0.0	1.0	1.0	0.0	0.0	0	0	0
7959121db060d	48	0	0.0	1.0	0.0	0.0	0.0	0	1	
77aa6a34f6da1	21	0	0.0	0.0	1.0	1.0	0.0	0	1	
...	...	...	...	...	...	...	...	...	...	...
e17cb00bd9677	66	0	0.0	0.0	1.0	0.0	0.0	0	2	

### 3.2 Rancangan Penelitian

Gambar 3.2 menunjukkan alur penelitian yang terstruktur, dimulai dari *Start* dan diakhiri dengan *End*, dengan tahapan-tahapan yang meliputi identifikasi masalah yang spesifik, studi literatur untuk memahami konteks dan metode terkait, pengumpulan data yang melibatkan data audio dan tabel riwayat pasien, serta *Data Cleaning* untuk mempersiapkan analisis data. Selanjutnya, data dibagi menjadi data latih dan data uji dengan perbandingan 80:20, diikuti dengan variasi model yang melibatkan pengaturan parameter seperti dimensi embedding ( $d_{model}$ ), dimensi *feed-forward* ( $d_{ff}$ ), dan jumlah *head* ( $h$ ). Proses *Audio Processing* dan ekstraksi fitur MFCC dilakukan terhadap data audio, sementara fitur-fitur dari kedua jenis data diubah menjadi *embedding* dalam tahap *Feature Embedding* dan diproses lebih lanjut dalam *Feature Processing*. Model kemudian dilatih menggunakan data latih dalam tahap *Training Model*, dievaluasi kinerjanya menggunakan data uji dalam tahap *Evaluasi Model*, dan diakhiri dengan penarikan Hasil dan Kesimpulan yang menjawab rumusan masalah penelitian.



Gambar 3.2 Flowchart rancangan penelitian

### 3.2.1 Pemrosesan Suara

Data suara yang digunakan pada penelitian ini berupa suara batuk *cough.wav* pasien yang berisi minimal 3 kali batuk, dengan durasi rekaman maksimal 15 detik. Dilakukan ekstraksi fitur menggunakan *Mel frequency Capstral Coefficients* (MFCC), yaitu dengan cara memfilter secara logaritmik pada frekuensi di atas 1000 Hz dan secara linier pada frekuensi di bawah 1000 Hz. MFCC dapat meningkatkan sensitivitas pada suara dengan frekuensi rendah dan sebaliknya, pada suara dengan frekuensi tinggi, MFCC dapat mengurangi sensitivitas dalam menangkap suara [32]. Algoritma MFCC dapat dilihat pada Algoritma 2.

Algoritma 1 berfungsi untuk mendeteksi dan memotong bagian-bagian sinyal audio yang mengandung suara batuk berdasarkan energi sinyalnya. Pertama, sinyal diubah menjadi satu channel jika berbentuk stereo, lalu dihitung nilai Root Mean Square (RMS) energy untuk setiap segmen kecil audio. Energi ini dinormalisasi, kemudian dibandingkan dengan ambang batas tertentu (*cough\_threshold*) untuk menentukan kapan sebuah kejadian (batuk) terjadi. Jika energi melewati ambang batas, bagian tersebut dianggap sebagai potensi batuk, dan dipotong dari sinyal asli dengan sedikit penambahan waktu sebelum dan sesudahnya (*padding*) untuk menangkap suara batuk secara lebih lengkap. Potongan-potongan audio hasil segmentasi ini kemudian dikembalikan sebagai daftar array sinyal-sinyal batuk.

---

### Algoritma 1 Segmentasi Suara Batuk

---

- 1: **Input:**  $signal, sr, cough\_threshold, min\_cough\_duration, padding$
- 2: **Inisialisasi:**  $hop\_length \leftarrow \text{int}(min\_cough\_duration \times sr)$
- 3: **if** jumlah channel pada  $signal > 1$  **then**
- 4:      $signal \leftarrow$  rata-rata nilai antar channel
- 5: **end if**
- 6:  $energy \leftarrow$  RMS energy dari  $signal$  dengan panjang jendela  $hop\_length$
- 7:  $normalized\_energy \leftarrow$  normalisasi( $energy$ )
- 8:  $cough\_threshold \leftarrow \max(normalized\_energy) \times$   
 $cough\_threshold$
- 9:  $min\_cough\_samples \leftarrow \text{round}(sr \times min\_cough\_duration)$
- 10:  $cough\_segments \leftarrow$  list kosong
- 11:  $event\_start \leftarrow \text{None}$
- 12: **for** setiap  $(i, value)$  dalam  $normalized\_energy$  **do**
- 13:     **if**  $value \geq cough\_threshold$  **then**
- 14:         **if**  $event\_start$  is None **then**
- 15:              $event\_start \leftarrow i \times hop\_length$
- 16:         **end if**
- 17:         **else**
- 18:             **if**  $event\_start$  is not None **then**
- 19:                  $cough\_duration \leftarrow i \times hop\_length - event\_start$
- 20:                 **if**  $cough\_duration \geq min\_cough\_samples$  **then**
- 21:                      $event\_end \leftarrow i \times hop\_length + padding \times sr$
- 22:                      $event\_start \leftarrow \max(event\_start - padding \times sr, 0)$
- 23:                     tambahkan segmen  $signal[event\_start : event\_end + 1]$  ke  $cough\_segments$
- 24:                 **end if**
- 25:              $event\_start \leftarrow \text{None}$
- 26:         **end if**
- 27:     **end if**
- 28: **end for**
- 29: **Return:**  $cough\_segments$

---

---

## Algoritma 2 Ekstraksi MFCC

---

```
1: Input:  $f_{path}, n_{mfcc}, L_{target}$ 
2:  $[a, s_r] \leftarrow \text{muatAudio}(f_{path})$ 
3:  $S_{cough} \leftarrow \text{segmentasiBatuk}(a, s_r)$ 
4: if  $S_{cough} = \emptyset$  then
5:    $a_{segmented} \leftarrow a$ 
6: else
7:    $a_{segmented} \leftarrow \text{gabung}(S_{cough})$ 
8: end if
9:  $M \leftarrow \text{MFCC}(a_{segmented})$ 
10:  $M \leftarrow M^T$  (bentuk  $(T, n_{mfcc})$ )
11: if panjang( $M$ ) >  $L_{target}$  then
12:    $M \leftarrow M[0 : L_{target}, :]$ 
13: else if panjang( $M$ ) <  $L_{target}$  then
14:    $M \leftarrow \text{pad}(M, (0, L_{target} - \text{panjang}(M)), \text{'constant'})$ 
15: end if
16: Return:  $M$ 
```

---

Algoritma Ekstraksi MFCC pada Algoritma 2 bertujuan untuk menghasilkan fitur audio yang seragam dari setiap rekaman batuk pasien. Proses dimulai dengan memuat file audio dan sampling rate, kemudian dilakukan segmentasi untuk mendeteksi bagian-bagian yang mengandung suara batuk. Jika tidak ditemukan segmen batuk, seluruh audio digunakan; jika ada, segmen batuk digabungkan menjadi satu sinyal. Selanjutnya, fitur MFCC diekstraksi dari sinyal hasil segmentasi dan matriks hasilnya ditransposisikan agar berbentuk  $(T, n_{mfcc})$ , di mana  $T$  adalah jumlah frame dan  $n_{mfcc}$  jumlah koefisien MFCC. Untuk memastikan panjang fitur konsisten, jika jumlah frame lebih dari  $L_{target}$  maka dipotong, sedangkan jika kurang akan dipenuhi dengan padding hingga mencapai  $L_{target}$ . Hasil akhir berupa matriks MFCC berukuran tetap

yang siap digunakan sebagai input pada model klasifikasi.

### 3.2.2 Pemrosesan Data Tabel Riwayat Pasien

Dilakukan *preprocessing* seperti pengecekan nilai kosong, penghapusan fitur yang tidak relevan untuk memastikan kualitas data yang baik agar model dapat menangkap informasi yang relevan pada data.

---

#### Algoritma 3 Pemrosesan Data

---

```
1: function PREPROCESSFEATURES( $r$ )
2:   Normalisasi fitur numerik
3:   Ekstraksi fitur kategorikal
4:   Gabungkan semua fitur menjadi vektor  $v_f$ 
5:   return  $v_f$ 
6: end function
7: function PROCESSRow( $r$ )
8:   Dapatkan ID kandidat:  $id \leftarrow r["candidateID"]$ 
9:   Hasilkan jalur audio:  $p_{audio} \leftarrow$ 
   JOIN( $FOLDER_{suara}, id, "cough.wav"$ )
10:  Ekstraksi MFCC:  $M \leftarrow \text{EXTRACTMFCC}(p_{audio})$ 
11:  Proses fitur:  $v_f \leftarrow \text{PREPROCESSFEATURES}(r)$ 
12:  One-hot encode label:  $l \leftarrow$ 
   ONEHOTENCODE( $r["disease"]$ , depth = 3)
13:  return ( $M, v_f, l$ )
14: end function
```

---

Algoritma Pemrosesan Data pada Algoritma 3 bertujuan untuk menyiapkan data input yang terdiri dari fitur audio dan fitur tabel riwayat pasien agar siap digunakan dalam pelatihan model klasifikasi. Proses diawali dengan fungsi PreprocessFeatures yang melakukan normalisasi pada fitur numerik, mengekstraksi

fitur kategorikal, lalu menggabungkan seluruh fitur menjadi satu vektor fitur  $v_f$ . Selanjutnya, fungsi `ProcessRow` akan mengambil ID kandidat dari data, membangkitkan jalur file audio batuk berdasarkan ID tersebut, mengekstraksi fitur MFCC dari audio, memproses fitur tabel menggunakan `PreprocessFeatures`, serta mengubah label penyakit menjadi bentuk one-hot encoding dengan tiga kelas. Hasil akhir dari proses ini adalah tuple berisi fitur MFCC ( $M$ ), vektor fitur tabel ( $v_f$ ), dan label one-hot ( $l$ ) yang siap digunakan sebagai input pada model pembelajaran mesin.

### 3.2.3 Pipeline dan Pembagian Data

Data dibagi menjadi data latih dan data uji dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Digunakan parameter `random_state` agar proses pembagian data bisa direproduksi ulang. Dilakukan *stratified sampling* berdasarkan kolom target yaitu `disease` untuk memastikan distribusi kelas tetap sama di kedua set.

---

**Algoritma 4** PipeLine Model dan Pembagian Data

---

```
1: function DATAGENERATOR( $D$ )
2:   for all  $r \in D$  do
3:      $(M, v_f, l) \leftarrow \text{PROCESSRow}(r)$ 
4:     Return:  $(M, v_f, l)$ 
5:   end for
6: end function
7: function CREATDATASET( $D, b_s$ )
8:   Definisikan signature dataset
9:   Buat dataset menggunakan DATAGENERATOR( $D$ )
10:  Batch, shuffle, dan prefetch dataset
11:  return  $D$ 
12: end function
13: function SPLITDATA( $D, s_{test}, s_{rand}$ )
14:   Bagi dataset menjadi set latih dan validasi dengan stratifikasi
      target
15:   return  $(D_{train}, D_{valid})$ 
16: end function
17:  $(D_{train}, D_{valid}) \leftarrow \text{SPLITDATA}(D_{data}, 0.2, 42)$ 
18:  $\mathcal{D}_{train} \leftarrow \text{CREATDATASET}(D_{train}, B)$ 
19:  $\mathcal{D}_{valid} \leftarrow \text{CREATDATASET}(D_{valid}, B)$ 
```

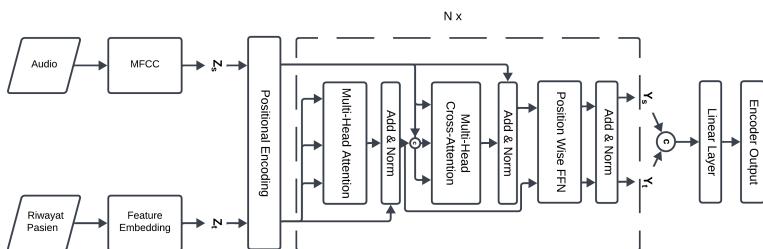
---

Algoritma Pipeline Model dan Pembagian Data pada Algoritma 4 bertujuan untuk menyiapkan alur pemrosesan data sebelum proses pelatihan model dimulai. Proses diawali dengan fungsi `DataGenerator` yang akan mengambil setiap baris data, memprosesnya menggunakan `ProcessRow` sehingga menghasilkan fitur MFCC, vektor fitur tabel, dan label one-hot. Selanjutnya, fungsi `CreateDataset` akan membentuk dataset dengan mendefinisikan struktur data, kemudian melakukan proses batching,

shuffling, dan prefetch agar data siap digunakan secara efisien dalam pelatihan model. Fungsi `SplitData` digunakan untuk membagi data menjadi data latih dan validasi dengan metode stratifikasi berdasarkan label, sehingga distribusi kelas tetap seimbang pada kedua set. Dengan demikian, pipeline ini memastikan seluruh data telah diproses, dibagi, dan disiapkan dalam format yang optimal untuk proses pelatihan dan evaluasi model klasifikasi.

### 3.2.4 Arsitektur

Arsitektur model yang digunakan adalah *multimodal transformer* [33] yang menggabungkan fitur audio dan riwayat pasien untuk klasifikasi penyakit. Model ini terdiri dari beberapa komponen utama: *embedding layer*, *positional encoding*, *multi-head attention* dan *multi-head cross-attention* serta *feedforward network* (FFN). Gambar 3.3 menunjukkan desain arsitektur model yang digunakan dalam penelitian ini.



