

**ANALISIS PERBANDINGAN PERFORMA ARSITEKTUR  
PANNS DENGAN PENDEKATAN INPUT BERBASIS *RAW*  
*WAVEFORM*, *LOG-MEL SPECTROGRAM*, DAN *HYBRID* UNTUK  
KLASIFIKASI SUARA BAHAYA**

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di  
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Institut  
Teknologi Sumatera

**Oleh:**

**Ramon Riping**

**122140078**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN  
2026**

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	<b>ii</b>
<b>DAFTAR TABEL</b> .....	<b>iii</b>
<b>DAFTAR GAMBAR</b> .....	<b>iv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian .....	5
1.4 Batasan Masalah .....	5
1.5 Manfaat Penelitian .....	5
1.6 Sistematika Penulisan .....	6
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>8</b>
<b>BAB III METODE PENELITIAN</b> .....	<b>9</b>
3.1 Alur Penelitian .....	9
3.2 Pengumpulan dan Pra-pemrosesan Data .....	12
3.2.1 Seleksi Kelas ( <i>Class Filtering</i> ) .....	12
3.2.2 Strategi Penyusunan Ulang Data ( <i>Fold Mapping</i> ) .....	13
3.2.3 Standarisasi Format Audio .....	14
3.2.4 Augmentasi Temporal ( <i>Random Cropping</i> ) .....	15
3.3 Konfigurasi Model .....	16
3.3.1 Modifikasi Arsitektur PANNs .....	17
3.3.2 Strategi <i>Freeze Base</i> .....	17
3.3.3 Parameter Konfigurasi Input .....	18
3.4 Eksperimen <i>Transfer Learning</i> .....	19
3.4.1 Lingkungan Implementasi dan Reprodusibilitas .....	19

3.4.2	Penanganan Ketidakseimbangan Data ( <i>Weight Penalty</i> ) . . . .	21
3.4.3	Konfigurasi <i>Optimizer</i> dan <i>Scheduler</i> . . . . .	22
3.4.4	Strategi Optimasi: <i>Early Stopping</i> dan <i>Checkpoint</i> . . . . .	23
3.4.5	Pemantauan dan Pencatatan Eksperimen . . . . .	23
3.4.6	Konfigurasi Parameter Pelatihan . . . . .	24
3.5	Analisis dan Evaluasi . . . . .	25
<b>DAFTAR PUSTAKA . . . . .</b>		<b>27</b>

## DAFTAR TABEL

Tabel 3.1	Konfigurasi Parameter Input Audio Model PANNs . . . . .	18
Tabel 3.2	Spesifikasi Perangkat Keras untuk Pelatihan . . . . .	20
Tabel 3.3	Daftar Perangkat Lunak dan Pustaka Pendukung . . . . .	20
Tabel 3.4	Parameter Konfigurasi Pelatihan . . . . .	25

## DAFTAR GAMBAR

Gambar 3.1	Diagram Alur Penelitian .....	9
Gambar 3.2	Ilustrasi Strategi Penyusunan Ulang Fold pada Dataset UrbanSound8K .....	14
Gambar 3.3	Ilustrasi Teknik <i>Random Cropping</i> Pada Setiap <i>Epoch</i> ...	16
Gambar 3.4	Ilustrasi <i>Transfer Learning</i> ( <i>Freeze</i> vs <i>Trainable Base</i> ) ..	18

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Keselamatan berlalu lintas di lingkungan perkotaan merupakan tantangan krusial yang dihadapi masyarakat saat ini, terutama bagi kelompok rentan seperti penyandang disabilitas [1]. Dalam aktivitas tersebut, indra pendengaran berperan sebagai mekanisme deteksi alami yang penting untuk mengetahui kondisi lingkungan di sekitar. Namun, fungsi indra tersebut tidak dimiliki oleh penyandang tunarungu, yang hanya bisa mengandalkan penglihatan mereka untuk memantau keadaan. Ketergantungan penuh pada aspek visual ini dapat menjadi kerentanan serius, mengingat mereka memiliki keterbatasan sudut pandang dan tidak dapat memantau kondisi di luar jangkauan penglihatan. Akibatnya, ancaman yang muncul dari titik buta (*blind spot*), seperti gonggongan dari anjing yang mengejar atau klakson kendaraan yang melaju kencang dari arah belakang, seringkali terlambat disadari akibat tidak adanya peringatan suara. Keterlambatan respon inilah yang secara signifikan meningkatkan risiko terjadinya kecelakaan fatal [2]. Maka dari itu, diperlukan mekanisme bantu yang dapat menggantikan peran indra pendengaran dalam mendeteksi ancaman yang muncul dari luar jangkauan visual.

Saat ini, Alat Bantu Dengar (ABD) merupakan perangkat yang umum digunakan untuk menunjang komunikasi verbal penyandang tunarungu. Meskipun efektif untuk komunikasi verbal jarak dekat, alat ini memiliki keterbatasan signifikan dalam konteks keselamatan di luar ruangan. Hal ini disebabkan oleh penurunan selektivitas frekuensi (*reduced frequency selectivity*) yang umum terjadi pada gangguan pendengaran sensorineural, sehingga menyulitkan pemisahan sinyal suara utama dari kebisingan latar belakang yang tumpang tindih [3]. Kondisi ini diperburuk oleh keterbatasan

teknis ABD, di mana sekadar amplifikasi sinyal suara tidak cukup untuk mengembalikan kemampuan pemilahan suara secara alami. Akibatnya, sinyal ancaman penting seringkali tertutup oleh suara-suara lainnya, yang berdampak pada hilangnya kewaspadaan situasional pengguna. Keterbatasan perangkat keras dalam memilah sinyal suara ini memunculkan kebutuhan teknologi bagi penyandang tunarungu agar dapat mengidentifikasi suara bahaya melalui pola sinyal suara, dan bukan sekadar amplifikasi sinyal.

