IMPLEMENTASI SISTEM PRA-KLASIFIKASI SUARA BURUNG MENGGUNAKAN METODE FUZZY DAN SUPERVISED LEARNING

Disusun Oleh:

M. Rifqi Dzaky Azhad / 103012330009

Fathan Arya Maulana / 10301230083

Latar Belakang

Penurunan populasi burung, dengan lebih dari 1.400 spesies terancam punah secara global dan 177 di Indonesia, menunjukkan pentingnya pemantauan menggunakan observasi langsung atau teknologi AI untuk konservasi keanekaragaman hayati.

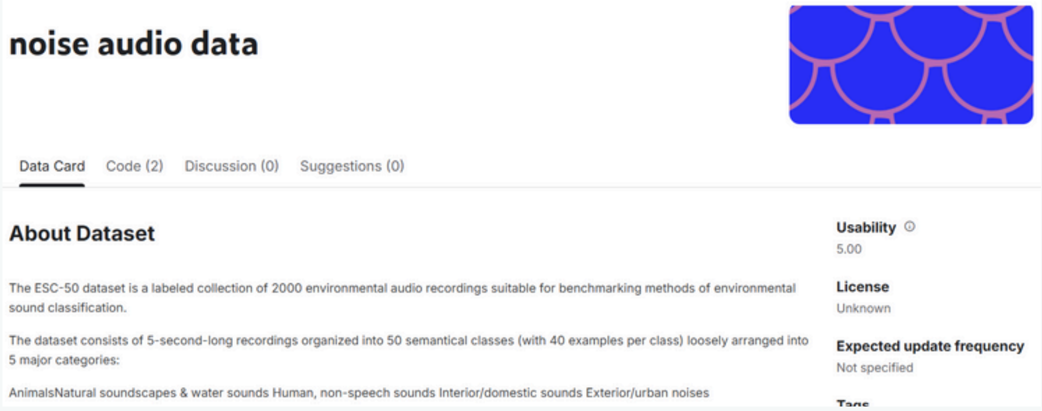
Rumusan Masalah

Monitoring burung secara langsung mengganggu perilaku alami, sehingga pengamatan pasif dengan mikrofon lebih efektif. Namun, metode klasifikasi suara yang ada butuh komputasi dan energi besar, menyulitkan penerapan monitoring real-time di lapangan.

Solusi

Proyek ini mengusulkan sistem monitoring pasif berbasis audio dengan model **pra-klasifikasi** untuk mendeteksi segmen berpotensi memiliki suara burung, sehingga hanya bagian relevan yang diproses oleh model klasifikasi. Pendekatan ini bertujuan mengurangi beban komputasi dan meningkatkan efisiensi sistem.

Xeno-Canto



Kaggle: noise audio data

Dataset

Dataset didapat dari 2 sumber yaitu Xeno-canto dan Noise Audio Data (Kaggle).

- Dataset Xeno-canto terdiri dari 30 file audio dengan total durasi 11 menit 12 detik.
- Dataset Noise Audio Data (Kaggle) terdiri dari 50 file audio dengan total durasi 4 menit 10 detik

Anotasi

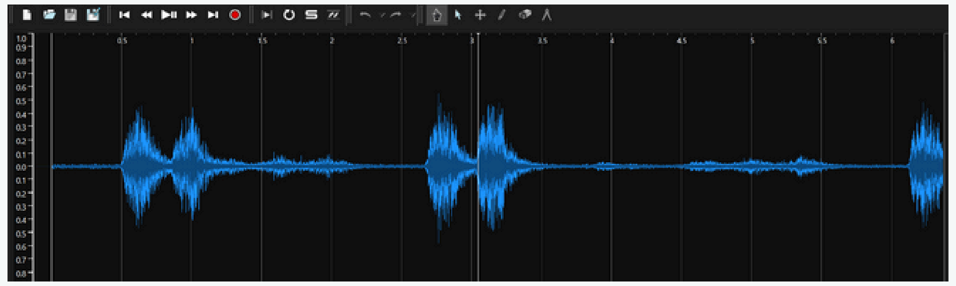
Dilakukan secara manual menggunakan aplikasi sonic visualizer. label diberikan dengan ketentuan:

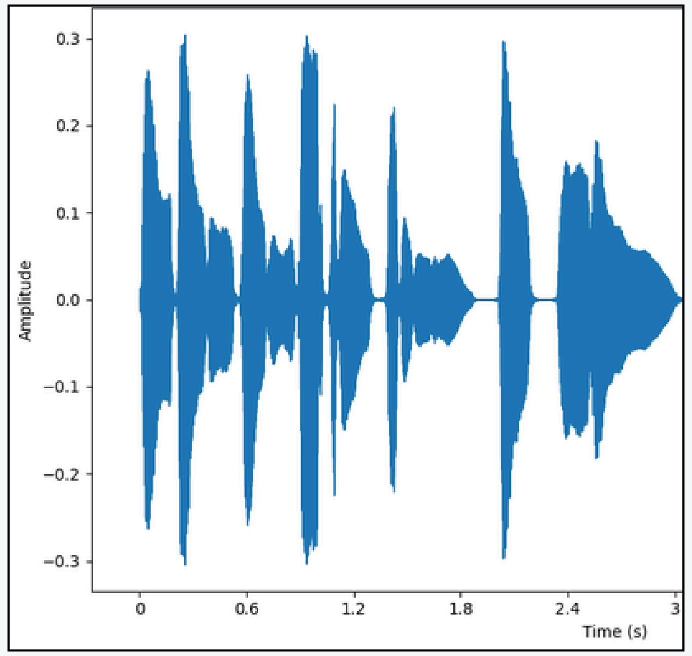
- 0 untuk suara burung
- 1 untuk suara event (keras)
- 2 untuk untuk suara non-event (hening)

Prapemrosesan

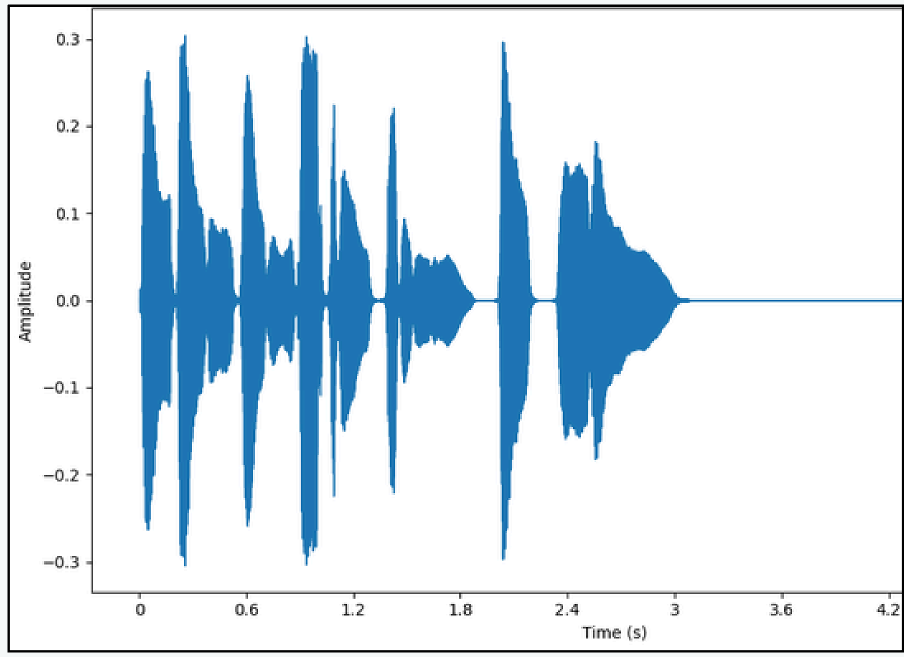
Prapemrosesan dilakukan dengan cara menyamakan karakteristik dari semua file audio.