**Gambar 3.3** Desain rancangan arsitektur model.

Input berupa audio dengan fitur MFCC berukuran  $T \times F$  (dengan  $T$  adalah jumlah frame dan  $F$  adalah jumlah koefisien MFCC) dan data riwayat pasien berukuran  $N \times d_{feat}$  ( $N$  adalah jumlah data riwayat,  $d_{feat}$  adalah dimensi fitur) akan diubah menjadi embedding

berukuran  $T \times d_{model}$  dan  $N \times d_{model}$  melalui *embedding layer* menghasilkan *embedding*  $Z_s$  dan  $Z_t$ . *Embedding* ini ditambahkan dengan *positional encoding* untuk memberikan informasi posisi pada data. Setelah *embedding*  $Z_s$  dan  $Z_t$  melewati *positional encoding*, *embedding*  $Z_t$  diteruskan ke layer *Multi Head Attention*, di mana mekanisme *self-attention* digunakan untuk menangkap hubungan antar fitur riwayat pasien. Kemudian, pada *Multi Head Cross-Attention Query embedding*  $Z_s$  akan melakukan *cross-attention* dengan *Key Value* gabungan  $Z_s$  dan  $Z_t$ . Ide dasar dari *cross-attention*[34] ini bahwa  $Z_s$  akan memperhatikan informasi temporal  $Z_t$  dari tabel riwayat pasien. Hasil kedua *Multi Head Attention* akan memasuki *layer feedforward network* (FFN) untuk pemetaan *non-linear*. Setelah melalui *feedforward network* (FFN), hasil akhir *encoder* berupa *embedding multimodal* dengan token [cls] dikombinasikan dan diproyeksikan menggunakan *linear layer*, diikuti oleh *softmax* untuk menghasilkan output akhir berupa nilai peluang untuk setiap kelas. Perhitungan lebih detail dijelaskan pada lampiran B.

### 3.2.5 Variasi Model

Arsitektur model yang digunakan mengadopsi desain encoder Transformer[24], serupa dengan prinsip pada ViT[26] dan BERT[29]. Didefinisikan empat konfigurasi berdasarkan kapasitas model: Base 1 (Base-1,  $N = 3$ ,  $d_{model} = 512$ ,  $d_{ff} = 1024$ ,  $h = 8$ ,  $P_{drop} = 0,1$ ), Base 2 (Base-2,  $N = 3$ ,  $d_{model} = 768$ ,  $d_{ff} = 1536$ ,  $h = 12$ ,  $P_{drop} = 0,2$ ), Big 1 (Big-1,  $N = 6$ ,  $d_{model} = 512$ ,  $d_{ff} = 1024$ ,  $h = 8$ ,  $P_{drop} = 0,2$ ), dan Big 2 (Big-2,  $N = 6$ ,  $d_{model} = 1024$ ,  $d_{ff} = 2048$ ,  $h = 16$ ,  $P_{drop} = 0,3$ ), di mana  $N$  adalah jumlah blok encoder,  $d_{model}$  adalah dimensi embedding,  $d_{ff}$  adalah dimensi feedforward,  $h$  adalah jumlah head pada attention, dan  $P_{drop}$  adalah tingkat dropout. Skema penamaan ini mencerminkan

kedalaman dan ukuran model, dengan “Base” dan “Big” menunjukkan jumlah lapisan encoder, sedangkan “1” dan “2” menunjukkan besar kecilnya dimensi model dan kapasitas pemrosesan.

### 3.2.6 Loss Function

Fungsi kerugian yang digunakan *Categorical Cross-Entropy* untuk setiap kelas 3.1:

$$L(y, \hat{y}) = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (3.1)$$

Keterangan:

$L(y, \hat{y})$  : nilai *Categorical Cross-Entropy*

$y$  : nilai target asli

$\hat{y}_i$  : nilai prediksi

$y_i$  adalah label sebenarnya (0 atau 1 untuk setiap kelas) dari *one-hot encoding* vektor target.  $\hat{y}_i$  adalah probabilitas yang diprediksi untuk kelas  $i$ .  $C$  merupakan jumlah kelas yang diprediksi.

### 3.2.7 Pelatihan Model

Penelitian ini akan menggunakan *optimizer* sama dengan yang digunakan Vaswani et al.[24], yaitu *optimizer* Adam [35] yang menggabungkan momentum dengan RMSprop. Pembahasan lebih detail mengenai optimizer Adam dijelaskan pada lampiran C. Model akan dilatih menggunakan *Notebook Kaggle* dengan GPU NVIDIA Tesla P100-PCIE-16GB untuk model *Base* dan *Big*. Beberapa kedalaman *layer* dan jumlah *head attention* akan diuji performanya dengan tetap memperhatikan batasan komputasi.

---

**Algoritma 5** Pelatihan Model

---

```
1: Inisialisasi:  $m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0, \beta_1 \leftarrow 0.9, \beta_2 \leftarrow 0.98, \epsilon \leftarrow 10^{-9}$ 
2: Inisialisasi: epochs, batch_size
3: while epoch → epochs do
4:   while step → N // batch_size do
5:      $lr \leftarrow (d_{embed}^{-0.5} \cdot \min(step^{-0.5}, step \times 4000^{-1.5}))$ 
6:      $t \leftarrow t + 1$ 
7:      $g_t \leftarrow \nabla_{\theta} \mathcal{L}(\theta_{t-1})$  (Gradien loss function)
8:      $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update momentum pertama)
9:      $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update momentum kedua)
10:     $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Koreksi bias momentum pertama)
11:     $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Koreksi bias momentum kedua)
12:     $\theta_t \leftarrow \theta_{t-1} - lr \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameter)
13:  end while
14: end while
15: Return: Parameter  $\theta_t$ 
```

---

### 3.3 Evaluasi Model

#### 3.3.1 Confusion Matrix

*Confusion matrix* adalah metode dalam pembelajaran mesin yang menyediakan representasi visual dari performa model dalam tugas klasifikasi [36]. Matriks ini merangkum prediksi yang benar dan salah yang dibuat oleh model, yang memungkinkan penghitungan berbagai metrik kinerja seperti akurasi, presisi, *recall*, dan *F1-score*. Matriks ini biasanya terdiri dari empat komponen: *True Positive*, *True Negarive*, *False Positive*, dan *False Negative* 3.2.

**Tabel 3.2** Confusion Matrix untuk 3 Kelas (a), Rumus metrik evaluasi (b)

		Prediksi			Metrik	Rumus
		C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>		
Aktual	C <sub>1</sub>	TP <sub>1</sub>	FP <sub>12</sub>	FP <sub>13</sub>	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
	C <sub>2</sub>	FN <sub>21</sub>	TP <sub>2</sub>	FP <sub>23</sub>	Precision	$\frac{TP}{TP+FP}$
	C <sub>3</sub>	FN <sub>31</sub>	FN <sub>32</sub>	TP <sub>3</sub>	Recall	$\frac{TP}{TP+FN}$

(a)

		F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
--	--	----------	---

(b)

### 3.3.2 Kurva ROC-AUC

Kurva ROC-AUC merupakan metode untuk mengevaluasi keakuratan algoritma klasifikasi, yang memberikan informasi tentang performanya di berbagai ambang batas (*threshold*)[37]. Kurva ini mengukur keseimbangan antara sensitivitas (*true positive rate*) dan spesifisitas (*false positive rate*), yang memungkinkan penilaian komprehensif terhadap efektivitas model [38].

$$\text{TPR/sensitivitas} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.2)$$

$$\text{FPR/spesifisitas} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (3.3)$$

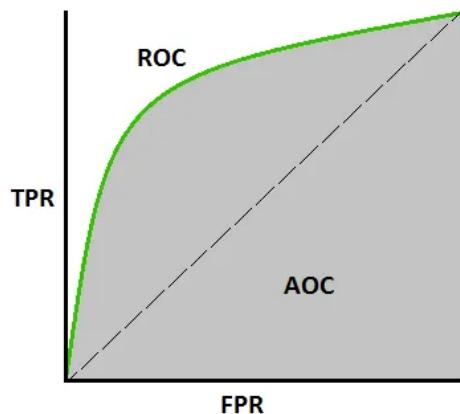
Keterangan :

TP : *True Positive* (Jumlah data positif yang diprediksi benar)

FN : *False Negative* (Jumlah data positif yang diprediksi salah)

FP : *False Positive* (Jumlah data negatif yang diprediksi salah)

TN : *True Negative* (Jumlah data negatif yang diprediksi benar)



**Gambar 3.4** Kurva ROC-AUC

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Data Cleaning dan Data Preprocessing**

Sebelum memulai pelatihan model, serangkaian proses pengecekan dataset dilakukan untuk mendapatkan kualitas dataset yang baik. Tahapan ini penting untuk menjamin validitas hasil penelitian serta memastikan model berjalan dengan baik selama proses pelatihan.

##### **4.1.1 Pemrosesan Tabel riwayat pasien**

1. **Pengecekan Data Kosong** dilakukan untuk mengidentifikasi dan menangani nilai-nilai yang hilang dalam data. Pada tabel 4.1 diketahui terdapat 148 data kosong pada kolom coldPresent.

**Tabel 4.1 Jumlah Baris Kosong per Kolom**

Nama Kolom	Jumlah Baris Kosong
candidateID	0
age	0
gender	0
tbContactHistory	0
wheezingHistory	0
phlegmCough	0
familyAsthmaHistory	0
feverHistory	0
coldPresent	148
packYears	0
disease	0

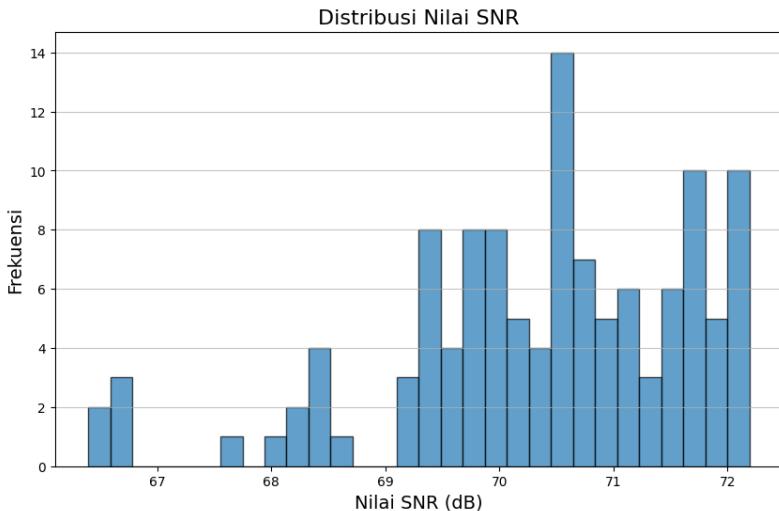
2. **Menghapus kolom dengan nilai kosong** untuk membersihkan dataset dari kolom yang tidak memiliki informasi yang berguna. Hal ini dapat meningkatkan efisiensi analisis data dan menghindari gangguan dari data yang tidak lengkap atau tidak relevan. Kolom coldPresent dihapus karena terdapat nilai kosong sebanyak 148 baris atau 27% dari data.

**Tabel 4.2** Jumlah Baris Kosong per Kolom setelah menghapus

Nama Kolom	Jumlah Baris Kosong
candidateID	0
age	0
gender	0
tbContactHistory	0
wheezingHistory	0
phlegmCough	0
familyAsthmaHistory	0
feverHistory	0
packYears	0
disease	0

#### 4.1.2 *Audio Processing*

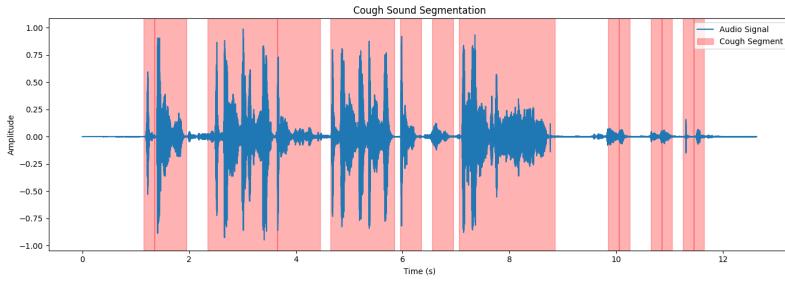
1. Signal to Noise Ratio (SNR) adalah rasio antara kekuatan sinyal suara batuk dengan kekuatan noise yang tidak diinginkan. SNR yang tinggi menunjukkan bahwa sinyal suara batuk lebih kuat dibandingkan noise, sehingga kualitas audio lebih baik. Sebaliknya, SNR yang rendah menunjukkan bahwa noise mengganggu sinyal suara batuk, sehingga kualitas audio menurun. Ditampilkan gambar 4.1 untuk mengetahui distribusi nilai SNR pada suara batuk.



**Gambar 4.1** Visualisasi distribusi SNR

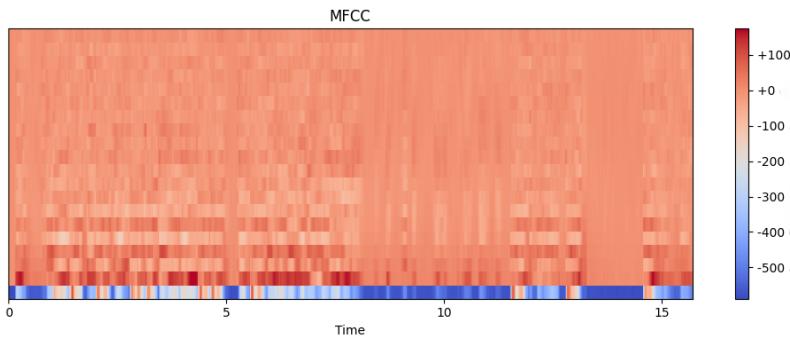
Berdasarkan histogram distribusi nilai SNR pada gambar 4.1, terlihat bahwa sebagian besar nilai SNR terkonsentrasi di rentang antara 69 dB hingga 72 dB, dengan nilai minimum 66.39 dB dan maksimum 72.19 dB dengan puncak frekuensi tertinggi berada di sekitar 70.5 dB. Distribusi ini menunjukkan bahwa kualitas sinyal audio secara umum sangat baik, dengan sebagian besar sampel memiliki rasio suara terhadap noise yang tinggi. Secara keseluruhan, berdasarkan tabel 2.2 nilai SNR berada pada rentang data  $\geq 60$ , menandakan bahwa mayoritas data memiliki SNR yang sangat baik.