Guna mengatasi keterbatasan ini, dikembangkanlah metode cerdas yang dikenal sebagai Klasifikasi Suara Lingkungan atau *Environmental Sound Classification* (ESC). Integrasi teknologi ini pada alat bantu dengar telah lama diteliti sebagai upaya meningkatkan kesadaran situasi pengguna [4]. Pada tahap awal pengembangannya, sistem ESC umumnya dibangun menggunakan metode *Machine Learning* konvensional seperti *Support Vector Machine* (SVM) atau *Random Forest* [5]. Namun, metode-metode klasik tersebut sangat bergantung pada proses ekstraksi fitur secara manual (*hand-crafted features*) yang kaku, sehingga performanya cenderung menurun drastis ketika dihadapkan dengan variasi kebisingan lingkungan yang dinamis. Kelemahan metode tersebut memicu pergeseran tren penelitian menuju pendekatan *Deep Learning*, khususnya *Convolutional Neural Networks* (CNN) yang menawarkan kemampuan untuk mempelajari fitur suara secara otomatis dan hirarkis langsung dari data [6]. Kemampuan adaptasi fitur inilah yang menjadikannya sebagai solusi yang jauh lebih andal dibandingkan metode konvensional. Walaupun menjanjikan akurasi yang lebih tinggi, metode *Deep Learning* membutuhkan dataset berskala masif untuk melatih fitur-fitur tersebut secara efektif. Ketergantungan ini menjadi kendala signifikan pada kasus dengan ketersediaan data yang terbatas, sehingga diperlukan strategi pembelajaran khusus agar model tetap memiliki performa yang *robust*.

Sebagai implementasi strategi tersebut, metode *Transfer Learning* menjadi solusi efektif untuk mengatasi kelangkaan data. Pendekatan ini memanfaatkan

*Pre-trained Audio Neural Networks* (PANNs) [7], yaitu sebuah kerangka kerja model *Deep Learning* skala besar yang telah dilatih sebelumnya (*pre-trained*) pada dataset AudioSet [8] untuk mengenali berbagai pola suara umum. Salah satu keunggulan PANNs terletak pada variasi arsitektur yang dirancang khusus untuk menangani dua jenis representasi input audio yang berbeda. Pertama adalah arsitektur berbasis satu dimensi yang mengolah *Raw Waveform*, yaitu sinyal gelombang suara mentah dalam domain waktu. Kedua adalah arsitektur berbasis dua dimensi yang memanfaatkan *Log-mel Spectrogram*, yaitu representasi visual yang memetakan intensitas energi frekuensi suara layaknya sebuah citra gambar. Selain itu, PANNs juga menyediakan arsitektur dengan pendekatan *Hybrid* yang menggabungkan kedua representasi tersebut, yang secara teoritis berpotensi memaksimalkan akurasi deteksi. Ketersediaan variasi ini memunculkan urgensi untuk mengevaluasi arsitektur mana yang paling optimal untuk diterapkan pada kasus ini, apakah berbasis domain waktu, domain frekuensi, atau penggabungan keduanya (*Hybrid*).

Meskipun Kong et al. [7] telah memaparkan tolak ukur kinerja model-model tersebut pada dataset masif AudioSet [8], performa tersebut belum tentu sebanding ketika diterapkan pada kasus penerapan spesifik dengan ketersediaan data yang terbatas (*data scarcity*) seperti pada kasus klasifikasi suara lingkungan perkotaan. Perbedaan karakteristik data ini memunculkan dugaan bahwa kompleksitas arsitektur model *Hybrid* dan *Log-mel Spectrogram* justru memiliki risiko *overfitting* yang lebih tinggi dibandingkan model *Raw Waveform* ketika dilatih pada dataset yang kecil. Selain itu, terdapat juga perbedaan mendasar pada jenis keluaran klasifikasi, di mana PANNs dilatih untuk mendeteksi banyak label sekaligus (*Multi-Label*), sedangkan penelitian ini dirancang untuk memprioritaskan identifikasi sumber bahaya paling dominan (*Single-Label*). Pendekatan *Single-Label* ini dipilih untuk menyelaraskan karakteristik dataset UrbanSound8K yang memiliki anotasi satu sumber suara dominan (*salient event*), serta guna menghindari ambiguitas peringatan agar pengguna dapat



mengambil keputusan responsif. Ketidakpastian inilah yang menjadi celah penelitian (*research gap*) yang belum terjamah. Oleh karena itu, penelitian ini menjadi krusial untuk mengevaluasi ulang adaptabilitas dan melakukan analisis komparasi ketiga arsitektur tersebut secara spesifik pada dataset UrbanSound8K.

Guna menjawab tantangan adaptabilitas pada dataset terbatas tersebut, penelitian ini bertujuan utama secara teknis untuk menginvestigasi dan membandingkan kinerja tiga pendekatan representasi input, yaitu *Raw Waveform*, *Log-mel Spectrogram*, dan *Hybrid* dalam mengklasifikasikan suara tanda bahaya yang mengancam keselamatan penyandang tunarungu. Studi komparasi ini diposisikan sebagai langkah fundamental untuk menemukan konfigurasi model yang paling *robust* (tahan uji) terhadap minimnya data, sekaligus meminimalisir kesalahan deteksi fatal. Dengan demikian, hasil evaluasi ini diharapkan dapat menjadi landasan teknis yang valid bagi pengembangan teknologi asistif yang benar-benar andal untuk menjamin keselamatan komunitas tunarungu.

## 1.2 Rumusan Masalah

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

1. Bagaimana pengaruh perbedaan representasi input (*Raw Waveform*, *Log-mel Spectrogram*, dan *Hybrid*) terhadap performa model *Pre-trained Audio Neural Networks* (PANNs) dalam mengklasifikasikan suara bahaya pada kondisi ketersediaan data yang terbatas?
2. Representasi input manakah yang menghasilkan model paling optimal berdasarkan metrik *F1-Score*, nilai *Loss*, dan analisis *Confusion Matrix* untuk meminimalisir kesalahan deteksi pada sistem keselamatan penyandang tunarungu?