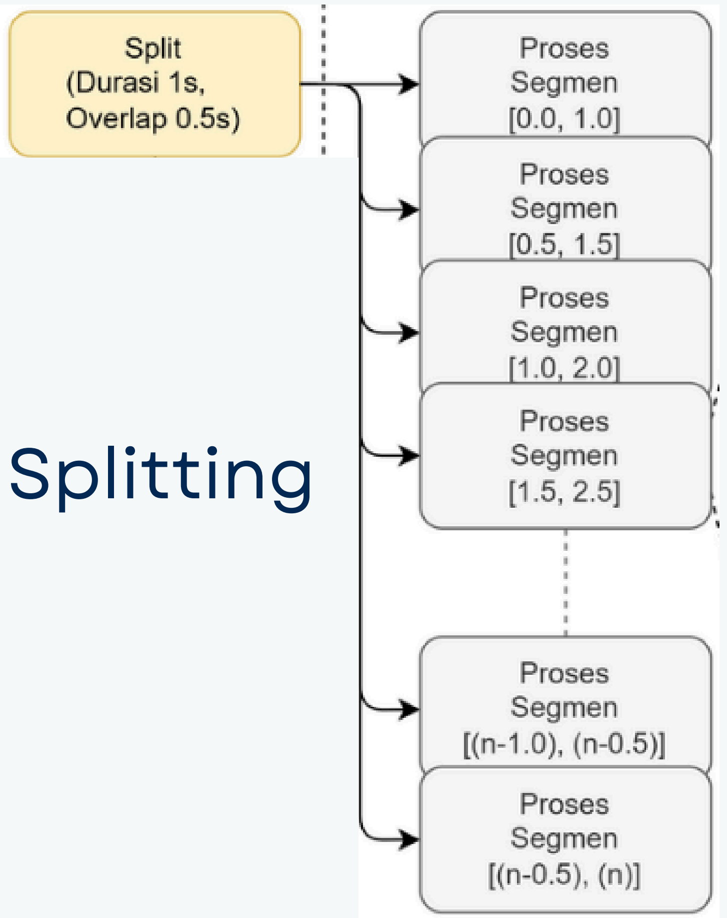
Downmixing





Padding





Splitting

Resampling, menyamakan sample rate menjadi 48 khz (S. Kahl birdNet anlyzer 2021)

```python
def prapemrosesan_downmixing(audio):
    if audio.ndim > 1:
        audio = np.mean(audio, axis=1)
    return audio.astype(np.float32)

def prapemrosesan_resampling(audio, sr):
    if sr == SAMPLE_RATE:
        return audio.copy(), SAMPLE_RATE

    ratio = SAMPLE_RATE / sr
    n_samples = int(np.round(len(audio) * ratio))

    x_old = np.linspace(0, 1, len(audio))
    x_new = np.linspace(0, 1, n_samples)
    return np.interp(x_new, x_old, audio), SAMPLE_RATE

def prapemrosesan_padding(audio):
    if np.mod(audio.shape[0], SEGMENT_DURATION) != 0:
        padding = SEGMENT_DURATION - (audio.shape[0] % SEGMENT_DURATION)
        audio = np.pad(audio, (0, padding))
    return audio

def prapemrosesan_splitting(audio):
    num_segments = int(np.floor((len(audio) - SEGMENT_DURATION) / OVERLAP_DURATION)) + 1
    segments = []

    for i in range(num_segments):
        start = int(i * OVERLAP_DURATION)
        end = int(start + SEGMENT_DURATION)
        segment = audio[start:end]
        if len(segment) < SEGMENT_DURATION:
            segment = np.pad(segment, (0, SEGMENT_DURATION - len(segment)), mode='constant')
        segments.append(segment)

    return np.array(segments)
```

Langkah	Deskripsi
Downmixing	Mengonversi audio stereo menjadi mono dengan merata-ratakan kedua channel: $x_{\text{mono}} = \frac{x_L + x_R}{2}$
Resampling	Mengubah sample rate menjadi 48 kHz untuk standarisasi sinyal audio [16].
Padding	Menyesuaikan durasi audio agar menjadi kelipatan detik tertentu, guna mempermudah proses segmentasi.
Splitting	Membagi audio menjadi segmen berdurasi 1 detik dengan overlap 0.5 detik. Segmen ke-i dimulai pada $i \times 0.5$ detik.

4 Langkah prapemrosesan

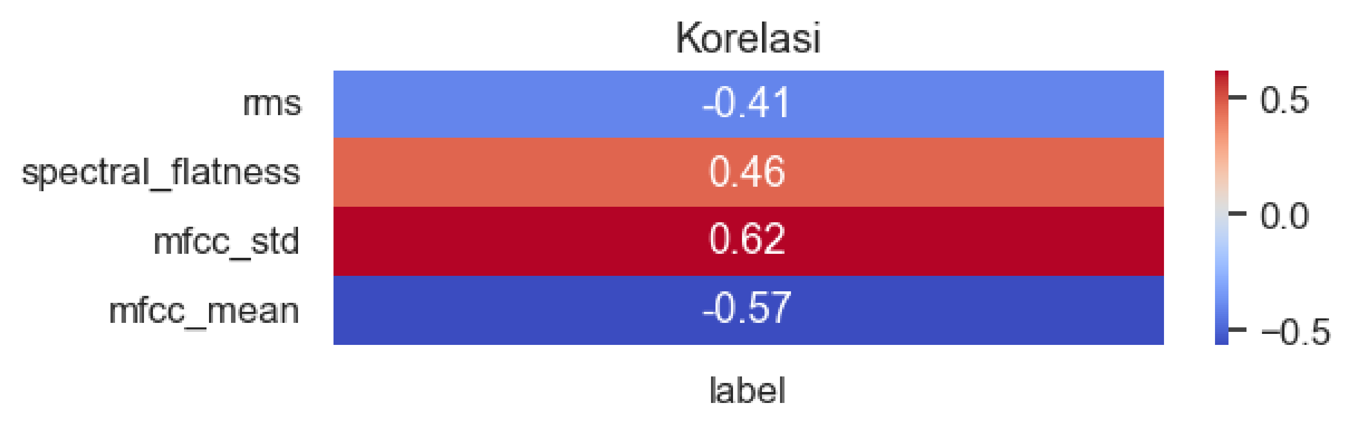
Fitur Ekstaksi

```python
def get_rms(segment):
    return np.mean(librosa.feature.rms(y=segment))

def get_spectral_flatness(segment):
    return np.mean(librosa.feature.spectral_flatness(y=segment))

def get_mfcc(segment):
    return librosa.feature.mfcc(y=segment, sr=SAMPLE_RATE)
```

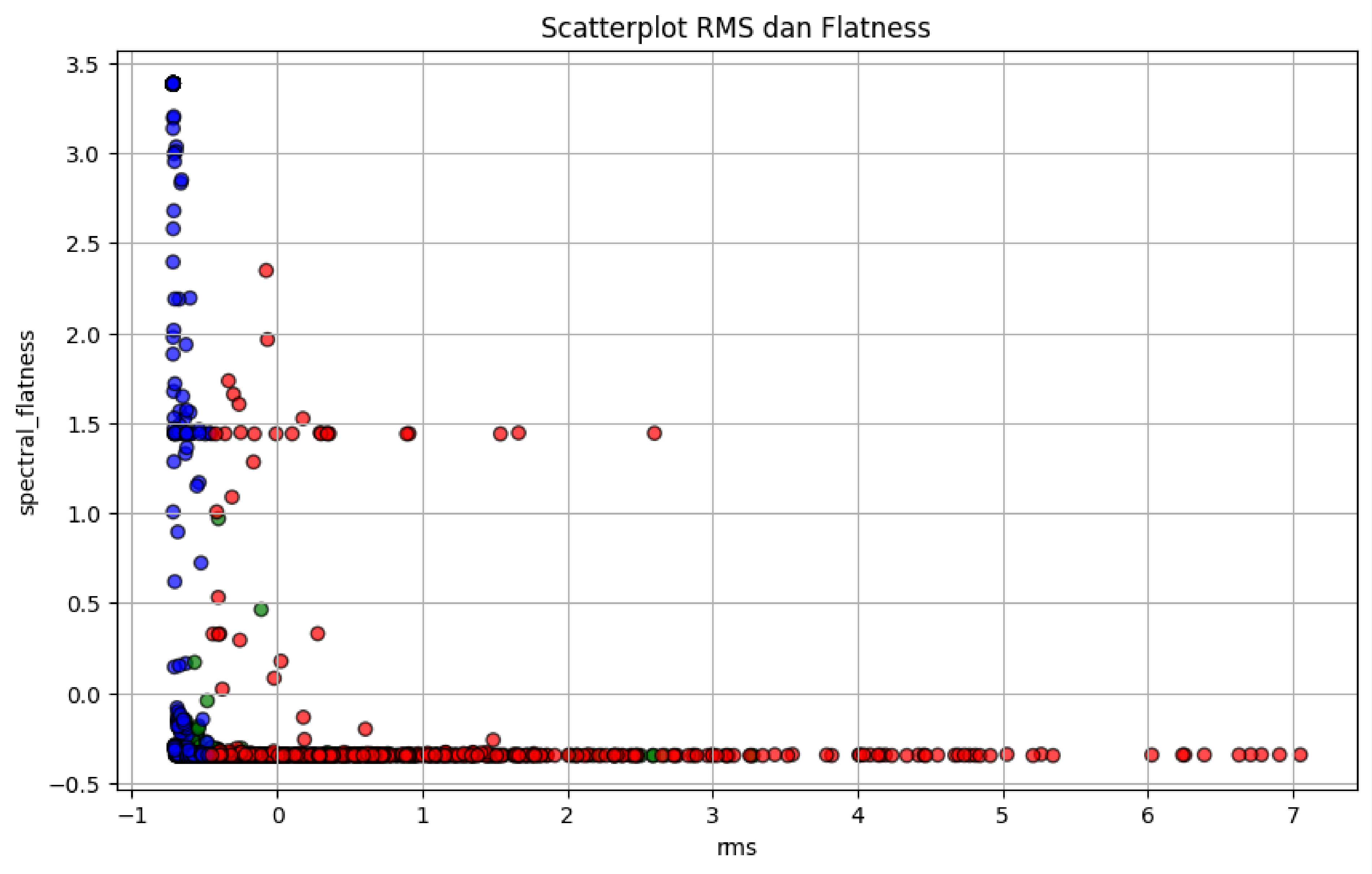
Fitur	**Rumus**
RMS	$\sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}$
Spectral Flatness	$\frac{\left(\prod_{n=1}^{N} X[n]\right)^{1/N}}{\frac{1}{N}\sum_{n=1}^{N} X[n]}$
MFCC	$\mathrm{DCT}(\log(\mathrm{MelFB} \cdot \lvert STFT(x)\rvert^2))$

Tiga fitur diekstraksi dari masing-masing segmen audio, yaitu RMS (Root Mean Squared), Spectral Flatness, dan MFCC (Mel-Frequency Cepstral Coefficients).