2. **Segmentasi Audio** dilakukan menggunakan Algoritma 1 untuk mendapatkan bagian batuk berdasarkan batas  $threshold=0.05$  dan durasi minimal 0.1 detik. Hasil segmentasi audio dapat dilihat pada gambar 4.2.



**Gambar 4.2** Segmentasi Suara batuk

3. **Ekstraksi MFCC** dilakukan setelah segmentasi, fitur suara diekstrak menggunakan MFCC dengan parameter `n_mfcc=20` dan `SEQ_LENGTH` sesuai dengan variasi model menggunakan Algoritma 2. Hasil ekstraksi audio menggunakan MFCC dapat dilihat pada gambar 4.3. Keluaran MFCC berupa matriks berukuran  $20 \times \text{SEQ\_LENGTH}$ .



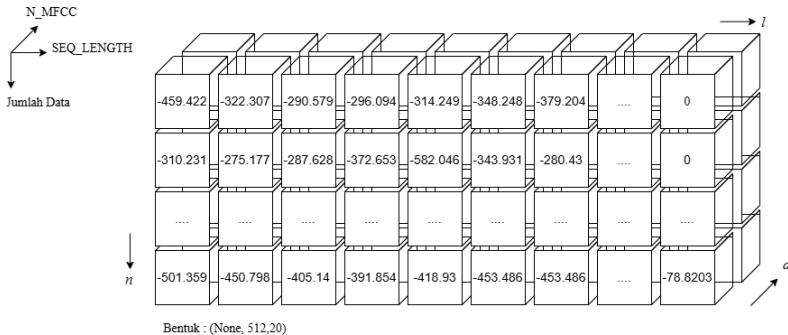
**Gambar 4.3** Visualisasi MFCC

## 4.2 Visualisasi Input Model

### 4.2.1 *Input Spatial Embedding*

Visualisasi input embedding spasial  $\mathbf{Z}_s = [\mathbf{z}_1^{(s)}, \mathbf{z}_2^{(s)}, \dots, \mathbf{z}_n^{(s)}]$  dilakukan untuk memeriksa apakah data yang digunakan sudah

sesuai dengan yang diharapkan. Gambar 4.4 menunjukkan visualisasi input model berupa nilai MFCC yang telah diekstraksi dari rekaman audio batuk. Setiap baris (**n**) mewakili fitur MFCC untuk setiap pasien, dan setiap kolom (**l**) mewakili frame sebanyak SEQ\_LENGTH dan kedalaman (**d**) mewakili jumlah koefisien MFCC.



**Gambar 4.4** Visualisasi *Input Spatial Embedding*

#### 4.2.2 *Input Temporal Embedding*

Visualisasi input embedding temporal  $\mathbf{Z}_t = [\mathbf{z}_1^{(t)}, \mathbf{z}_2^{(t)}, \dots, \mathbf{z}_n^{(t)}]$  dilakukan untuk memeriksa apakah data yang digunakan sudah sesuai dengan yang diharapkan. Gambar 4.5 menunjukkan visualisasi input model berupa nilai tabel riwayat pasien. Setiap baris (**n**) mewakili fitur riwayat untuk setiap pasien, dan setiap kolom (**l**) mewakili variabel dalam riwayat pasien.

	Jumlah Kolom	55	1	0	0	0	1	0
	Jumlah Data	65	0	0	1	0	0	560
		....	....	....	....	....	....	....
		18	0	0	0	0	0	0
n								

Bentuk : (None, 8, 1)

**Gambar 4.5** Visualisasi *Input Temporal Embedding*

Sebelum masuk ke dalam *encoder*, kedua input spasial  $\mathbf{z}_i^{(s)}$  dan temporal  $\mathbf{z}_i^{(t)}$  diubah menjadi vektor embedding dengan dimensi yang sesuai dengan model menggunakan *layer linear*. Input spasial diubah menjadi vektor embedding spasial  $\mathbf{z}_i^{(s)}$  dengan dimensi  $d_{model}$ , sedangkan input temporal diubah menjadi vektor embedding temporal  $\mathbf{z}_i^{(t)}$  dengan dimensi  $d_{model}$ . Proses ini dilakukan untuk memastikan bahwa kedua jenis input memiliki representasi yang konsisten dan dapat diproses oleh model Transformer.

## 4.3 Evaluasi Model

### 4.3.1 Variasi Model

Tabel 4.3 menunjukkan bahwa model Transformer dengan konfigurasi **Big-1 (d\_model= 512, d\_ff= 1024, h = 8, dropout= 0.2)** memberikan performa terbaik dengan akurasi 82.57% dan F1-score 81.95%, meskipun jumlah parameternya (20.1 juta) masih tergolong efisien. Sebaliknya, peningkatan ukuran model secara drastis, seperti pada konfigurasi dengan  $d_{model}=1024$  dan 80 juta parameter, justru menurunkan performa, hal tersebut dapat diakibatkan oleh overfitting atau kesulitan dalam mengikuti pola selama proses pelatihan. Selain itu, model Base dengan parameter

lebih sedikit justru dapat menghasilkan performa yang cukup kompetitif, terutama konfigurasi pertama yang mencapai akurasi 81.65%. Hal ini menunjukkan bahwa peningkatan kompleksitas model tidak selalu sebanding dengan peningkatan kinerja. Detail rangkuman model dan jumlah parameter terdapat pada lampiran D.

**Tabel 4.3** Variasi model

Model	N	$d_{model}$	$d_{ff}$	h	$P_{drop}$	Akurasi	F1-Score	Param
Base-1	3	512	1024	8	0.1	81.65	81.52	10.6 juta
Base-2	3	768	1536	12	0.2	77.98	75.83	23.8 juta
Big-1	6	512	1024	8	0.2	<b>82.57</b>	<b>81.95</b>	20.1 juta
Big-2	6	1024	2048	16	0.3	77.98	75.60	80 juta

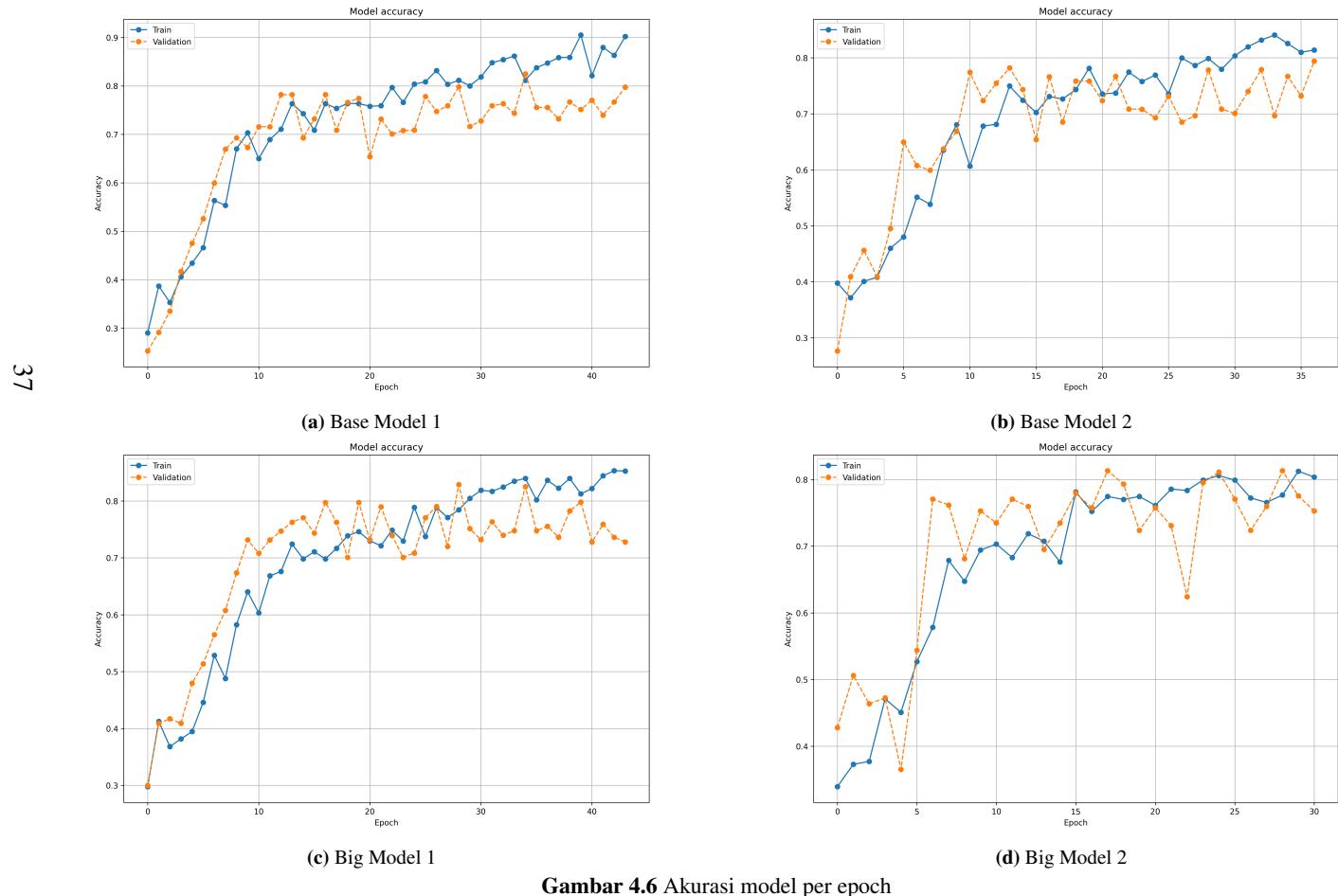
### 4.3.2 Akurasi dan Loss Pelatihan

Selama proses pelatihan, akurasi dan loss terus dimonitor dengan menggunakan *Early Stopping* dengan meminimalkan `val_loss` dengan parameter `patience = 15` dan `restore_best_weights = True` yang berarti model akan memonitor nilai loss validasi dengan nilai terendah selama 15 epoch terakhir. Jika selama 15 epoch terakhir tidak terdapat penurunan loss, maka proses pelatihan akan dihentikan dan model akan memulihkan bobot terbaik. *Early Stopping* digunakan selama pelatihan model untuk mencegah overfitting. *Early Stopping* memungkinkan proses pelatihan dihentikan setelah performa model pada data validasi mulai menurun, sehingga mempertahankan model yang berkinerja baik pada data yang tidak terlihat.

#### 1. Akurasi

Keempat Gambar 4.6 menunjukkan performa akurasi model selama pelatihan dan validasi. Base Model 1 memiliki akurasi pelatihan tinggi namun akurasi validasi stagnan,

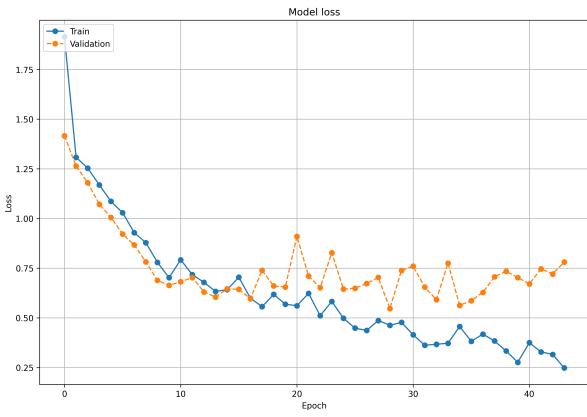
menunjukkan overfitting. Base Model 2 menunjukkan peningkatan akurasi yang lebih stabil namun performanya lebih rendah secara keseluruhan. Big Model 1 memiliki akurasi pelatihan yang tinggi namun akurasi validasi fluktuatif, mengindikasikan overfitting atau ketidakstabilan generalisasi. Sementara itu, Big Model 2 menunjukkan keseimbangan terbaik antara akurasi pelatihan dan validasi, dengan hasil yang tinggi dan stabil pada keduanya, sehingga merupakan model dengan performa generalisasi terbaik di antara keempat model yang diuji.