### 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Menganalisis pengaruh perbedaan representasi input (*Raw Waveform*, *Log-mel Spectrogram*, dan *Hybrid*) terhadap performa model *Pre-trained Audio Neural Networks* (PANNs) dalam mengklasifikasikan suara bahaya pada kondisi ketersediaan data yang terbatas.
2. Mengevaluasi pendekatan representasi input yang menghasilkan model paling optimal berdasarkan metrik *F1-Score*, nilai *Loss*, dan analisis *Confusion Matrix* untuk meminimalisir kesalahan deteksi pada sistem keselamatan penyandang tunarungu.

### 1.4 Batasan Masalah

Batasan masalah yang didefinisikan dalam penelitian ini adalah sebagai berikut :

1. Dataset yang digunakan adalah dataset sekunder terstandarisasi, yaitu UrbanSound8K, tanpa melakukan perekaman data primer secara manual.
2. Lingkup klasifikasi dibatasi pada kategori suara lingkungan yang merepresentasikan indikator bahaya atau peringatan bagi keselamatan fisik di jalan raya.
3. Fokus penelitian terbatas pada eksperimen pelatihan (*training*) dan evaluasi performa model *Deep Learning*, serta tidak mencakup perancangan perangkat keras (*hardware*), pengembangan antarmuka pengguna (*User Interface*), maupun implementasi sistem secara *real-time*.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Memberikan bukti nyata terkait efektivitas metode *Transfer Learning* pada arsitektur PANNs serta perbandingan performa antara representasi input *Raw Waveform*, *Log-mel Spectrogram*, dan *Hybrid* dalam mengatasi

keterbatasan dataset.

2. Berkontribusi dalam pengembangan teknologi asistif berbasis AI yang dapat meningkatkan keselamatan dan kemandirian mobilitas penyandang tunarungu melalui deteksi suara bahaya yang akurat.
3. Menjadi referensi bagi penelitian selanjutnya atau pengembang aplikasi dalam menentukan konfigurasi model yang paling optimal untuk diterapkan pada sistem peringatan dini.

## **1.6 Sistematika Penulisan**

Sistematika penulisan berisi pembahasan apa yang akan ditulis disetiap Bab. Sistematika pada umumnya berupa paragraf yang setiap paragraf mencerminkan bahasan setiap Bab.

### **Bab I**

Bab ini berisikan penjelasan latar belakang dari topik penelitian yang berlangsung, rumusan masalah dari masalah yang dihadapi pada penjelasan di latar belakang, tujuan dari penelitian, batasan dari penelitian, manfaat dari hasil penelitian, dan sistematika penulisan tugas akhir.

### **Bab II**

Bab ini membahas mengenai tinjauan pustaka dari penelitian terdahulu dan dasar teori yang berkaitan dengan penelitian ini.

### **Bab III**

Bab ini berisikan penjelasan alur kerja penelitian, alat dan data yang digunakan, metode yang digunakan, dan metrik pengujian.

### **Bab IV**

Bab ini membahas hasil implementasi dan pengujian dari penelitian yang dilakukan, serta analisis dan evaluasi yang dapat dipetik dari hasil.

**Bab V**

Bab ini membahas kesimpulan dari hasil penelitian dan juga saran untuk penelitian selanjutnya.

## **BAB II**

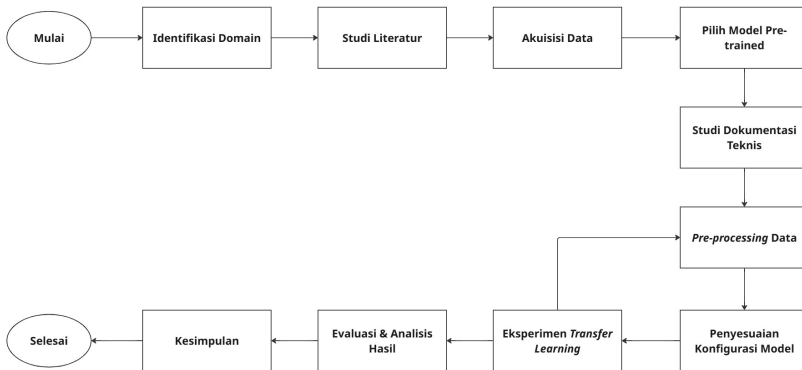
### **TINJAUAN PUSTAKA**

## BAB III

### METODE PENELITIAN

#### 3.1 Alur Penelitian

Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis guna memastikan model klasifikasi suara ancaman bahaya yang dikembangkan dapat bekerja optimal pada domain keselamatan penyandang tunarungu. Diagram alur penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Penjelasan rinci mengenai tahapan penelitian adalah sebagai berikut:

##### 1. Identifikasi Domain Penelitian

Tahap awal dilakukan dengan mencari referensi domain penelitian untuk klasifikasi suara. Berdasarkan pertimbangan ketersediaan dataset dan urgensi model klasifikasi, domain yang dipilih adalah domain keselamatan publik yang difokuskan untuk alat bantu penyandang tunarungu.

##### 2. Studi Literatur

Dilakukan kajian literatur mendalam untuk menemukan urgensi penelitian, khususnya mengenai kebutuhan teknologi asistif yang mampu mengurangi

risiko kecelakaan bagi penyandang tunarungu melalui pengenalan sinyal bahaya. Ini menjadi landasan teoritis bagaimana teknologi klasifikasi suara dapat menjadi solusi yang efektif dalam mengatasi permasalahan tersebut.

### 3. Identifikasi dan Akuisisi Data

Akuisisi data dilakukan dengan mengambil dataset sekunder yang sudah terstandarisasi, yaitu UrbanSound8K [5]. Akuisisi data primer tidak dilakukan demi menghindari *device bias* akibat perekaman dataset dengan perangkat non-standar (seperti smartphone) serta kendala regulasi keamanan pada perekaman kelas *gun\_shot*. Dari dataset tersebut, dilakukan seleksi kelas sinyal bahaya dengan strategi antisipasi terhadap kendala ketidakseimbangan data (*imbalanced data*) yang telah disiapkan sejak awal.