Exploratory Data Analysis



Nilai Korelasi (Pearson’s Correlation)



Scatter Plot RMS x Flatness. Suara hening (biru), suara event (merah), suara burung (hijau)

Model

Pra-klasifikasi

model yang digunakan untuk klasifikasi suara event dan non-event, model yang digunakan:

- Fuzzy Mamdani
- Logistic Regression
- Decision Tree

Klasifikasi

model yang digunakan untuk klasifikasi suara burung dan non-suara burung, model yang digunakan:

- Light Gradient Boost Machine (LGBM)
- Multi Layer Preceptron (MLP)

Mamdani Fuzzy Inference System

1

Nilai Linguistik:

1. Fuzzy: Low, Medium, dan High
2. Flatness: Rigid dan Smooth
3. Label: Non-Event dan Event

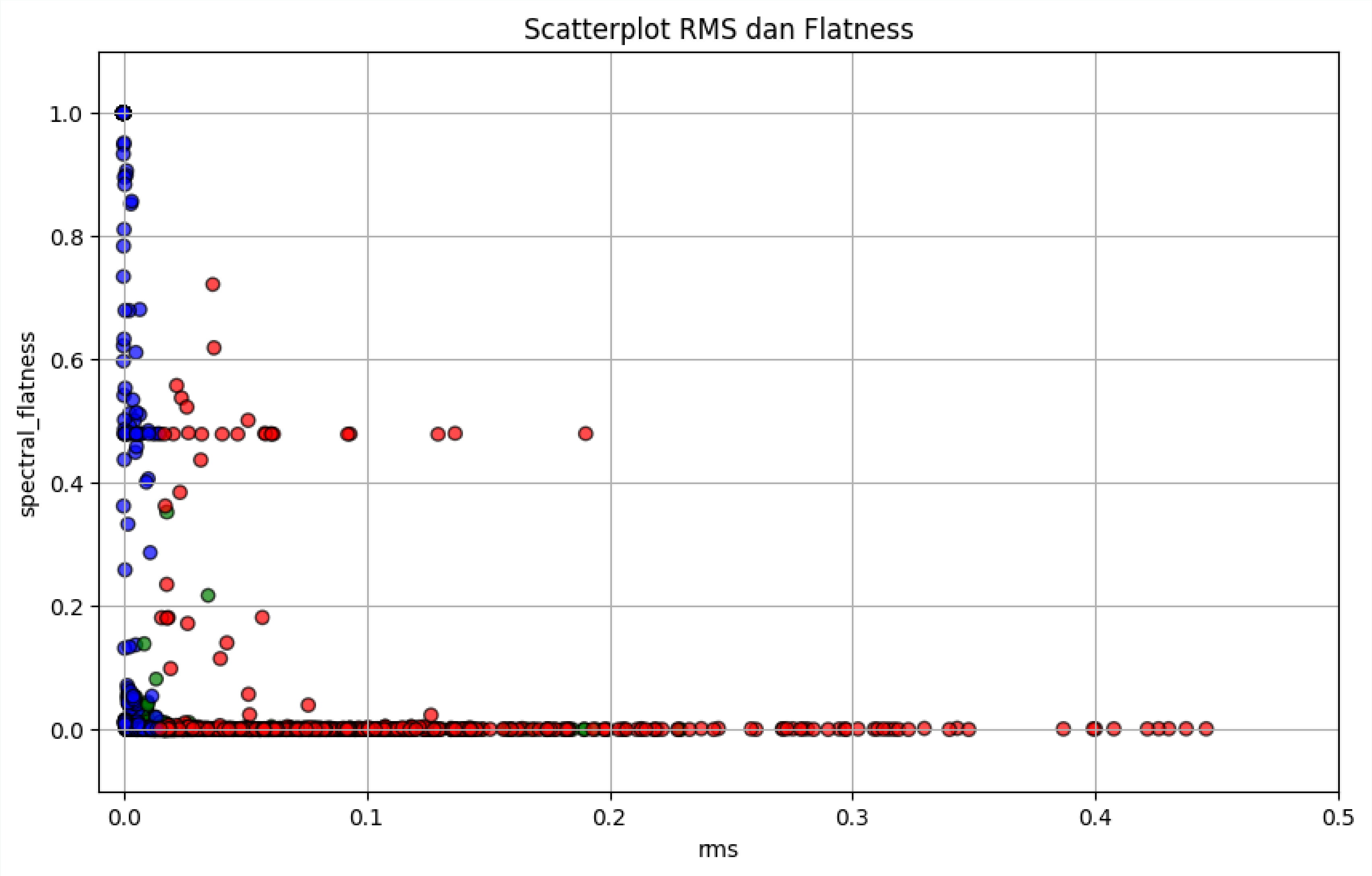
Variabel	Label Keanggotaan
RMS	Low Medium High
Flatness	Rigid Smooth
Label Variabel	Event Non-event

Mamdani Fuzzy Inference System

2

Batas nilai Linguistik

- RMS: [-1, 8]
- Flatness: [-1, 4]
- Label: [1, 2]

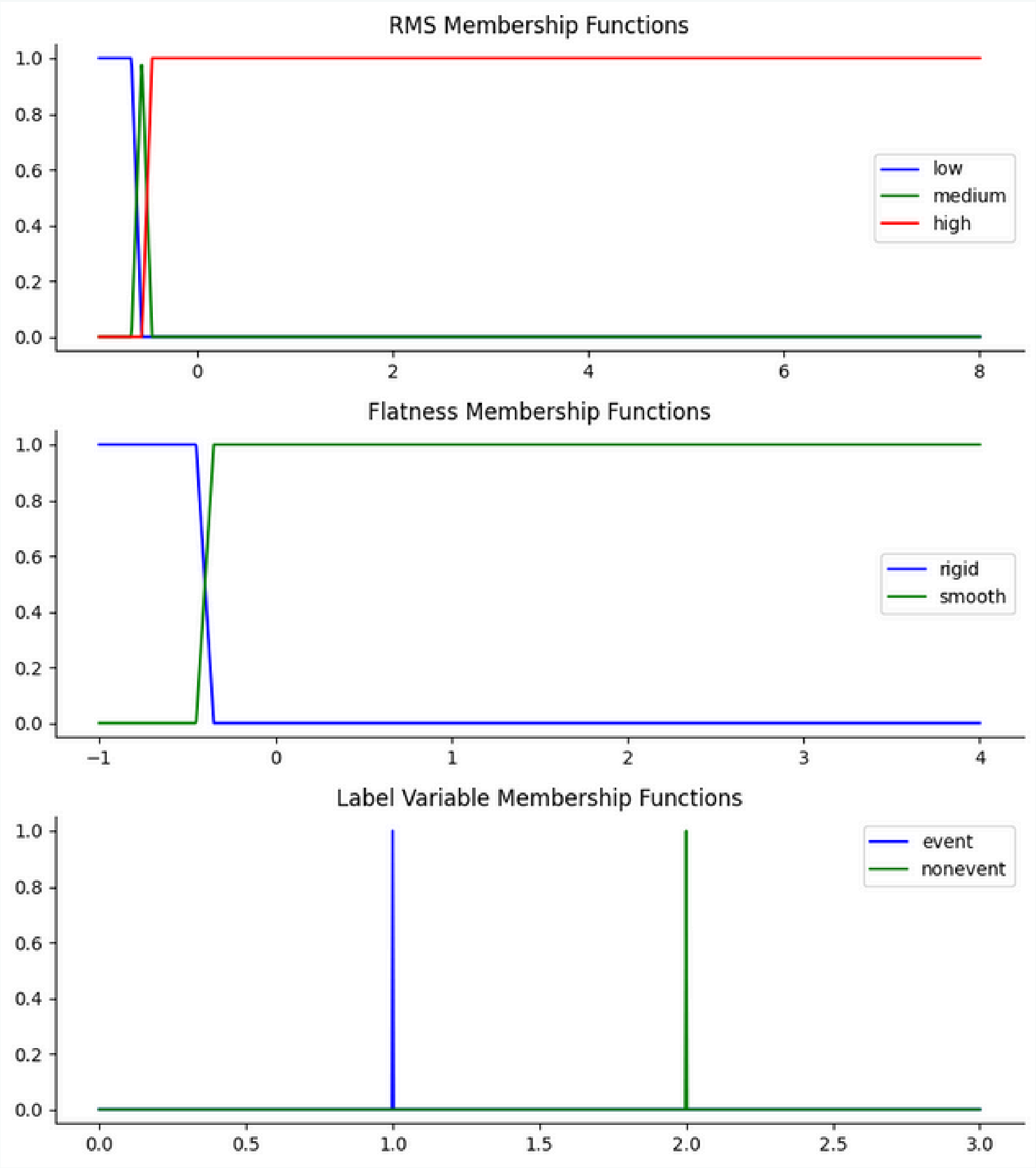


Mamdani Fuzzy Inference System

3

Fungsi Keanggotaan

Trapezoidal	(trapmf),
$[-1, -1, -0.67, -0.565]$	
Triangular	(trimf),
$[-0.67, -0.565, -0.46]$	
Trapezoidal	(trapmf),
$[-0.565, -0.46, 8, 8]$	
Trapezoidal	(trapmf),
$[-1, -1, -0.45, -0.35]$	
Trapezoidal	(trapmf),
$[-0.45, -0.35, 4, 4]$	
Triangular	(trimf),
$[1, 1, 1]$	
Triangular	(trimf),
$[2, 2, 2]$	




```python
rms['mf']['low'] = trapmf(rms['universe'], [-1, -1, -0.67, -0.565])
rms['mf']['medium'] = trimf(rms['universe'], [-0.67, -0.565, -0.46])
rms['mf']['high'] = trapmf(rms['universe'], [-0.565, -0.46, 8, 8])

flatness['mf']['rigid'] = trapmf(flatness['universe'], [-1, -1, -0.45, -0.35])
flatness['mf']['smooth'] = trapmf(flatness['universe'], [-0.45, -0.35, 4, 4])

label_var['mf']['event'] = trimf(label_var['universe'], [1, 1, 1])
label_var['mf']['nonevent'] = trimf(label_var['universe'], [2, 2, 2])
```