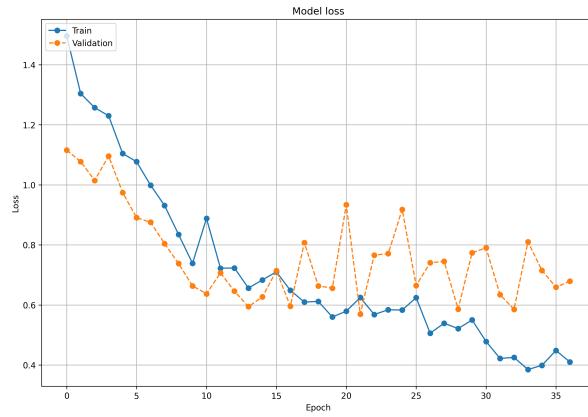
Gambar 4.6 Akurasi model per epoch

## 2. Loss

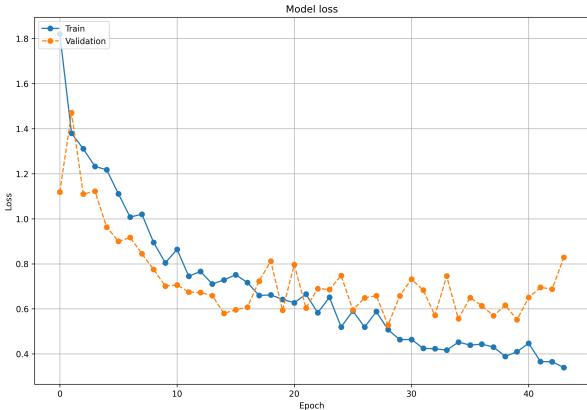
Keempat Gambar 4.7 menunjukkan tren penurunan *loss* selama pelatihan, namun dengan pola validasi yang berbeda. Base Model 1 dan Big Model 1 mengalami overfitting yang cukup jelas, ditandai dengan loss pelatihan yang terus menurun sementara loss validasi stagnan dan fluktuatif. Base Model 2 menunjukkan penurunan loss pelatihan yang stabil, namun loss validasi masih cukup fluktuatif dan tidak menunjukkan peningkatan signifikan. Big Model 2 memiliki kinerja terbaik, dengan penurunan loss pelatihan yang konsisten dan loss validasi yang lebih stabil serta mengikuti tren yang serupa, menandakan generalisasi model yang baik. Dengan demikian, Big Model 2 kembali menunjukkan performa paling seimbang di antara semua model.



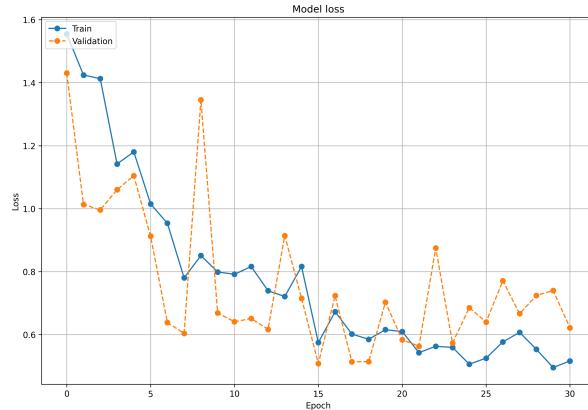
(a) Base Model 1



(b) Base Model 2



(c) Big Model 1

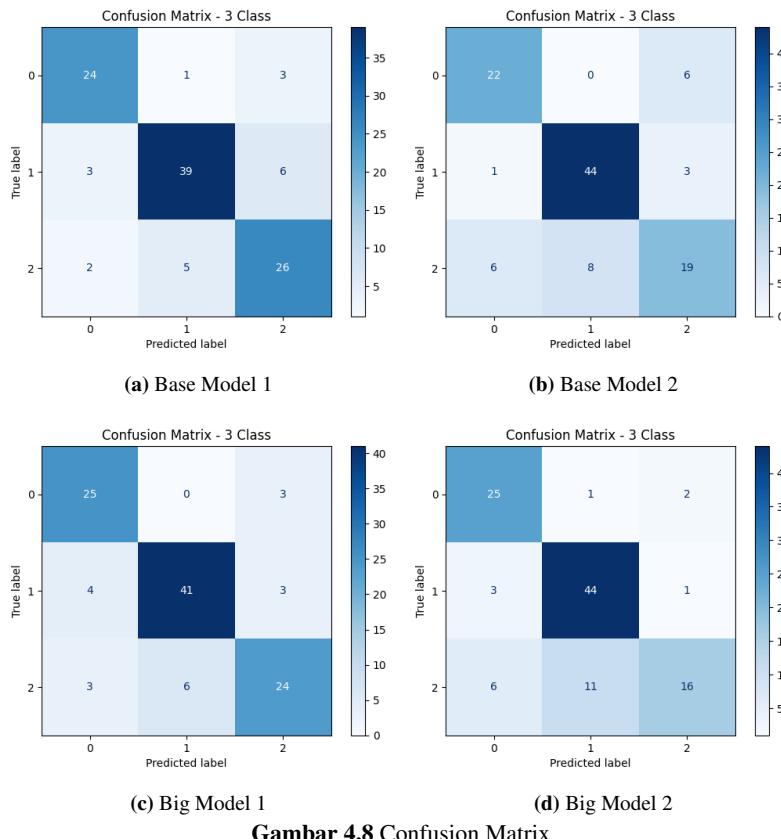


(d) Big Model 2

Gambar 4.7 Loss model per epoch

### 4.3.3 Confusion Matrix

Penggunaan *confusion matrix* dalam evaluasi model memberikan wawasan tentang kinerja model dengan nilai *true positive*, *true negative*, *false positive*, dan *false negative*. Berikut ditampilkan hasil visualisasi *confusion matrix* dari keempat model pada gambar 4.8.



**Gambar 4.8** Confusion Matrix

Gambar 4.8 menunjukkan hasil *confusion matrix* untuk keempat model Base-1, Base-2, Big-1 dan Big-2. Keempat model menunjukkan hasil prediksi yang baik dengan nilai diagonal *True*

*Positive*(TP) yang tinggi. Selanjutnya dihitung metrik dari *confusion matrix* menggunakan persamaan pada tabel 3.2 berupa akurasi, presisi, *recall* dan *f1-score*. Hasil perhitungan metrik evaluasi ditampilkan pada tabel 4.4.

**Tabel 4.4** Akurasi, Presisi, Recall, dan F1-Score dari 4 Model

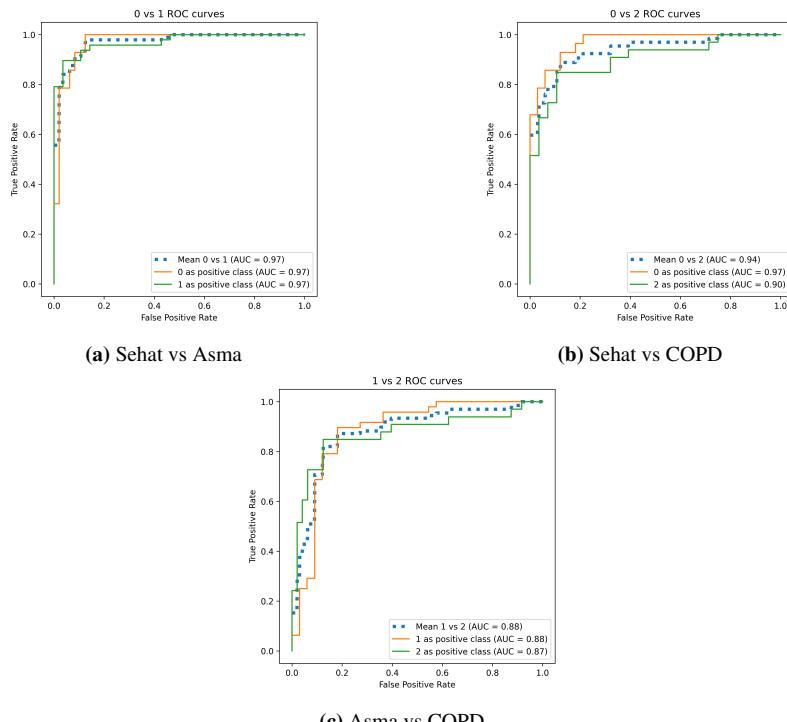
Model	Akurasi	Presisi	Recall	F1-Score
Base Model 1	81.65	81.24	81.92	81.52
Base Model 2	77.98	76.11	75.94	75.83
Big Model 1	<b>82.57</b>	<b>81.79</b>	<b>82.48</b>	<b>81.95</b>
Big Model 2	77.98	78.77	76.48	75.60

Tabel 4.4 menunjukkan bahwa **Big Model 1** memiliki performa terbaik di antara keempat model, dengan nilai **akurasi tertinggi (82.57%)**, **presisi (81.79%)**, **recall (82.48%)**, dan **F1-score (81.95%)**, yang menunjukkan keseimbangan yang sangat baik antara kemampuan model dalam mendekripsi kelas positif dan menghindari kesalahan klasifikasi. **Base Model 1** menempati posisi kedua dengan nilai yang cukup kompetitif, terutama pada recall (81.92%) dan F1-score (81.52%), menunjukkan bahwa meskipun lebih sederhana, model ini cukup andal. **Base Model 2** dan **Big Model 2** menunjukkan performa yang lebih rendah secara keseluruhan, dengan nilai akurasi yang sama (77.98%), tetapi Big Model 2 unggul sedikit dalam presisi (78.77%) dan recall (76.48%) dibandingkan Base Model 2. Hal ini mengindikasikan bahwa kompleksitas tambahan pada Big Model 2 tidak memberikan peningkatan signifikan dalam kinerja dan bahkan menunjukkan hasil F1-score terendah (75.60%). Secara keseluruhan, Big Model 1 menjadi model paling optimal dari sisi keseimbangan performa klasifikasi.

#### 4.3.4 Kurva AUC-ROC

##### 1. Base Model 1

Ditampilkan kurva AUC-ROC dari model Base-1 untuk mengetahui performa model di berbagai *threshold*. Berikut ditampilkan hasil visualisasi kurva AUC-ROC dari model Base-1 pada Gambar 4.9.



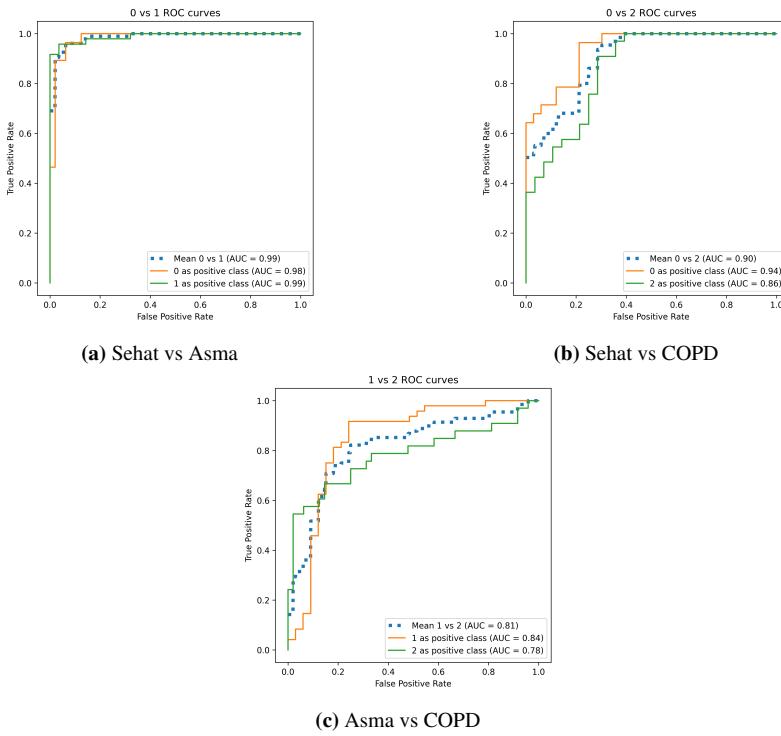
Gambar 4.9 Kurva AUC-ROC antar kelas Base Model 1

Kurva AUC-ROC Base Model 1 pada gambar 4.9 menunjukkan performa klasifikasi yang sangat baik untuk membedakan kelas sehat dengan asma (AUC = 0.97) dan sehat dengan COPD (AUC = 0.94), menandakan bahwa model mampu mengenali kondisi sehat dengan sangat akurat.

Namun, performa menurun saat membedakan antara asma dan COPD ( $AUC = 0.88$ ), yang menunjukkan bahwa model kesulitan membedakan dua kondisi penyakit yang memiliki karakteristik serupa. Secara keseluruhan, model unggul dalam klasifikasi antara kondisi sehat dan sakit, namun masih memerlukan peningkatan dalam membedakan jenis penyakit secara lebih spesifik.

## 2. Base Model 2

Ditampilkan kurva AUC-ROC dari model Base-2 untuk mengetahui performa model di berbagai *threshold*. Berikut ditampilkan hasil visualisasi kurva AUC-ROC dari model Base-2 pada gambar 4.10.

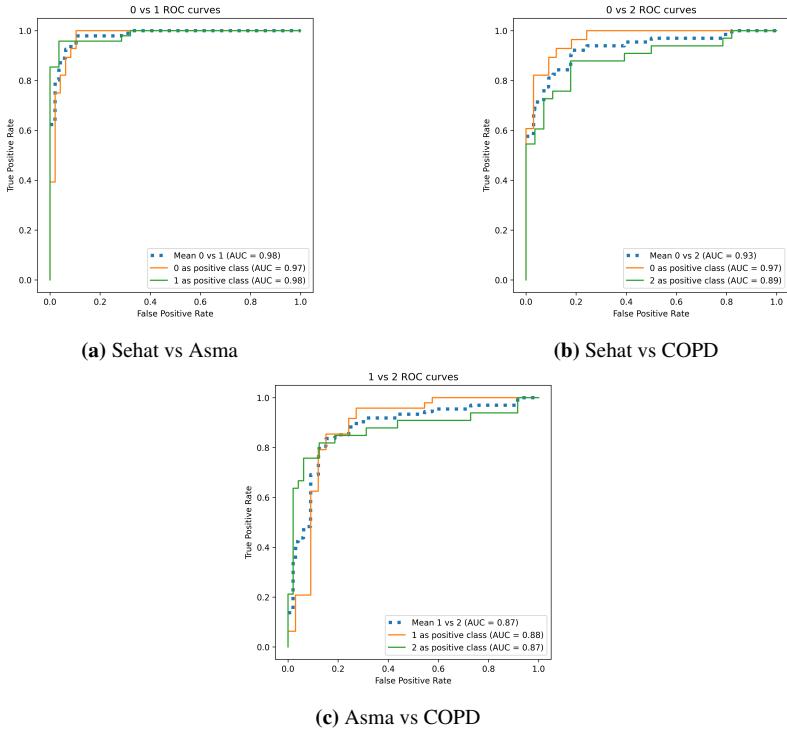


**Gambar 4.10** Kurva AUC-ROC antar kelas Base Model 2

Kurva AUC-ROC Base Model 2 pada gambar 4.10 menunjukkan performa yang sangat baik dalam membedakan kelas sehat dengan asma ( $AUC = 0.99$ ), menandakan kemampuan deteksi yang hampir sempurna. Namun, performa menurun untuk pasangan sehat vs COPD ( $AUC = 0.90$ ) dan lebih rendah lagi untuk asma vs COPD ( $AUC = 0.81$ ), terutama saat COPD menjadi kelas positif ( $AUC = 0.78$ ). Ini menunjukkan bahwa meskipun model sangat efektif dalam membedakan individu sehat dari yang sakit, kemampuannya untuk membedakan antara dua kondisi penyakit masih terbatas. Kesulitan ini disebabkan oleh kesamaan karakteristik antara asma dan COPD, serta representasi fitur yang kurang optimal untuk kelas COPD[39].

