



Seminar Hasil Tugas Akhir Genap 2024 – 2025

Analisis Perbandingan Feature-based dan Image-based untuk Deteksi Deepfake Speech Menggunakan Machine Learning dan Deep Learning

Laboratorium Statistika Komputasi dan Sains Data

Dosen Pembimbing
Prof. Drs. Nur Iriawan, M.Ikomp., Ph.D.
NIP 19621015 198803 1 002

Dosen co-Pembimbing
Adatul Mukarromah, S.Si., M.Si.
NIP 19800418 200312 2 001

Dosen Penguji I
Dr. Irhamah, S.Si., M.Si.
NIP 19780406 200112 2 002

Dosen Penguji II
T. Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D
NIP 19950520 202406 2 003

Disusun oleh:
Galih Fitriatmo
NRP 5003211087

SLIDE 1



www.its.ac.id/statistika

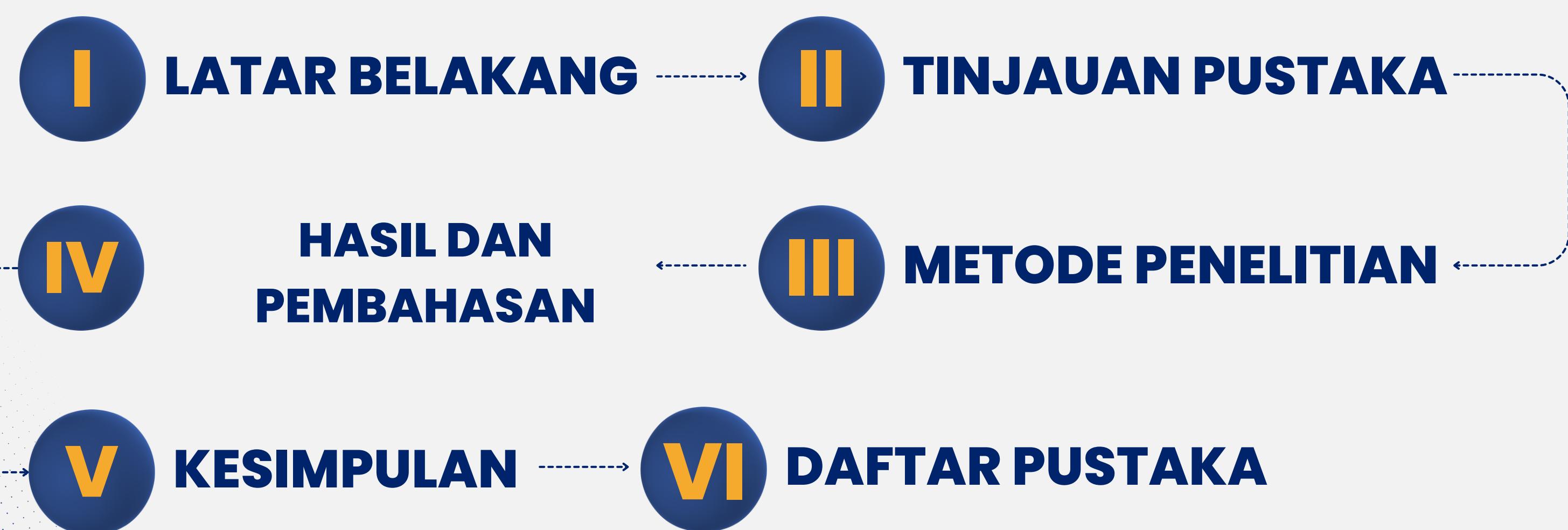


@its_statistics

INSTITUT TEKNOLOGI SEPULUH NOPEMBER, Surabaya - Indonesia



OUTLINE PEMBAHASAN





LATAR BELAKANG

SLIDE 3



www.its.ac.id/statistika



@its_statistics

INSTITUT TEKNOLOGI SEPULUH NOPEMBER, Surabaya - Indonesia

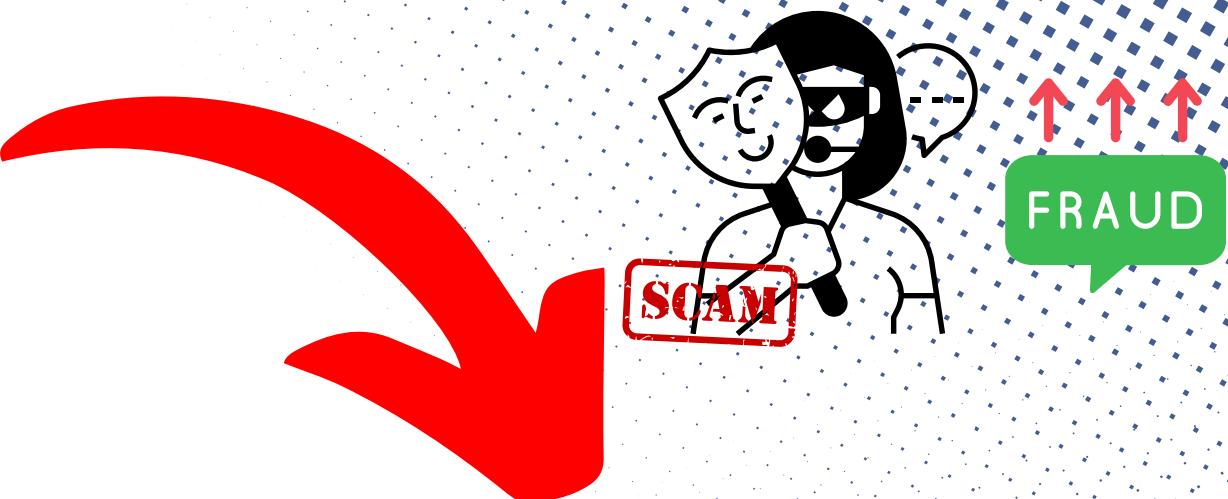
PESATNYA PERKEMBANGAN AI



DEEP FAKE SPEECH



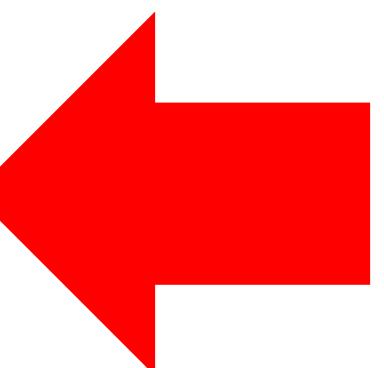
DISALAHGUNAKAN?!!



Sebagai Pengguna Medsos Pasti Familiar



penyebaran fitnah, informasi palsu, dan konten propokatif yang dapat merugikan.



Menurut data dari VIDA, mencatat bahwa terdapat **lonjakan signifikan** sebesar 1.550 % dalam kasus penipuan berbasis kecerdasan buatan di sektor keuangan Indonesia.

Data dari Bitget menyatakan bahwa angka kerugian **global** yang disebabkan oleh **deepfake** pada **Q1 2024** sebesar **US\$6,28 miliar**. Angka ini menyumbang hampir setengah dari total penipuan dengan deepfake **sepanjang 2022** sebesar **US\$13.81 miliar**.

PERLU ADANYA SOLUSI YANG EFEKTIF UNTUK MENDETEKSI DEEPCODE SPEECH



MEMBANDINGKAN PENDEKATAN KLASIFIKASI DENGAN FEATURE-BASED DAN IMAGE-BASED

FEATURE-BASED

Metode dengan pendekatan feature-based memiliki keunggulan dalam efisiensi komputasi dengan memanfaatkan berbagai fitur numerik hasil ekstraksi dari sinyal audio.

IMAGE-BASED

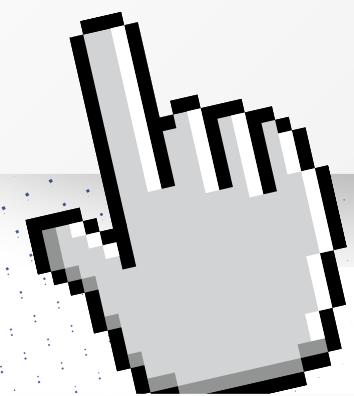
Metode dengan pendekatan image-based memiliki keunggulan dalam kemampuan menangkap pola kompleks di domain waktu-frekuensi.

(Singh, Singh, & Nathwani, 2021)

TUJUAN

1

Mengetahui **karakteristik fitur numerik** dan **fitur mel-spectrogram** antara kelas spoofed dan bonafide yang digunakan dalam klasifikasi deepfake speech.



2

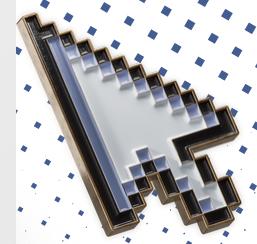
Mengevaluasi **efektivitas ketepatan klasifikasi dan efisiensi waktu prediksi** dalam mendeteksi deepfake speech menggunakan pendekatan **feature-based classifier**.

3

Mengevaluasi **efektivitas ketepatan klasifikasi dan efisiensi waktu prediksi** dalam mendeteksi deepfake speech menggunakan pendekatan **image-based classifier**.

4

Memperoleh **perbandingan performa** antara pendekatan **feature-based classifier** dan **image-based classifier** dalam efektivitas ketepatan klasifikasi dan efisiensi waktu prediksi dalam deteksi deepfake speech



MANFAAT

Penelitian ini bermanfaat untuk

- **Memberikan wawasan tentang kinerja** pendekatan feature-based dan image-based classifier dalam klasifikasi deepfake speech.
- Mendukung **pengembangan sistem deteksi yang efektif** untuk mencegah penyalahgunaan teknologi, seperti penyebaran informasi palsu, penipuan suara, dan manipulasi identitas digital

BATASAN

Penelitian ini dibatasi pada

- Data Logical Access (LA) yang diambil dari dataset ASVSpoof 2019
- Suara berbasis bahasa Inggris
- Dihasilkan dari teknik spoofing yaitu Text-to-Speech (TTS) dan Voice Conversion (VC).
- Karena keterbatasan komputasi, dilakukan undersampling pada kelas spoofed dengan rasio 2:1 terhadap bonafide untuk setiap subset data guna menurunkan beban pemrosesan.



TINJAUAN PUSTAKA

SLIDE 8



www.its.ac.id/statistika



@its_statistics

INSTITUT TEKNOLOGI SEPULUH NOPEMBER, Surabaya - Indonesia

PENELITIAN TERDAHULU DENGAN IMAGE BASED-CLASSIFIER

(Bartusiak & Delp, 2022)

- Dataset: ASV spoof 2019
- Metode: CNN
- Hasil: Menggunakan image-based approach dengan metransformasi suara menjadi spectrogram dengan **performa klasifikasi 85.99% menggunakan CNN.**

(Mcubaa, Singha, Ikuesanb, & Hein, 2023)

- Dataset: The Baidu Silicon Valley AI Lab
- Metode: FG-LCNN, ResNet, VGG-16, Custom CNN
- Hasil: Melakukan beberapa percobaan dengan menggunakan beberapa fitur ekstraksi gambar didapatkan **bahwa Mel-spectrogram lebih stabil dalam memprediksi deepfake speech** yang diterapkan pada beberapa algoritma.

(Khochare, Joshi, Yenarkar, Suratkar, & Kazi, 2021)

- Dataset: Fake or Real (FoR) Dataset
- Metode: TCN, STN
- Hasil: Menggunakan mel-spectrogram sebagai input model didapatkan hasil akurasi tertinggi yaitu **menggunakan TCN dengan akurasi 92%,** sedangkan STN sebesar 80%.

PENELITIAN TERDAHULU DENGAN FEATURE BASED-CLASSIFIER

(Khochare, Joshi, Yenarkar, Suratkar, & Kazi, 2021)

- Dataset: Fake-or-Real (For) Dataset
- Metode: SVM, RF, KNN, XGBoost, LGBM
- Hasil: Menggunakan feature based approach dengan mengekstraksi suara menjadi fitur RMSE, Spectral centroid, Spectral spread, Spectral rolloff, ZCR, Chroma (1-12), MFCC (1-20) didapatkan hasil akurasi tertinggi yaitu dengan menggunakan SVM sebesar 85%.

(Borrelli, Bestagini, Antonacci, Sarti, & Tubaro, 2021)

- Dataset: ASV Spoof 2019
- Metode: RF, SVM
- Hasil: Menggunakan fitur eks-traksi Short Term dan Long Term didapatkan akurasi sebesar **72%** dengan menggunakan **SVM**.

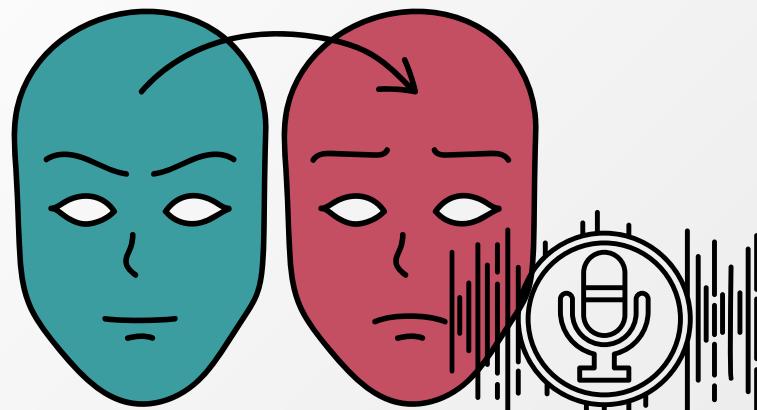
(Barua, Rahim, Parizat, Noor, & Jannah, 2021)

- Dataset: Fake-or-Real (FoR) Dataset
- Metode: SVM, CNN-LSTM
- Hasil: Menggunakan feature based approach dengan mengekstraksi suara menjadi MFCC, STFT, CQT, CENS, ZCR, Spectral Centroid, Spectral rolloff, dan RMSE didapatkan hasil terbaik yaitu dengan menggunakan **CNN-LSTM** dengan akurasi **98.33%** dan SVM sebesar 92.92%.

(Liu, Yan, Wang, Yan, & Chen, 2021)

- Dataset: IMDbTop250 Movies Dataset, QQ Music Songs dataset
- Metode: CNN, SVM
- Hasil: Menggunakan ekstraksi fitur MFCC didapatkan bahwa metode **CNN memiliki performa yang lebih baik dibanding dengan metode SVM**.

Deepfake Speech



Deepfakes adalah media sintetis yang dihasilkan menggunakan metode deep learning. Berdasarkan tipe deepfake speech:
(Korshunov & Marcel, 2018)

Text To Speech

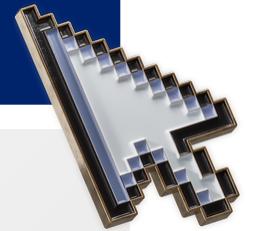
Pada TTS generation,

- Model dilatih untuk **menghasilkan audio menggunakan teks sebagai input**
- Memungkinkan komputer untuk **menghasilkan suara** yang terdengar seperti suara manusia, menggunakan teks tertulis sebagai sumber.

Voice Conversion

Sementara itu, voice conversion

- Melibatkan **konversi dari satu suara ke suara lainnya**.
- Mengubah karakteristik suara, tanpa mengubah konten percakapan yang disampaikan (Almutairi & Elgibreen, 2022).



ASVspoof 2019

- ASVspoof 2019 adalah dataset dan kompetisi internasional untuk mengembangkan sistem anti-spoofing pada verifikasi suara otomatis (ASV).
- Dataset Logical Access (LA) fokus pada serangan digital berbasis Text-to-Speech (TTS) dan Voice Conversion (VC). Terbagi menjadi 3 subset:
 - Train & Validation: 6 known attacks (A01–A06)
 - Test: 13 unknown attacks (A07–A19) → **uji kemampuan generalisasi**
- Prinsip **disjoint speaker** antar subset → tidak ada speaker yang tumpang tindih.
- Dirancang agar sistem anti-spoofing tidak hanya mengenali suara spesifik, tetapi juga pola serangan secara umum.

Attack ID	Jenis Serangan	Metode Teknologi
A01	TTS	<i>Neural waveform model</i>
A02	TTS	<i>Vocoder</i>
A03	TTS	<i>Vocoder</i>
A04	TTS	<i>Waveform concatenation</i>
A05	VC	<i>Vocoder</i>
A06	VC	<i>Spectral filtering</i>
A07	TTS	<i>Vocoder + GAN</i>
A08	TTS	<i>Neural waveform</i>
A09	TTS	<i>Vocoder</i>
A10	TTS	<i>Neural waveform</i>
A11	TTS	<i>Griffin-Lim</i>
A12	TTS	<i>Neural waveform</i>
A13	TTS_VC	<i>Waveform concatenation + filtering</i>
A14	TTS_VC	<i>Vocoder</i>
A15	TTS_VC	<i>Neural waveform</i>
A16	TTS	<i>Waveform concatenation</i>
A17	VC	<i>Waveform filtering</i>
A18	VC	<i>Vocoder</i>
A19	VC	<i>Spectral filtering</i>

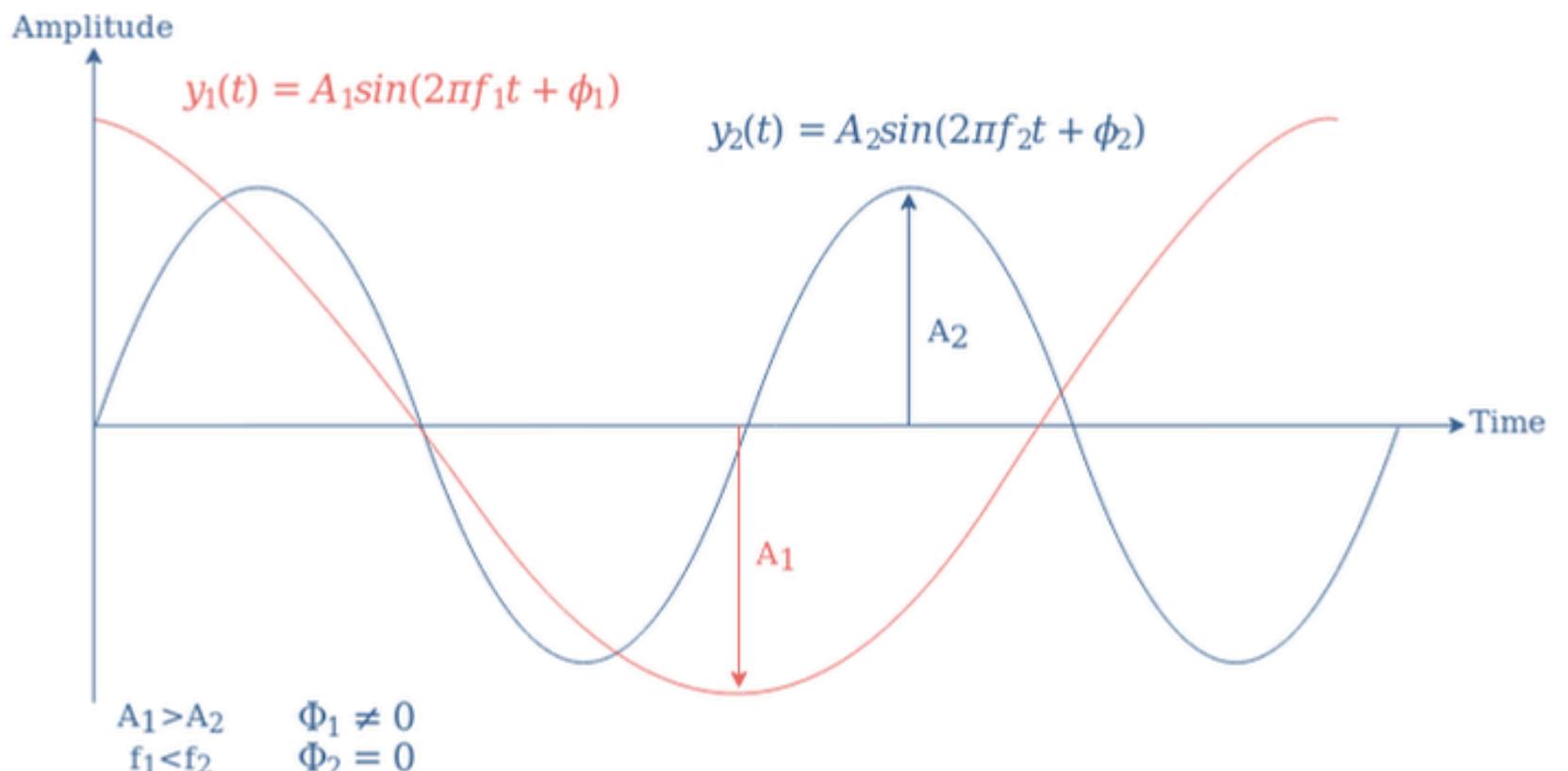
Sinyal Audio

Sinyal audio sering kali direpresentasikan sebagai **kombinasi dari berbagai sinyal sinusoidal**, di mana setiap komponen memiliki frekuensi, amplitudo, dan fase yang berbeda (Proakis & Manolakis, 2006)

$$a(t) = A \sin(2 \pi k t + \varphi).$$

Keterangan:

- $a(t)$: Nilai amplitudo sinyal pada saat ke- t
- A : Amplitudo sinyal
- \sin : Fungsi sinus
- $2\pi k$: Komponen frekuensi sudut
- t : Waktu dalam detik
- φ : Fase



Feature Extraction

Mel-Spectrogram

Skala Mel adalah sebuah skala yang dikembangkan untuk mengukur persepsi manusia terhadap frekuensi suara (Stevens & Volkmann, 1940).