### 4. Pemilihan Model *Pre-trained*

Mengingat keterbatasan data berisiko menyebabkan kegagalan pelatihan *from scratch*, digunakan pendekatan *Transfer Learning* dengan memanfaatkan model *Pre-trained*. Tiga varian arsitektur PANNs (*Pre-trained Audio Neural Networks*) [7] dipilih untuk mewakili domain input yang berbeda, yaitu *Res1dNet31* (domain waktu), *ResNet38* (domain frekuensi), dan *Wavegram-Logmel-CNN (hybrid)*. Arsitektur PANNs dipilih karena telah dilatih pada dataset masif (AudioSet) dan memiliki performa yang teruji.

### 5. Studi Dokumentasi Teknis

Tahap ini mempelajari karakteristik data dan model yang digunakan. Fokus utama dalam tahap ini adalah memahami mekanisme pembagian data (fold) pada dataset guna mencegah kebocoran data (*data leakage*) dan mempelajari variasi representasi input pada arsitektur PANNs untuk menentukan skenario komparasi yang tepat.

## 6. *Pre-processing Data*

Serangkaian proses dilakukan untuk mengubah data mentah menjadi format yang siap latih, meliputi penanganan struktur *fold*, seleksi kelas, dan penyesuaian format audio, serta penerapan teknik augmentasi temporal berupa *random cropping* untuk menstandarisasi durasi input.

## 7. **Penyesuaian Konfigurasi Model**

Dilakukan modifikasi pada arsitektur model agar sesuai dengan tujuan klasifikasi 4 kelas bahaya, serta penerapan strategi untuk menangani ketidakseimbangan data.

## 8. **Eksperimen *Transfer Learning***

Tahapan ini menjadi tahapan inti di mana model dilatih ulang (*fine-tuning*) untuk mengenali karakteristik suara spesifik. Proses ini mencakup eksperimen *hyperparameter* dan pelatihan ketiga variasi model untuk menemukan konfigurasi parameter pelatihan yang paling optimal. Proses ini bersifat iteratif. Apabila performa model belum mencapai target yang diharapkan, maka akan dilakukan penyesuaian ulang pada tahap *pre-processing* data untuk melakukan penyesuaian data dan melakukan pelatihan kembali hingga diperoleh model yang optimal.

## 9. **Evaluasi dan Analisis Hasil**

Performa model hasil *fine-tuning* dievaluasi menggunakan metrik *F1-Score*, nilai *Loss*, dan analisis *Confusion Matrix*. Evaluasi ini bertujuan untuk membandingkan dan menganalisis performa antara ketiga model dalam melakukan klasifikasi suara ancaman bahaya.

## 10. **Kesimpulan**

Berdasarkan hasil evaluasi, dilakukan perbandingan komparatif untuk menyimpulkan model mana yang memiliki performa paling unggul dan stabil.



### 3.2 Pengumpulan dan Pra-pemrosesan Data

Sumber data utama dalam penelitian ini adalah dataset publik **UrbanSound8K** [5]. Dataset ini diunduh dalam format terkompresi (.zip). Setelah diekstraksi, struktur dataset terdiri dari 10 folder (masing-masing mewakili satu *fold*) beserta satu file metadata (.csv) yang memuat informasi nama file audio, *class ID*, dan *fold* asal.

Namun, dataset mentah ini memerlukan serangkaian penyesuaian agar relevan dengan konteks keselamatan penyandang tunarungu. Mengingat dataset ini awalnya berisi 10 kelas suara umum di perkotaan yang tersebar dalam 10 *fold*, penelitian ini memfokuskan pada seleksi kelas bahaya spesifik serta menyusun ulang data menjadi 5 fold eksperimen. Sebagai langkah antisipasi terhadap keterbatasan data setelah seleksi kelas, strategi augmentasi temporal diterapkan untuk memperkaya variasi data latih. Oleh karena itu, tahap pra-pemrosesan data menjamin integritas data dan mencegah kebocoran informasi (*data leakage*) selama proses pelatihan, sebagaimana dijabarkan pada tahapan berikut.

#### 3.2.1 Seleksi Kelas (*Class Filtering*)

Tidak seluruh kelas pada dataset UrbanSound8K relevan dengan konteks keselamatan tunarungu. Oleh karena itu, dilakukan penyaringan untuk hanya mengambil 4 kelas prioritas yang merepresentasikan sinyal bahaya, yaitu:

1. ***gun\_shot* (Tembakan)**

Suara tembakan senjata api merupakan sinyal bahaya dengan tingkat fatalitas yang sangat tinggi. Walaupun kejadiannya jarang terjadi, khususnya di Indonesia, deteksi suara ini menjadi langkah mitigasi risiko yang baik dalam melindungi penyandang tunarungu.

2. ***siren* (Sirine)**

Suara sirine menandakan keberadaan kendaraan prioritas (ambulans, pemadam kebakaran, atau patroli polisi) yang sering kali melaju dengan kecepatan tinggi dan memiliki hak prioritas jalan. Dengan deteksi suara

tersebut, pengguna dapat mengambil langkah untuk segera menepi dan menghindari jalur lintasan kendaraan tersebut.

### 3. *dog bark* (Gonggongan Anjing)

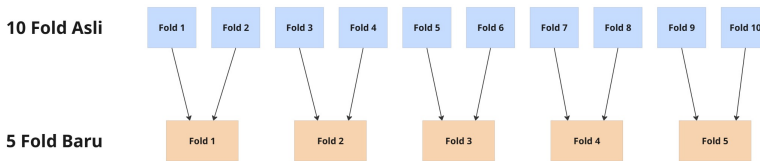
Suara anjing yang menggonggong berpotensi menjadi sinyal bahaya, terutama jika anjing tersebut berupa anjing liar yang dapat menyerang secara tiba-tiba. Dengan mendeteksi suara ini, pengguna dapat langsung mengetahui di mana posisi anjing tersebut dan mengambil langkah untuk antisipasi ancaman.

### 4. *car horn* (Klakson Mobil)

Suara klakson mobil menjadi sinyal penting pada lalu lintas, yang salah satu fungsinya adalah sebagai peringatan dari pengemudi kendaraan tersebut kepada pejalan kaki yang berada di jalur lintasnya. Dengan deteksi ini, pengguna dapat lebih waspada terhadap situasi lalu lintas di sekitarnya.