### 3. Big Model 1

Ditampilkan kurva AUC-ROC dari model Big-1 untuk mengetahui performa model di berbagai *threshold*. Berikut ditampilkan hasil visualisasi kurva AUC-ROC dari model Big-1 pada gambar 4.11.

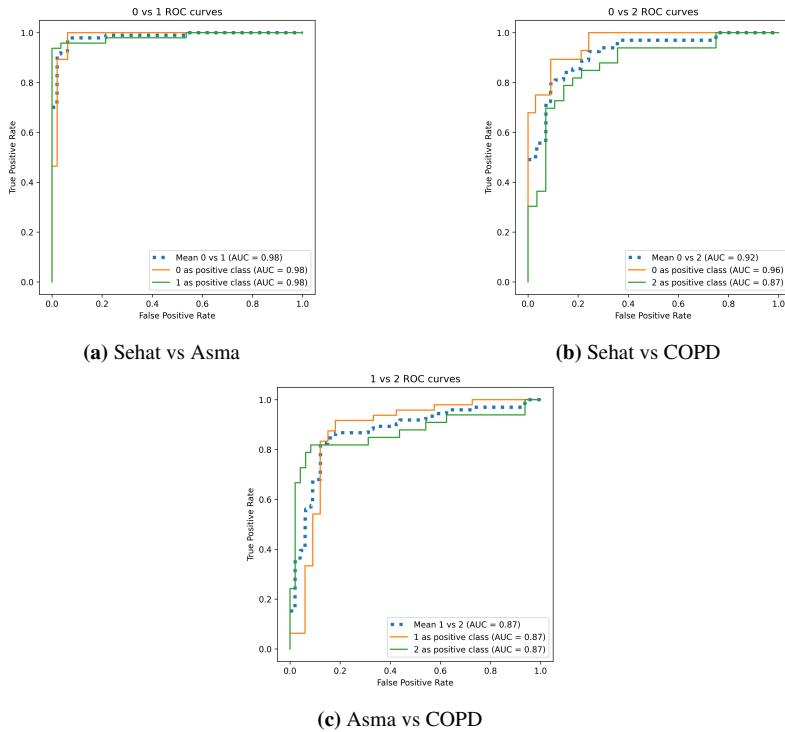


**Gambar 4.11** Kurva AUC-ROC antar kelas Big Model 1

Kurva AUC-ROC dari Big Model 1 pada gambar 4.11 menunjukkan performa klasifikasi yang sangat baik dalam membedakan antara kelas sehat dan asma dengan AUC sebesar 0.98, serta antara sehat dan COPD dengan AUC sebesar 0.93 [40]. Hal ini mengindikasikan bahwa model mampu mengenali individu sehat dari penderita penyakit dengan akurasi tinggi. Untuk pasangan asma vs COPD, AUC sebesar 0.87 menunjukkan performa yang cukup baik namun tidak seakurat dua pasangan lainnya, hal ini disebabkan karena tumpang tindih karakteristik antara kedua kondisi tersebut[39]. Secara keseluruhan, Big Model 1 menunjukkan keseimbangan yang baik dalam mendeteksi ketiga kelas.

#### 4. Big Model 2

Ditampilkan kurva AUC-ROC dari model Big-2 untuk mengetahui performa model di berbagai *threshold*. Berikut ditampilkan hasil visualisasi kurva AUC-ROC dari model Big-2 pada gambar 4.12.



Gambar 4.12 Kurva AUC-ROC antar kelas Big Model 2

Kurva AUC-ROC dari Big Model 2 pada gambar 4.12 menunjukkan performa klasifikasi yang sangat baik dalam membedakan antara kelas sehat dan asma dengan AUC sebesar 0.98, mencerminkan kemampuan tinggi model dalam mengidentifikasi kedua kondisi tersebut. Untuk pasangan sehat dan COPD, AUC sebesar 0.92 juga menunjukkan performa yang kuat, meskipun sedikit lebih rendah dibanding

klasifikasi sehat vs asma. Sementara itu, klasifikasi antara asma dan COPD memiliki AUC sebesar 0.87, yang menunjukkan performa cukup baik namun hasil tersebut bisa disebabkan karena kesamaan karakteristik antara kedua kondisi[39]. Secara keseluruhan, Big Model 2 menunjukkan kapabilitas klasifikasi yang konsisten tinggi di semua kombinasi kelas.

#### 4.4 Studi Ablasi

Studi ablation dilakukan untuk mengevaluasi kontribusi masing-masing komponen dalam model terhadap performa keseluruhan.

##### 4.4.1 Positional Encoding

Studi ablation pada positional encoding dilakukan untuk mengevaluasi pengaruhnya terhadap performa model. Ditampilkan tabel 4.5 yang membandingkan performa model dengan Sinusoidal Positional Encoding dan Trainable Positional Encoding.

**Tabel 4.5** Perbandingan Sinusoidal Positional Encoding dengan Trainable Positional Encoding.

Metrik	Sinusoidal	Trainable
Akurasi	66.06%	82.57%
Presisi	66.22%	81.79%
Recall	61.68%	82.48%
F1-Score	62.36%	81.95%

Tabel 4.5 menunjukkan bahwa model dengan Trainable Positional Encoding secara konsisten mengungguli Sinusoidal Positional Encoding pada semua metrik evaluasi, yaitu akurasi, presisi, recall,

dan F1-score. Model dengan Trainable Positional Encoding mencapai akurasi 82.57%, presisi 81.79%, recall 82.48%, dan F1-score 81.95%, jauh lebih tinggi dibandingkan dengan model berbasis Sinusoidal yang hanya mencapai kisaran 61–66% di semua metrik. Hal ini mengindikasikan bahwa positional encoding yang dapat dilatih memungkinkan model untuk memahami representasi posisi secara lebih fleksibel dan sesuai dengan karakteristik data, sehingga menghasilkan performa klasifikasi yang jauh lebih baik[29].

#### 4.4.2 *Cross-Attention*

Studi ablation pada *cross-attention* dilakukan untuk mengevaluasi pengaruhnya terhadap performa model. Tabel 4.6 menunjukkan perbandingan performa model dengan dan tanpa cross-attention.

**Tabel 4.6** Perbandingan model dengan Cross-Attention dan tanpa Cross-Attention

Model	Akurasi	F1-Score	Parameter
Tanpa Cross-Attention	82.57%	81.50%	20.1 juta
Dengan Cross-Attention	<b>82.57%</b>	<b>81.95%</b>	<b>20.1 juta</b>

Tabel 4.6 menunjukkan bahwa model dengan *cross-attention* mencapai akurasi 82.57% dan F1-score 81.95%, sedangkan model tanpa *cross-attention* memiliki akurasi yang sama (82.57%) tetapi F1-score sedikit lebih rendah (81.50%). Hal ini menunjukkan bahwa meskipun *cross-attention* tidak meningkatkan akurasi, namun memberikan kontribusi positif terhadap keseimbangan antara presisi dan recall, yang tercermin pada peningkatan F1-score. Dengan demikian, *cross-attention* berperan penting dalam meningkatkan kemampuan model untuk menangkap hubungan antar fitur spasial dan temporal, sehingga meningkatkan

performa klasifikasi secara keseluruhan[34].

#### 4.4.3 *Shared-weights Feed-Forward Network (FFN)*

Selain penggunaan *cross-attention*, studi ablation juga dilakukan pada *shared-weights* feed-forward network (FFN) untuk mengevaluasi pengaruhnya terhadap performa model. Tabel 4.7 menunjukkan perbandingan performa model dengan dan tanpa *shared-weights* feed-forward network.

**Tabel 4.7** Perbandingan model dengan Shared-Weights Feed-Forward Network

Model	Akurasi	F1-Score	Parameter
Tanpa Shared-Weights FFN	80.73%	78.45%	26.4 juta
Dengan Shared-Weights FFN	<b>84.40%</b>	<b>84.11%</b>	<b>20.1 juta</b>

Tabel 4.7 menunjukkan bahwa model dengan *shared-weights* feed-forward network mencapai akurasi 84.40% dan F1-score 84.11%, sedangkan model tanpa *shared-weights* memiliki akurasi yang lebih rendah (80.73%) dan F1-score (78.45%). Hal ini menunjukkan bahwa penggunaan *shared-weights* feed-forward network tidak hanya meningkatkan akurasi, tetapi juga memberikan peningkatan signifikan pada keseimbangan antara presisi dan recall, yang tercermin pada peningkatan F1-score. Dengan demikian, penggunaan *shared-weights* feed-forward network berkontribusi positif terhadap performa model secara keseluruhan, memungkinkan model untuk menangkap pola yang lebih kompleks dalam data dengan jumlah parameter yang lebih efisien[41].

#### 4.4.4 Layer Encoder ( $N$ )

Studi ablation pada ukuran layer encoder ( $N$ ) menunjukkan bahwa penambahan jumlah layer encoder dari 3 menjadi 6 memberikan

peningkatan performa yang signifikan. Hal ini menunjukkan bahwa model dengan lebih banyak layer dapat menangkap pola yang lebih kompleks dalam data, sehingga meningkatkan akurasi dan F1-score.

**Tabel 4.8** Perbandingan Ukuran Layer Encoder ( $N$ )

Jumlah layer encoder ( $N$ )	Akurasi	F1-Score	Parameter
3	81.65%	81.52%	<b>10.6 juta</b>
6	<b>82.57%</b>	<b>81.95%</b>	20.1 juta

Tabel 4.8 menunjukkan bahwa model dengan 6 layer encoder mencapai akurasi tertinggi (82.57%) dan F1-score (81.95%), sedangkan model dengan 3 layer encoder memiliki performa yang sedikit lebih rendah dengan akurasi 81.65% dan F1-score 81.52%. Peningkatan jumlah layer encoder memungkinkan model untuk menangkap informasi yang lebih kompleks dalam data, sehingga meningkatkan kemampuan generalisasi model terhadap data yang tidak terlihat[42].

#### 4.4.5 Dimensi Embedding ( $d_{model}$ )

Studi ablation pada ukuran embedding ( $d_{model}$ ) menunjukkan bahwa peningkatan dimensi embedding dari 512 hingga 1024 tidak memberikan peningkatan performa yang signifikan. Hal ini mengindikasikan bahwa model dengan dimensi embedding yang lebih besar tidak selalu menghasilkan performa yang lebih baik, dan ukuran 512 sudah cukup optimal untuk menangkap informasi penting dalam data.

**Tabel 4.9** Perbandingan Ukuran Embedding ( $d_{model}$ )

Ukuran Embedding ( $d_{model}$ )	Akurasi	F1-Score	Parameter
512	<b>82.57%</b>	<b>81.95%</b>	<b>20.1 juta</b>
768	77.98%	75.83%	23.8 juta
1024	77.98%	75.60%	80 juta

Tabel 4.9 menunjukkan bahwa model dengan ukuran embedding 512 mencapai akurasi tertinggi (82.57%) dan F1-score (81.95%), sedangkan model dengan ukuran embedding 1024 mengalami penurunan performa yang signifikan, dengan akurasi hanya 77.98% dan F1-score 75.60%. Hal ini menunjukkan bahwa ukuran embedding yang lebih besar tidak selalu berkontribusi positif terhadap performa model, dan dalam kasus ini, ukuran 512 sudah cukup optimal. Hal ini bisa disebabkan oleh kompleksitas model yang meningkat tanpa peningkatan informasi yang relevan, sehingga menyebabkan overfitting atau kesulitan dalam generalisasi pada data yang tidak terlihat. Selain itu, ukuran data yang terbatas juga dapat menjadi faktor yang mempengaruhi performa model dengan ukuran embedding yang lebih besar, model tidak memiliki cukup data untuk belajar pola yang kompleks secara efektif[43].

## 4.5 Prediksi Data baru

Prediksi model pada data baru (*test*) dilakukan untuk menguji model pada data yang belum pernah digunakan oleh model dalam proses pelatihan. Hal ini bertujuan untuk mengetahui performa generalisasi model pada data baru. Berikut ditampilkan hasil evaluasi *F1-score* pada 55% dan 45% data *test* pada tabel 4.10.