Transformasi Fourier Jangka Pendek

STFT memungkinkan analisis sinyal dalam domain waktu dan frekuensi secara bersamaan.

$$X(k, n_f) = \sum_{m=0}^{M-1} a(m + n_f H) \cdot \omega(m) \cdot e^{-\frac{i2\pi mk}{M}}$$

1

Mendapatkan Spektrum Daya

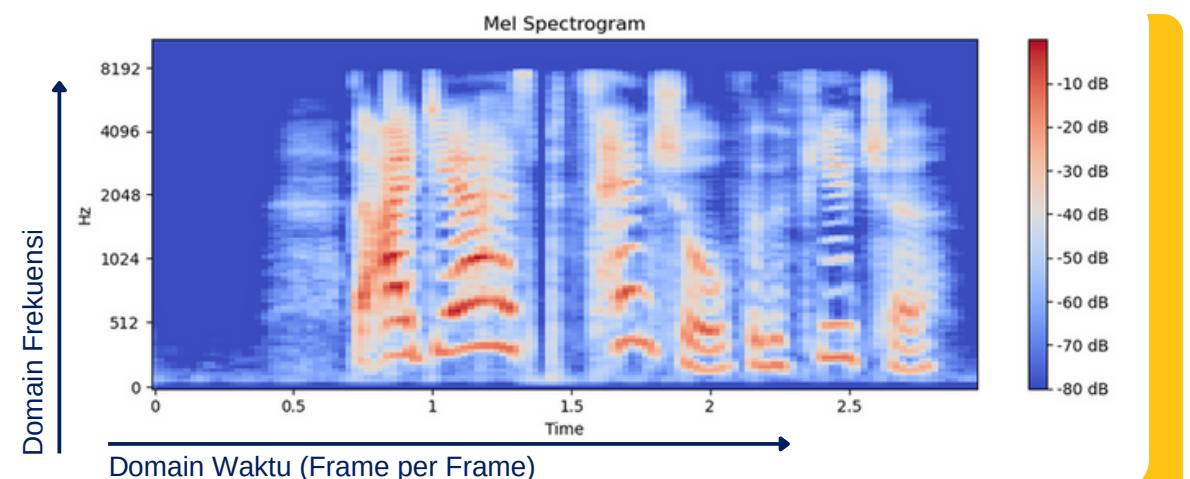
$$Y(k, n_f) = |X(k, n_f)|^2,$$

2

Penentuan Jumlah Mel-bands dan Pemetaan ke skala Mel

$$m_S(k) = 2595 \cdot \log_{10} \left(1 + \frac{k}{700 \text{ Hz}} \right)$$

3



Transformasi Logaritmik

$$\text{LogMel}(k', n_f) = \log(|X'(k', n_f)|^2 + \epsilon)$$

5

Penerapan Mel-Filter Bank

$$|X'(k', n_f)| = \sum_{k=0}^{K-1} |X(k, n_f)| \cdot H(k, k').$$

4

Feature Extraction

Chromagram

Chromagram yaitu representasi distribusi energi suara berdasarkan 12 nada musik (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) dalam satu oktaf.

$$v_{cm}(j_{cm}, n_f) = \sum_{o=1}^{o_u} \left(\frac{1}{k_u(o, j_{cm}) - k_l(o, j_{cm}) + 1} \sum_{k=k_l(o, j_{cm})}^{k_u(o, j_{cm})} |X(k, n_f)| \right),$$

Spektrum daya yang dihasilkan dari STFT kemudian dipetakan dipetakan ke pita kromatik (j_{cm}) (Lerch, 2012).

Spectral Centroid

Spectral centorid (SC) adalah fitur yang **menunjukkan lokasi pusat massa energi dalam spektrum frekuensi**, memberikan gambaran di mana energi sinyal terkonsentrasi (Lerch, 2012).

$$v_{sc}(n_f) = \frac{\sum_{k=0}^{K-1} k \cdot |X(k, n_f)|^2}{\sum_{k=0}^{K-1} |X(k, n_f)|^2}$$

Keterangan:

$X(k, n_f)$: Koefisien fourier pada frekuensi ke- k dan frame ke- n
k	: Urutan frekuensi ke- k
K	: Jumlah frekuensi total dalam audio
n_f	: Urutan frame ke- n
N_f	: Jumlah frame dalam audio

Mel-Frequency Cepstral Coefficients (MFCCs)

Menghasilkan koefisien numerik (fitur suara) yang **merekpresentasikan karakteristik utama sinyal suara dalam domain Mel Cepstral** (Lerch, 2012).

$$v_{MFCC}^j(n_f) = \sum_{k'=1}^{K'} \log(|X'(k', n_f)|) \cdot \cos\left(j \cdot \left(k' - \frac{1}{2}\right) \frac{\pi}{K'}\right)$$

M	: Jumlah sampel dalam <i>frame</i>
m	: Urutan sampel ke- m
H	: Panjang <i>hop</i> (jarak antar window)
$\omega(m)$: Fungsi <i>windowing</i> pada sampel ke- m
$Y(k, n_f)$: Spektrum daya pada frekuensi ke- k dan <i>frame</i> ke- n
$ X(k, n_f) $: Magnitudo spektrum asli pada frekuensi ke- k dan <i>frame</i> ke- n
$H(k, k')$: Respon filter mel untuk frekuensi ke- k pada filter ke- k'
k'	: Urutan filter mel ke- k'
$\text{LogMel}(k', n_f)$: Transformasi logaritmik spektrum daya dalam skala mel untuk filter ke- k'
$v_{MFCC}^j(n_f)$: Nilai koefisien MFCC ke- j pada <i>frame</i> ke- n
K'	: Jumlah total filter dalam <i>mel filter bank</i>
$v_{sc}(n_f)$: Nilai <i>spectral centroid</i> pada <i>frame</i> ke- n

Feature Extraction

Spectral Rolloff

Spectral rolloff (SR) didefinisikan sebagai **indeks frekuensi di mana akumulasi magnitudo mencapai persentase tertentu (R) dari jumlah total magnitudo** (Lerch, 2012).

$$v_{SR}(n_f) = \rho \left| \sum_{k=0}^{\rho} |X(k, n_f)| \right| = R \cdot \sum_{k=0}^{K-1} |X(k, n_f)|,$$

Spectral Spread

Spectral spread (SS) mengukur penyebaran kekuatan spektrum di sekitar **Spectral centorid** (Lerch, 2012).

$$v_{SS}(n_f) = \sqrt{\frac{\sum_{k=0}^{K-1} (k - v_{SC}(n_f))^2 \cdot |X(k, n_f)|^2}{\sum_{k=0}^{K-1} |X(k, n_f)|^2}}$$

$v_{SS}(n_f)$: Nilai *spectral spread* pada frame ke- n

$v_{SR}(n_f)$: Nilai *spectral rolloff* pada frame ke- n

R : Faktor skala spektral

ρ : Indeks frekuensi ke- ρ yang memenuhi persamaan

Zero Crossing Rate

Zero crossing rate (ZCR) mengukur **seberapa sering sinyal audio melewati nilai nol** (positif ke negatif atau sebaliknya) tanda (suara bernada tinggi atau noise), sedangkan nilai rendah (Bird & Lotfi, 2023).

$$ZCR(n_f) = \frac{1}{2M} \sum_{m=1}^{M-1} |sgn(a[m]) - sgn(a[m-1])|,$$

Root Mean Square Energy

Root mean square energy (RMS) adalah metrik untuk **mengukur energi rata-rata pada suatu sinyal** (Bird & Lotfi, 2023).

$$RMS(n_f) = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} a[m]^2},$$

ρ

$ZCR(n_f)$

$sgn()$

$RMS(n_f)$

: Indeks frekuensi ke- ρ yang memenuhi persamaan

: Nilai *zero crossing rate* pada frame ke- n

: Fungsi tanda

: Nilai *root mean square energy* pada frame ke- n

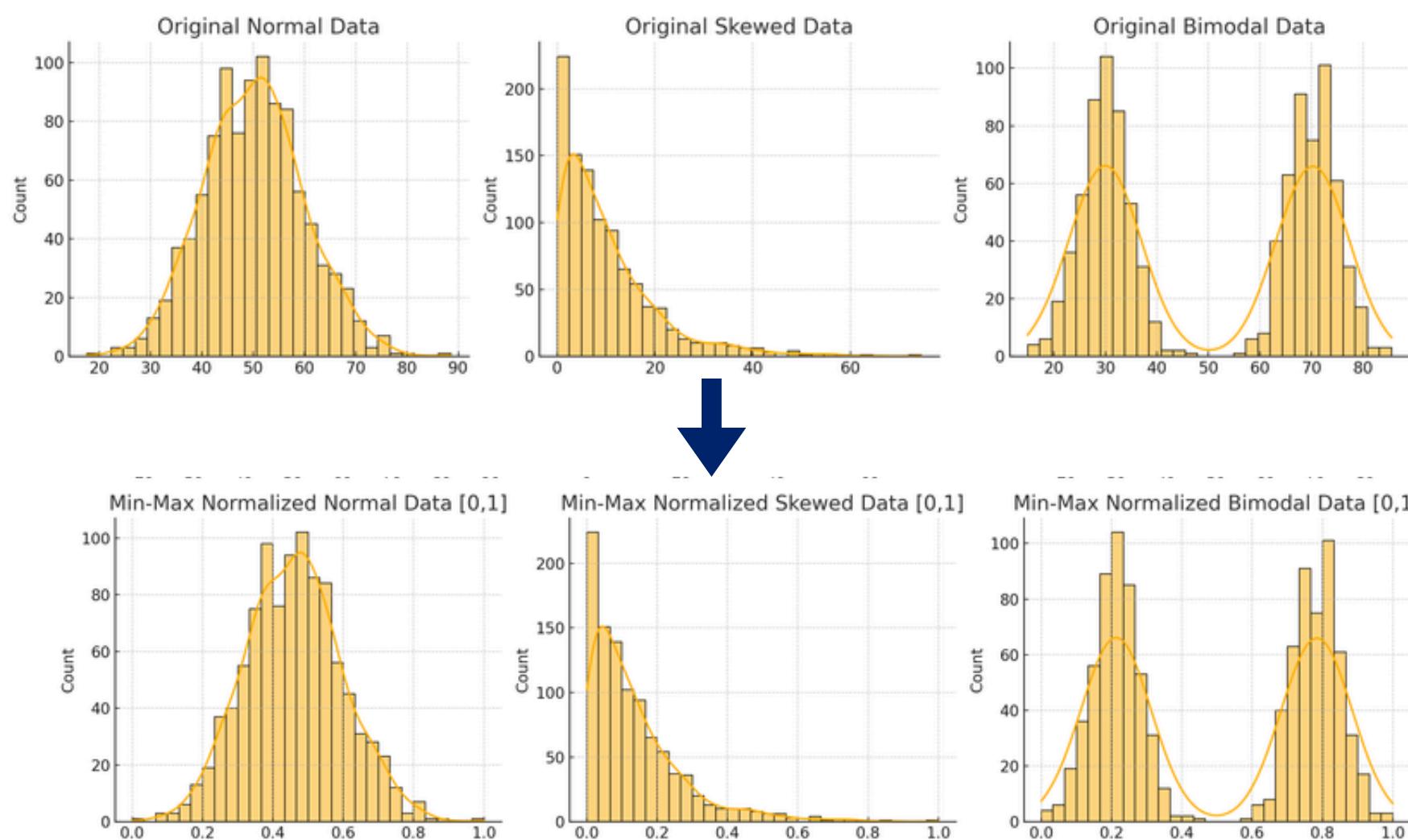
Normalisasi Data

Mengubah skala agar berada dalam interval nilai tertentu (Han, Kamber, & Pei, 2012).

Min-Max Normalization

Mengubah rentang menjadi [new_min, new_max]

$$x'_{mm} = \frac{x_i - \min(x)}{\max(x) - \min(x)} (new_max - new_min) + new_min$$



Z-Test Two-Sampel Independent

Metode ini sering digunakan untuk mengetahui apakah dua kelompok berbeda memiliki rata-rata yang berbeda secara signifikan dalam berbagai konteks analisis data (Walpole, Myers, Myers, & Ye, 2012).

Syarat : Varians populasi diketahui / $n>30$ (CLT)

Hipotesis

Hipotesis nol (H_0): $\mu_1 = \mu_2$, yang menyatakan bahwa tidak terdapat perbedaan rata-rata antara dua populasi.

Hipotesis alternatif (H_1): $\mu_1 \neq \mu_2$, yang menyatakan bahwa terdapat perbedaan rata-rata antara dua populasi.

Statistik Uji

$$Z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Keputusan

Tolak H_0 jika nilai Z hitung lebih besar dari Z tabel ($Z > Z_a$) atau lebih kecil dari -Z tabel ($Z < -Z_a$).

Feature Selection

Feature selection adalah teknik memilih subset fitur dari seluruh fitur yang tersedia, dengan tujuan meningkatkan kinerja prediktif model, mengurangi waktu komputasi, serta meningkatkan interpretabilitas model yang dihasilkan (Chandrashekhar & Sahin, 2014)

RFE

Recursive Feature Elimination (RFE) adalah metode seleksi fitur yang bekerja secara iteratif dengan melatih model secara berulang dan menghapus fitur dengan kontribusi paling rendah pada setiap langkah, hingga diperoleh subset fitur yang paling relevan terhadap target.

3

Dipilih jumlah fitur yang menghasilkan akurasi tertinggi. Fitur yang terpilih merupakan subset fitur paling relevan dalam pemodelan deepfake speech.

1

Model dilatih dengan keseluruhan fitur kemudian dilakukan klasifikasi pada data validasi dan dicatat akurasinya. Proses ini dilakukan dengan crossvalidation sebanyak 5 kali dan repetition sebanyak 6 kali, total 30 kali proses.

2

Fitur dengan tingkat kepentingan paling kecil dieliminasi dan proses pelatihan dan klasifikasi diulangi. Proses ini dilakukan hingga semua fitur tereliminasi.

Support Vector Machine (SVM)

Ide utama dari SVM adalah **menemukan fungsi pemisah atau hyperplane**, yang dapat membagi dua kelas dengan cara yang paling optimal (Hindle, Prasetyo, & Hafner, 2012).

Support Vector Machine Linear

Fungsi Pemisah Hyperplane:

$$\mathbf{w}^T \mathbf{x} + b = 0$$

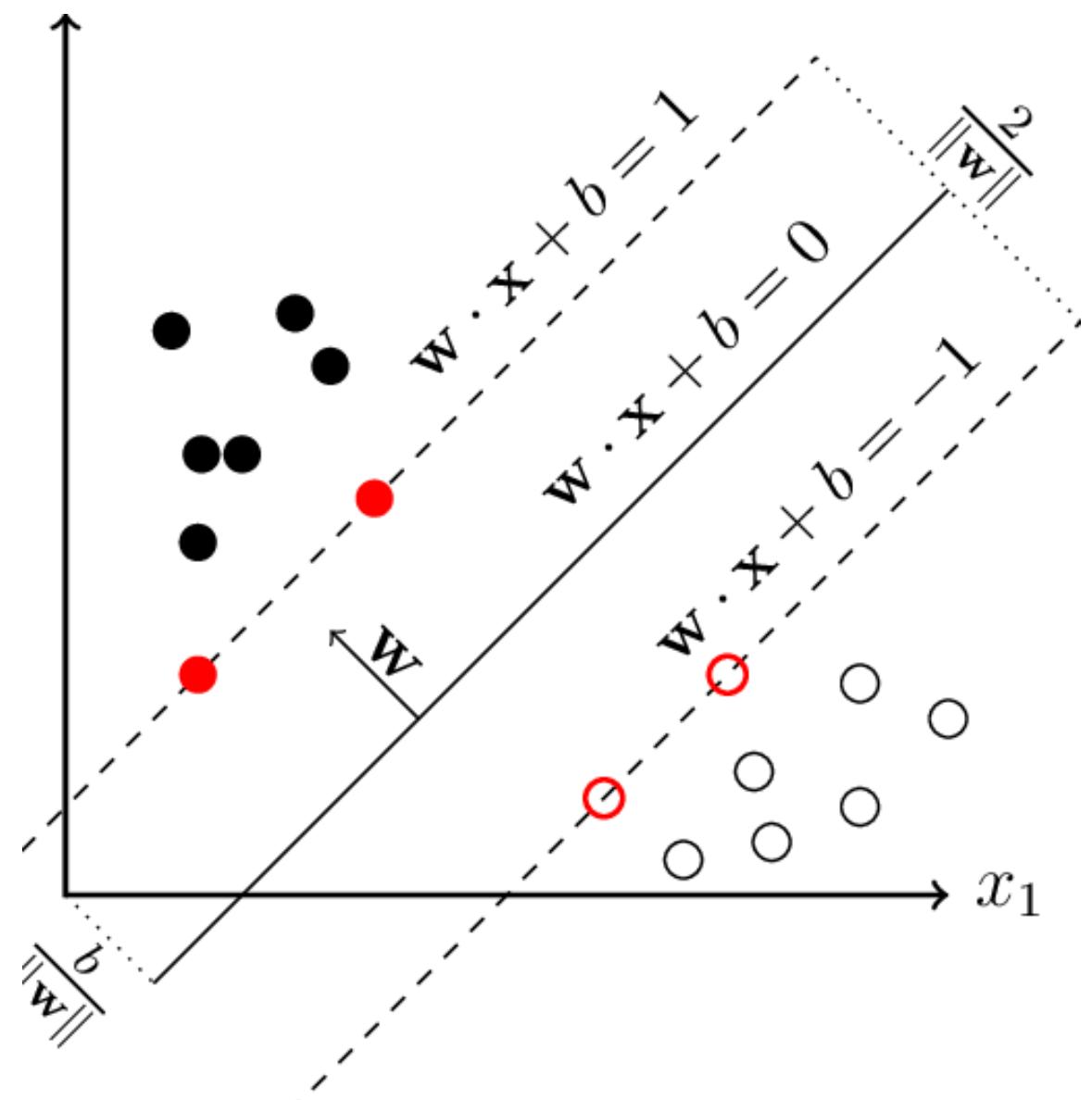
Untuk mendapatkan fungsi pemisah yang paling optimal, maka perlu **mendapatkan margin maksimum**. Margin maksimum adalah jarak total antara dua batas margin.

$$\text{margin} = \frac{2}{\|\mathbf{w}\|},$$

Selain mendapatkan margin maksimum, maka perlu juga **memastikan bahwa fungsi constrain terpenuhi**.