## 3.2.2 Strategi Penyusunan Ulang Data (*Fold Mapping*)

Dataset UrbanSound8K secara bawaan terbagi ke dalam 10 *fold*. Untuk efisiensi eksperimen tanpa melanggar aturan independensi data, penelitian ini menyusun ulang 10 *fold* tersebut menjadi 5 *fold* eksperimen baru. Penyusunan ini dilakukan secara berurutan (misalnya Fold 1 dan 2 menjadi Fold Baru 1) tanpa pengacakan data antar *fold*. Teknik penyusunan tersebut memastikan data latih dan data uji tidak tercampur pada fold yang sama, sehingga teknik penyusunan ini menghindari adanya kebocoran data (*data leakage*) yang menyebabkan model menjadi *overfitting*.



Gambar 3.2 Ilustrasi Strategi Penyusunan Ulang Fold pada Dataset UrbanSound8K

Adapun penyusunan ini didasari oleh dua pertimbangan, yaitu rasio pembagian data yang ideal dan efisiensi sumber daya komputasi. Skema *5-fold* menghasilkan proporsi 80% data latih dan 20% data uji. Proporsi ini memberikan model evaluasi yang lebih representatif terhadap variasi data daripada skema *10-fold* bawaan yang menyisakan 10% data uji. Selain memberikan porsi data yang ideal, skema tersebut berdampak mengurangi waktu komputasi pelatihan model tanpa mengurangi validitas pengujian.

Dikarenakan skema baru tersebut, maka mekanisme validasi yang digunakan selanjutnya pada model adalah *5-fold cross validation*. Pada iterasi setiap pengujianya, satu *fold* dialokasikan sebagai data uji secara bergantian dan empat *fold* sisanya akan digunakan sebagai data latih model. Mekanisme rotasi ini diterapkan agar model selalu diuji dengan data yang berbeda dari yang sudah pernah dipelajari saat proses latih, sehingga hasil evaluasi mencerminkan kemampuan generalisasi yang sebenarnya.

### 3.2.3 Standarisasi Format Audio

Sebelum data audio dapat diproses oleh model, diperlukan standarisasi format pada data tersebut mengingat data mentah memiliki format yang berbeda-beda. Ketidaksesuaian format input dapat menyebabkan kegagalan pada proses ekstraksi fitur, bahkan pelatihan model. Oleh karena itu, standarisasi format menjadi langkah penting untuk memastikan bahwa semua data audio memiliki

format yang konsisten dan sesuai dengan kebutuhan model, di mana langkah-langkah tersebut mencakup :

#### 1. **Konversi *Channel* Audio (*Down-mixing*)**

Langkah ini diterapkan karena terdapat beberapa data audio dengan format stereo (dua *channel*). Semua data audio dikonversi menjadi format mono (satu *channel*) untuk menyederhanakan representasi suara dan mengurangi kompleksitas komputasi.

#### 2. **Penyesuaian Sampling Rate (*Resampling*)**

Setelah audio diubah menjadi format mono, dilakukan penyesuaian sampling rate (*resampling*). Sampling rate pada dataset UrbanSound8K bervariasi, sehingga sampling rate pada data diubah menjadi 32.000 Hz agar sesuai dengan standar input pada model PANNs.

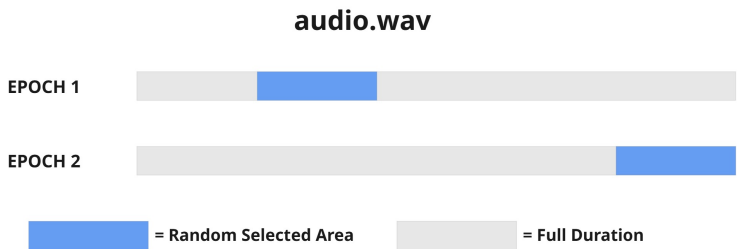
### 3.2.4 **Augmentasi Temporal (*Random Cropping*)**

Seleksi kelas bahaya yang telah dilakukan menghasilkan jumlah data yang relatif sedikit pada kelas yang terpilih, yang berpotensi membuat model kekurangan data untuk latihan dan uji. Oleh karena itu, teknik augmentasi hadir sebagai solusi untuk memperkaya variasi data tanpa perlu menambah jumlah sampel dengan dataset lain. Teknik ini juga mempersingkat waktu dan sumber daya komputasi dikarenakan model hanya mempelajari sebagian fitur penting dari seluruh bagian fitur pada data aslinya.

Sebelum melakukan proses augmentasi, durasi audio akan diperiksa terlebih dahulu apakah sesuai dengan standar yang telah ditetapkan, yaitu 5 detik. Durasi 5 detik ini dipilih karena dataset UrbanSound8K memiliki data audio yang berdurasi rata-rata 4 detik, sehingga 5 detik merupakan durasi yang cukup aman untuk dijadikan standar dalam penelitian ini. Jika audio asli berdurasi kurang dari 5 detik, maka audio akan ditambahkan *padding* hingga durasinya mencapai 5 detik. Jika audio asli berdurasi lebih dari 5 detik, maka audio akan dipotong hingga berdurasi 5 detik. Setelah standarisasi durasi selesai, maka

data siap untuk diproses menggunakan teknik augmentasi.

Teknik augmentasi yang diterapkan adalah *Random Cropping*, yaitu pemotongan acak pada segmen audio sepanjang durasi tertentu. Pada setiap *epoch* pelatihan, segmen 1 detik akan diambil secara acak dari durasi 5 detik tersebut. Ini memberikan variasi temporal pada model untuk belajar mengenali karakteristik suara yang berbeda dari berbagai posisi dalam rekaman asli pada setiap *epoch*. Teknik *Random Cropping* ini tidak diterapkan pada data uji dikarenakan berpotensi mengubah distribusi data uji, sehingga model tidak dapat mencapai nilai evaluasi secara maksimal. Strategi ini memastikan bahwa data uji memiliki dimensi yang konsisten, sehingga performa model dapat dievaluasi secara stabil dan tolak ukur pengujian dapat direproduksi dengan akurat.