**Tabel 4.10** Prediksi data baru

Model	F1-score 55%	F1-score 45%
Base Model 1	79.08%	78.84%
Base Model 2	78.34%	<b>82.55%</b>
Big Model 1	<b>79.45%</b>	78.93%
Big Model 2	78.41%	80.59%

Tabel 4.10 memperlihatkan performa keempat model dalam memprediksi data test berdasarkan metrik F1-score pada dua skenario pembagian data uji 55% dan 45%. Pada skenario **55% data uji**, **Big Model 1** mencatat F1-score tertinggi sebesar **79.45%**, disusul oleh Base Model 1 (79.08%), Big Model 2 (78.41%), dan Base Model 2 (78.34%). Sementara itu, pada skenario **45% data uji**, **Base Model 2** menunjukkan performa terbaik dengan **F1-score sebesar 82.55%**, mengungguli Big Model 2 (80.59%), Base Model 1 (78.84%), dan Big Model 1 (78.93%). Hasil ini menunjukkan bahwa performa model bisa bervariasi tergantung pada proporsi data uji, dan meskipun Big Model 1 unggul saat jumlah data uji lebih besar, Base Model 2 justru lebih baik ketika jumlah data uji lebih kecil. Hal ini mengindikasikan bahwa model dengan kompleksitas lebih rendah seperti Base Model 2 memiliki generalisasi yang lebih stabil terhadap data baru dalam beberapa kondisi. Hasil evaluasi ini tidak dapat merepresentasikan model secara keseluruhan karena hanya menggunakan F1-score, diperlukan metrik evaluasi lain seperti akurasi, presisi, dan recall. Deskripsi data test dapat dilihat pada lampiran E.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis dan evaluasi model, didapatkan kesimpulan sebagai berikut:

1. Penerapan konsep *Video Vision Transformer* (ViViT) pada suara batuk penyakit paru-paru dan riwayat pasien menggunakan metode *Cross Attention* menghasilkan performa model yang baik sehingga dapat mengklasifikasikan penyakit paru-paru dengan akurat. Selain itu, *Trainable Positional Encoding* memungkinkan model untuk memahami representasi posisi secara lebih fleksibel dan sesuai dengan karakteristik data, sehingga menghasilkan performa klasifikasi yang jauh lebih baik.
2. Performa model terbaik didapatkan pada varian model Big 1 dengan blok encoder ( $N = 6$ ), Dimensi Embedding ( $d_{model} = 512$ ), head attention ( $h = 8$ ) menghasilkan akurasi 82.57%, presisi 81.79%, recall 82.48% dan F1-score 81.95% pada semua kelas.

#### **5.2 Saran**

Saran dari penulis yang dapat dipertimbangkan untuk penelitian selanjutnya adalah sebagai berikut:

1. Melakukan eksperimen dengan menggunakan ekstraksi fitur atau augmentasi suara lainnya selain MFCC seperti Linear Predictive Coding (LPC) dan Pitch-Synchronous Zero-crossing peak-amplitude (PS-ZCPA).

2. Menggunakan dataset dengan jenis penyakit paru-paru yang lebih beragam agar dapat mengakomodir berbagai kondisi penyakit paru-paru lainnya.

## DAFTAR PUSTAKA

- [1] A. Chanda dan G. Singh, “Lung tissue simulants”, di dalam *Soft Tissue Simulants*. Singapore: Springer Nature Singapore, 2024, hlmn. 59–70. sumber: [https://doi.org/10.1007/978-981-97-3060-5\\_6](https://doi.org/10.1007/978-981-97-3060-5_6).
- [2] L. F. Australia, Mei 2023. sumber: <https://lungfoundation.com.au/lung-health/lung-disease/what-is-lung-disease/>.
- [3] M. A. Fadhilah, “Chronic obstructive pulmonary disease”, *Jurnal Medika Nusantara*, vol. 2, no. 2, hlmn. 117–125, Mei 2024. sumber: <https://jurnal.stikeskesdam4dip.ac.id/index.php/Medika/article/view/1127>.
- [4] WHO, *Chronic obstructive pulmonary disease (COPD)* — *who.int*, [https://www.who.int/news-room/fact-sheets/detail/chronic-obstructive-pulmonary-disease-\(coppd\)](https://www.who.int/news-room/fact-sheets/detail/chronic-obstructive-pulmonary-disease-(coppd)), [Accessed 27-11-2024], 2024.
- [5] W. EMRO, *WHO EMRO | Chronic obstructive pulmonary disease (COPD) | Health topics* — *emro.who.int*, <https://www.emro.who.int/health-topics/chronic-obstructive-pulmonary-disease-coppd/index.html>, [Accessed 27-11-2024], 2024.
- [6] *Pasien PPOK RI 19 Juta di 2024, Diprediksi Terus Meningkat - Gaya Hidup* — *bloombergtechnoz.com*, <https://www.bloombergtechnoz.com/detail-news/55485/pasien-ppok-ri-19-juta-di-2024-diprediksi-terus-meningkat>, [Accessed 27-11-2024].
- [7] J. Heitmann, A. Gланетас, J. Doenz, dkk., “Deepbreath—automated detection of respiratory pathology from lung auscultation in 572 pediatric outpatients across 5 countries”, *NPJ digital medicine*, vol. 6, no. 1, hlmn. 104, 2023.

- [8] Y. Ma, X. Xu, dan Y. Li, “Lungrn+ nl: An improved adventitious lung sound classification using non-local block resnet neural network with mixup data augmentation.”, di dalam *Interspeech*, 2020, hlmn. 2902–2906.
- [9] K. He, X. Zhang, S. Ren, dan J. Sun, “Deep residual learning for image recognition”, di dalam *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, hlmn. 770–778.
- [10] S. Gairola, F. Tom, N. Kwatra, dan M. Jain, “Respirenet: A deep neural network for accurately detecting abnormal lung sounds in limited data setting”, di dalam *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2021, hlmn. 527–530.
- [11] B. Elizalde, S. Deshmukh, dan H. Wang, *Natural language supervision for general-purpose audio representations*, 2024. arXiv: 2309.05767 [cs.SD]. sumber: <https://arxiv.org/abs/2309.05767>.
- [12] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, dan C. Schmid, *Vivit: A video vision transformer*, 2021. arXiv: 2103.15691 [cs.CV]. sumber: <https://arxiv.org/abs/2103.15691>.
- [13] H. Lin, X. Cheng, X. Wu, dan D. Shen, “Cat: Cross attention in vision transformer”, di dalam *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022, hlmn. 1–6.
- [14] H. Xue dan F. D. Salim, “Exploring self-supervised representation ensembles for covid-19 cough classification”, di dalam *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Ser. KDD ’21,

- ACM, Agt. 2021, hlmn. 1944–1952. sumber: <http://dx.doi.org/10.1145/3447548.3467263>.
- [15] L. Xiao, L. Fang, Y. Yang, dan W. Tu, “Lungadapter: Efficient adapting audio spectrogram transformer for lung sound classification”, di dalam *Proc. Interspeech 2024*, 2024, hlmn. 4738–4742.
  - [16] V. Basu dan S. Rana, “Respiratory diseases recognition through respiratory sound with the help of deep neural network”, di dalam *2020 4th International Conference on Computational Intelligence and Networks (CINE)*, 2020, hlmn. 1–6.
  - [17] C. Barstow dan D. Forbes, “Respiratory conditions: Chronic obstructive pulmonary disease”, *FP essentials*, vol. 486, hlmn. 26–32, Nov. 2019. sumber: <http://europepmc.org/abstract/MED/31710455>.
  - [18] D. Singh, M. Miravitles, dan C. Vogelmeier, “Chronic obstructive pulmonary disease individualized therapy: Tailored approach to symptom management”, *Advances in therapy*, vol. 34, hlmn. 281–299, 2017.
  - [19] D. M. Mannino, “Chronic obstructive pulmonary disease: Epidemiology and evaluation”, *Hospital physician*, vol. 37, no. 10, hlmn. 22–40, 2001.
  - [20] L. D. Benton, “Childhood respiratory conditions: Asthma”, *FP essentials*, vol. 513, hlmn. 11–19, Feb. 2022. sumber: <http://europepmc.org/abstract/MED/35143150>.
  - [21] M. Malarvili, T. A. Howe, S. Ramanathan, M. Alexie, dan O. P. Singh, “Chapter two - asthma: The disease and issues in monitoring the asthmatic attack”, di dalam *Systems and Signal Processing of Capnography as a Diagnostic Tool for Asthma Assessment*, M. Malarvili, T. A. Howe, S. Ramanathan, M. Alexie, dan O. P. Singh, timed., Academic

- Press, 2023, hlmn. 25–50. sumber: <https://www.sciencedirect.com/science/article/pii/B9780323857475000073>.
- [22] S. M. Al Sasongko, S. Tsaury, S. Ariessaputra, dan S. Ch, “Mel frequency cepstral coefficients (mfcc) method and multiple adaline neural network model for speaker identification”, *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 4, hlmn. 2306–2312, 2023.
  - [23] J. Sueur, “Mel-frequency cepstral and linear predictive coefficients”, di dalam *Sound Analysis and Synthesis with R*. Cham: Springer International Publishing, 2018, hlmn. 381–398. sumber: [https://doi.org/10.1007/978-3-319-77647-7\\_12](https://doi.org/10.1007/978-3-319-77647-7_12).
  - [24] A. Vaswani, N. Shazeer, N. Parmar, dkk., *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. sumber: <https://arxiv.org/abs/1706.03762>.
  - [25] Y. Gong, Y.-A. Chung, dan J. Glass, *Ast: Audio spectrogram transformer*, 2021. arXiv: 2104.01778 [cs.SD]. sumber: <https://arxiv.org/abs/2104.01778>.
  - [26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, dkk., *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. sumber: <https://arxiv.org/abs/2010.11929>.
  - [27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, dan S. Zagoruyko, *End-to-end object detection with transformers*, 2020. arXiv: 2005.12872 [cs.CV]. sumber: <https://arxiv.org/abs/2005.12872>.
  - [28] H. Lu, G. Yang, N. Fei, dkk., *Vdt: General-purpose video diffusion transformers via mask modeling*, 2023. arXiv: 2305.13311 [cs.CV]. sumber: <https://arxiv.org/abs/2305.13311>.
  - [29] J. Devlin, M.-W. Chang, K. Lee, dan K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language*

- understanding*, 2019. arXiv: 1810.04805 [cs.CL]. sumber: <https://arxiv.org/abs/1810.04805>.
- [30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, dkk., “Language models are unsupervised multitask learners”, *OpenAI blog*, vol. 1, no. 8, hlmn. 9, 2019.
  - [31] H. Kaushik, *Medical sound classification challenge*, <https://kaggle.com/competitions/airs-ai-in-respiratory-sounds>, Kaggle, 2024.
  - [32] M. Avadhani, Jahnavi, A. P. Bidargaddi, dan T. S, “Multi-class urban sound classification with deep learning architectures”, di dalam *2024 5th International Conference for Emerging Technology (INCET)*, 2024, hlmn. 1–7.
  - [33] P. Xu, X. Zhu, dan D. A. Clifton, *Multimodal learning with transformers: A survey*, 2023. arXiv: 2206.06488 [cs.CV]. sumber: <https://arxiv.org/abs/2206.06488>.
  - [34] M. Gheini, X. Ren, dan J. May, *Cross-attention is all you need: Adapting pretrained transformers for machine translation*, 2021. arXiv: 2104.08771 [cs.CL]. sumber: <https://arxiv.org/abs/2104.08771>.
  - [35] K. Ahn, Z. Zhang, Y. Kook, dan Y. Dai, *Understanding adam optimizer via online learning of updates: Adam is ftrl in disguise*, 2024. arXiv: 2402.01567 [cs.LG]. sumber: <https://arxiv.org/abs/2402.01567>.
  - [36] E. Manai, M. Mejri, dan J. Fattahi, “Confusion matrix explainability to improve model performance: Application to network intrusion detection”, di dalam *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2024, hlmn. 1–5.
  - [37] J. Li, “Area under the roc curve has the most consistent evaluation for binary classification”, *PLOS ONE*, vol. 19,

- no. 12, H. Qin, Ed., e0316019, Des. 2024. sumber: <http://dx.doi.org/10.1371/journal.pone.0316019>.
- [38] A. D. Rahajoe, Agussalim, R. Mumpuni, dkk., “Optimization of binary classification based on receiver operating characteristic area under the curve for supervised machine learning”, di dalam *2023 IEEE 9th Information Technology International Seminar (ITIS)*, 2023, hlmn. 1–6.
  - [39] T. Schiffer, S. Korn, L. Cervis, dan R. Buhl, “COPD and the asthma-copd-overlap-syndrome”, *European Respiratory Journal*, vol. 44, no. Suppl 58, 2014. eprint: [https://publications.ersnet.org//content/erj/44/Suppl\\_58/P1762.full.pdf](https://publications.ersnet.org//content/erj/44/Suppl_58/P1762.full.pdf). sumber: [https://publications.ersnet.org//content/erj/44/Suppl\\_58/P1762](https://publications.ersnet.org//content/erj/44/Suppl_58/P1762).
  - [40] N. Barnes, L. A. Hunt, dan M. Mayo, “Classification models in intensive care outcome prediction-can we improve on current models?”, di dalam *AIH+ CARE@ AUS-AI*, Citeseer, 2013, hlmn. 5–21.
  - [41] O. Chang dan H. Lipson, *Balanced and deterministic weight-sharing helps network performance*, 2023. arXiv: 2312.08401 [cs.LG]. sumber: <https://arxiv.org/abs/2312.08401>.
  - [42] X. Liu, K. Duh, L. Liu, dan J. Gao, *Very deep transformers for neural machine translation*, 2020. arXiv: 2008.07772 [cs.CL]. sumber: <https://arxiv.org/abs/2008.07772>.
  - [43] L. Brigato dan L. Iocchi, “A close look at deep learning with small data”, di dalam *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, hlmn. 2490–2497.
  - [44] K. S. Rao dan K. Manjunath, *Speech recognition using articulatory and excitation source features*. Springer, 2017.
  - [45] H. A. Feldman, N. Kaiser, dan J. A. Peacock, “Power spectrum analysis of three-dimensional redshift surveys”, *arXiv preprint astro-ph/9304022*, 1993.