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$$

\mathbf{x}_D	: Vektor observasi ke- D
\mathbf{w}	: Vektor bobot pada <i>support vector machine</i>
b	: Nilai bias pada <i>support vector machine</i>



Support Vector Machine (SVM)

Support Vector Machine Non-Linear

Data yang digunakan dalam analisis sering kali **tidak dapat dipisahkan secara linier**, sehingga sulit untuk menemukan hyperplane linier. Untuk mengatasi masalah ini, data dapat **ditransformasikan ke ruang dengan dimensi yang lebih tinggi**. Transformasi ini dilakukan **menggunakan kernel trick**, yaitu metode untuk menghitung hubungan atau kesamaan antara data langsung di ruang dimensi tinggi tanpa perlu memetakan data secara eksplisit (Tan, Hijazi, & Nohuddin, 2023).

Fungsi Kernel Trick:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Fungsi kernel yang umum digunakan:

- | | |
|---|--|
| 1. <i>Radial Basis Function</i> | : $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ ^2)$ |
| 2. <i>Linear Kernel</i> | : $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ |
| 3. <i>Polynomial kernel of degree q</i> | : $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^\lambda$ |
| 4. <i>Sigmoid kernel</i> | : $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\tau \mathbf{x}_i^T \mathbf{x}_j + \vartheta)$ |

- | | |
|--------------------------------------|----------------------------------|
| $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ | : Fungsi kernel antar vektor |
| $\phi(\mathbf{x}_i)$ | : Fungsi pemetaan ke ruang fitur |

Random Forest

Decision Tree

Algoritma ini bekerja dengan **membagi dataset** menjadi subset berdasarkan kondisi tertentu, **hingga dataset mencapai pembagian yang homogen** (Han, Kamber, & Pei, 2012).

Metode CART:

$$\text{Gini}(D_T) = 1 - \sum_{\ell=1}^L P_\ell^2, \quad \text{-> mengukur impurity}$$

$$P_\ell = \frac{D_{T\ell}}{D_T}.$$

Keterangan:

- $\text{Gini}(D_T)$: Gini index pada partisi data D node ke-T
- P_ℓ : Probabilitas kelas ke- ℓ
- $D_{T\ell}$: Banyaknya sampel yang ada pada *node T* untuk kelas ke- ℓ
- D_T : Banyaknya sampel yang ada pada *node T*
- L : Banyaknya kelas

Jika diberi pembobotan probabilitas untuk setiap kelas:

$$\text{Gini}(D_T)' = 1 - \sum_{\ell=1}^L B_\ell P_\ell^2$$

Misalkan terjadi binary split yang dipisahkan oleh X_j variabel sehingga membagi D_T menjadi D_{TL} dan D_{TR} .

$$\text{Gini}_{X_j}(D_T)' = \frac{D_{TL}}{D_T} \text{Gini}(D_{TL})' + \frac{D_{TR}}{D_T} \text{Gini}(D_{TR})',$$

$$\text{Gini}(D_{TL})' = 1 - \sum_{\ell=1}^L B_\ell \left(\frac{D_{T\ell,\ell}}{D_{TL}} \right)^2,$$

$$\text{Gini}(D_{TR})' = 1 - \sum_{\ell=1}^L B_\ell \left(\frac{D_{T\ell,\ell}}{D_{TR}} \right)^2.$$

Keterangan:

- D_{TR} : Banyaknya data pada *node kanan* dari hasil partisi
- D_{TL} : Banyaknya data pada *node kiri* dari hasil partisi
- $\text{Gini}(D_{TL})'$: Gini index terboboti pada partisi *node kiri*
- $\text{Gini}(D_{TR})'$: Gini index terboboti pada partisi *node kanan*
- B_ℓ : Bobot untuk kelas ke- ℓ

Goodness of Split:

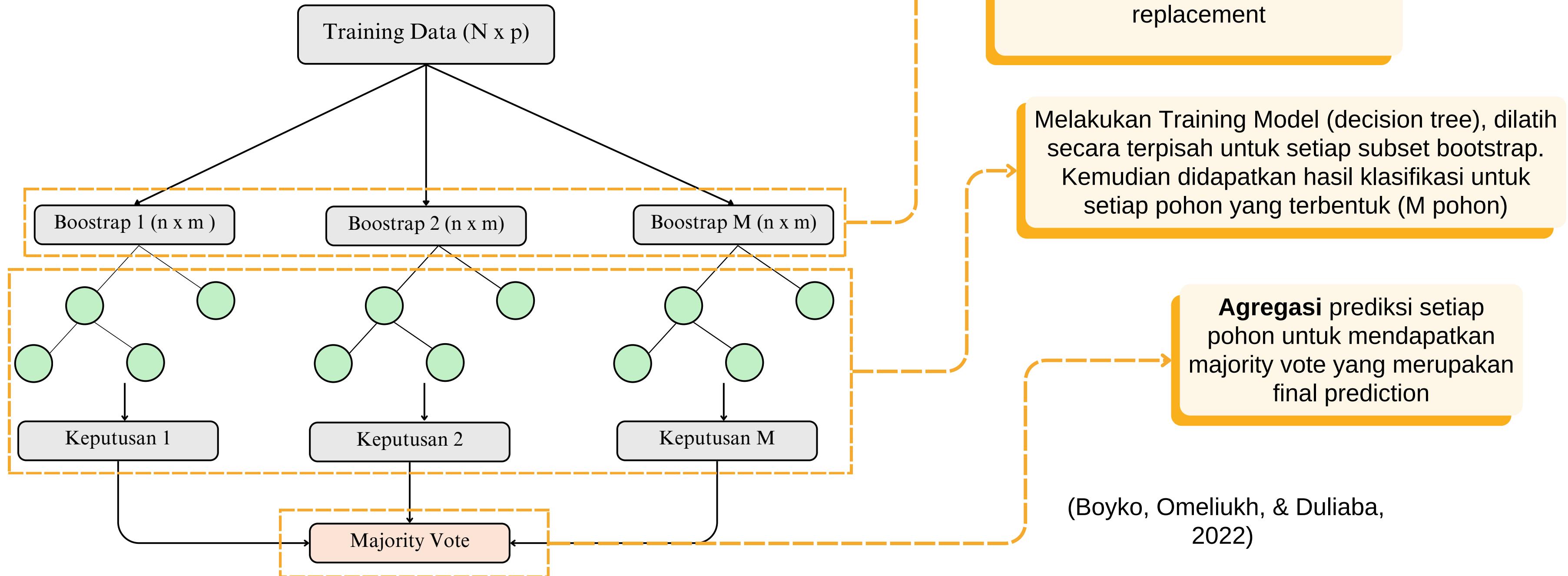
$$\Delta \text{Gini}(X_j) = \text{Gini}(D_T)' - \text{Gini}_{X_j}(D_T)'$$

* semakin besar semakin baik

Random Forest

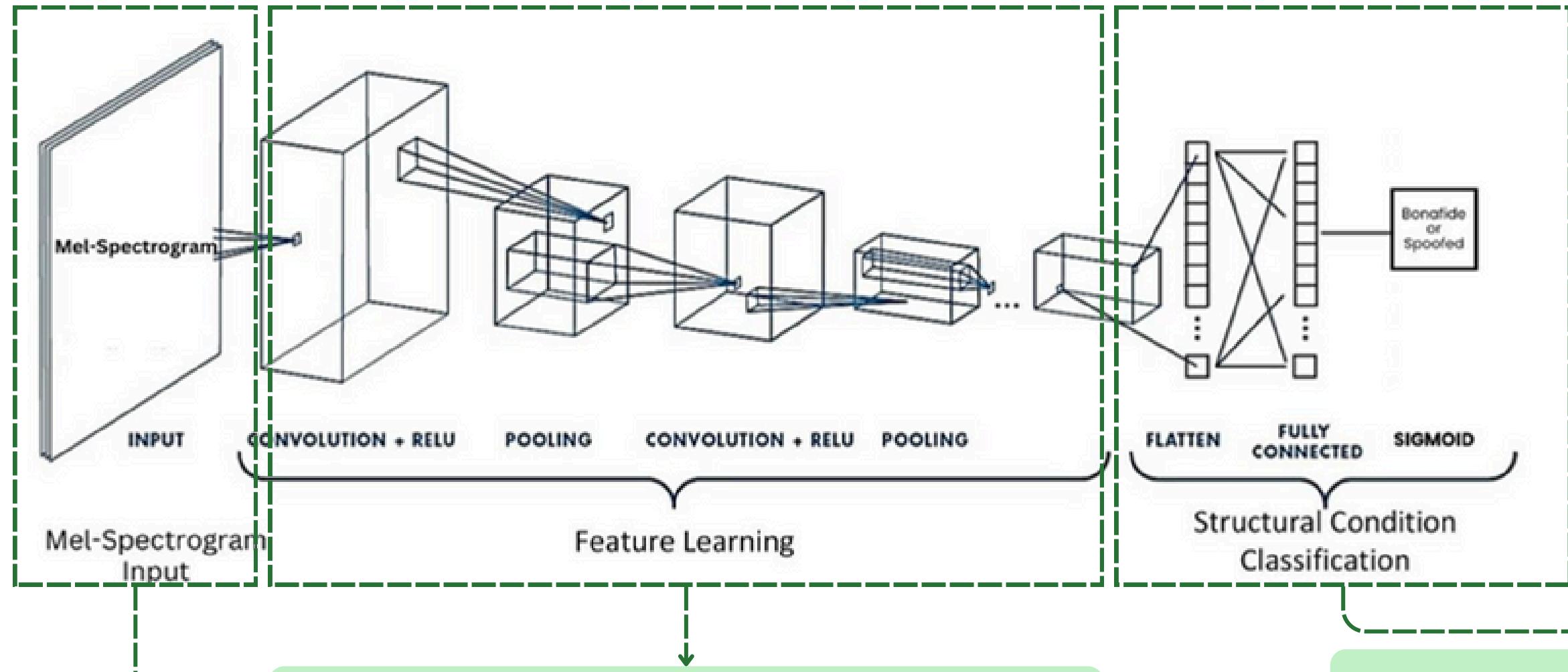
Random Forest

Random forest merupakan pengembangan dari decision tree yang menerapkan pendekatan bagging (Bootstrap Aggregating).



Convolutional Neural Network

Convolutional neural network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengelola data dua dimensi. CNN memiliki kemampuan untuk mengekstraksi pola-pola atau fitur secara otomatis dari data visual (Aloysius & Geetham, 2017).



Input menerima data mentah dan kemudian melewatkannya ke tahap feature learning

Mengekstrak fitur-fitur penting dari data input, seperti pola-pola lokal yang relevan untuk analisis lebih lanjut dan **menyederhanakan** serta **mengurangi** dimensi data tanpa kehilangan informasi yang signifikan.

Lapisan ini bertugas untuk **mengklasifikasikan** data berdasarkan fitur yang telah dipelajari, menghasilkan prediksi ke dalam beberapa kategori.

(Landini, 2021)

SLIDE 24

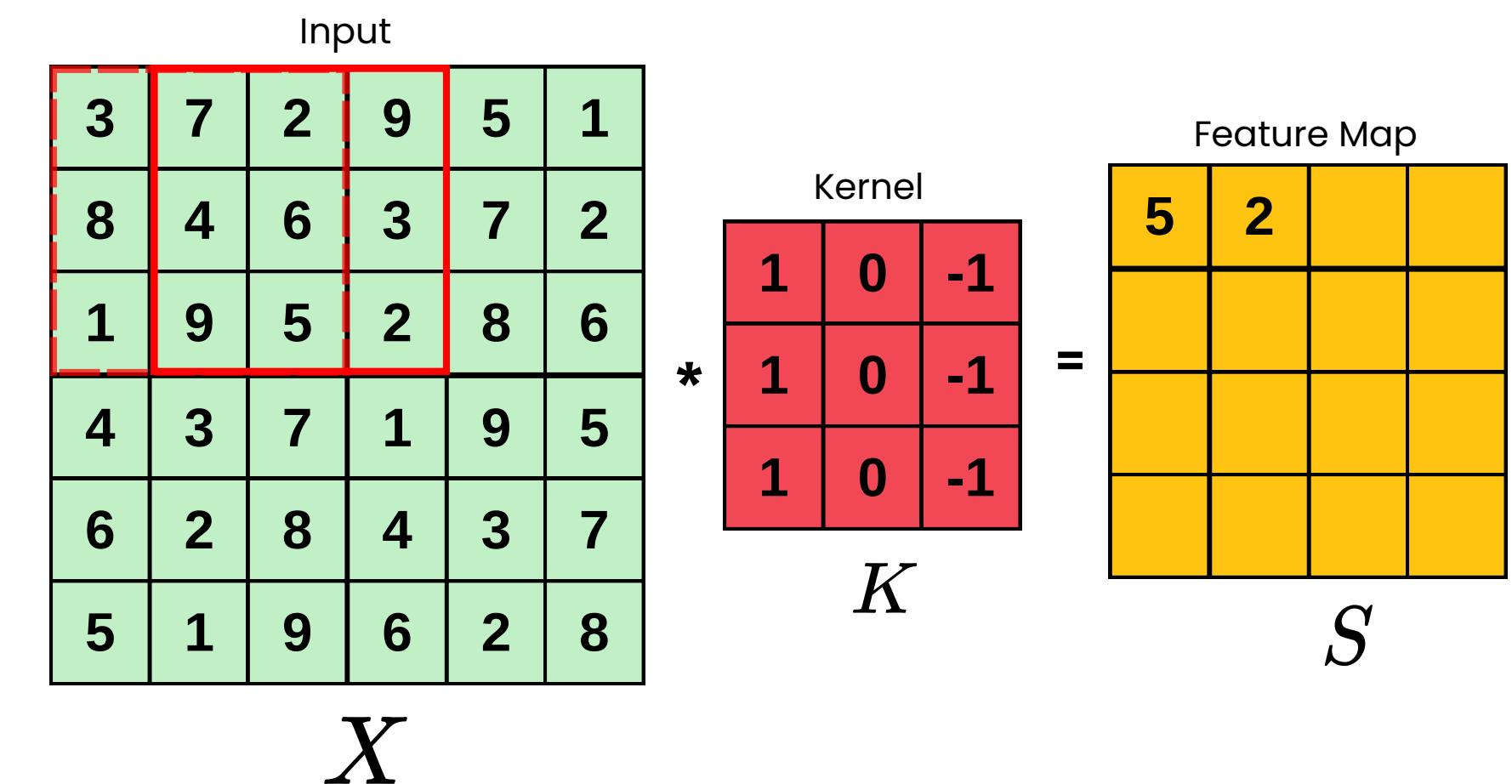
Convolutional Layer

Convolutional layer adalah lapisan inti dari CNN yang digunakan untuk **mengekstraksi** fitur-fitur penting dari data input (Goodfellow, Bengio, & Courville, 2016).

$$S_{h,p,q} = f \left(\sum_{d=0}^{r-1} \sum_{e=0}^{r-1} \tilde{X}_{h,p \cdot s+d, q \cdot s+e} \cdot \tilde{K}_{d,e} + \tilde{b}_h \right)$$

Keterangan:

- $\tilde{S}_{h,p,q}$: Output convolutional layer lapisan ke- h , baris ke- p , kolom ke- q
- $\tilde{X}_{h,p \cdot s+d, q \cdot s+e}$: Nilai piksel lapisan ke- h , posisi $(p \cdot s + d, q \cdot s + e)$
- s : Stride, jumlah pergeseran wilayah pooling bergerak pada setiap operasi
- \tilde{K} : Matriks kernel konvolusi berukuran $r \times r$
- $\tilde{K}_{d,e}$: Nilai matriks \tilde{K} baris ke- d dan kolom ke- e
- \tilde{b}_h : Nilai bias pada feature map convolutional layer lapisan ke- h



Pooling Layer

Fungsi utama dari pooling layer adalah untuk **mengurangi dimensi data (downsampling)**, tetapi dengan mempertahankan informasi penting dari hasil konvolusi (Lee, Gallagher, & Tu, 2015).

Max Pooling Layer

Max pooling berfungsi untuk mengekstrak fitur dengan mengambil **nilai maksimal** dari semua nilai piksel untuk setiap jendela.

$$P_{max,h,p,q} = \max \{ \tilde{X}_{h,d,e} | d \in [p \cdot s, p \cdot s + l - 1], e \in [q \cdot s, q \cdot s + l - 1] \},$$

Average Pooling Layer

Average pooling berfungsi untuk mengekstraksi fitur dengan **mengambil nilai rata-rata** dari semua piksel untuk setiap jendela.

$$P_{avg,h,p,q} = \frac{1}{k^2} \sum_{d=p \cdot s}^{p \cdot s + l - 1} \sum_{e=q \cdot s}^{q \cdot s + l - 1} \tilde{X}_{h,d,e}.$$

Keterangan:

$P_{h,p,q}$: Output pooling lapisan ke- h , baris ke- p , kolom ke- q

$X_{h,d,e}$: Nilai piksel lapisan ke- h , baris ke- d , kolom ke- e

l : Ukuran jendela pooling (*pool size*)

s : *Stride*, seberapa jauh pergeseran wilayah pooling bergerak pada setiap operasi

Max Pooling

29	12	45	98
3	56	78	21
34	19	5	88
50	2	90	14

(a)

56	98
50	90

Average Pooling

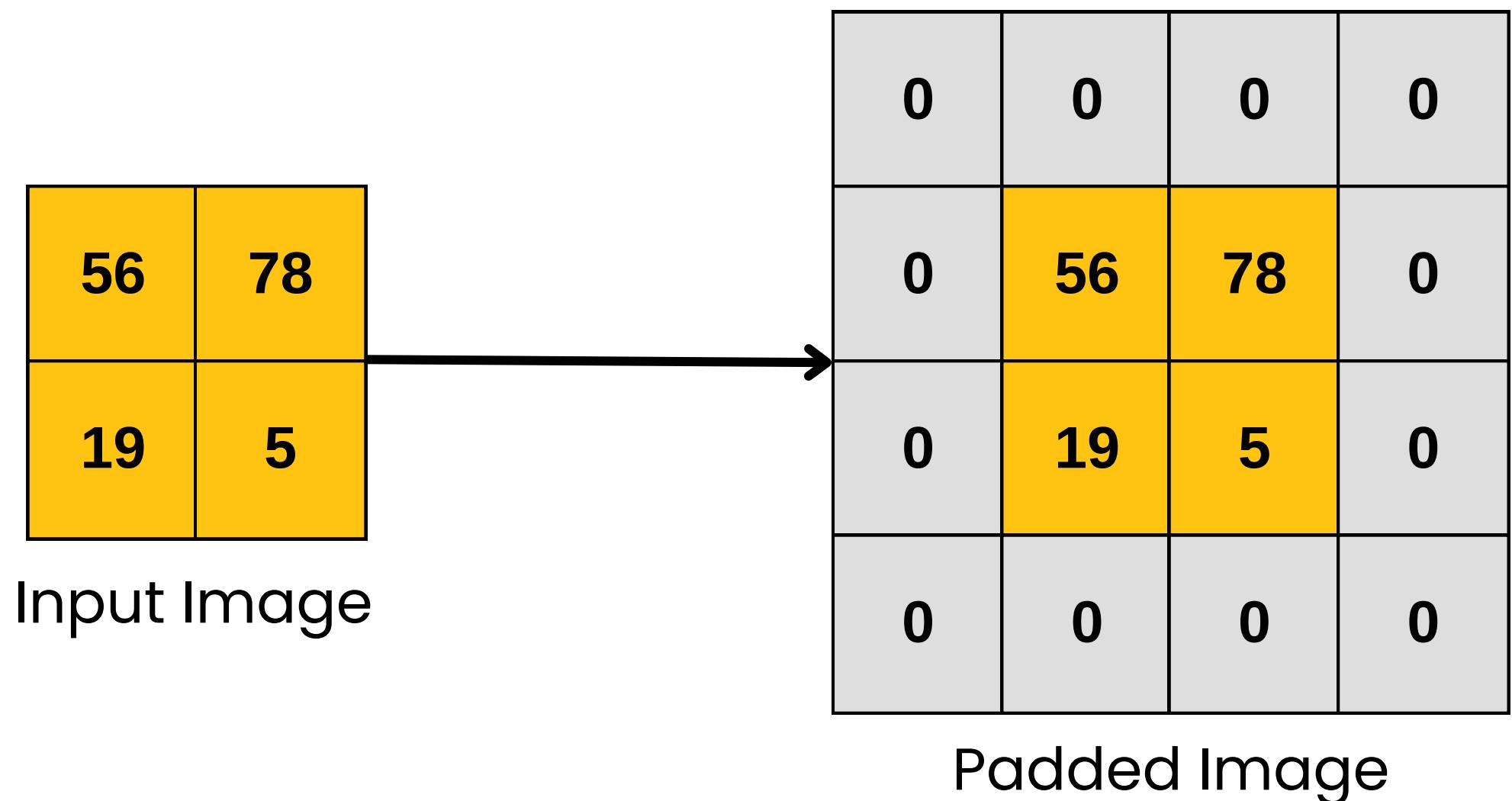
29	12	45	98
3	56	78	21
34	19	5	88
50	2	90	14

(b)

22.75	60.50
22.25	49.25

Padding

Padding atau Zero Padding adalah proses **menambahkan piksel yang bernilai 0** pada setiap sisi dari data input. Tujuan utama penggunaan padding adalah untuk mencegah pengurangan dimensi output secara signifikan setelah proses konvolusi.