Mamdani Fuzzy Inference System

4

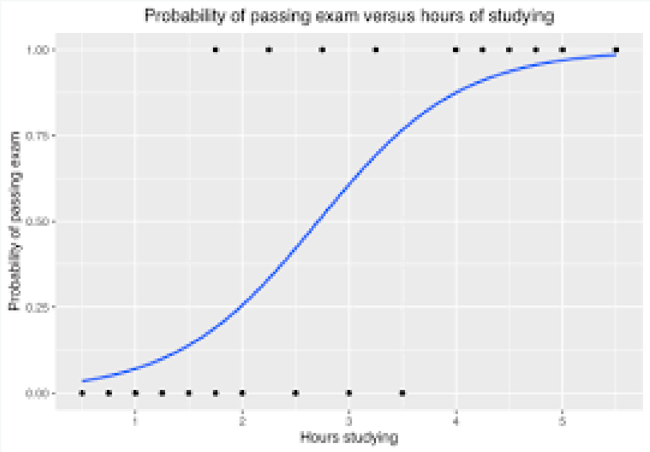
Fuzzy Rule

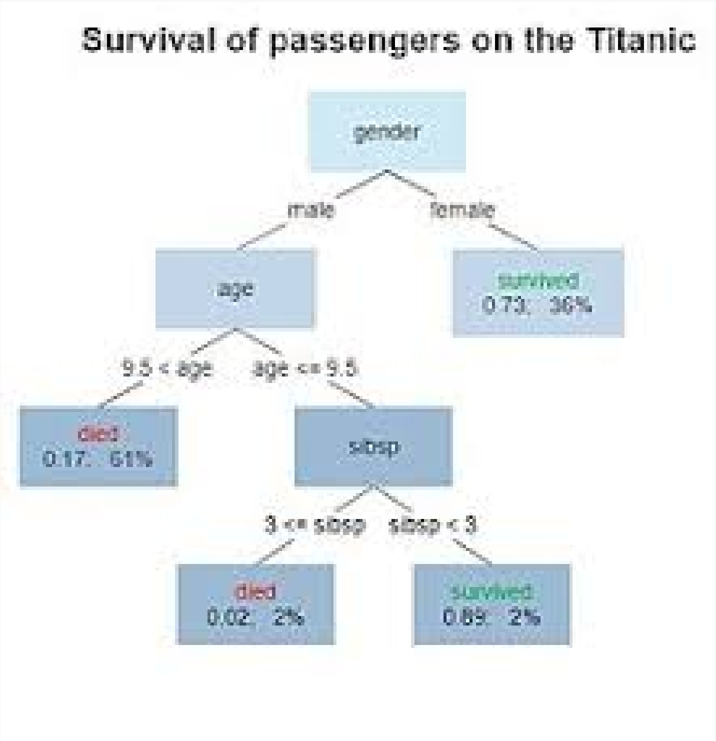
		Flatness	
		Smooth	Rigid
RMS	Low	non-event	non-event
	Medium	non-event	non-event
	High	event	event

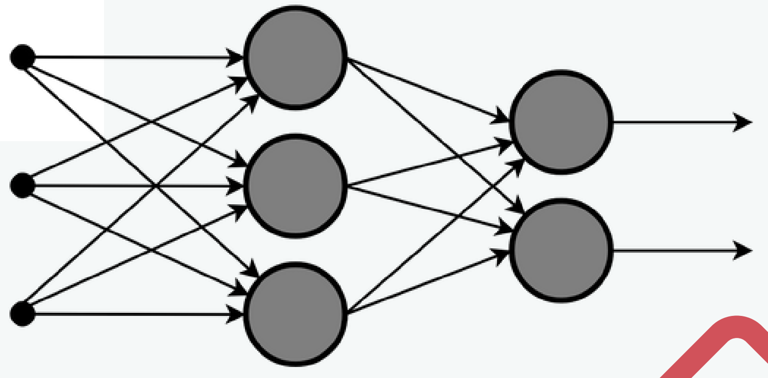
```
rule1 = Rule('smooth', 'low', 'nonevent')
rule2 = Rule('smooth', 'medium', 'nonevent')
rule3 = Rule('smooth', 'high', 'event')
rule4 = Rule('rigid', 'low', 'nonevent')
rule5 = Rule('rigid', 'medium', 'nonevent')
rule6 = Rule('rigid', 'high', 'event')
```