- [46] H. Yin, V. Hohmann, dan C. Nadeu, “Acoustic features for speech recognition based on gammatone filterbank and instantaneous frequency”, *Speech Communication*, vol. 53, no. 5, hlmn. 707–715, 2011, Perceptual and Statistical Audition. sumber: <https://www.sciencedirect.com/science/article/pii/S01676393100009%2019>.
- [47] G. Strang, “The discrete cosine transform”, *SIAM review*, vol. 41, no. 1, hlmn. 135–147, 1999.
- [48] D. Hendrycks dan K. Gimpel, *Gaussian error linear units (gelus)*, 2023. arXiv: 1606.08415 [cs.LG]. sumber: <https://arxiv.org/abs/1606.08415>.

# LAMPIRAN

## LAMPIRAN A

### *Mel frequency Capstral Coefficients (MFCC)*

#### A.1 Pre-Emphasis

Pre-Emphasis merupakan salah satu praktik pra-pemrosesan umum dalam bidang pemrosesan sinyal yang digunakan untuk mengompensasi frekuensi tinggi sinyal yang ditekan selama produksi sinyal. Pra-penekanan merupakan langkah pertama selama adaptasi MFCC, yang dapat diadopsi hanya dengan menerapkan filter high-pass dengan pengaturan  $[1, -0.97]$ . Proses penyaringan mengubah distribusi energi di seluruh frekuensi, serta tingkat energi keseluruhan. Formula Pre-Emphasis dapat dilihat pada persamaan A.1.

$$\hat{y}[n] = x[n] - \alpha \cdot x[n - 1] \quad (\text{A.1})$$

Keterangan:

$x[n]$  : sinyal input

$\hat{y}[n]$  : sinyal keluaran

$\alpha$  : faktor pre-emphasis

Keluaran  $\hat{y}[n]$  akan menjadi seperti persamaan A.2

$$\{x(0), x(1) - \alpha x(0), \dots, x(n - 1) - \alpha x(n - 2)\} \quad (\text{A.2})$$

#### A.2 Framing dan Windowing

Ide di balik pemisahan sinyal menjadi "frame" yang berbeda adalah memecah sinyal data mentah menjadi frame yang sinyalnya cenderung lebih stasioner. Untuk karakteristik akustik yang stabil, suara perlu diperiksa dalam jangka waktu yang cukup singkat. Oleh karena itu, pengukuran spektral jangka pendek biasanya dilakukan

selama jendela 20 ms, dan setiap bingkai tumpang tindih 10 ms dengan bingkai berikutnya. Tumpang tindih bingkai sebesar 10 ms memungkinkan karakteristik temporal sinyal audio dilacak. Dengan tumpang tindih bingkai audio, representasi suara akan kira-kira terpusat pada beberapa bingkai. Pada setiap frame, jendela diterapkan untuk mempersempit sinyal ke arah batas frame. Secara umum, jendela Hanning dan Hamming [44] adalah salah satu metode yang paling banyak digunakan. Jendela ini dapat meningkatkan harmonik, menghaluskan tepi, dan mengurangi efek tepi saat melakukan DFT pada sinyal. Framing dan windowing diterapkan dengan persamaan A.3.

$$x_n[m] = x[n] \cdot w[n] \quad (\text{A.3})$$

Keterangan:

$w[n]$  : fungsi windowing

$x_n[m]$  : hasil windowing

$x[n]$  : sinyal input

### A.3 Discrete Fourier Transform (DFT)

*Discrete Fourier Transform* (DFT) banyak digunakan untuk menghitung spektrum daya. Spektrum daya dapat dideskripsikan sebagai distribusi dari daya pada komponen frekuensi pada sinyal [45]. Spektrum daya masing-masing frame dapat ditentukan dengan persamaan A.4.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \quad (\text{A.4})$$

Keterangan:

$j$  : bilangan imajiner

$N$  : panjang sinyal

$x[n]$  : sinyal input

dengan  $x(n)$  adalah sinyal diskrit dan  $N$  adalah panjang dari sinyal.

#### A.4 Mel-Frequency Filter Bank

*Mel-Frequency Filter Bank* adalah bank filter yang dibangun berdasarkan persepsi nada. Filter Mel awalnya dikembangkan untuk *speech recognition* dan seperti persepsi telinga manusia terhadap ucapan, filter ini menargetkan ekstraksi representasi nonlinier dari sinyal ucapan. *Mel-Frequency Filter Bank* konvensional dibangun dari 40 filter segitiga [46]. Fungsi transfer (TF) dari masing-masing filter ke- $m$  dapat dihitung melalui persamaan A.5,

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k < f(m) \\ 1 & k = f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) < k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (\text{A.5})$$

dengan  $f(m)$  adalah pusat frekuensi dari filter segitiga dan  $\sum_m^{M-1} H_m(k) = 1$ . Skala Mel terhadap frekuensi respons dan sebaliknya dihitung dengan persamaan A.6 dan A.7.

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (\text{A.6})$$

$$f = 700 \left( 10^{\frac{m}{2595}} - 1 \right) \quad (\text{A.7})$$

Keterangan:

- $f$  adalah frekuensi
- $m$  adalah skala mel

### A.5 Discrete Cosine Transform (DCT)

*Discrete Cosine Transform* (DCT) menyatakan *finite sequence* dari titik data mengenai penjumlahan fungsi kosinus yang berosilasi pada frekuensi yang berbeda. DCT diperkenalkan oleh Nasir Ahmed pada tahun 1972. Dalam proses MFCC, DCT diterapkan pada bank filter Mel untuk memilih koefisien yang paling akurat atau untuk memisahkan hubungan dalam besaran spektral logaritma dari bank filter [47]. DCT dihitung dengan persamaan A.8

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi j nk}{N}\right), \quad k = 0, 1, \dots, N - 1 \quad (\text{A.8})$$

Keterangan:

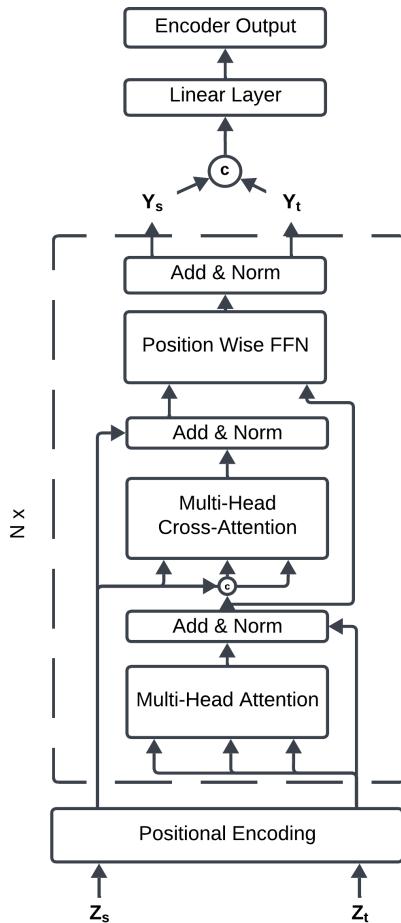
$x_n$  : sinyal diskrit

$N$  : panjang sinyal

$X_k$  : koefisien MFCC

## LAMPIRAN B

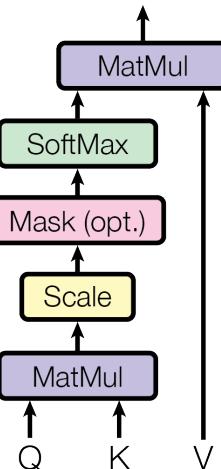
### Perhitungan *Encoder Cross-Attention*



Gambar B.1 Encoder Cross Attention

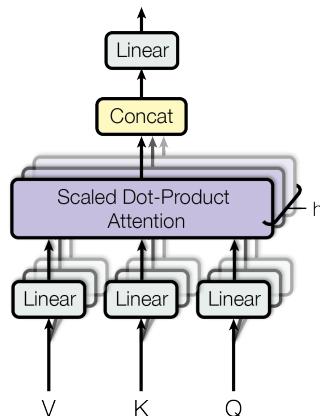
## B.1 *Self-Attention* dan *Multi-Head Attention*

Proses dari *Self-Attention* atau yang memiliki nama lain *Scaled Dot-Product Attention* dapat dilihat pada Gambar B.2.



**Gambar B.2** *Self-Attention* atau *Scaled Dot-Product Attention*

Proses dari *Multi-Head Attention* dapat dilihat pada Gambar B.3.



**Gambar B.3** *Multi-Head Attention*

Inisiasi nilai variabel diperlukan untuk penentuan nilai awal pada pelatihan. Nilai yang digunakan adalah nilai yang sama dengan paper "*Attention Is All You Need*" dengan beberapa penyesuaian [24]. Nilai variabel inisiasi dapat dilihat pada tabel B.1.

**Tabel B.1** Tabel Variabel untuk Encoder Transformer Model Big 1

<b>Nama Variabel</b>	<b>Nilai Variabel</b>
N	6
SEQ_LENGTH	512
NUM_FEATURE	8
D_MODELS	512
NUM_HEADS	8
DROPOUT_RATE	0.2
EPOCHS	100

Matriks fitur riwayat pasien  $Z_t$  berukuran  $8 \times 1$  dimasukkan ke dalam layer linier agar memiliki dimensi yang sama dengan matriks hasil ekstraksi fitur MFCC sehingga memiliki ukuran  $8 \times 512$ . Matriks fitur suara  $Z_s$  hasil ekstraksi fitur MFCC berukuran  $512 \times 512$  dan matriks fitur riwayat pasien berukuran  $8 \times 512$  ditambahkan dengan token [cls] menjadi masing-masing berukuran  $513 \times 512$  dan  $9 \times 512$ .

Matriks fitur riwayat pasien lalu masuk ke dalam mekanisme *Multi-Head Attention* dan *Layer Norm* dengan hasil matriks berukuran  $9 \times 512$ . Matriks ini selanjutnya di *concat* dengan matriks fitur suara yang telah ditambah token [cls] berukuran  $513 \times 512$  sehingga menghasilkan matriks berukuran  $521 \times 512$ .

Matriks fitur suara yang telah di *concat* dengan matriks fitur riwayat pasien yang telah melalui mekanisme *Multi-Head Attention* lalu masuk ke dalam mekanisme *Multi-Head Cross Attention*. Operasi yang digunakan pada *Multi-Head Cross Attention* sama dengan

*Multi-Head Attention*, perbedaannya terletak pada input berupa gabungan antara fitur suara dan fitur riwayat pasien. Setelah melewati *Layer Norm* ukuran matriks hasil *Multi-Head Cross Attention* menjadi  $513 \times 512$ .

## B.2 Feed-Forward Network

Matriks hasil *Multi-Head Attention* dan *Multi-Head Cross Attention* masing-masing masuk ke dalam mekanisme *Feed-Forward Network*. mekanisme *Feed-Forward Network* terdiri dari tiga operasi utama yaitu *Linear Transformation*, *Activation Function* dan *Layer Normalization*. Transformasi Linear diimplementasikan sebagai *fully connected layer*, atau juga dikenal sebagai *dense layer*, yang menghubungkan setiap neuron masukan ke setiap neuron keluaran. Langkah berikutnya dalam operasi *Feed-Forward Network* adalah menerapkan fungsi aktivasi. Fungsi ini merupakan fungsi nonlinier yang memungkinkannya mempelajari pola yang lebih kompleks. Fungsi aktivasi yang digunakan pada model ini adalah GELU (*Gaussian Error Linear Unit*) yang memiliki kecepatan dan konvergensi yang lebih baik karena telah menggabungkan properti dari dropout, zoneout, dan ReLUs [48]. Persamaan fungsi aktivasi GELU dapat dilihat pada persamaan B.1.

$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2} \left[ 1 + \text{erf}(x/\sqrt{2}) \right] \quad (\text{B.1})$$

Fungsi aktivasi GELU dapat diaproksimasi dengan persamaan B.2

$$0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)]) \quad (\text{B.2})$$

atau persamaan B.3.

$$x\sigma(1.702x) \quad (\text{B.3})$$

Tahap terakhir pada *Feed-Forward Network* adalah *Layer Normalization*. *Layer Normalization* adalah teknik yang menormalkan masukan di seluruh dimensi fitur (bukan dimensi batch), menstabilkan jaringan dan mempercepat pelatihan. Output dari *Feed-Forward Network* berupa kedua matriks dengan ukuran  $513 \times 512$  dan  $9 \times 512$ .

### B.3 Output Encoder $Y_s$ dan $Y_t$

Proses dari keseluruhan di atas merupakan proses dalam satu blok encoder, proses ini diulang sebanyak N\_BLOCK. Didapatkan vektor  $Y_s$  dan  $Y_t$  dengan ukuran  $1 \times 512$  yang kemudian di *concat* secara horizontal/kolom sehingga didapatkan vektor berukuran  $1 \times 1024$ . Vektor ini kemudian masuk ke dalam fungsi aktivasi GELU dan layer Linear sehingga didapatkan vektor berukuran  $1 \times 512$ . Fungsi aktivasi Softmax digunakan untuk mengakomodir klasifikasi multi-kelas dengan menghasilkan probabilitas setiap kelas dengan jumlah semua probabilitas berjumlah tepat 1. Fungsi aktivasi Softmax dapat dilihat pada persamaan B.4.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad \text{for } i = 1, \dots, K \quad (\text{B.4})$$

Keterangan:

$\sigma$  : fungsi softmax

$\mathbf{z}$  : input vektor

$K$  : jumlah kelas

$e^{z_i}$  : fungsi eksponensial untuk vektor input

$e^{z_j}$  : fungsi eksponensial untuk vektor output

Fungsi B.4 menghasilkan vektor  $1 \times 3$  yang berisi nilai probabilitas dari ketiga kelas. Nilai probabilitas terbesar menunjukkan prediksi kelas dari data.