Long Short Term Memory

Recurrent Neural network adalah jaringan neural yang **menggunakan konsep memori**. Mekanisme ini memberikan kemampuan seperti memori, sehingga **memungkinkan untuk memahami korelasi dari urutan data**. LSTM adalah jenis RNN yang dirancang untuk **mengatasi masalah vanishing gradient dan juga exploding gradient** melalui cell state dan gate (Barua, Rahim, Parizat, Noor, & Jannah, 2021).

Forget Gate

Proses ini menentukan apakah informasi dari cell state \mathbf{c}_{t-1} sebelumnya akan dilupakan atau dipertahankan, tergantung pada nilai f_{gt}

$$f_{gt} = \sigma(\mathbf{z}_t \cdot \mathbf{U}_f + \mathbf{h}_{t-1} \cdot \mathbf{W}_f)$$

Input Gate

Input gate menentukan informasi baru apa yang akan disimpan pada long-term memory

$$i_{gt} = \sigma(\mathbf{z}_t \cdot \mathbf{U}_i + \mathbf{h}_{t-1} \cdot \mathbf{W}_i)$$

Informasi dari input gate akan dikombinasikan dengan kandidat cell state baru untuk memperbarui cell state

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{z}_t \cdot \mathbf{U}_c + \mathbf{h}_{t-1} \cdot \mathbf{W}_c),$$

Kemudian, cell state diperbarui dengan menggabungkan informasi baru dari kandidat cell state dan informasi lama dari cell state sebelumnya.

$$\mathbf{c}_t = f_{gt} \odot \mathbf{c}_{t-1} + i_{gt} \odot \tilde{\mathbf{c}}_t.$$

Output Gate

Output gate bertindak sebagai filter yang menentukan bagian dari cell state yang akan diteruskan sebagai hidden state ke langkah waktu berikutnya atau ke lapisan selanjutnya

$$o_{gt} = \sigma(\mathbf{z}_t \cdot \mathbf{U}_o + \mathbf{h}_{t-1} \cdot \mathbf{W}_o)$$

Kemudian, output gate akan dikombinasikan dengan cell state yang menggunakan fungsi aktivasi tanh untuk menormalisasi cell state.

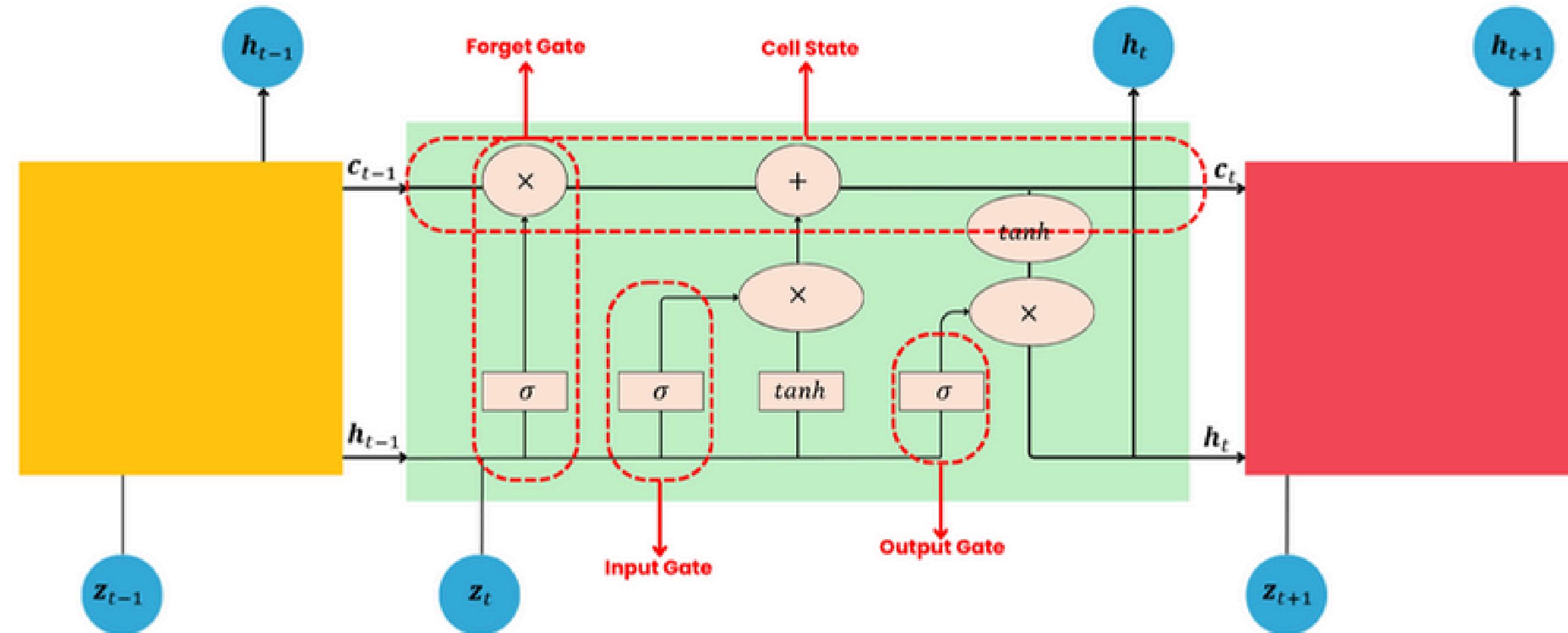
$$\mathbf{h}_t = o_{gt} \odot \tanh(\mathbf{c}_t)$$

Hidden state ini merupakan output akhir dari LSTM atau dapat diteruskan ke sel berikutnya

Keterangan:

\mathbf{z}_t	: Vektor input pada timestamp saat ini ($1 \times m$)
\mathbf{h}_t	: Vektor hidden state pada timestamp saat ini ($1 \times n$)
\mathbf{c}_t	: Vektor cell state pada timestamp saat ini ($1 \times n$)
$\tilde{\mathbf{c}}_t$: Vektor kandidat cell state pada timestamp saat ini ($1 \times n$)
\mathbf{U}	: Matriks bobot input ($m \times n$)
\mathbf{W}	: Matriks bobot hidden state ($n \times n$)
σ	: Fungsi aktivasi sigmoid
\tanh	: Fungsi aktivasi tanh

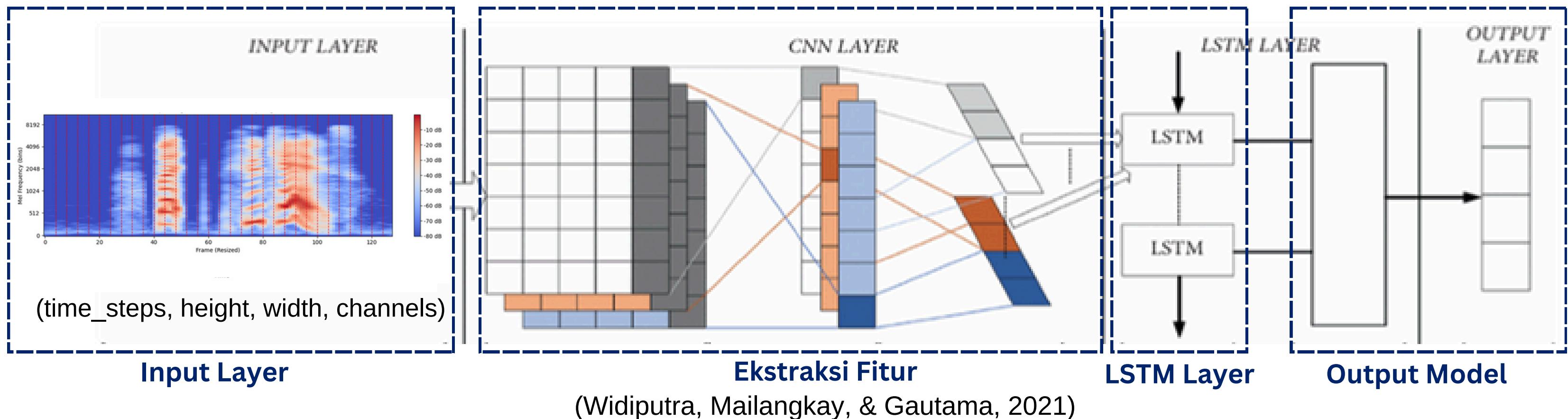
Long Short Term Memory



(Barua, Rahim, Parizat, Noor, & Jannah, 2021)

CNN-LSTM

CNN-LSTM adalah model hybrid yang menggabungkan antara Convolutional Neural Network (CNN) dan Long Short-Term Memory (LSTM). Pendekatan ini memanfaatkan keunggulan dari **CNN dalam menganalisis data spasial** dan **LSTM dalam menganalisis data temporal**. CNN-LSTM akan memanfaatkan informasi antar sequence dalam membuat prediksi.

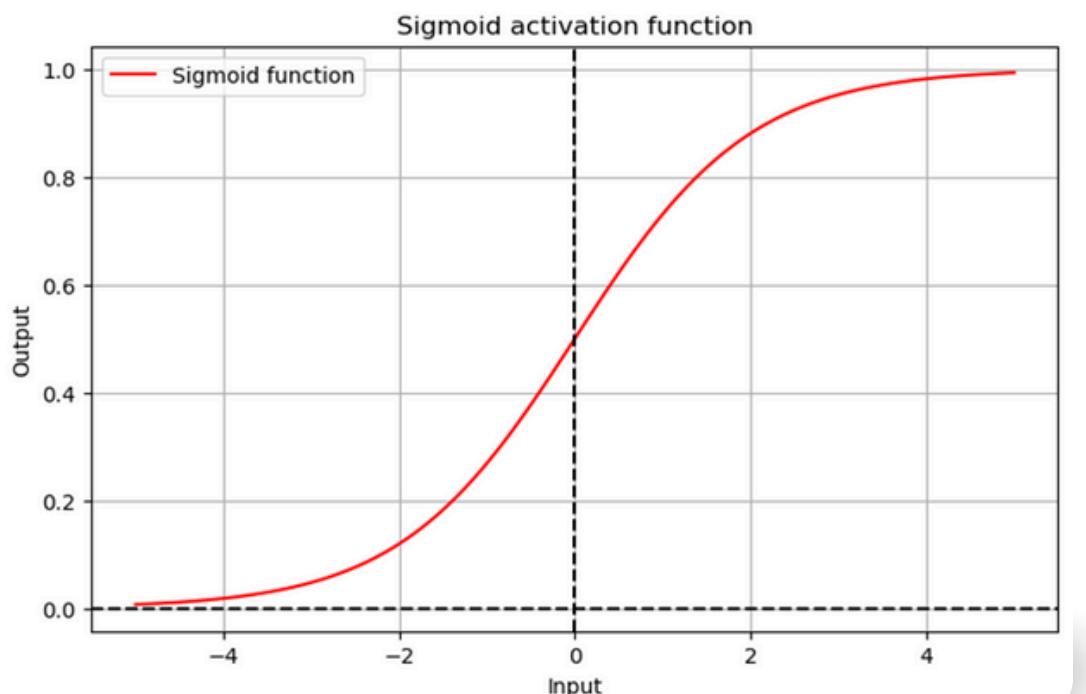


Fungsi Aktivasi

Fungsi Aktivasi Sigmoid

Fungsi Sigmoid berguna sebagai probabilitas output pada binary classification.

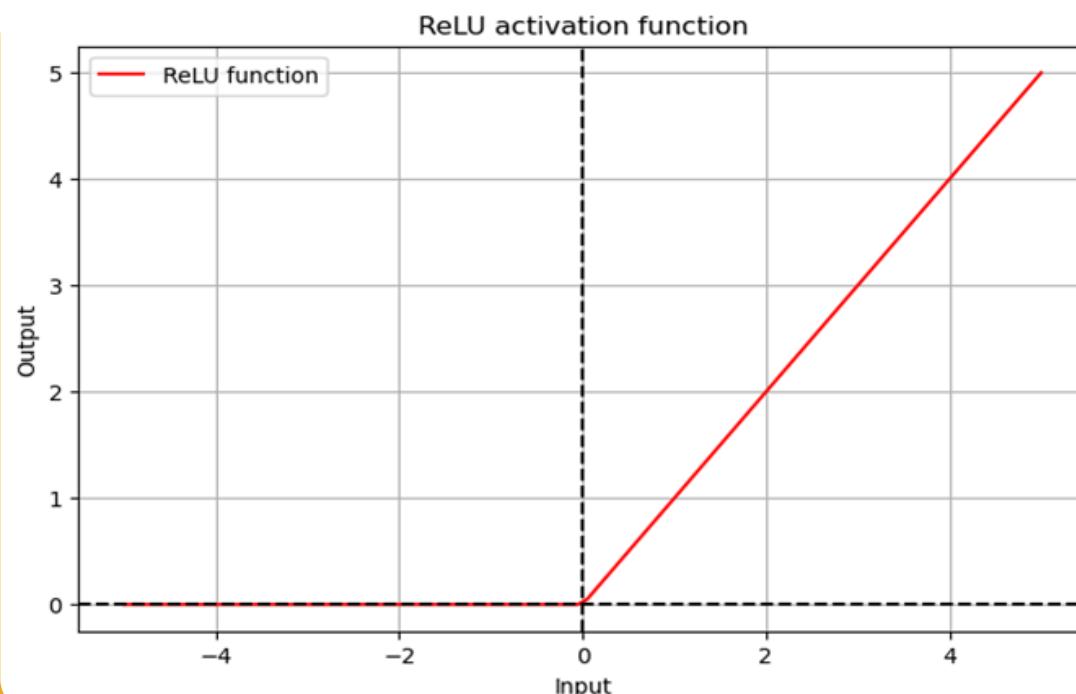
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Fungsi Aktivasi ReLU

Fungsi ReLU memungkinkan model untuk mencapai konvergensi lebih cepat, terutama ketika menggunakan metode gradient descent sehingga sering digunakan pada hidden layer.

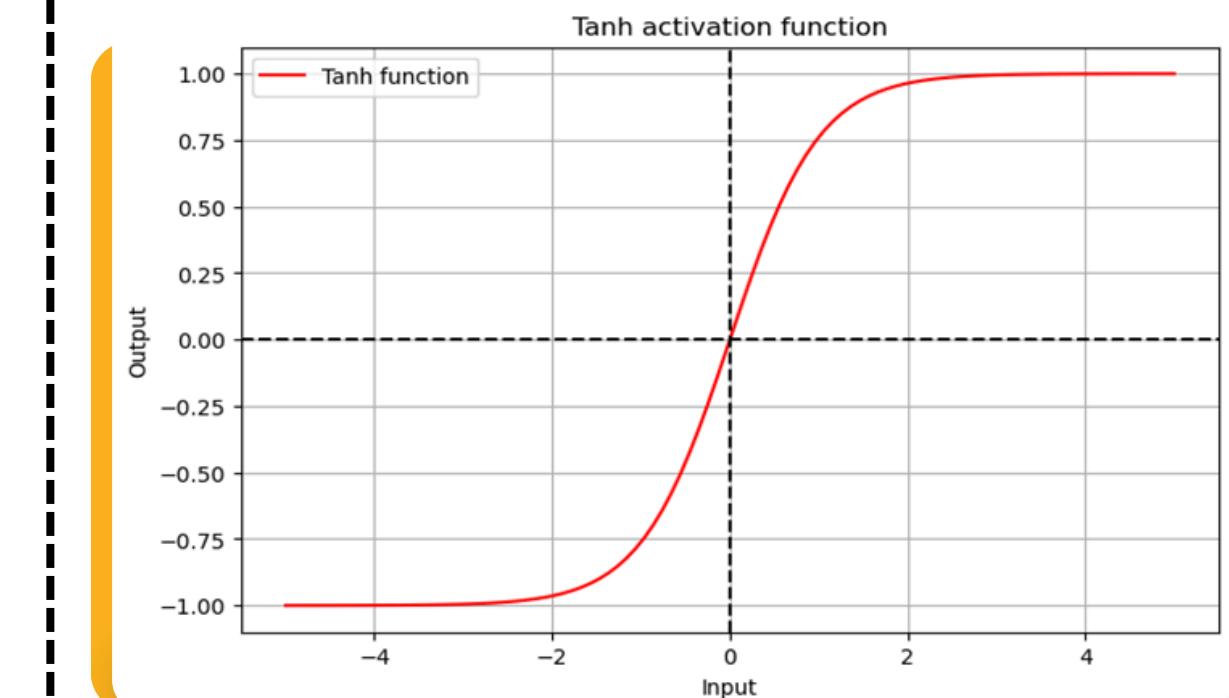
$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{jika } x \geq 0 \\ 0, & \text{jika } x < 0 \end{cases}$$



Fungsi Aktivasi Tanh

Fungsi Tanh digunakan dalam situasi di mana representasi simetris di sekitar nol lebih diinginkan, seperti pada hidden layers atau model sekuensial seperti LSTM

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



(Ding, Qian, & Zhou, 2018).