Gambar 3.3 Ilustrasi Teknik *Random Cropping* Pada Setiap *Epoch*

3.3 Konfigurasi Model

Model yang digunakan dalam penelitian ini adalah model PANNs [7] dengan 3 arsitektur berbeda, yaitu *Res1dNet31*, *ResNet38*, dan *Wavegram-Logmel-CNN*. Ketiga arsitektur ini telah terbukti memiliki performa yang baik pada tugas klasifikasi audio berdasarkan masing-masing representasi inputnya. Namun, agar model dapat berfungsi optimal sesuai dengan konteks klasifikasi 4 kelas bahaya pada dataset UrbanSound8K, diperlukan beberapa penyesuaian konfigurasi model, yaitu sebagai berikut.

### 3.3.1 Modifikasi Arsitektur PANNs

Modifikasi beberapa komponen pada arsitektur PANNs diperlukan pada lapisan keluaran (*output layer*). Langkah ini dilakukan mengingat jumlah kelas pada *output layer* yang belum sesuai dengan target kelas suara yang ditentukan dan penyesuaian jenis output klasifikasi PANNs yang awalnya *multi-label*. Oleh karena itu, dilakukan dua penyesuaian utama:

#### 1. Penyesuaian Jumlah Neuron pada Lapisan Akhir

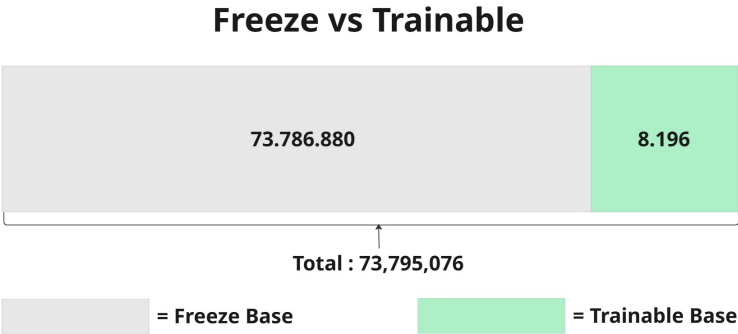
Jumlah *neuron* pada lapisan akhir disesuaikan dari 527 *node* menjadi 4 *node* (sesuai jumlah kelas pilihan).

#### 2. Penyesuaian Fungsi Aktivasi pada Lapisan Akhir

Pada model asli PANNs, lapisan akhir menggunakan fungsi aktivasi *sigmoid* yang mengeluarkan output berupa probabilitas tiap kelasnya sendiri (cocok untuk *multi-label*). Namun, klasifikasi model yang ditentukan pada penelitian ini adalah klasifikasi *Single-Label*, sehingga terjadi modifikasi pada lapisan akhir menjadi lapisan *Linear* yang mengeluarkan angka mentah (*logits*). Nilai *logits* ini nantinya akan dikonversi menjadi distribusi probabilitas tunggal oleh fungsi kerugian (*Loss Function*) selama fase pelatihan. Dengan perubahan ini, model akan memilih satu kelas pemenang dengan probabilitas tertinggi daripada kelas lainnya.

### 3.3.2 Strategi Freeze Base

Mengingat penelitian ini menggunakan metode *Transfer Learning*, lapisan dasar fitur (*base model*) pada model PANNs dibekukan agar model tidak menghapus ingatan fitur dasar yang telah dipelajari dari dataset besar sebelumnya (AudioSet) [8]. Strategi ini memastikan model memiliki inisialisasi bobot yang baik dan hanya perlu mempelajari pola baru yang spesifik pada dataset target.



Gambar 3.4 Ilustrasi *Transfer Learning* (Freeze vs Trainable Base)

**3.3.3 Parameter Konfigurasi Input**

Selain melakukan modifikasi arsitektur, konfigurasi parameter input juga disesuaikan dengan standar yang ditetapkan pada PANNs untuk menjamin kompatibilitas dimensi fitur antara data input dengan bobot *pre-trained*, sehingga ekstraksi fitur dapat berjalan optimal tanpa adanya kesalahan dimensi. Rincian konfigurasi tersebut meliputi:

Tabel 3.1 Konfigurasi Parameter Input Audio Model PANNs

Parameter	Nilai	Keterangan
Sampling Rate	32.000 Hz	Frekuensi pencuplikan sinyal audio
Window Size	1024	Ukuran jendela FFT ( <i>Fast Fourier Transform</i> )
Hop Size	320	Jarak pergeseran antar jendela waktu
Mel-bins	64	Jumlah filter bank pada skala Mel (dimensi fitur vertikal)
F-min	50 Hz	Batas frekuensi minimum
F-max	14.000 Hz	Batas frekuensi maksimum

Penggunaan *sampling rate* 32 kHz dipilih merujuk pada spesifikasi standar PANNs [7]. Angka ini juga telah memenuhi kriteria Teorema Nyquist untuk menangkap frekuensi hingga 16 kHz. Rentang ini dinilai efisien karena sebagian besar energi spektral pada suara bahaya (seperti sirine, klakson, dan tembakan) berada pada rentang 50 Hz hingga 14.000 Hz, sehingga parameter ini

dapat mempertahankan informasi vital sekaligus mereduksi beban komputasi dibandingkan standar audio 44.1 kHz.

### **3.4 Eksperimen *Transfer Learning***

Setelah model dikonfigurasi sesuai kebutuhan penelitian, tahap selanjutnya adalah melakukan eksperimen *Transfer Learning* dengan arsitektur PANNs. Tahapan ini melibatkan konfigurasi hyperparameter pelatihan secara eksperimental dan penerapan beberapa strategi pelatihan yang bertujuan untuk mendapatkan hasil evaluasi yang optimal. Beberapa strategi teknis yang diterapkan mencakup pengondisian lingkungan implementasi, penerapan standar reproduksibilitas, penanganan ketidakseimbangan data, serta mekanisme kontrol pelatihan untuk memastikan model mempelajari data dengan baik dan menghindari *overfitting*. Penggunaan WandB sebagai alat pemantauan dan pencatatan eksperimen pelatihan model secara *real-time* termasuk ke dalam strategi untuk memastikan transparansi dan menjaga validitas dalam merekam hasil eksperimen. Rincian beberapa langkah strategi teknis tersebut dijabarkan sebagai berikut.