Supervised Model

1. Logistic Regression
2. Descision Tree
3. Light Gradient Boost Machine
4. Multi Layer Preceptron







Metrik

Metrik Evaluasi

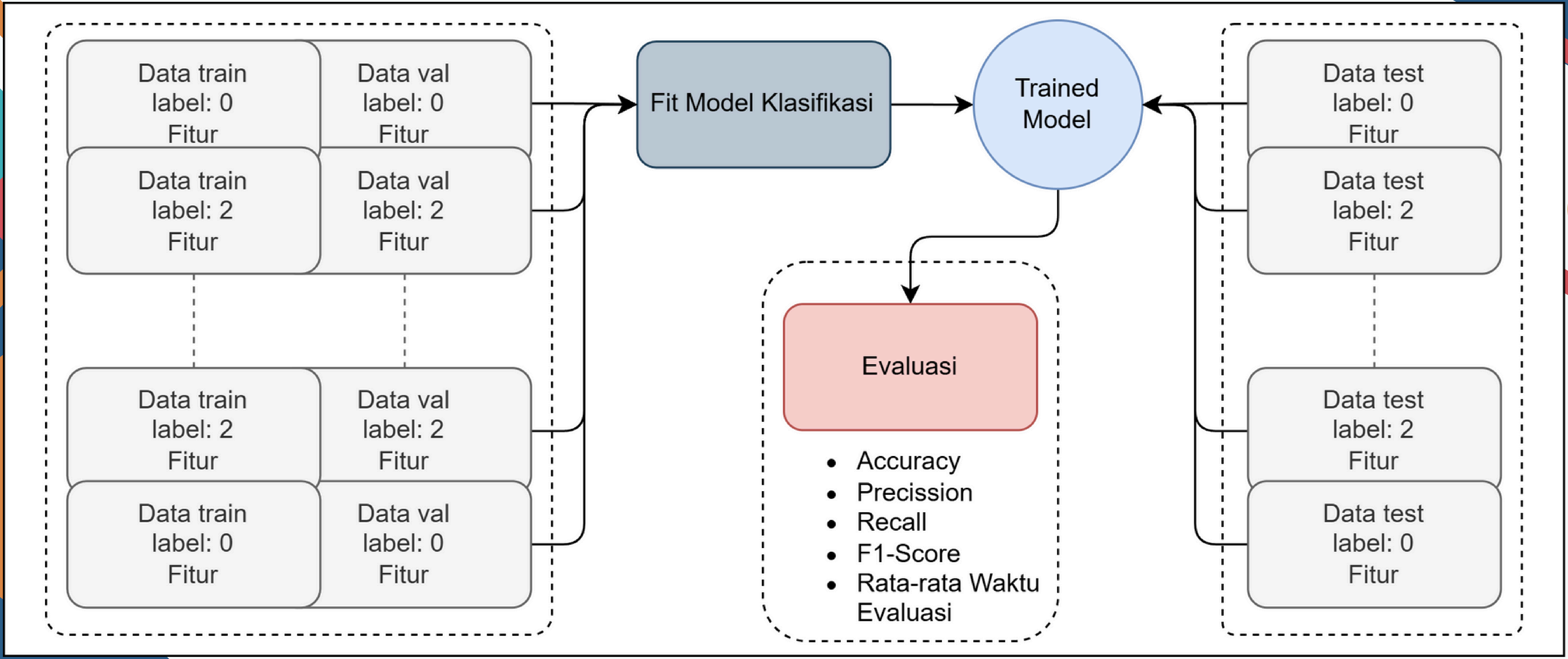
- Akurasi
- Presisi
- Recall
- F1-Score

* Presisi, Recall, dan F1-Score menggunakan Weighted Averages

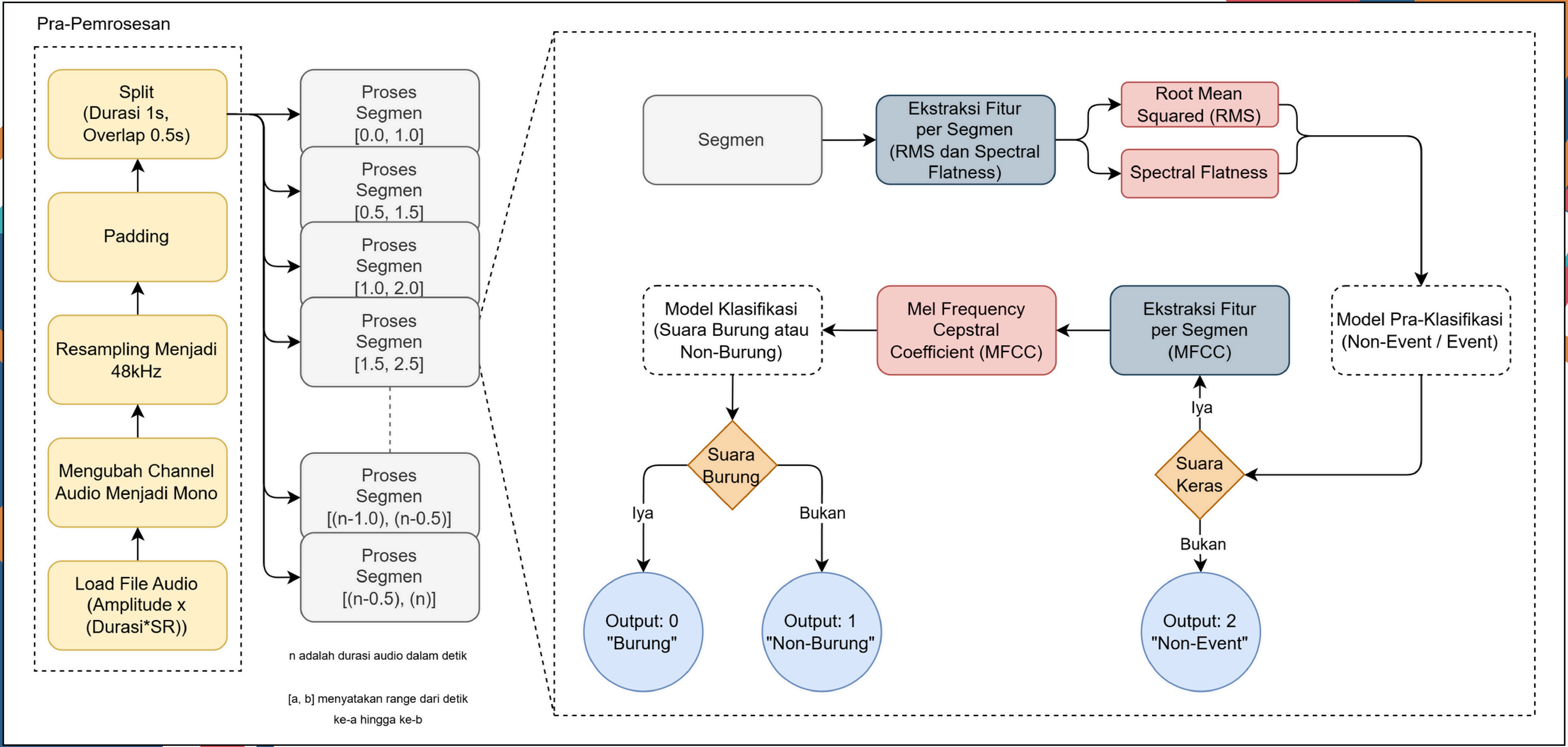
Metrik Effisiensi

- Rata-rata Rasio Runtime-Durasi
- Median Rasio Runtime-Durasi

Desain Sistem - Training



Desain Sistem - Inference



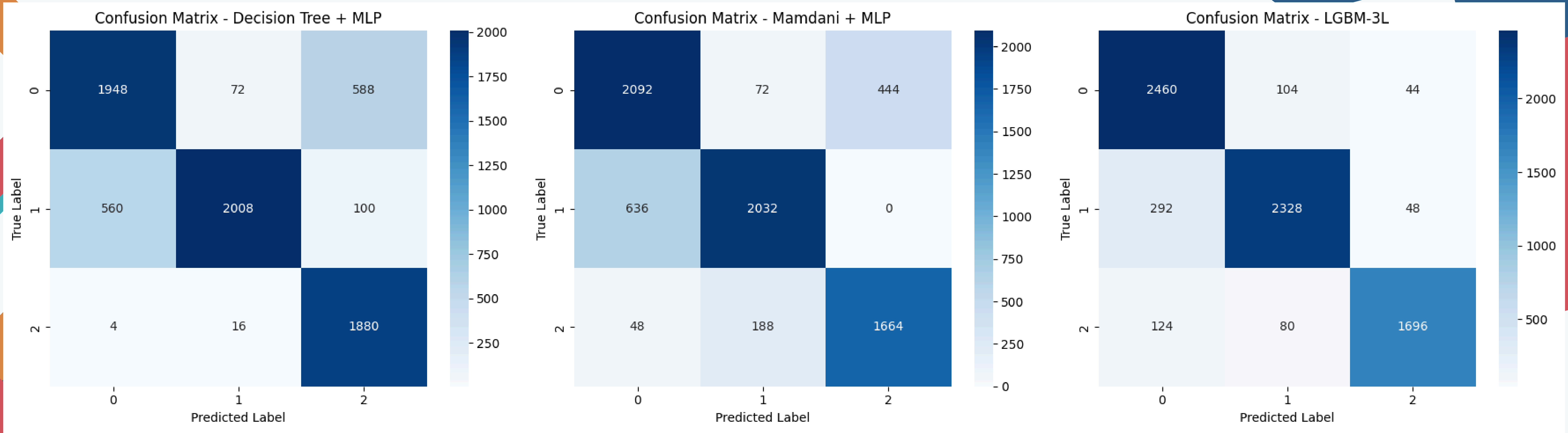
Metrik Hasil

Pra-Klasifikasi	Klasifikasi	Metrik				Waktu Eksekusi (s)	
		Akurasi	Recall	Presisi	F1-Score	Rata-rata	Median
LR	MLP-2L	0.799	0.800	0.802	0.800	0.552	0.262
M-FIS	MLP-2L	0.807	0.813	0.810	0.809	0.564	0.269
DT	MLP-2L	0.813	0.830	0.822	0.815	**0.523**	**0.239**
LR	LGBM-2L	0.826	0.824	0.824	0.824	0.610	0.265
M-FIS	LGBM-2L	0.834	0.838	0.832	0.833	0.637	0.283
DT	LGBM-2L	0.836	0.850	0.839	0.835	0.589	0.285
-	MLP-3L	0.875	0.875	0.890	0.876	0.591	0.233
-	LGBM-3L	**0.904**	**0.904**	**0.907**	**0.904**	0.675	0.289

Presisi, Recall, dan F1-Score menggunakan Weighted Averages

Waktu Eksekusi merupakan rasion (runtime)/(durasi asli)