Binary Cross Entropy

Binary Cross Entropy adalah fungsi loss yang umum digunakan pada pemodelan klasifikasi biner.

Loss Function

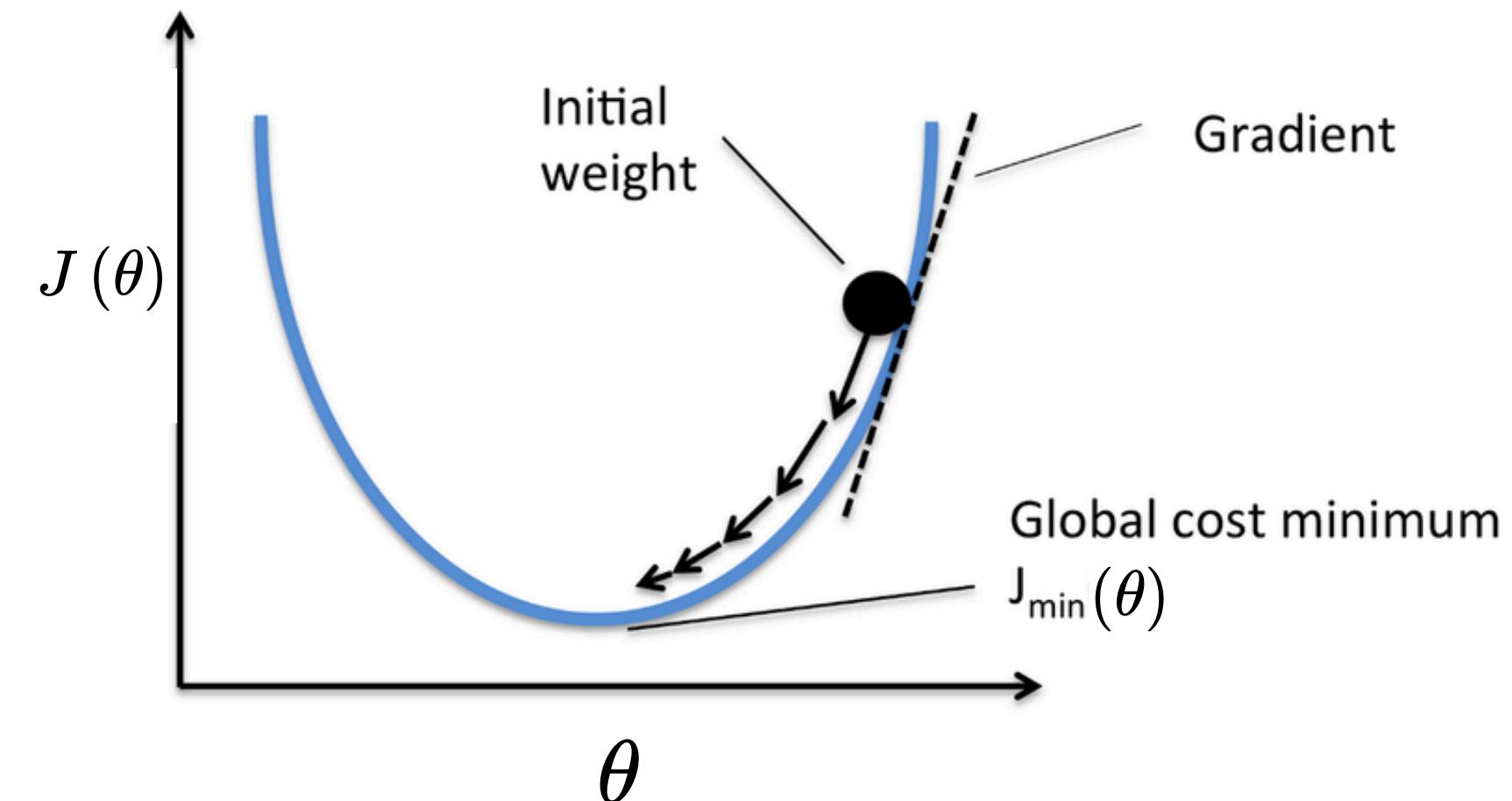
$$L(y_i, \hat{y}_i) = -[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)].$$

Mengukur kesalahan pada satu sampel data

Cost Function

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)],$$

Mengukur rata-rata kesalahan keseluruhan data



$L(y_i, \hat{y}_i)$: Loss function binary cross entropy
$J(\theta)$: Cost function binary cross entropy
y_i	: Label kelas sebenarnya
\hat{y}_i	: Probabilitas hasil prediksi

Meminimalkan Error

Gradient Cost Function
Mendekati Nol

Pembaruan Parameter
dengan Iterasi

Optimizer untuk melakukan pembaruan parameter.

- Menggabungkan keuntungan momentum dan adaptasi learning rate secara bersamaan.
- Membuat algoritma pembelajaran lebih cepat dan efektif (Soyander, 2020)

Gradient Cost Function

$$\mathbf{g}^{(t)} = \nabla J(\boldsymbol{\theta}^{(t)}) = \frac{\partial J(\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}^{(t)}},$$

Chain Rule

$$\frac{\partial J(\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}^{(t)}} = \frac{\partial J(\boldsymbol{\theta}^{(t)})}{\partial \hat{y}_i^{(t)}} \cdot \frac{\partial \hat{y}_i^{(t)}}{\partial z_i^{(t)}} \cdot \frac{\partial z_i^{(t)}}{\partial \boldsymbol{\theta}^{(t)}}$$

$$\frac{\partial J(\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}^{(t)}} = \frac{1}{N} \sum_{i=1}^N (y_i^{(t)} - \hat{y}_i^{(t)}) \cdot \mathbf{x}_i^{(t)}$$

Momen pertama dan kedua

$$\dot{\mathbf{m}}^{(t)} = \beta_1 \dot{\mathbf{m}}^{(t-1)} + (1 - \beta_1) \mathbf{g}^{(t)},$$

$$\dot{\mathbf{v}}^{(t)} = \beta_2 \dot{\mathbf{v}}^{(t-1)} + (1 - \beta_2) \mathbf{g}^{(t)^2}$$

Koreksi bias

$$\hat{\mathbf{m}}^{(t)} = \frac{\dot{\mathbf{m}}^{(t)}}{(1 - \beta_1^t)},$$

$$\hat{\mathbf{v}}^{(t)} = \frac{\dot{\mathbf{v}}^{(t)}}{(1 - \beta_2^t)}$$

Hasil Akhir

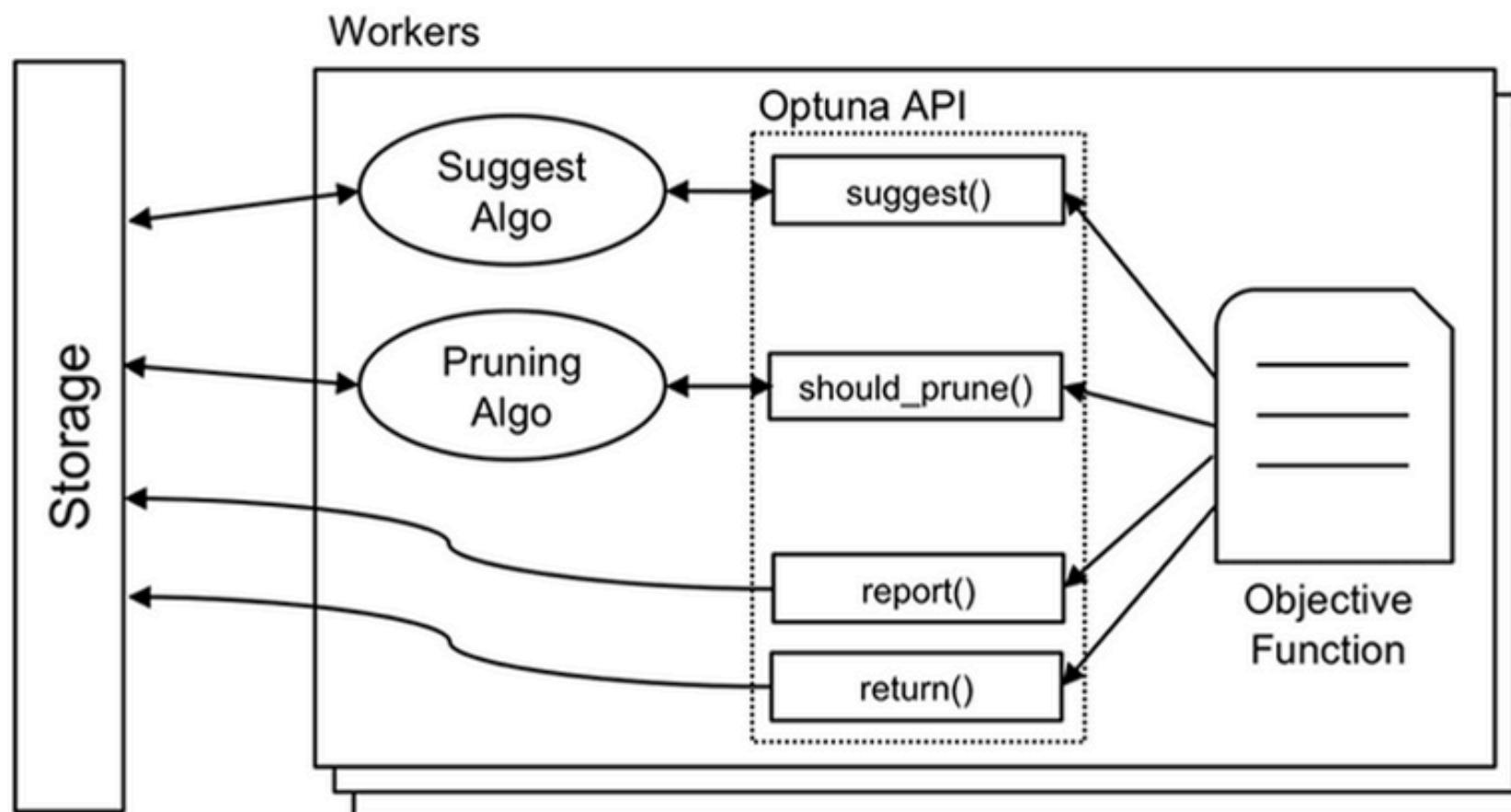
$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \alpha \frac{\hat{\mathbf{m}}^{(t)}}{\sqrt{\hat{\mathbf{v}}^{(t)}} + \epsilon}$$

*Hingga parameter konvergen \rightarrow gradient mendekati nol

$J(\boldsymbol{\theta})$: Cost function binary cross entropy
y_i	: Label kelas sebenarnya
\hat{y}_i	: Probabilitas hasil prediksi
$\boldsymbol{\theta}^{(t)}$: Parameter pada perhitungan neural network pada iterasi ke- t
α	: Learning rate
β_1 & β_2	: Exponential decay rate
$\dot{\mathbf{m}}^{(t)}$: Momen pertama pada iterasi ke- t
$\dot{\mathbf{v}}^{(t)}$: Momen kedua pada iterasi ke- t
z_i	: Transformasi linear dari input
$\hat{\mathbf{m}}^{(t)}$: Estimasi koreksi bias momen pertama
$\hat{\mathbf{v}}^{(t)}$: Estimasi koreksi bias momen kedua
ϵ	: Epsilon regularisasi Adam optimizer

Memilih konfigurasi hyperparameter dengan penyesuaian adaptif berdasarkan hasil evaluasi percobaan sebelumnya.

(Akiba, Sano, Yanase, Ohta, & Koyama, 2019)



Algoritma Optuna

1. Memulai Trial dengan suggest()
2. Menjalankan Fungsi Objektif
3. Monitoring Selama Training dengan report()
4. Pruning dengan should_prune()
5. Menyimpan Hasil Trial
6. Mengulang dengan Pembelajaran dari Trial Sebelumnya dengan Suggest Algo

Evaluasi Ketepatan Klasifikasi

Untuk mengevaluasi hasil klasifikasi, confusion matrix sering digunakan untuk menilai berapa banyak data yang diklasifikasikan dengan benar atau salah oleh model (Han, Kamber, & Pei, 2012).

<i>Actual Class</i>	<i>Predicted Class</i>	
	<i>negative</i>	<i>positive</i>
<i>negative</i>	TN	FP
<i>positive</i>	FN	TP

Akurasi

Akurasi didefinisikan sebagai parameter yang menunjukkan seberapa akurat model dalam melakukan klasifikasi secara keseluruhan

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Presisi

Nilai Presisi berguna untuk menghindari kesalahan prediksi positif palsu (false positive), yaitu sampel negatif diklasifikasikan menjadi positif

$$\text{Presisi} = \frac{TP}{TN + FP}.$$

Recall

Nilai Recall berguna untuk menghindari kesalahan prediksi negatif palsu (false negative), yaitu sampel positif diklasifikasikan menjadi negatif

$$\text{Recall} = \frac{TP}{TP + FN}.$$

F1-Score

F1-Score adalah rata-rata harmonis dari presisi dan recall, sehingga memberikan gambaran kebaikan yang seimbang tentang kinerja model.

$$\text{F1 - Score} = \frac{2 \times \text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}}.$$



METODE PENELITIAN

SPESIFIKASI KOMPUTASI



Lenovo V14 G2 ITL

OPERATING SYSTEM

Windows 10 64-bit

PROCESSOR (CPU)

Intel i5-1135G7

MEMORY (RAM)

8 GB

GPU

Iris Xe Graphics

SUMBER DATA



Automatic Speaker Verification and
Spoofing Countermeasures Challenge

SUMBER DATA

Dataset sekunder dari platform open source ASVspoof 2019.
www.asvspoof.org/index2019.html

JENIS DATASET

Dataset berbasis Logical Access (LA). Tipe Konversi TTS & VC

JUMLAH DATA

Total: 121.461 sampel audio.
Bonafide: 12.483 sampel & Spoofed: 108.978 sampel

VARIABEL PENELITIAN

Variabel Penelitian Pada Metode Pendekatan Feature-Based

Variabel	Keterangan	Skala
Y	Audio (0: <i>bonafide</i> , 1: <i>spoofed</i>)	Nominal
\dot{X}_1	<i>Chroma Feature #1</i> ($v_{cm}(1)$)	Rasio
\vdots	\vdots	\vdots
\dot{X}_{12}	<i>Chroma Feature #12</i> ($v_{cm}(12)$)	Rasio
\dot{X}_{13}	<i>MFCC Coefficient #1</i> ($v_{MFCC}(1)$)	Rasio
\vdots	\vdots	\vdots
\dot{X}_{32}	<i>MFCC Coefficient #20</i> ($v_{MFCC}(20)$)	Rasio
\dot{X}_{33}	<i>Spectral centroid</i> (v_{sc})	
\dot{X}_{34}	<i>Spectral spread</i> (v_{ss})	Rasio
\dot{X}_{35}	<i>Spectrall Rolloff</i> (v_{sr})	Rasio
\dot{X}_{36}	<i>Zero crossing rate</i> (ZCR)	Rasio
\dot{X}_{37}	<i>Root Mean Square</i> (RMS)	Rasio

Variabel Penelitian Pada Metode Pendekatan Image-Based

Variabel	Keterangan	Skala
Y	Kategori jenis audio (0: <i>bonafide</i> , 1: <i>spoofed</i>)	Nominal
$\tilde{X}_{r,n_f,k'}$	Nilai dari Intensitas Energi pada Piksel Gambar <i>Mel-spectrogram</i> untuk Audio ke- r , Frame ke- n_f , dan Mel Bands ke- k'	Rasio

STRUKTUR DATA

Struktur data pada Metode Pendekatan Feature-Based

No	\dot{X}_1	\dot{X}_2	\dot{X}_3	\dot{X}_4	\dot{X}_5	\dot{X}_6	...	\dot{X}_{17}	\dot{X}_{18}	...	\dot{X}_{37}	Y
1	$\dot{X}_{1,1}$	$\dot{X}_{1,2}$	$\dot{X}_{1,3}$	$\dot{X}_{1,4}$	$\dot{X}_{1,5}$	$\dot{X}_{1,6}$...	$\dot{X}_{1,17}$	$\dot{X}_{1,18}$...	$\dot{X}_{1,37}$	Y_1
2	$\dot{X}_{2,1}$	$\dot{X}_{2,2}$	$\dot{X}_{2,3}$	$\dot{X}_{2,4}$	$\dot{X}_{2,5}$	$\dot{X}_{2,6}$...	$\dot{X}_{2,17}$	$\dot{X}_{2,18}$...	$\dot{X}_{2,37}$	Y_2
3	$\dot{X}_{3,1}$	$\dot{X}_{3,2}$	$\dot{X}_{3,3}$	$\dot{X}_{3,4}$	$\dot{X}_{3,5}$	$\dot{X}_{3,6}$...	$\dot{X}_{3,17}$	$\dot{X}_{3,18}$...	$\dot{X}_{3,37}$	Y_3
4	$\dot{X}_{4,1}$	$\dot{X}_{4,2}$	$\dot{X}_{4,3}$	$\dot{X}_{4,4}$	$\dot{X}_{4,5}$	$\dot{X}_{4,6}$...	$\dot{X}_{4,17}$	$\dot{X}_{4,18}$...	$\dot{X}_{4,37}$	Y_4
:	:	:	:	:	:	:	...	:	:	...	:	:
r	$\dot{X}_{r,1}$	$\dot{X}_{r,2}$	$\dot{X}_{r,3}$	$\dot{X}_{r,4}$	$\dot{X}_{r,5}$	$\dot{X}_{r,6}$...	$\dot{X}_{r,17}$	$\dot{X}_{r,18}$...	$\dot{X}_{r,37}$	Y_r
:	:	:	:	:	:	:	...	:	:	...	:	:
N	$\dot{X}_{N,1}$	$\dot{X}_{N,2}$	$\dot{X}_{N,3}$	$\dot{X}_{N,4}$	$\dot{X}_{N,5}$	$\dot{X}_{N,6}$...	$\dot{X}_{N,17}$	$\dot{X}_{N,18}$...	$\dot{X}_{N,37}$	Y_N