#### **3.4.1 Lingkungan Implementasi dan Reproduksibilitas**

Seluruh rangkaian eksperimen pelatihan dijalankan pada satu perangkat komputasi terdedikasi guna menjaga konsistensi performa dan waktu komputasi. Mengingat perangkat keras memiliki dampak signifikan terhadap efisiensi proses *fine-tuning* model, rincian spesifikasi perangkat keras (*hardware*) yang digunakan dijabarkan pada Tabel 3.2.



Tabel 3.2 Spesifikasi Perangkat Keras untuk Pelatihan

Komponen	Spesifikasi	Keterangan
Processor (CPU)	Intel Core i7-7700HQ	Bertanggung jawab atas <i>pre-processing</i> data, augmentasi dan manajemen <i>dataloader</i> .
Kartu Grafis (GPU)	NVIDIA GeForce GTX 1050 (4 GB VRAM)	Berperan dalam akselerasi komputasi tensor selama pelatihan model.
Memori (RAM)	8 GB DDR4	Menunjang alokasi data sementara dengan kecepatan transfer 2400 MHz.
Penyimpanan	1 TB HDD	Digunakan sebagai repositori penyimpanan dataset utama.

Selain perangkat keras, lingkungan perangkat lunak juga dikonfigurasi menggunakan pustaka (*library*) standar industri untuk menjamin reliabilitas eksperimen pelatihan. Implementasi dilakukan menggunakan bahasa pemrograman **Python** dengan kerangka kerja utama **PyTorch**. Rincian lengkap *library* yang digunakan beserta fungsinya ditampilkan pada Tabel 3.3.

Tabel 3.3 Daftar Perangkat Lunak dan Pustaka Pendukung

Perangkat Lunak	Fungsi Utama
Python	Bahasa pemrograman dasar untuk seluruh implementasi teknis penelitian.
PyTorch	Kerangka kerja utama untuk pembangunan, pelatihan, dan evaluasi model <i>Deep Learning</i> .
NumPy	Pustaka komputasi numerik untuk manipulasi matriks dan array.
Scikit-learn	Pustaka untuk perhitungan metrik evaluasi ( <i>F1-Score</i> ) dan <i>Confusion Matrix</i> .
Matplotlib & Seaborn	Pustaka visualisasi data untuk pembuatan grafik hasil pelatihan.
Weights & Biases	Platform pemantauan ( <i>monitoring</i> ) metrik pelatihan secara <i>real-time</i> .

Guna memastikan eksperimen dapat direproduksi ulang (*reproducible*) pada aspek logika program, dilakukan penguncian benih acak (*Random Seed*)

bernilai tetap (42) pada tiga komponen pustaka utama, yaitu:

1. **Python Native Random**

Mengunci pengacakan pada operasi dasar.

2. **NumPy**

Menjamin konsistensi pada operasi manipulasi matriks dan pembagian data (*fold splitting*).

3. **PyTorch (CPU & CUDA)**

Mengunci inisialisasi bobot awal jaringan saraf agar start pelatihan selalu sama.

Meskipun inisialisasi bobot dan pembagian data telah dikunci secara deterministik, konfigurasi  `cudnn.deterministic`  pada *backend* GPU tetap dinonaktifkan (*False*). Keputusan ini diambil untuk memprioritaskan efisiensi sumber daya dan kecepatan komputasi, mengingat penegakan determinisme penuh pada operasi konvolusi GPU dapat meningkatkan waktu pelatihan secara signifikan. Variasi hasil numerik mikroskopis yang mungkin timbul akibat optimasi perangkat keras ini dinilai dapat diabaikan (*negligible*) dan tidak mempengaruhi validitas tren performa model secara keseluruhan.

### 3.4.2 Penanganan Ketidakseimbangan Data (*Weight Penalty*)

Meskipun dataset UrbanSound8K memiliki distribusi data yang relatif terstruktur, proses seleksi kelas dan penyusunan ulang *fold* berpotensi menghasilkan ketidakseimbangan jumlah sampel antar kelas. Ketidakseimbangan ini dapat memicu bias pada model, di mana model akan cenderung memprediksi kelas mayoritas (jumlah sampel banyak) dan mengabaikan kelas minoritas karena kontribusinya terhadap nilai *loss* yang kecil.

Untuk memitigasi risiko tersebut, diterapkan strategi *Weight Penalty* atau pemberian bobot penalti pada fungsi kerugian (*Loss Function*). Strategi ini bekerja dengan memberikan bobot *loss* yang besar pada kelas dengan jumlah

sampel sedikit, dan sebaliknya. Hal ini memaksa model untuk memberikan perhatian yang setara pada semua kelas selama proses pembaruan gradien tanpa memperhitungkan jumlah sampelnya.

Bobot penalti ( $W_j$ ) untuk setiap kelas ke- $j$  dihitung secara otomatis sebelum pelatihan dimulai menggunakan formulasi berikut:

$$W_j = \frac{N}{C \times N_j} \quad (\text{Rumus 3.1})$$

Keterangan:

- $W_j$  : Bobot skalar yang dihasilkan untuk kelas  $j$ .
- $N$  : Total jumlah sampel dalam seluruh dataset latih.
- $C$  : Jumlah kelas total (4 kelas).
- $N_j$  : Jumlah sampel spesifik pada kelas  $j$ .

Sebagai ilustrasi, jika kelas *gun\_shot* memiliki jumlah sampel yang jauh lebih sedikit dibandingkan *car\_horn*, maka nilai  $N_j$  yang kecil pada penyebut akan menghasilkan nilai  $W_j$  yang besar. Nilai bobot ini kemudian diaplikasikan sebagai argumen parameter pada fungsi *Cross Entropy Loss*. Secara matematis, jika model melakukan kesalahan prediksi pada kelas *gun\_shot*, nilai *loss* akan dikalikan dengan faktor  $W_j$  yang besar tersebut. Dengan begitu, model akan dihukum lebih berat dan dipaksa untuk belajar mengenali fitur kelas minoritas tersebut.