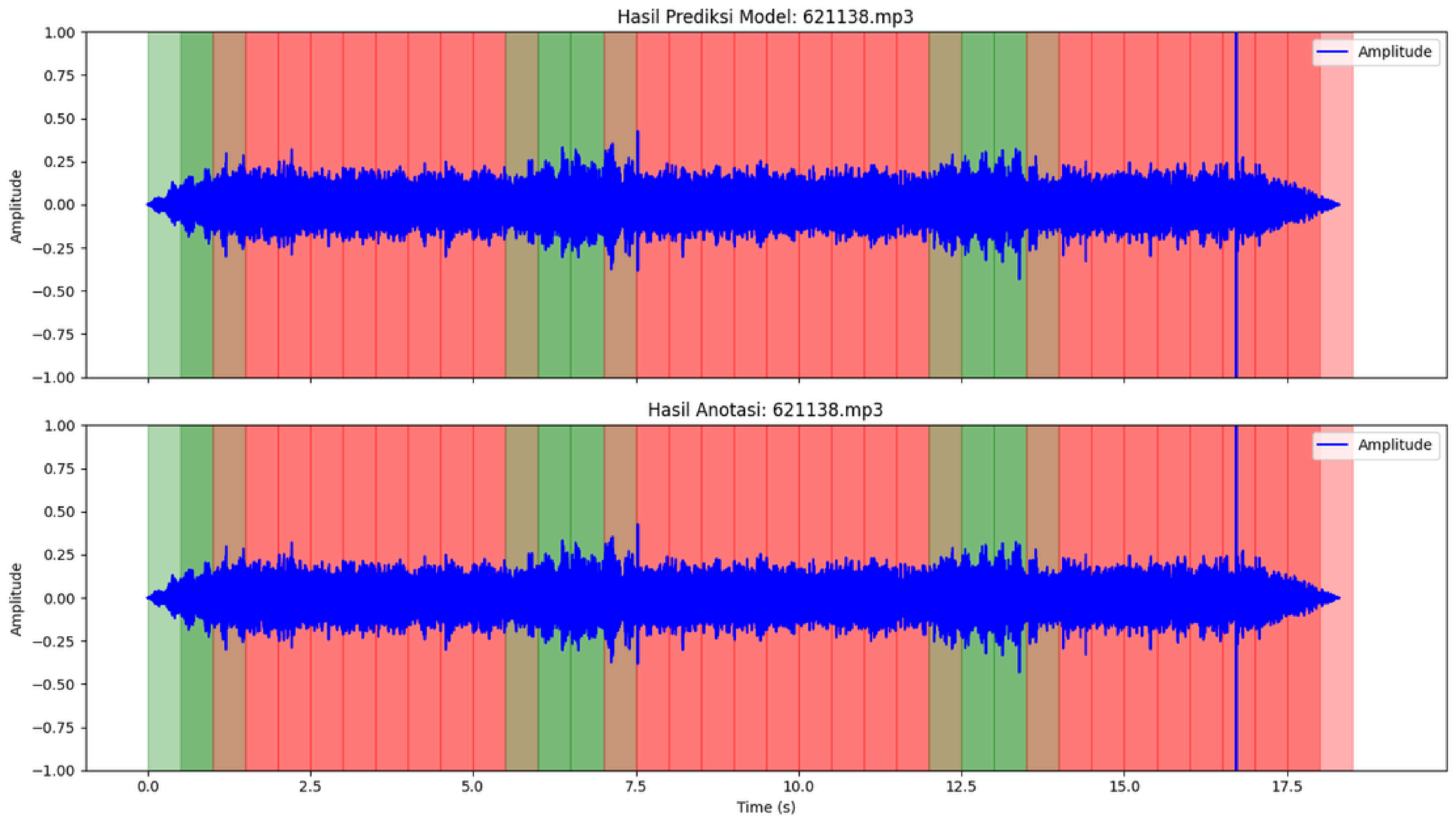
Confusion Matrix



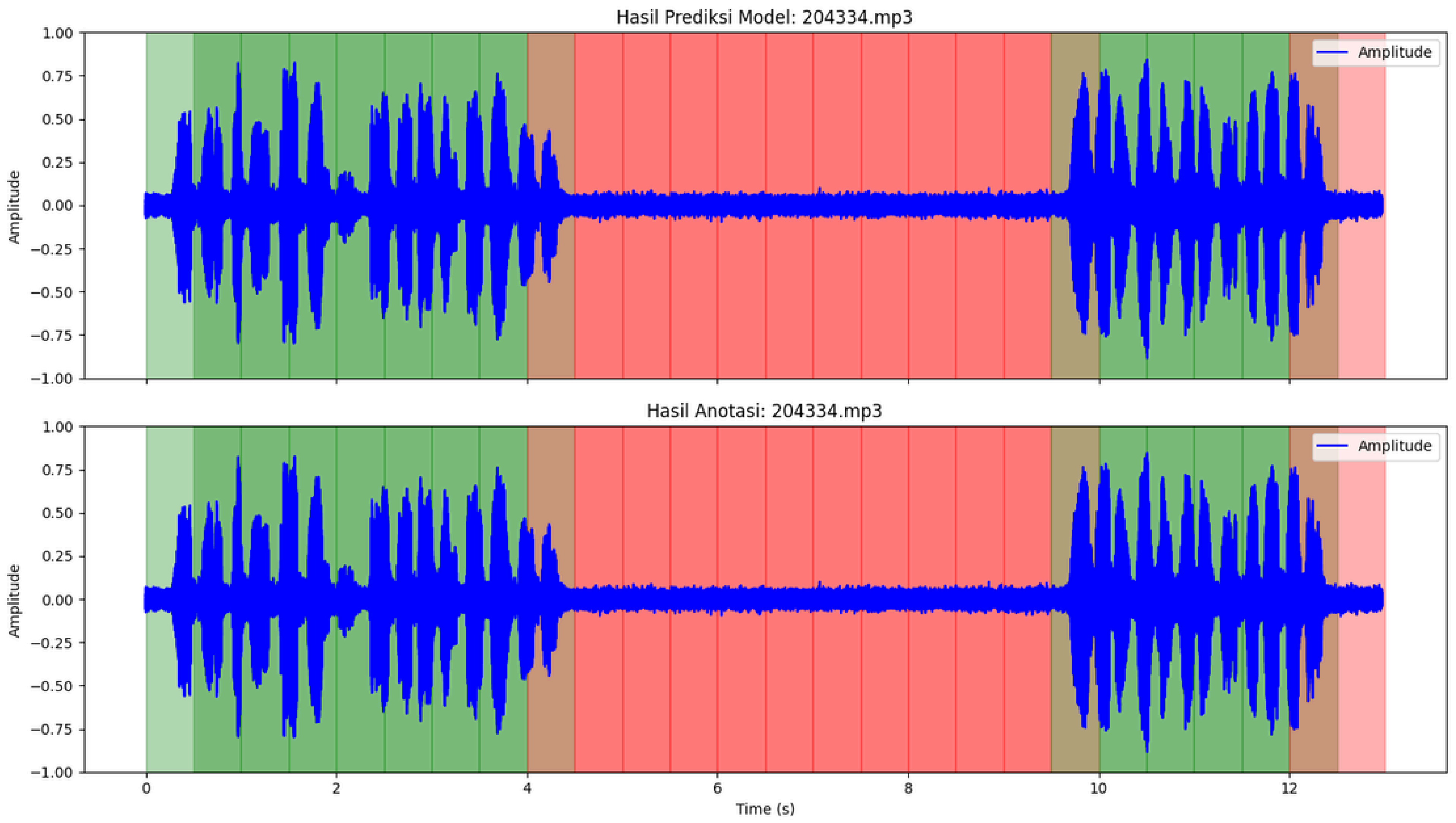
Confussion Matrix dengan label:

- 0: Suara Burung
- 1: Suara Event (Keras)
- 2: Suara Non-Event (Background)