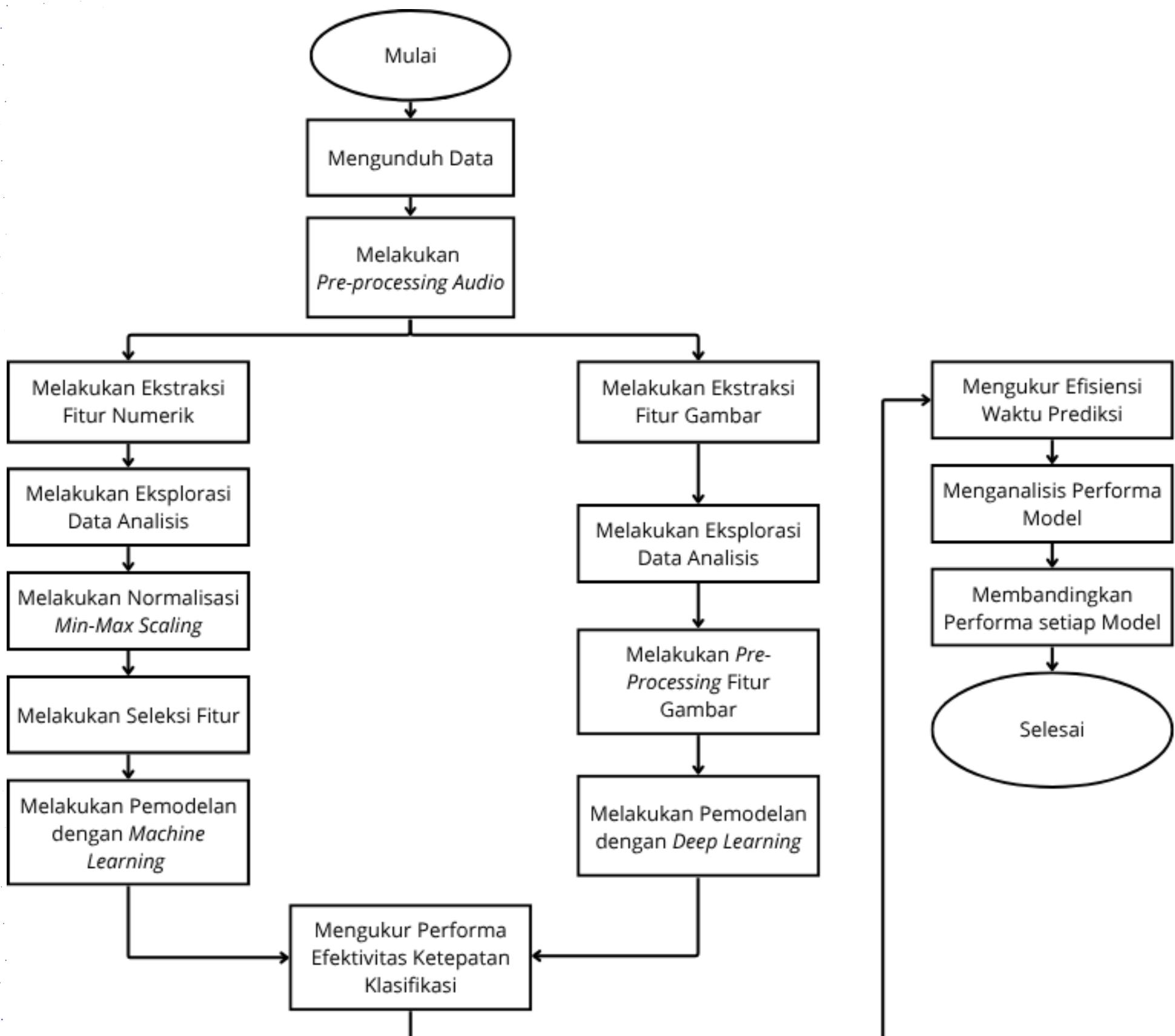
Struktur data pada Metode Pendekatan Image-Based (CNN-LSTM)

No	Piksel Mel-spectrogram (X)			Y
	Time steps = 1	Time steps = 2	...	
1	$\begin{pmatrix} \tilde{X}_{1,1,1} & \dots & \tilde{X}_{1,1,4} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{1,128,1} & \dots & \tilde{X}_{1,128,4} \end{pmatrix}$	$\begin{pmatrix} \tilde{X}_{1,1,5} & \dots & \tilde{X}_{1,1,8} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{1,128,5} & \dots & \tilde{X}_{1,128,8} \end{pmatrix}$...	$\begin{pmatrix} \tilde{X}_{1,1,125} & \dots & \tilde{X}_{1,1,128} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{1,128,125} & \dots & \tilde{X}_{1,128,128} \end{pmatrix} Y_1$
2	$\begin{pmatrix} \tilde{X}_{2,1,1} & \dots & \tilde{X}_{2,1,4} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{2,128,1} & \dots & \tilde{X}_{2,128,4} \end{pmatrix}$	$\begin{pmatrix} \tilde{X}_{2,1,5} & \dots & \tilde{X}_{2,1,8} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{2,128,5} & \dots & \tilde{X}_{2,128,8} \end{pmatrix}$...	$\begin{pmatrix} \tilde{X}_{2,1,125} & \dots & \tilde{X}_{2,1,128} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{2,128,125} & \dots & \tilde{X}_{2,128,128} \end{pmatrix} Y_2$
:	:	:
N	$\begin{pmatrix} \tilde{X}_{N,1,1} & \dots & \tilde{X}_{N,1,4} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{N,128,1} & \dots & \tilde{X}_{N,128,4} \end{pmatrix}$	$\begin{pmatrix} \tilde{X}_{N,1,5} & \dots & \tilde{X}_{N,1,8} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{N,128,5} & \dots & \tilde{X}_{N,128,8} \end{pmatrix}$...	$\begin{pmatrix} \tilde{X}_{N,1,125} & \dots & \tilde{X}_{N,1,128} \\ \vdots & \ddots & \vdots \\ \tilde{X}_{N,128,125} & \dots & \tilde{X}_{N,128,128} \end{pmatrix} Y_N$

Struktur data pada Metode Pendekatan Image-Based (CNN)

No	Nilai dari Intensitas Energi pada Piksel Gambar Mel-spectrogram (\tilde{X})	Label (Y)
1	$\begin{pmatrix} \tilde{X}_{1,1,1} & \tilde{X}_{1,1,2} & \dots & \tilde{X}_{1,1,128} \\ \tilde{X}_{1,2,1} & \tilde{X}_{1,2,2} & \dots & \tilde{X}_{1,2,128} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{X}_{1,128,1} & \tilde{X}_{1,128,2} & \dots & \tilde{X}_{1,128,128} \end{pmatrix}$	Y_1
2	$\begin{pmatrix} \tilde{X}_{2,1,1} & \tilde{X}_{2,1,2} & \dots & \tilde{X}_{2,1,128} \\ \tilde{X}_{2,2,1} & \tilde{X}_{2,2,2} & \dots & \tilde{X}_{2,2,128} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{X}_{2,128,1} & \tilde{X}_{2,128,2} & \dots & \tilde{X}_{2,128,128} \end{pmatrix}$	Y_2
:	⋮	⋮
N	$\begin{pmatrix} \tilde{X}_{N,1,1} & \tilde{X}_{N,1,2} & \dots & \tilde{X}_{N,1,128} \\ \tilde{X}_{N,2,1} & \tilde{X}_{N,2,2} & \dots & \tilde{X}_{N,2,128} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{X}_{N,128,1} & \tilde{X}_{N,128,2} & \dots & \tilde{X}_{N,128,128} \end{pmatrix}$	Y_N

LANGKAH ANALISIS





IV

HASIL DAN PEMBAHASAN

SLIDE 42

Pre-Processing

Undersampling Masalah Data Terlalu Besar

Attack ID	Training	Validation	Testing
A01	3.800	3.716	—
A02	3.800	3.716	—
A03	3.800	3.716	—
A04	3.800	3.716	—
A05	3.800	3.716	—
A06	3.800	3.716	—
A07	—	—	4.914
A08	—	—	4.914
A09	—	—	4.914
A10	—	—	4.914
A11	—	—	4.914
A12	—	—	4.914
A13	—	—	4.914
A14	—	—	4.914
A15	—	—	4.914
A16	—	—	4.914
A17	—	—	4.914
A18	—	—	4.914
A19	—	—	4.914
Bonafide	2.580	2.548	7.355

$$2580 \times 2 \div 6 = 860$$

$$\frac{\text{Spoofed}}{\text{Bonafide}} = \frac{2}{1}$$

Attack ID	Training	Validation	Testing
A01	860	849	—
A02	860	849	—
A03	860	849	—
A04	860	849	—
A05	860	849	—
A06	860	849	—
A07	—	—	1.132
A08	—	—	1.132
A09	—	—	1.132
A10	—	—	1.132
A11	—	—	1.132
A12	—	—	1.132
A13	—	—	1.132
A14	—	—	1.132
A15	—	—	1.132
A16	—	—	1.132
A17	—	—	1.132
A18	—	—	1.132
A19	—	—	1.132
Bonafide	2.580	2.548	7.355

Pre-Processing

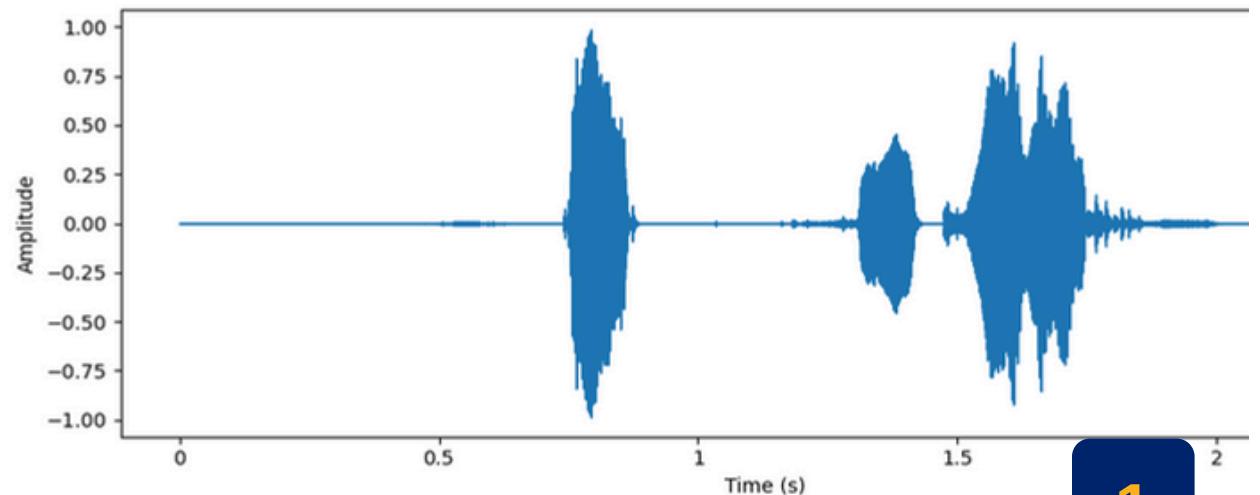
Final Data Setelah Undersampling

<i>Attack ID</i>	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
A01	860	849	—
A02	860	849	—
A03	860	849	—
A04	860	849	—
A05	860	849	—
A06	860	849	—
A07	—	—	1.132
A08	—	—	1.132
A09	—	—	1.132
A10	—	—	1.132
A11	—	—	1.132
A12	—	—	1.132
A13	—	—	1.132
A14	—	—	1.132
A15	—	—	1.132
A16	—	—	1.132
A17	—	—	1.132
A18	—	—	1.132
A19	—	—	1.132
<i>Bonafide</i>	2.580	2.548	7.355

Kelas	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
<i>Spoofed</i>	5.160	5.094	14.716
<i>Bonafide</i>	2.580	2.548	7.355
Total	7.740	7.678	22.071

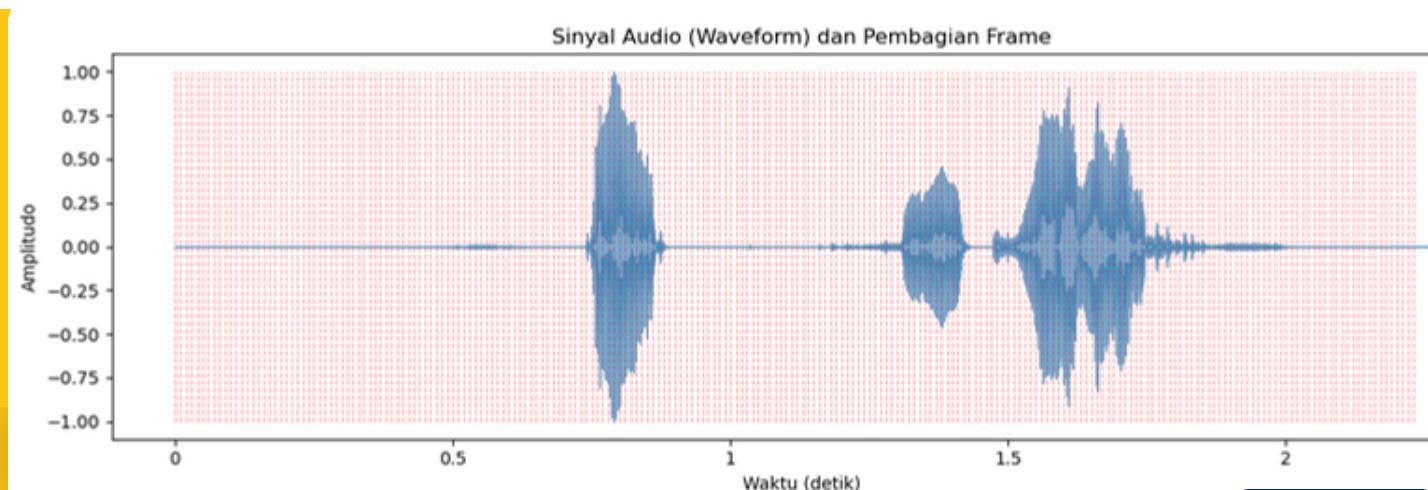
Ekstraksi Numerik

Waveform Audio



1

Membagi Audio menjadi Frame-Frame



2

Melakukan Agregasi Fitur Numerik

Setiap frame hasil ekstraksi kemudian **dilakukan agregasi** untuk menyederhanakan representasi fitur, yaitu dengan menghitung **rata-rata** dari setiap fitur numerik. Proses ini menghasilkan satu vektor fitur representatif untuk setiap file audio, yang kemudian dinormalisasi sebelum digunakan sebagai input bagi model klasifikasi.

4

Ekstraksi Fitur Numerik untuk Setiap Frame

Setiap frame menghasilkan nilai dari proses ekstraksi untuk masing-masing fitur, sehingga membentuk sebuah matriks berukuran 71×37 , di mana 71 merepresentasikan jumlah frame untuk audio tersebut, dan 37 merupakan jumlah fitur numerik yang diekstraksi dari setiap frame.

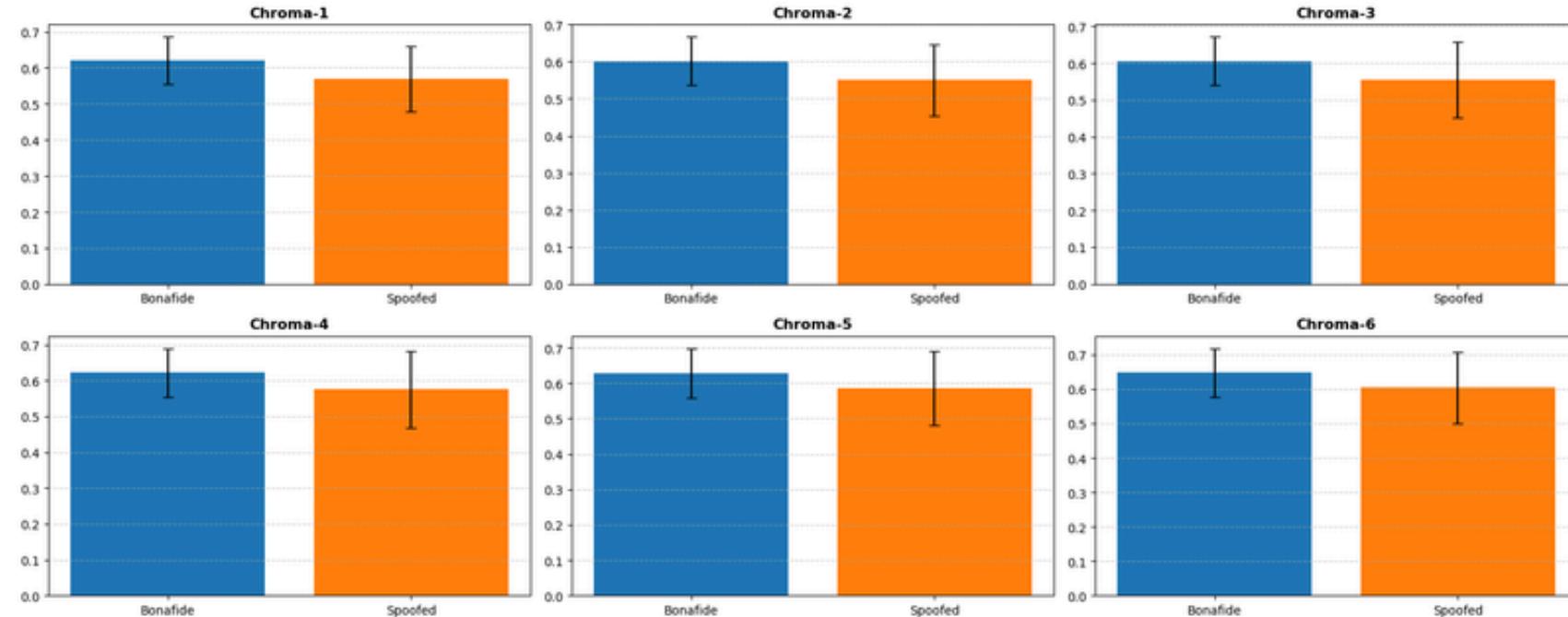
3

Hasil Ekstraksi

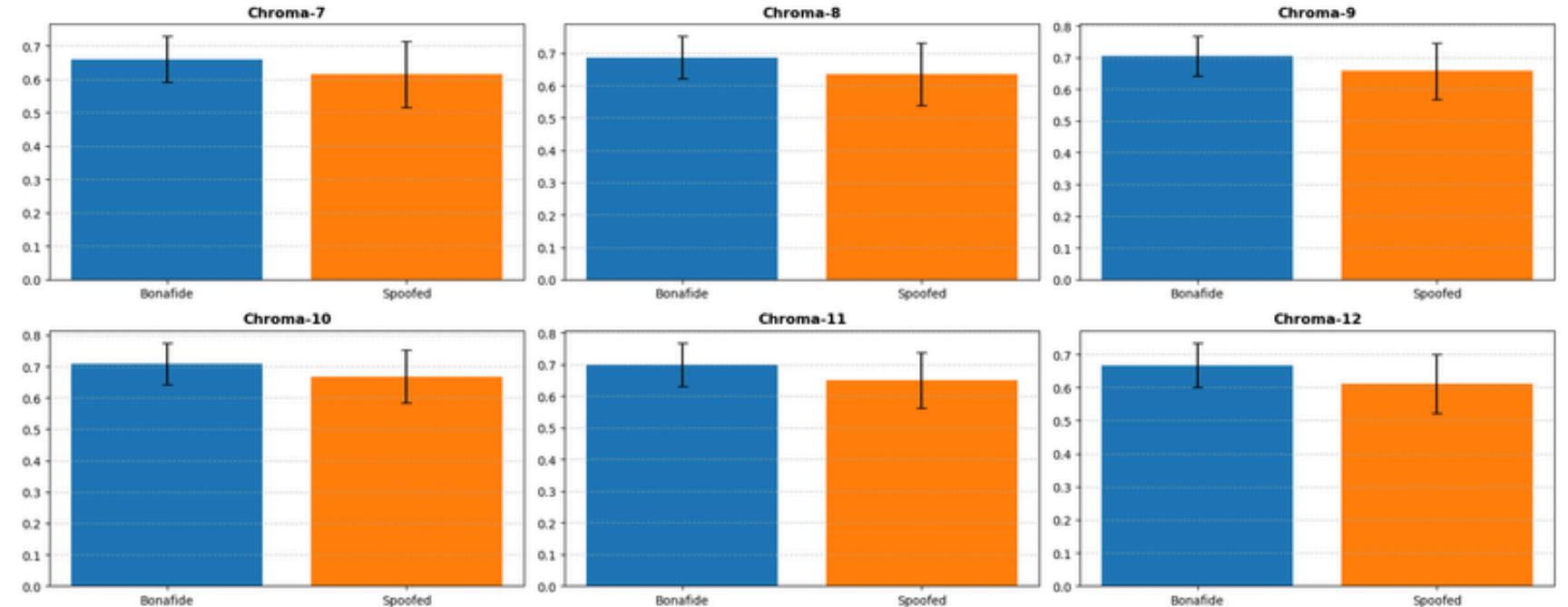
No	Fitur	Nilai
1	Chroma-1	0,3888
2	Chroma-2	0,3565
3	Chroma-3	0,3207
4	Chroma-4	0,4281
5	Chroma-5	0,5101
6	Chroma-6	0,5535
7	Chroma-7	0,4568
8	Chroma-8	0,4238
9	Chroma-9	0,4871
10	Chroma-10	0,4092
11	Chroma-11	0,4166
12	Chroma-12	0,3728
13	MFCC-1	-367,5911
14	MFCC-2	40,7555
15	MFCC-3	-19,8804
16	MFCC-4	11,5345
17	MFCC-5	-21,6711
18	MFCC-6	-4,9921
19	MFCC-7	-14,2397
20	MFCC-8	0,4120
21	MFCC-9	-4,8298
22	MFCC-10	-10,8160
23	MFCC-11	-2,6851
24	MFCC-12	-4,8315
25	MFCC-13	-3,7721
26	MFCC-14	-9,9427
27	MFCC-15	-7,4301
28	MFCC-16	-5,8291
29	MFCC-17	-6,4241
30	MFCC-18	0,5640
31	MFCC-19	-0,4950
32	MFCC-20	-4,7686
33	SC	2096,6777
34	SS	1851,4492
35	SR	4315,9111
36	ZCR	0,1442
37	RMS	0,0595