## LAMPIRAN C

### *Optimizer Adam*

#### C.1 Optimizer Adam pada Transformers

Adam menggabungkan kelebihan dari dua algoritma optimisasi, yaitu momentum dan RMSProp, dengan mengestimasi rata-rata dan varians dari gradien. Dalam model Transformers, parameter  $\theta$  mencakup bobot dan bias untuk lapisan *self-attention* dan *multi-head attention* (seperti bobot proyeksi query, key dan value), *Feed-Forward Network* dan parameter *Layer Normalization*. Setiap parameter diperbarui menggunakan persamaan C.2

#### C.2 Persamaan Oprimizer Adam

1. Menghitung gradien dari loss  $L$  dengan parameter  $\theta$  dengan persamaan C.1:

$$g_t = \nabla_{\theta} L(\theta_t) \quad (\text{C.1})$$

2. Perbarui estimasi momentum bias dengan persamaan C.2 dan C.3:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (\text{C.2})$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (\text{C.3})$$

3. Terapkan koreksi bias pada momentum dengan persamaan C.4:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (\text{C.4})$$

4. Perbarui parameter dengan persamaan C.5:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (\text{C.5})$$

Keterangan:

$g_t$  : fungsi gradien

$L(\theta_t)$  : fungsi Loss

$m_t$  : momentum

$v_t$  : velocity

$\eta$  : learning rate

$\beta_1, \beta_2$  : hyperparameter peluruhan momen

$\epsilon$  : konstanta kecil untuk mencegah pembagian dengan nol

### C.3 Contoh Perhitungan

Asumsikan :

- Learning rate  $\eta = 0.001$ ,
- $\beta_1 = 0.9, \beta_2 = 0.999$ ,
- $\epsilon = 10^{-8}$ ,
- Gradien pada timestep  $t$ :  $g_t = 0.02$ ,
- Estimasi momentum sebelumnya:  $m_{t-1} = 0.01, v_{t-1} = 0.0001$ ,
- $t = 10$ .

Langkah-langkah perhitungan:

1. Hitung estimasi momen bias:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t = 0.9 \cdot 0.01 + (1 - 0.9) \cdot 0.02 = 0.011$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 = 0.999 \cdot 0.0001 + (1 - 0.999) \cdot (0.02)^2 = 0.0001004$$

2. Terapkan koreksi bias:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} = \frac{0.011}{1 - 0.9^{10}} \approx 0.0111$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} = \frac{0.0001004}{1 - 0.999^{10}} \approx 0.000102$$

3. Hitung pembaruan parameter:

$$\Delta\theta = \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Substitusikan nilai:

$$\Delta\theta = 0.001 \cdot \frac{0.0111}{\sqrt{0.000102} + 10^{-8}} \approx 0.00109$$

4. Perbarui parameter:

$$\theta_{t+1} = \theta_t - \Delta\theta$$

## LAMPIRAN D

### Rangkuman Model

#### D.1 Model Base 1

Tabel D.1 Rangkuman Encoder Block Model Base 1

Layer ( <i>Type</i> )	Bentuk Output	Parameter
(Dense)	(None, 512, 256)	5,376
(Dense)	(None, 512, 512)	131,584
(Dense)	(None, 8, 512)	1,024
(PositionalEncoding)	(None, 9, 512)	4,608
(PositionalEncoding)	(None, 513, 512)	262,656
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 512)	524,800

*Dilanjutkan pada halaman berikutnya*

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(Dense)	(None, 512)	524,800
<b>Total parameter</b>		<b>10,394,368</b>

**Tabel D.2** Rangkuman Classifier Model Base 1

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 512)	262,656
(Dense)	(None, 3)	1,539
<b>Total parameter</b>		<b>264,195</b>

**Tabel D.3** Rangkuman Model Base 1

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
TransformerEncoderBlock	(None, 512)	10,394,368
Classifier	(None, 3)	264,195
<b>Total parameter</b>		<b>10,658,563</b>

## D.2 Model Base 2

**Tabel D.4** Rangkuman Encoder Block Model Base 2

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 768, 256)	5,376
(Dense)	(None, 768, 768)	197,376
(Dense)	(None, 8, 768)	1,536
(PositionalEncoding)	(None, 9, 768)	6,912
(PositionalEncoding)	(None, 769, 768)	590,592
(MultiHeadAttention)	(None, 9, 768)	2,362,368
(MultiHeadAttention)	(None, 9, 768)	2,362,368
(MultiHeadAttention)	(None, 9, 768)	2,362,368
(MultiHeadAttention)	(None, 769, 768)	2,362,368
(MultiHeadAttention)	(None, 769, 768)	2,362,368
(MultiHeadAttention)	(None, 769, 768)	2,362,368
(LayerNormalization)	(None, 9, 768)	1,536
(LayerNormalization)	(None, 9, 768)	1,536
(LayerNormalization)	(None, 9, 768)	1,536
(LayerNormalization)	(None, 769, 768)	1,536
(LayerNormalization)	(None, 769, 768)	1,536
(LayerNormalization)	(None, 769, 768)	1,536
(Dense)	(None, 769, 1536)	1,181,184
(Dense)	(None, 769, 1536)	1,181,184
(Dense)	(None, 769, 1536)	1,181,184
(Dense)	(None, 769, 768)	1,180,416
(Dense)	(None, 769, 768)	1,180,416
(Dense)	(None, 769, 768)	1,180,416
(LayerNormalization)	(None, 769, 768)	1,536
(LayerNormalization)	(None, 769, 768)	1,536
(LayerNormalization)	(None, 769, 768)	1,536
(Dense)	(None, 768)	1,180,416

*Dilanjutkan pada halaman berikutnya*

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
<b>Total parameter</b>		<b>23,256,576</b>

**Tabel D.5** Rangkuman Model Classifier

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 768)	590,592
(Dense)	(None, 3)	2,307
<b>Total parameter</b>		<b>592,899</b>

**Tabel D.6** Rangkuman Model Base 2

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
TransformerEncoderBlock	(None, 768)	23,256,576
Classifier	(None, 3)	592,899
<b>Total parameter</b>		<b>23,849,475</b>

### D.3 Model Big 1

**Tabel D.7** Rangkuman Encoder Block Model Big 1

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 512, 256)	5,376
(Dense)	(None, 512, 512)	131,584
(Dense)	(None, 8, 512)	1,024
(PositionalEncoding)	(None, 9, 512)	4,608
(PositionalEncoding)	(None, 513, 512)	262,656
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624

*Dilanjutkan pada halaman berikutnya*

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 9, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(MultiHeadAttention)	(None, 513, 512)	1,050,624
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 9, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312
(Dense)	(None, 513, 1024)	525,312

*Dilanjutkan pada halaman berikutnya*

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(Dense)	(None, 513, 512)	524,800
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(LayerNormalization)	(None, 513, 512)	1,024
(Dense)	(None, 512)	524,800
<b>Total parameter</b>		<b>19,857,664</b>

**Tabel D.8** Rangkuman Classifier Model Big 1

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 512)	262,656
(Dense)	(None, 3)	1,539
<b>Total parameter</b>		<b>264,195</b>

**Tabel D.9** Rangkuman Model Big 1

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
TransformerEncoderBlock	(None, 512)	19,857,664
Classifier	(None, 3)	264,195
<b>Total parameter</b>		<b>20,121,859</b>

#### D.4 Model Big 2

**Tabel D.10** Rangkuman Encoder Block Model Big 2

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 1024, 256)	5,376
(Dense)	(None, 1024, 1024)	263,168
(Dense)	(None, 8, 1024)	2,048
(PositionalEncoding)	(None, 9, 1024)	9,216
(PositionalEncoding)	(None, 1025, 1024)	1,049,600
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 9, 1024)	4,198,400
(MultiHeadAttention)	(None, 1025, 1024)	4,198,400
(MultiHeadAttention)	(None, 1025, 1024)	4,198,400
(MultiHeadAttention)	(None, 1025, 1024)	4,198,400
(MultiHeadAttention)	(None, 1025, 1024)	4,198,400
(MultiHeadAttention)	(None, 1025, 1024)	4,198,400
(LayerNormalization)	(None, 9, 1024)	2,048
(LayerNormalization)	(None, 9, 1024)	2,048
(LayerNormalization)	(None, 9, 1024)	2,048
(LayerNormalization)	(None, 9, 1024)	2,048
(LayerNormalization)	(None, 9, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048

*Dilanjutkan pada halaman berikutnya*

<b>Layer (<i>Type</i>)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 2048)	2,099,200
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(Dense)	(None, 1025, 1024)	2,098,176
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(LayerNormalization)	(None, 1025, 1024)	2,048
(Dense)	(None, 1024)	2,098,176
<b>Total parameter</b>		<b>79,031,552</b>

**Tabel D.11** Rangkuman Classifier Model Big 2

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
(Dense)	(None, 1024)	1,049,600
(Dense)	(None, 3)	3,075
<b>Total parameter</b>		<b>1,052,675</b>

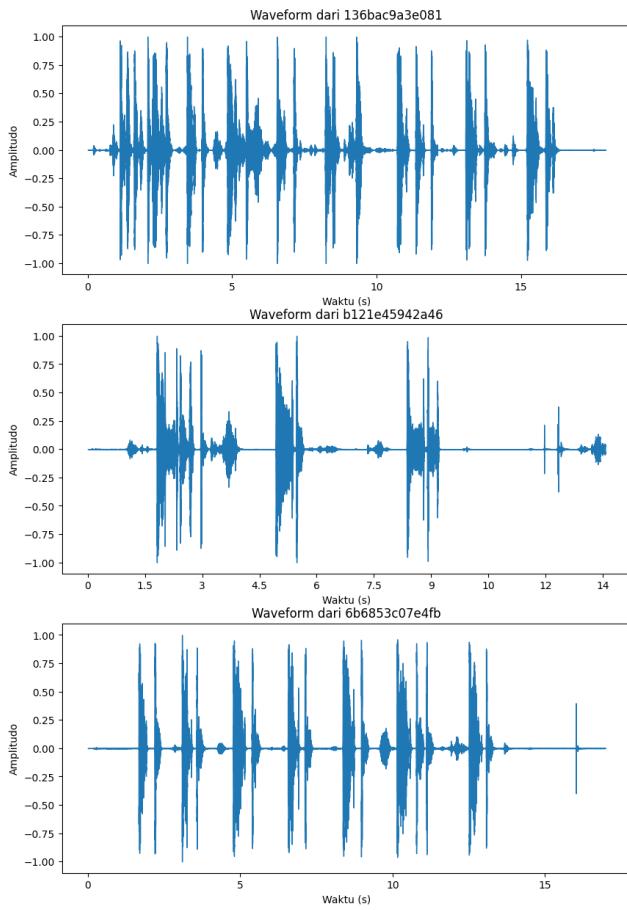
**Tabel D.12** Rangkuman Model Big 2

<b>Layer (Type)</b>	<b>Bentuk Output</b>	<b>Parameter</b>
TransformerEncoderBlock	(None, 1024)	79,031,552
Classifier	(None, 3)	1,052,675
<b>Total parameter</b>		<b>80,084,227</b>

## **LAMPIRAN E**

### **Data Test**

Data test berisi data suara dan riwayat pasien yang terdiri dari 338 entri, masing-masing mewakili satu individu. Setiap entri mencakup atribut demografis dan klinis yang relevan dengan kondisi pernapasan atau penyakit paru. Atribut yang tersedia meliputi: `candidateID` sebagai pengenal unik pasien, `age` (usia), `gender` (jenis kelamin), serta sejumlah faktor risiko seperti `tbContactHistory` (riwayat kontak dengan penderita tuberkulosis), `wheezingHistory` (riwayat mengi), `phlegmCough` (batuk berdahak), `familyAsthmaHistory` (riwayat asma dalam keluarga), `feverHistory` (riwayat demam), `coldPresent` (gejala flu saat ini), dan `packYears` (jumlah tahun konsumsi rokok dalam satuan "pack-years"). Data test akan digunakan untuk memprediksi `disease` menggunakan keempat model, hasilnya dapat dilihat pada tabel 4.10.



**Gambar E.1** Sampel audio data test

**Tabel E.1** Tabel riwayat pasien data test

<b>candidateID</b>	<b>age</b>	<b>gender</b>	<b>tbContact</b>	<b>wheezing</b>	<b>phlegm</b>	<b>Cough</b>	<b>familyAsthma</b>	<b>feverHistory</b>	<b>packYears</b>
136bac9a3e081	23	0	0.0	0.0	0.0	0.0	0	9	
b121e45942a46	26	1	0.0	0.0	0.0	1.0	0	0	
6b6853c07e4fb	26	1	1.0	0.0	0.0	0.0	1	0	
71de185eac888	55	0	0.0	0.0	0.0	0.0	1	0	
25deed742f133	16	0	0.0	0.0	0.0	0.0	1	0	
1de4591779d31	43	1	0.0	0.0	0.0	0.0	1	0	
102efeabb10a5	24	1	0.0	1.0	0.0	0.0	0	0	
522d1f8600a13	63	1	0.0	0.0	0.0	0.0	0	0	
e41530046a74e	30	0	0.0	1.0	0.0	0.0	0	0	
6337b96a160eb	42	1	0.0	0.0	0.0	0.0	0	0	
911b03c5b0338	63	1	0.0	0.0	0.0	0.0	0	0	
2a86de00c05a2	62	0	0.0	0.0	0.0	0.0	0	9	
...	...	...	...	...	...	...	...	...	...
56baf88e9259b	65	0	0.0	0.0	1.0	0.0	0	0	