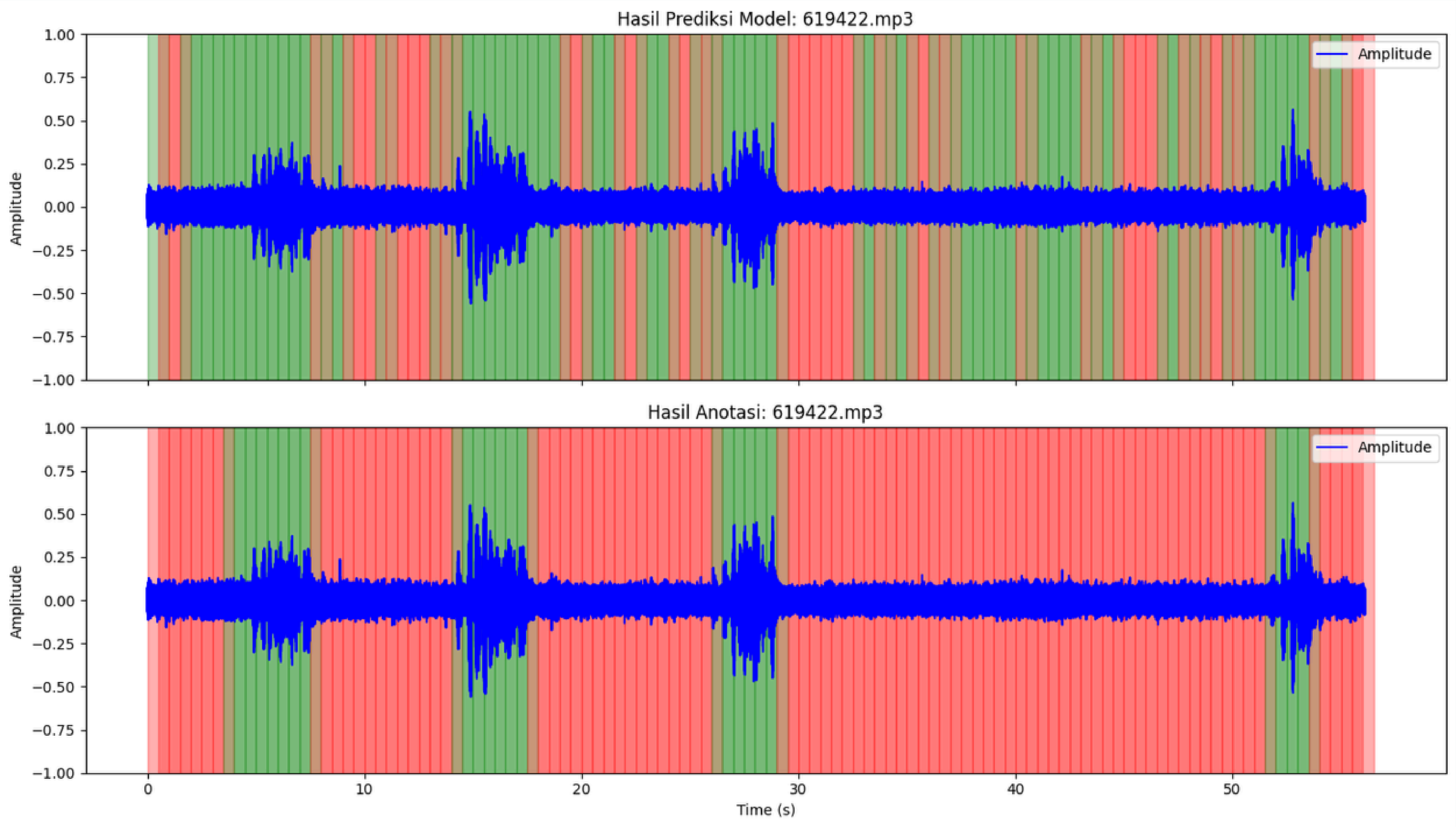
Contoh Hasil



Contoh Model akurat mengidentifikasi suara burung

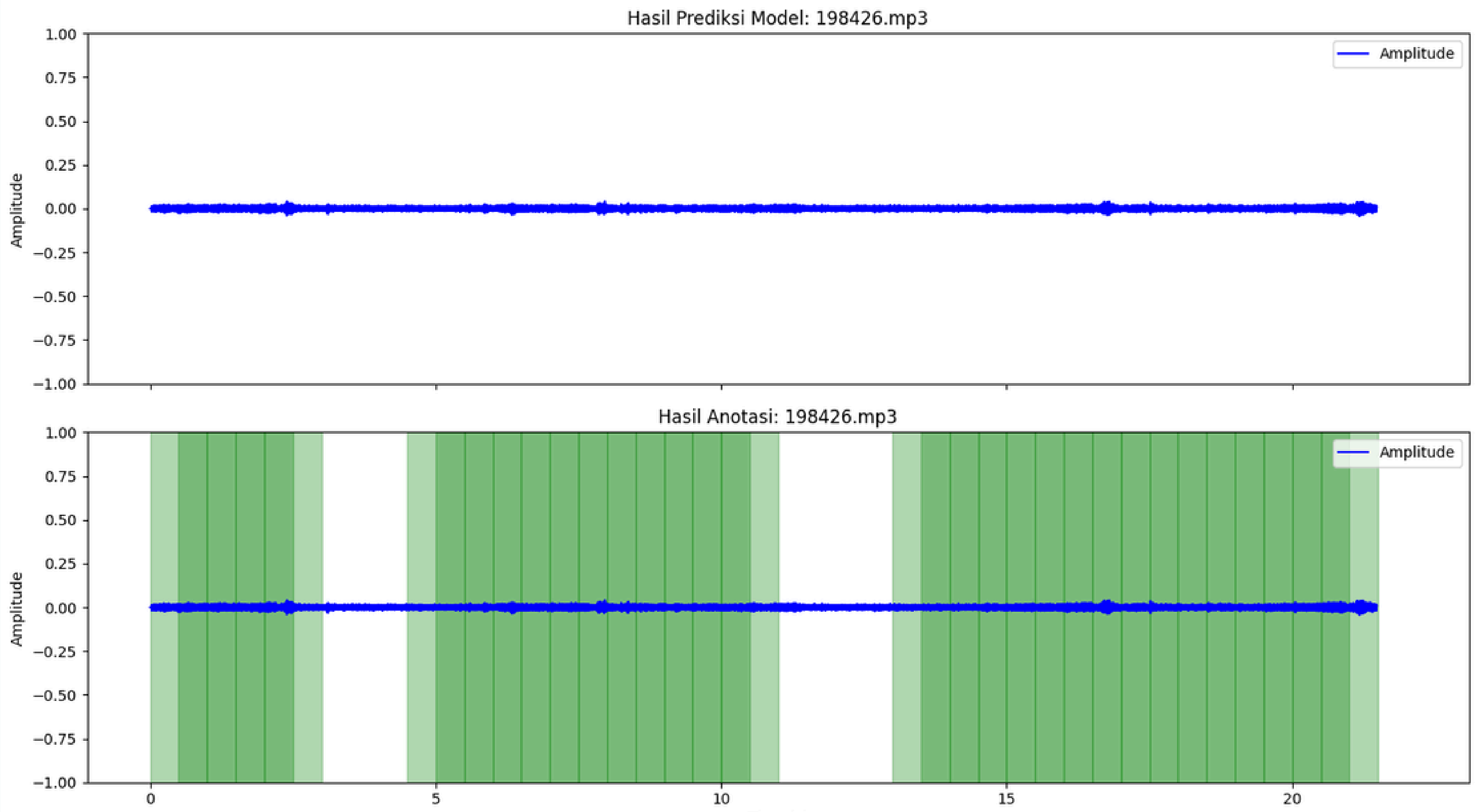


Contoh Hasil

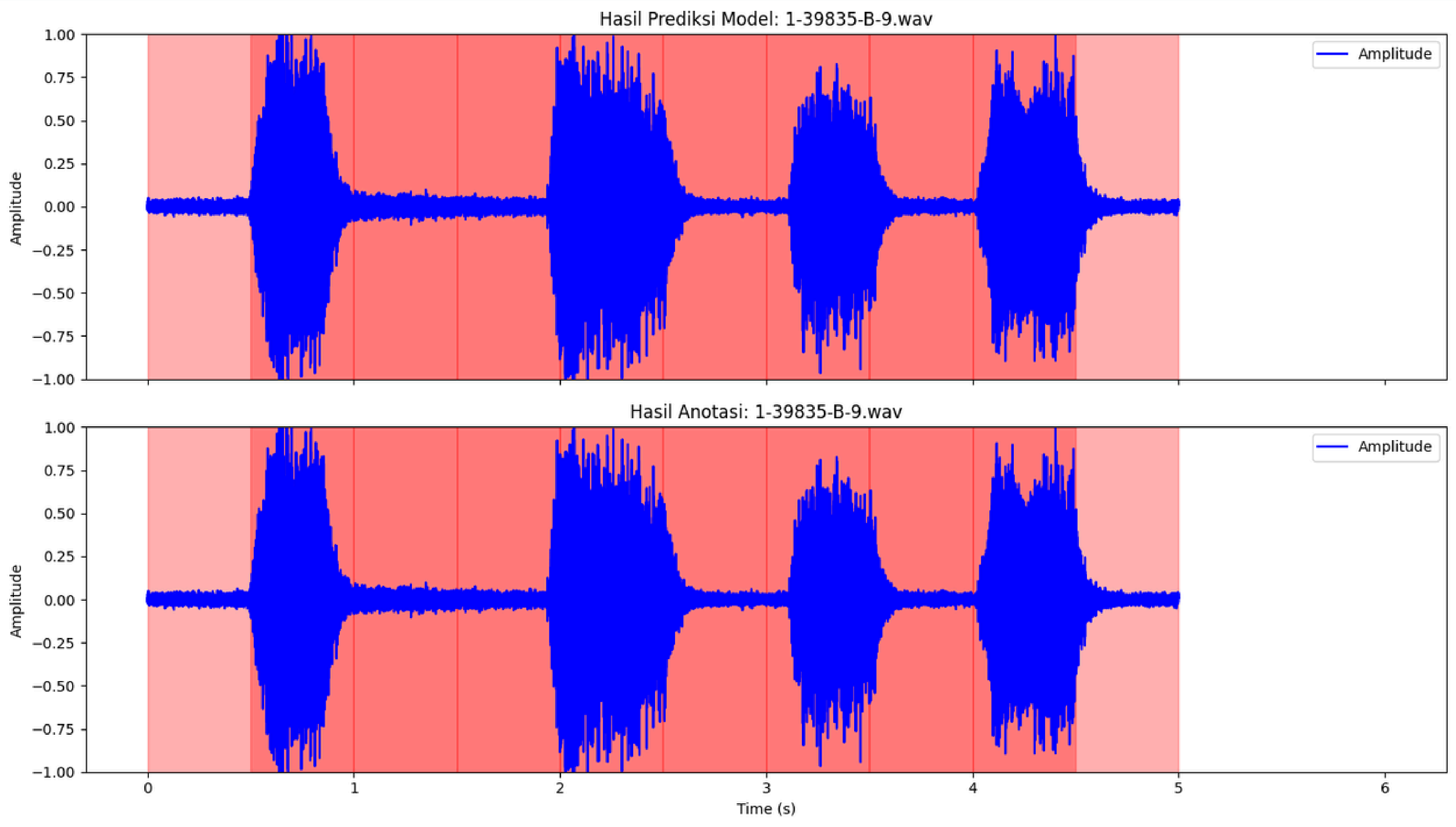


Contoh Model kesusahan membedakan suara background dan suara burung

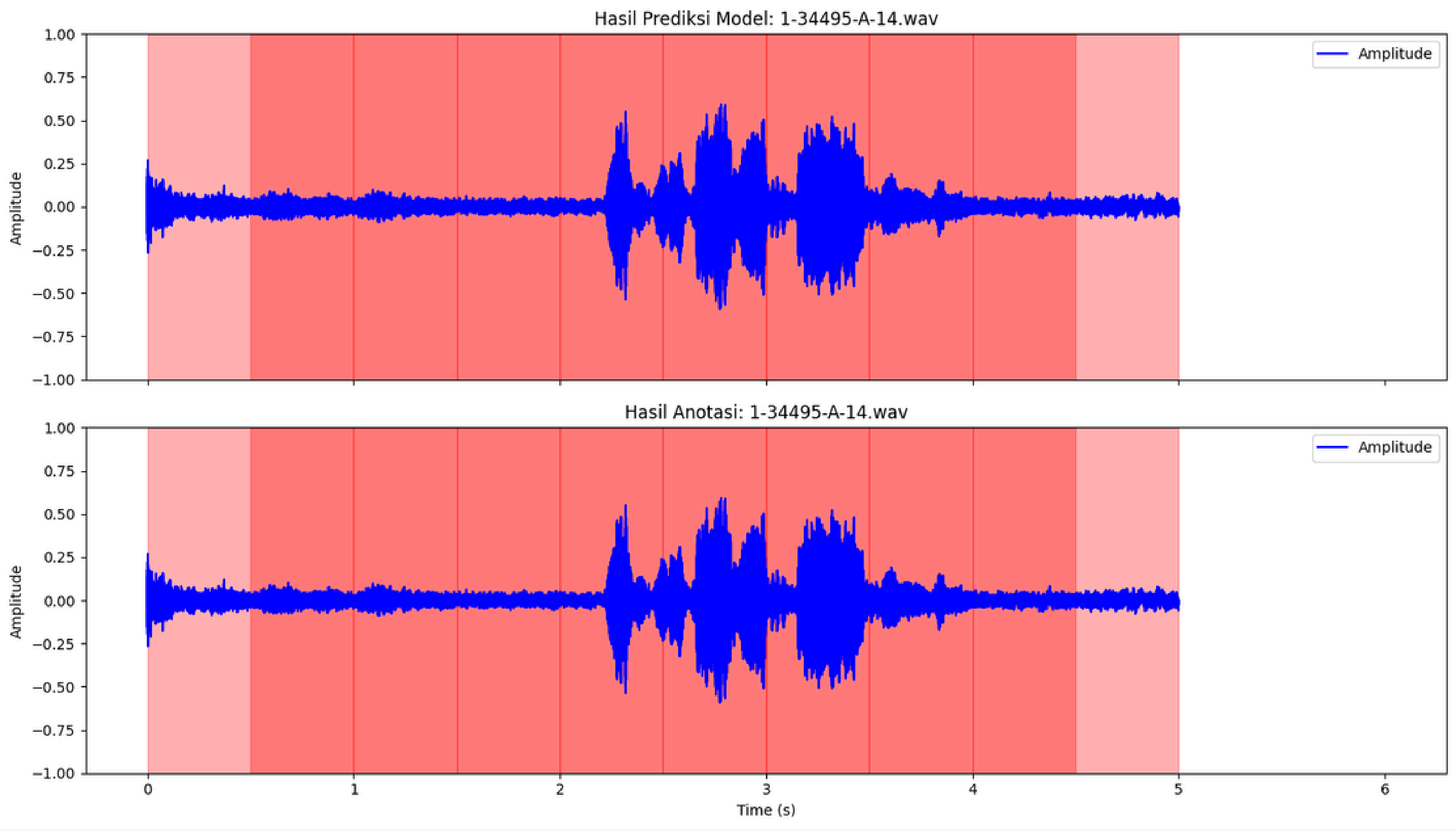
Contoh Model tidak dapat membedakan suara non-event dengan suara event / suara burung