Ekstraksi Numerik

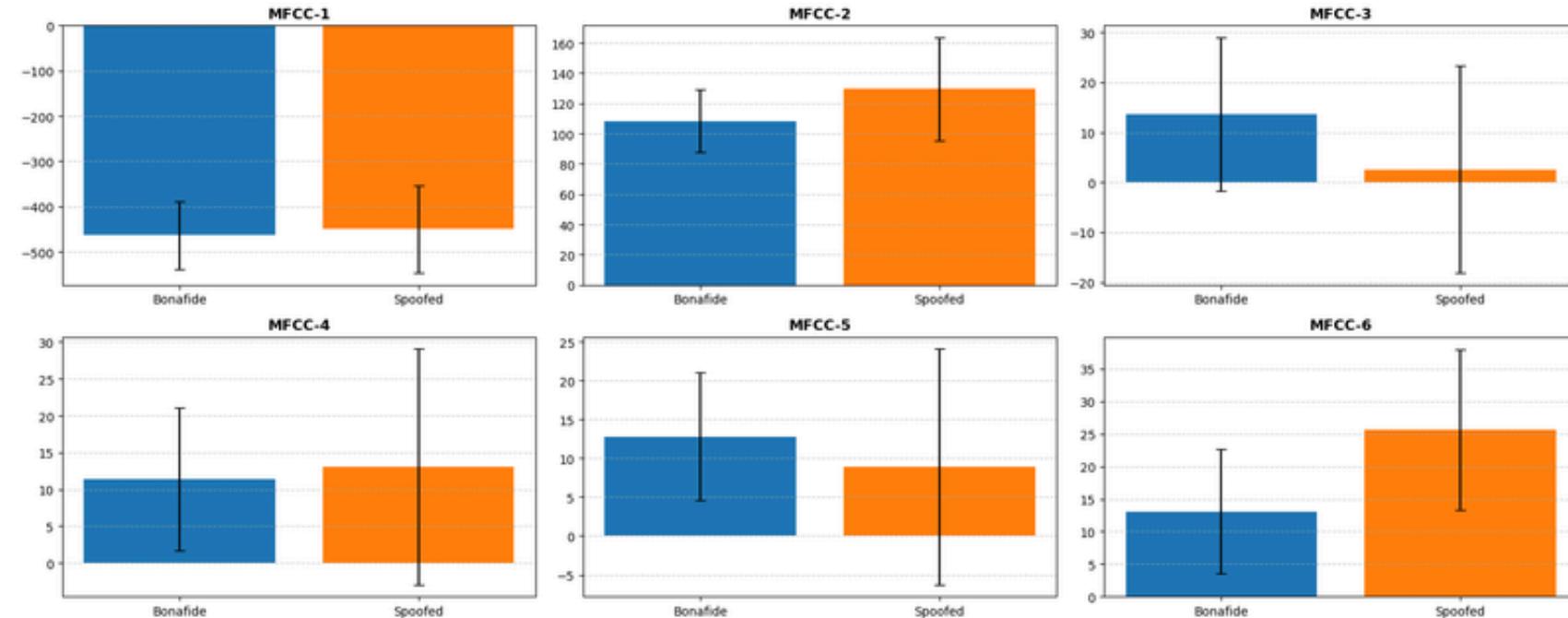
Bar Chart Mean ± Std per Feature (1-6)



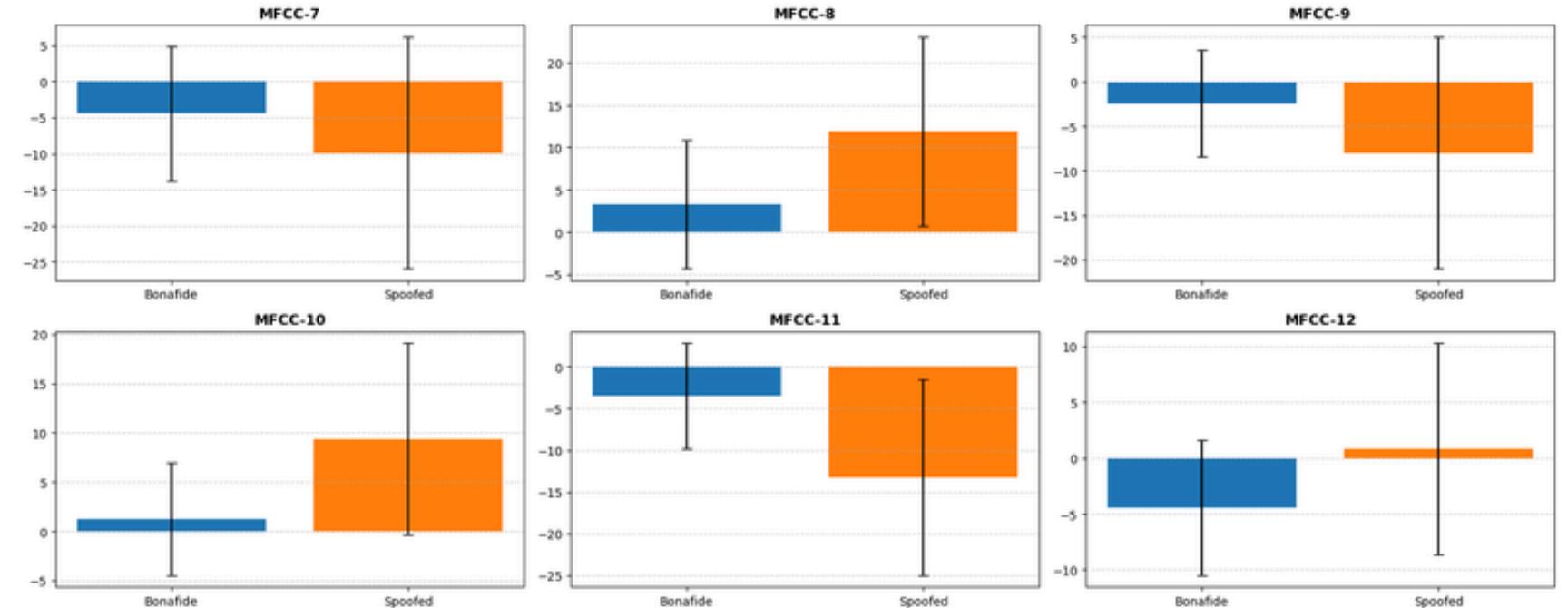
Bar Chart Mean ± Std per Feature (7-12)



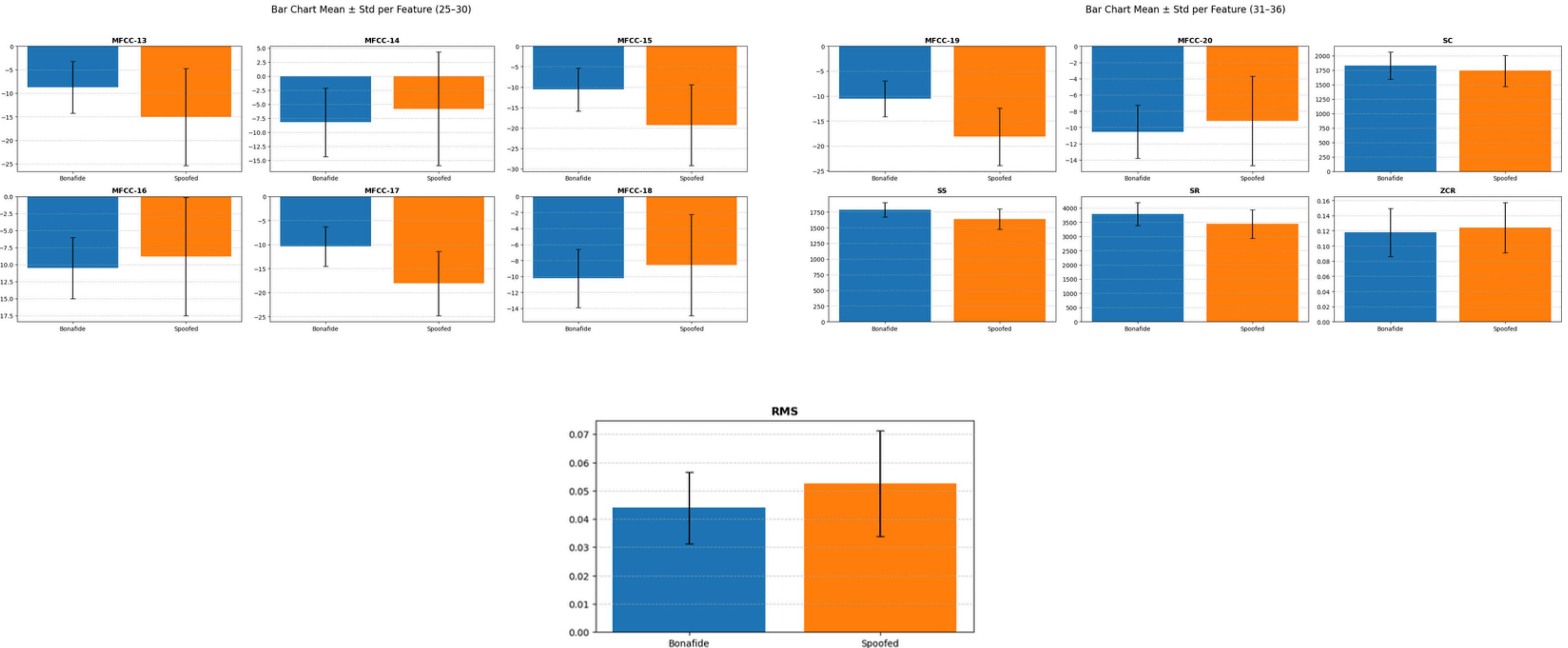
Bar Chart Mean ± Std per Feature (13-18)



Bar Chart Mean ± Std per Feature (19-24)



Ekstraksi Numerik



Ekstraksi Numerik

Z-test

Two Sample Independent

Apakah kedua kelas memiliki rata-rata yang berbeda secara statistik?

Menyusun Hipotesis

Hipotesis Nol untuk fitur ke- i ($H_0^{(i)}$):

$$H_0: \mu_{spoofed,i} = \mu_{bonafide,i}$$

Hipotesis Alternatif untuk fitur ke- i ($H_1^{(i)}$):