### 3.4.3 Konfigurasi *Optimizer* dan *Scheduler*

Proses pembaruan bobot model (*weight update*) dilakukan menggunakan algoritma optimasi *optimizer AdamW* (*Adam with Weight Decay*). Pemilihan AdamW didasari oleh kemampuannya dalam memisahkan mekanisme pembaruan bobot dengan *weight decay*, sehingga lebih efektif dalam mencegah *overfitting* dibandingkan algoritma Adam standar. Pelatihan dilakukan dalam *batch size* sebesar 8 selama maksimal 50 *epoch*, dengan pengaturan *Learning*

*Rate* awal sebesar  $5e-4$  dan nilai *Weight Decay* sebesar  $1e-4$ .

Untuk mengoptimalkan proses adaptasi pelatihan model, diterapkan pula penyesuaian laju pembelajaran dinamis menggunakan *scheduler ReduceLROnPlateau*. Mekanisme ini dirancang untuk memantau metrik *Validation Loss*. Apabila nilai kerugian validasi tersebut tidak mengalami penurunan selama 3 *epoch* berturut-turut (*patience*), maka nilai *Learning Rate* akan direduksi secara otomatis sebesar setengah dari nilai sebelumnya (faktor 0,5). Strategi ini membantu model untuk menemukan detail kecil global dengan lebih presisi (konvergensi) saat mendekati akhir pelatihan.

### 3.4.4 Strategi Optimasi: *Early Stopping* dan *Checkpoint*

Mengingat pelatihan dilakukan pada dataset terbatas, risiko *overfitting* pada *epoch* lanjutan menjadi sangat tinggi. Untuk mencegah hal tersebut, diterapkan dua mekanisme pencegahan otomatis:

#### 1. *Early Stopping*

Pelatihan model akan dihentikan secara paksa lebih awal apabila metrik evaluasi tidak mencatatkan rekor performa baru selama 5 *epoch* berturut-turut. Ini mencegah model menghafal data latih secara berlebihan.

#### 2. *Model Checkpointing*

Selama proses pelatihan berjalan, sistem secara dinamis hanya akan menyimpan (*save*) parameter bobot model pada *epoch* yang berhasil mencetak nilai *F1-Score* validasi tertinggi (*best model*). Jika performa pada *epoch* selanjutnya memburuk, model tidak akan disimpan, sehingga hasil akhir yang dievaluasi dipastikan adalah versi model yang paling optimal.

### 3.4.5 Pemantauan dan Pencatatan Eksperimen

Dalam memantau pergerakan metrik evaluasi pelatihan (termasuk *Train/Validation Loss*, *F1-Score*, dan pergerakan *Learning Rate*), metrik tersebut

direkam dan divisualisasikan secara *real-time* menggunakan alat bantu berupa *dashboard* **Weights & Biases (WandB)**. Penggunaan alat ini memastikan transparansi rekam jejak pelatihan model. Di akhir setiap pengujian *fold*, sistem juga secara otomatis menggenerasi laporan grafik riwayat pelatihan dengan menampilkan metrik yang dijadikan bahan evaluasi.

#### **3.4.6 Konfigurasi Parameter Pelatihan**

Rincian skenario pelatihan tadi telah membawa beberapa hyperparameter yang akan dijadikan sebagai tolak ukur eksperimen pada training model. Hyperparameter yang dikendalikan dalam eksperimen ini disajikan pada Tabel 3.4. Nilai-nilai yang ditetapkan dalam tabel ini ditetapkan berdasarkan studi literatur dan eksperimen awal untuk mendapatkan konvergensi yang optimal.

Tabel 3.4 Parameter Konfigurasi Pelatihan

Kategori	Konfigurasi / Nilai
<b>Hyperparameter Training</b>	
Batch Size	8
<i>Epoch</i>	50
Num Workers	2
Learning Rate Awal	$0.0005 (5e^{-4})$
<b>Optimizer</b>	
Tipe	AdamW
Weight Decay	$0.0001 (1e^{-4})$
<b>Learning Rate Scheduler</b>	
Tipe	ReduceLROnPlateau
Faktor Pengurangan	0.5
Patience	3
Target Metrik	Validation Loss
<b>Early Stopping</b>	
Patience	5
Target Metrik	Validation F1-Score $\leq$ Best Validation F1-Score
<b>Reproducibilitas</b>	
Random Seed	42

### 3.5 Analisis dan Evaluasi

Setelah proses pelatihan selesai, kinerja model diukur menggunakan empat indikator utama:

#### 1. *F1-Score*

Digunakan sebagai metrik utama untuk mengukur akurasi dan sensitivitas model secara harmonis, memastikan semua kelas bahaya terdeteksi dengan baik.

#### 2. *Nilai Loss*

Digunakan untuk memantau proses konvergensi model selama pelatihan dan mendeteksi indikasi *overfitting* atau *underfitting*.

### 3. *Confusion Matrix*

Digunakan untuk melihat detail distribusi prediksi benar dan salah pada setiap kelas spesifik.

## DAFTAR PUSTAKA

- [1] World Health Organization. *Global Status Report on Road Safety 2023*. Geneva: World Health Organization, 2023. ISBN: 9789240086517.
- [2] Birgitta Thorslund et al. “Effects of hearing loss on traffic safety and mobility”. *European Transport Research Review* 5 (2013), pp. 113–121.
- [3] Brian C. J. Moore. “Perceptual Consequences of Cochlear Hearing Loss and their Implications for the Design of Hearing Aids”. *Ear and Hearing* 17.2 (1996), pp. 133–161.
- [4] Michael Büchler et al. “Sound classification in hearing aids inspired by auditory scene analysis”. *The Journal of the Acoustical Society of America* 118.3 (2005), pp. 2057–2057.
- [5] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. “A Dataset and Taxonomy for Urban Sound Research”. *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM. 2014, pp. 1041–1044.
- [6] Karol J Piczak. “Environmental sound classification with convolutional neural networks”. *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2015, pp. 1–6.
- [7] Qiuqiang Kong et al. “PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition”. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2880–2894.
- [8] Jort F Gemmeke et al. “Audio set: An ontology and human-labeled dataset for audio events”. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 776–780.