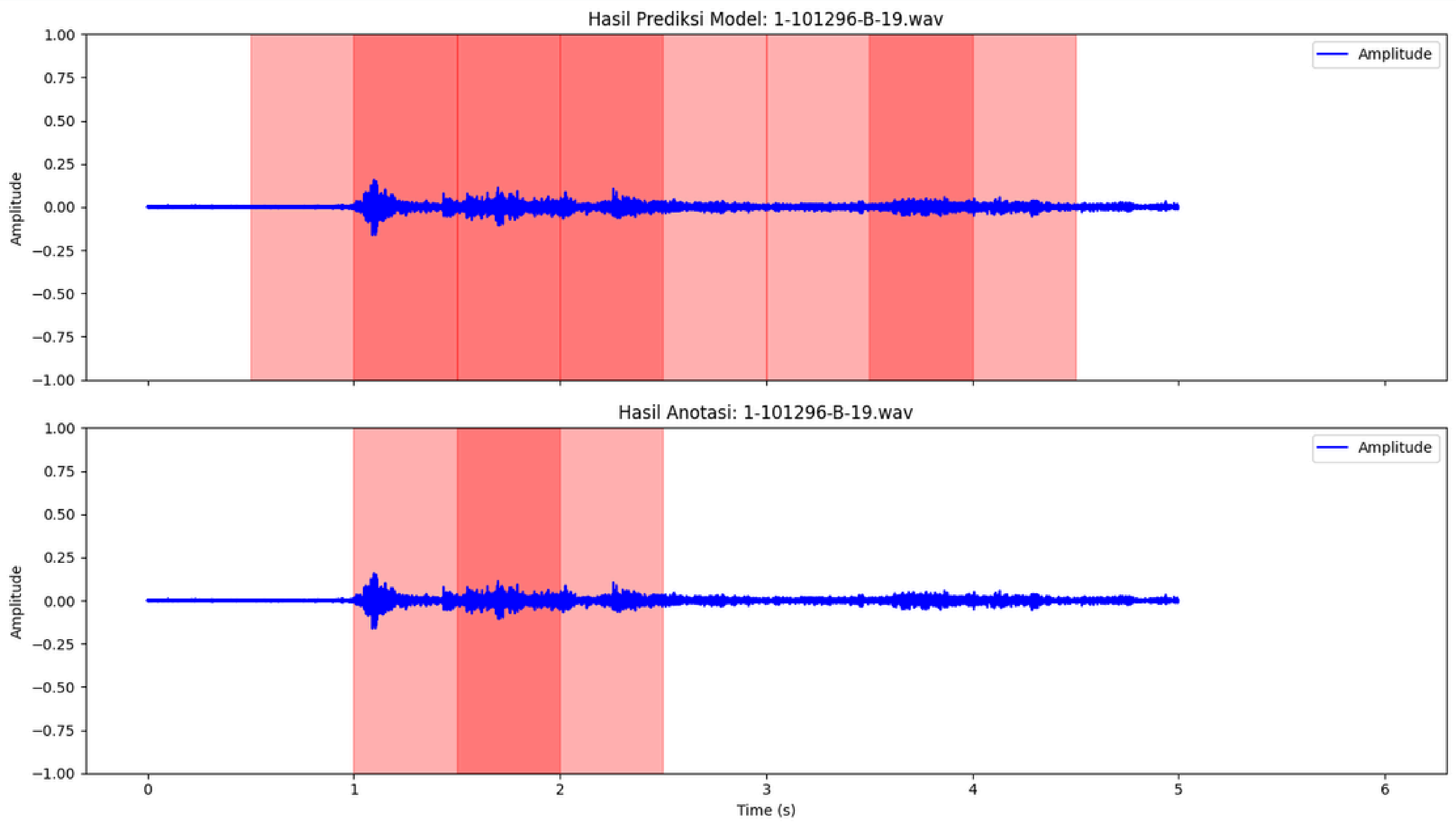
Contoh Hasil



Contoh model akurat dalam memprediksi suara event

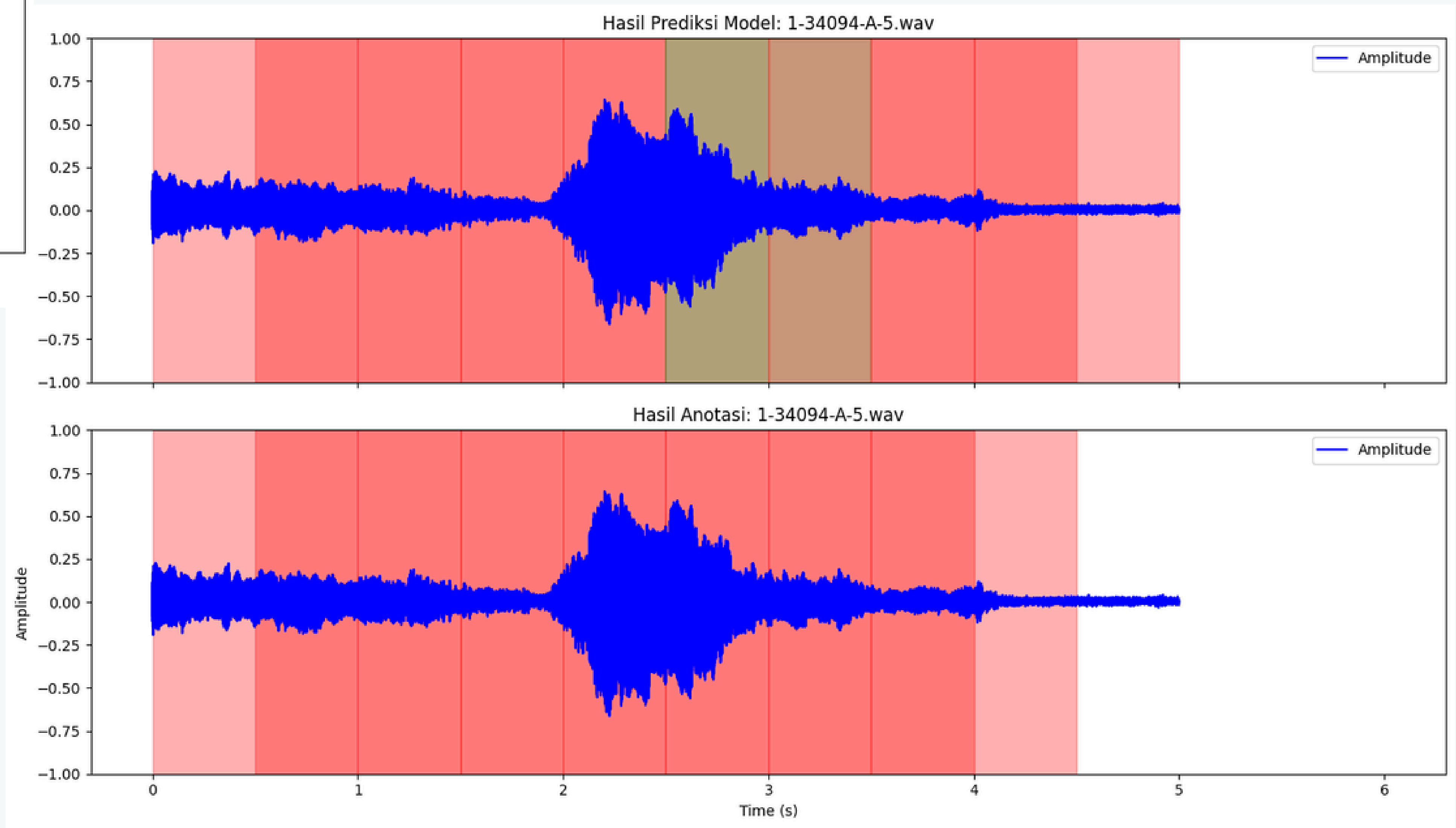


Contoh Hasil



Contoh Model terlalu general dalam mengidentifikasi suara event

Contoh Model salah memprediksi suara non-burung



Kesimpulan

- Pra-klasifikasi dengan Decision Tree berhasil menurunkan waktu eksekusi hingga 22% dibandingkan tanpa pra-klasifikasi (DT + MLP-2L: 0,523 detik vs. MLP-3L: 0,675 detik).
- Pra-klasifikasi dengan Mamdani FIS memiliki nilai akurasi 80.7% dengan MLP dan 83.4% dengan LGBM, serta menurunkan waktu sekitar 6% dibandingkan dengan tanpa pra-klasifikasi.
- Pendekatan pra-klasifikasi terbukti dapat meningkatkan efisiensi komputasi, serta memiliki akurasi yang bisa berkompetisi dengan model lain.

TERIMA KASIH

Referensi

[1] A. Lees, S. Bezeng, K. Rosenberg et al. (2022) Global bird
populations in decline, says research. Published by Manchester
Metropolitan University, May 5, 2022. [Online]. Available: https:
//www.mmu.ac.uk/news-and-events/news/story/?id=14791

[5] I. Susanto. (2025, April) Status burung di indone-
sia 2025, ada harapan di tengah ancaman. Accessed:
2025-05-20. [Online]. Available: https://www.kompas.id/artikel/
status-burung-di-indonesia-2025-ada-harapan-di-tengah-ancaman

[6] Freelancer Community. (2024, December) Understanding bird language
through ai: Unlocking the secrets of avian communication. Last Modi-
fied: 20 January 2025, Accessed: 2025-05-20.

[9] E. H. Mamdani, "Application of fuzzy algorithms for control of simple
dynamic plant," Proceedings of the Institution of Electrical Engineers,
vol. 121, no. 12, pp. 1585–1588, 1974.

[12] A. D. Putri and A. Maulana, "Penerapan metode mamdani fuzzy logic
untuk menentukan pembelian alat berat dalam proyek migas di pt smoe
indonesia," Jurnal Desain Dan Analisis Teknologi, 2023.

[13] J. Toshqorgonov, "Noise audio data," https://www.kaggle.com/datasets/
javohirtoshqorgonov/noise-audio-data, n.d., accessed: 2025-05-10.

[14] Xeno-canto, "Xeno-canto: Sharing bird sound recordings," https://
xeno-canto.org, n.d., accessed: 2025-05-10.

[15] C. Cannam, C. Landone, M. Barthet, D. Stowell, J. C. Garc´ıa,
M. Mauch, and C. Sapp, "Sonic visualiser," London, versi 5.2.1.
[Online]. Available: https://www.sonicvisualiser.org/

[16] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," Ecological Informatics, vol. 61, 2021.