$$H_1: \mu_{spoofed,i} \neq \mu_{bonafide,i}$$

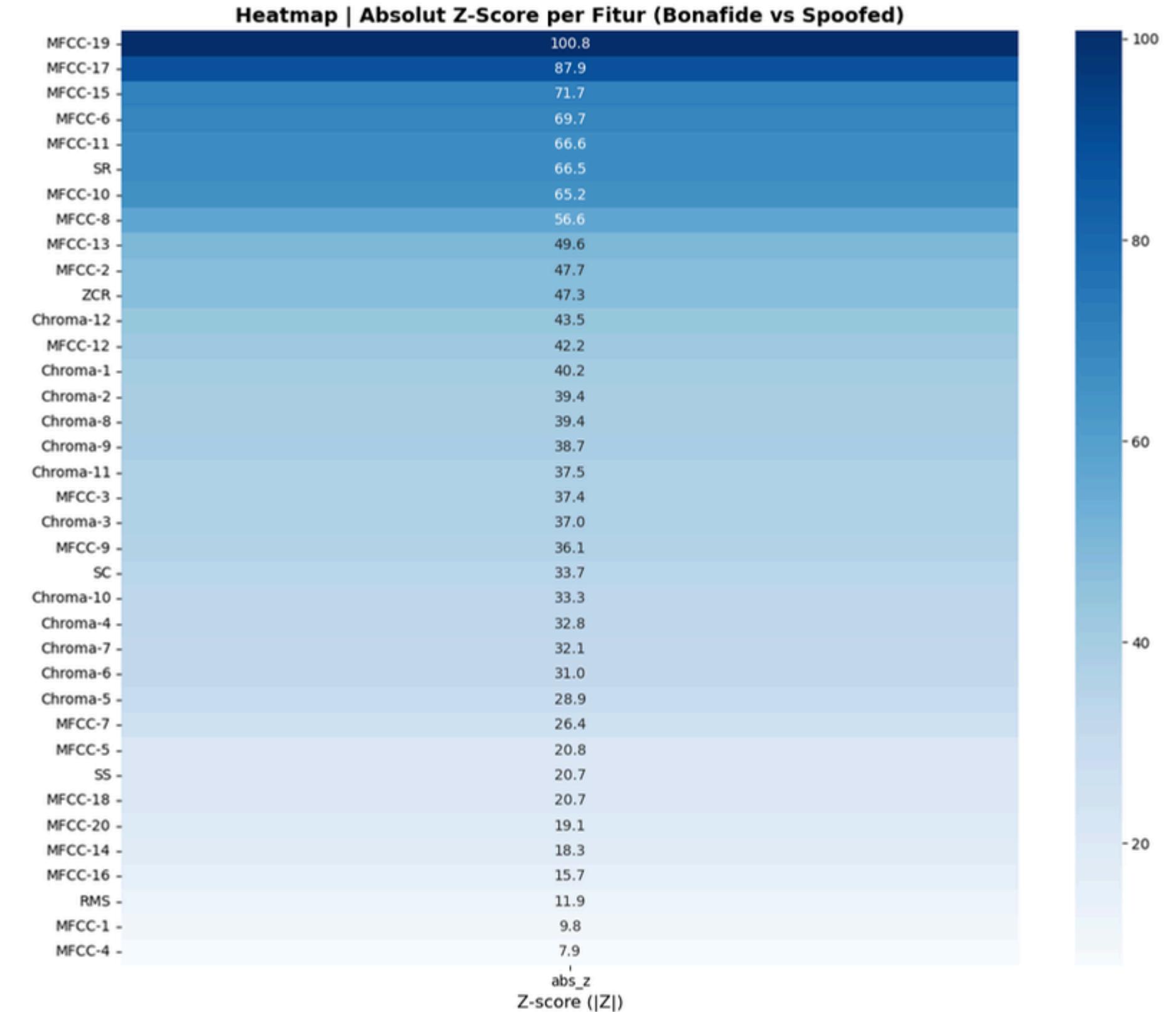
Titik Kritis

$$\alpha = 0,05$$

$$|Z_{table}| = 1,96$$

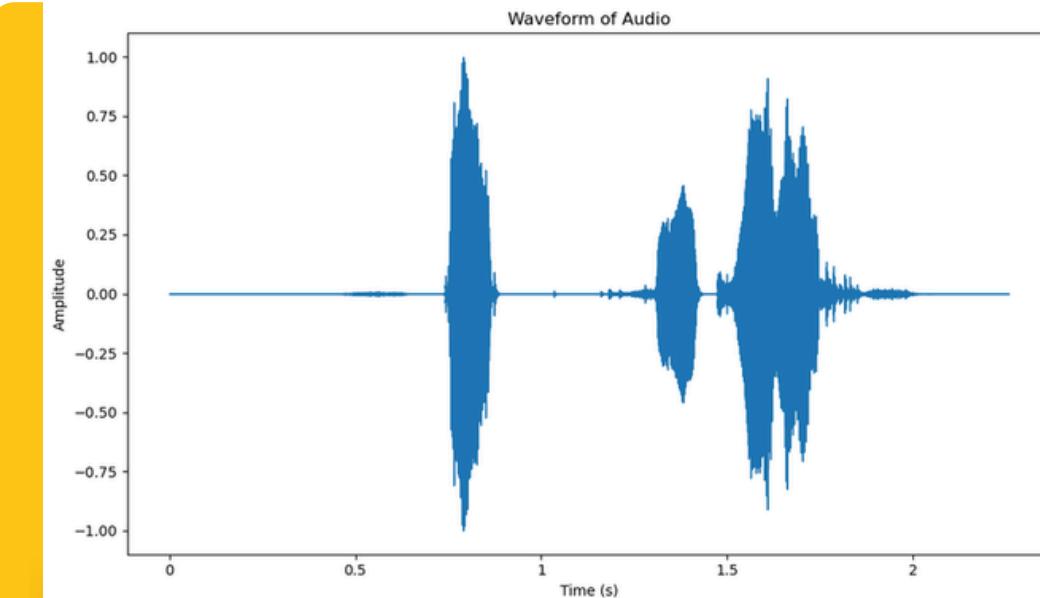
Keputusan

Untuk semua fitur $Z_{hit} > Z_{table}$, sehingga didapatkan keputusan Tolak H_0

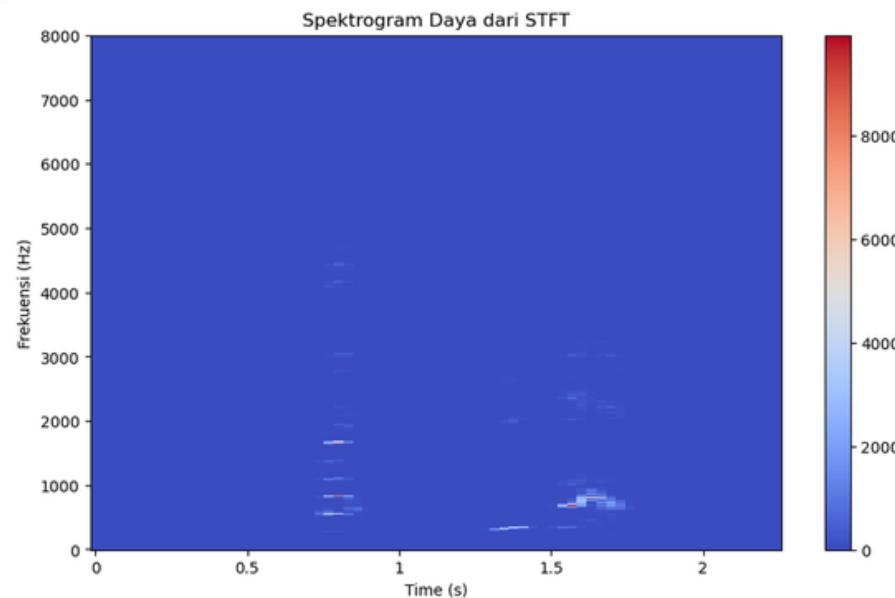


Ekstraksi Image

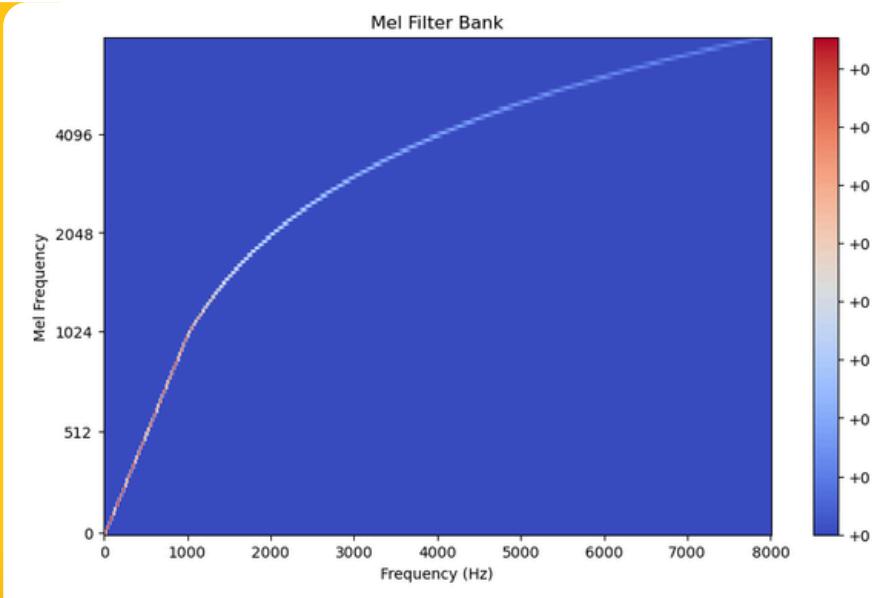
Waveform Audio



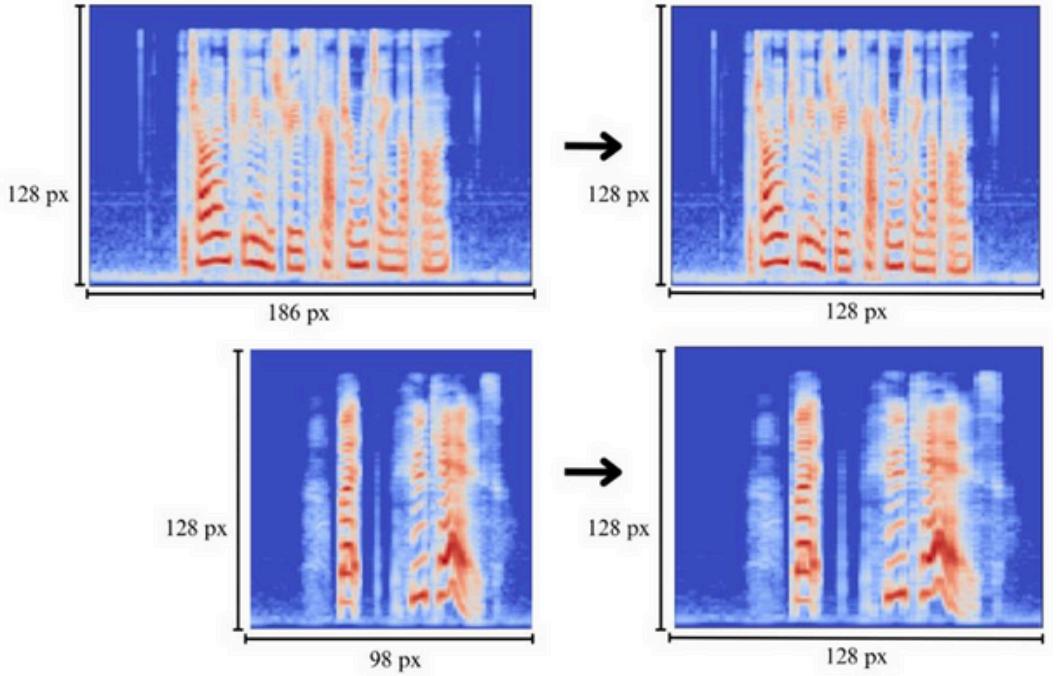
Spectrogram tanpa Log



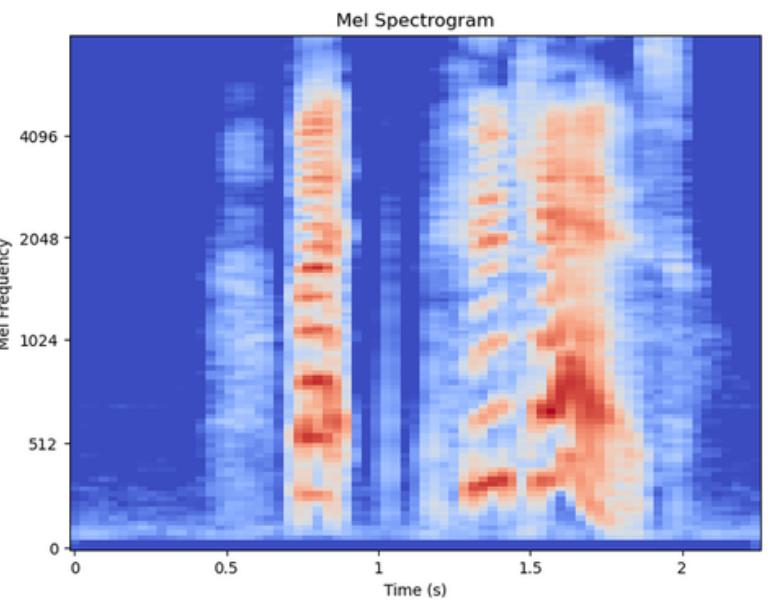
Mel Filter Bank



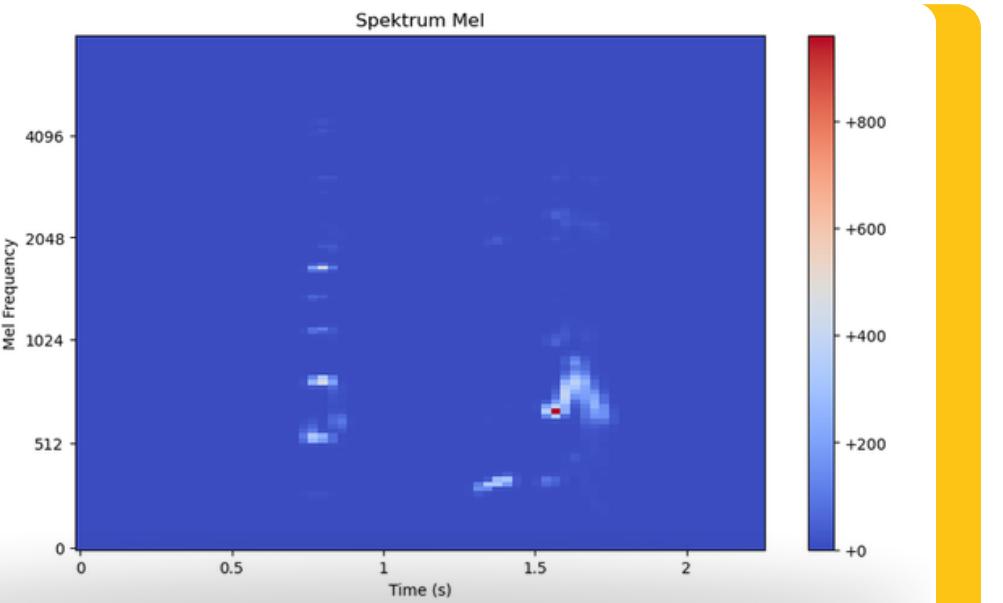
Resize Mel-Spectrogram + Normalisasi



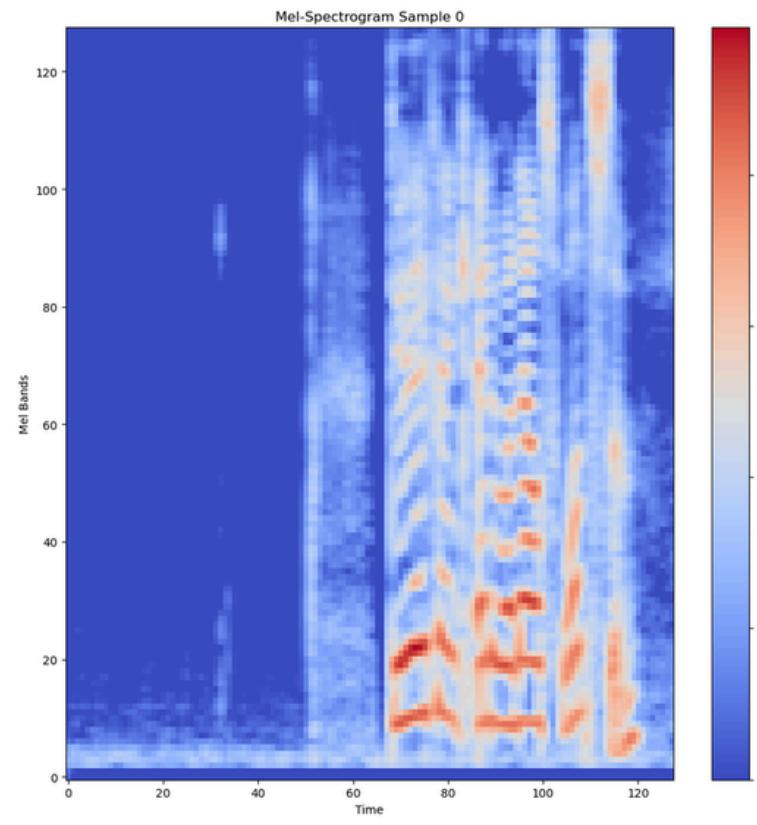
Mel-Spectrogram



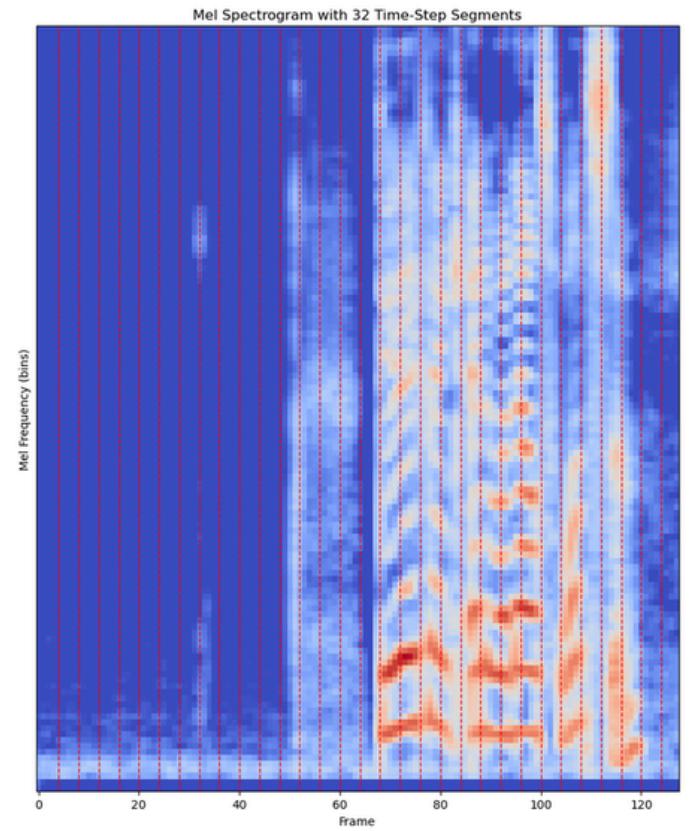
Mel-Spectrogram tanpa Log



Ekstraksi Image



CNN



CNN –
LSTM

frame	mel_1	mel_2	mel_3	...	mel_126	mel_127	mel_128
1	0.073	0.003	0.000	...	0.000	0.000	0.000
2	0.103	0.027	0.002	...	0.000	0.000	0.007
3	0.244	0.265	0.264	...	0.205	0.251	0.274
...
126	0.000	0.000	0.000	...	0.000	0.000	0.000
127	0.000	0.000	0.000	...	0.000	0.000	0.000
128	0.000	0.000	0.000	...	0.000	0.000	0.000

frame	mel_1	mel_2	mel_3	mel_126	mel_127	mel_128	
1	0.073	0.003	0.000	...	0.000	0.000	0.000
2	0.103	0.027	0.002	...	0.000	0.000	0.007
3	0.244	0.265	0.264	...	0.205	0.251	0.274
4	0.282	0.301	0.284	...	0.260	0.272	0.311
5	0.249	0.319	0.298	...	0.324	0.289	0.262
6	0.235	0.265	0.251	...	0.241	0.248	0.178
7	0.097	0.069	0.122	...	0.262	0.223	0.108
8	0.041	0.023	0.005	...	0.260	0.192	0.172
9	0.029	0.033	0.040	...	0.260	0.230	0.262
10	0.039	0.077	0.091	...	0.194	0.235	0.277
11	0.003	0.038	0.054	...	0.190	0.181	0.216
12	0.000	0.000	0.000	...	0.143	0.205	0.213
...
125	0.000	0.000	0.000	...	0.000	0.000	0.000
126	0.000	0.000	0.000	...	0.000	0.000	0.000
127	0.000	0.000	0.000	...	0.000	0.000	0.000
128	0.000	0.000	0.000	...	0.000	0.000	0.000

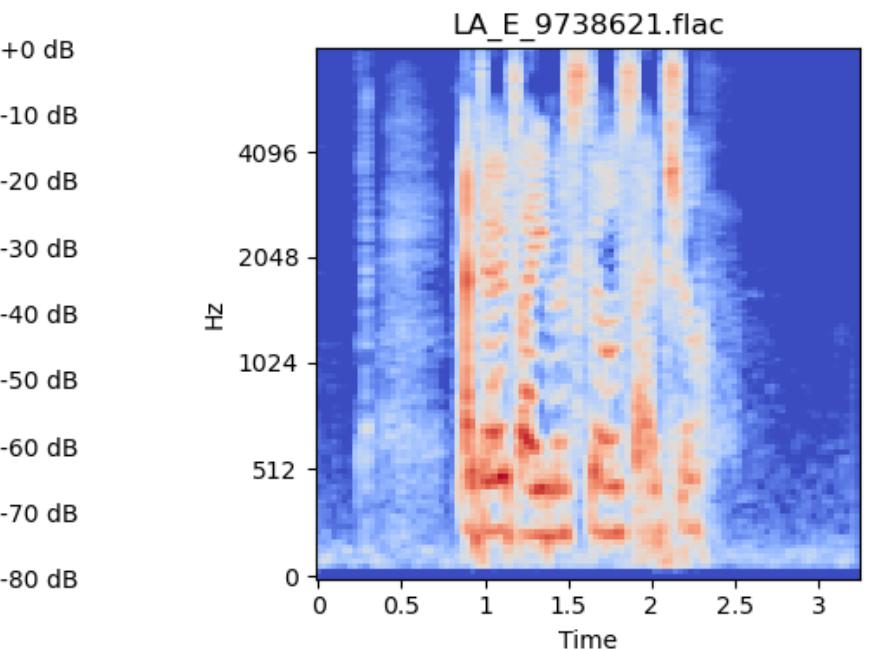
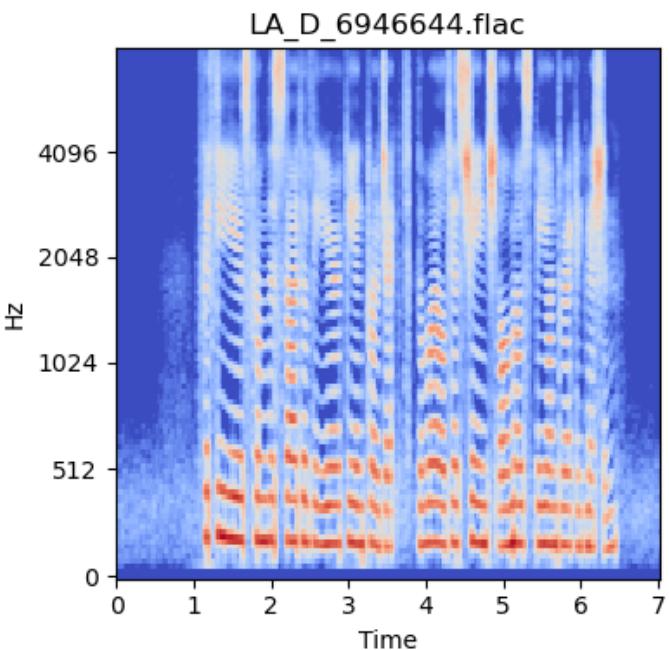
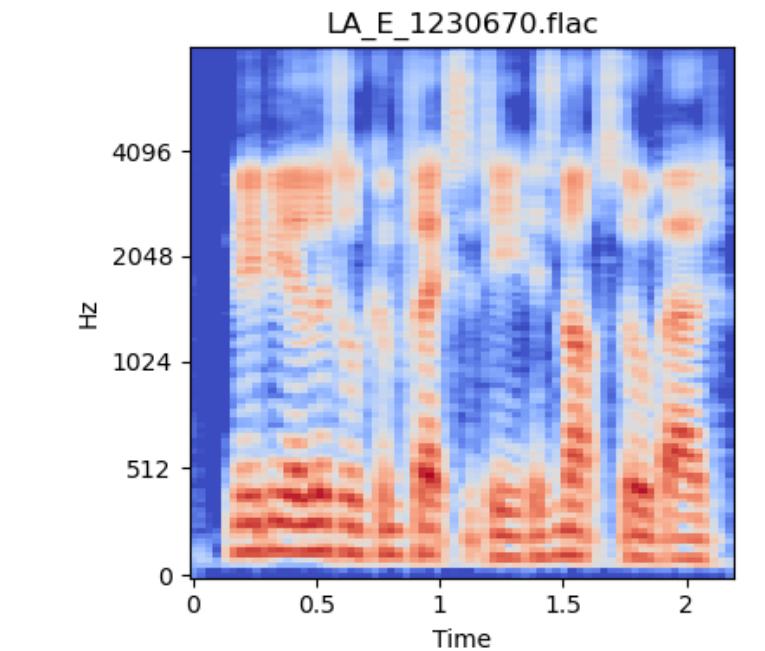
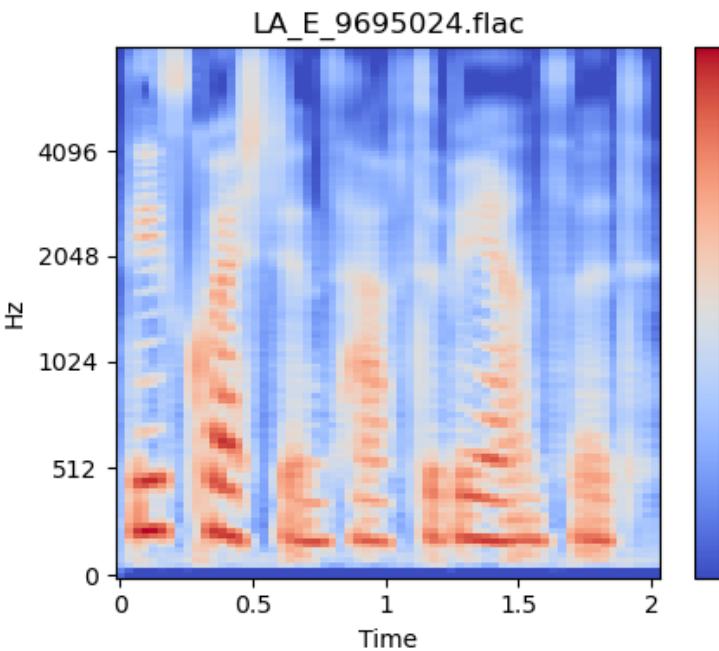
Semakin terang gambar pada mel-spectrogram menunjukkan intensitas energi yang semakin tinggi

SLIDE 50

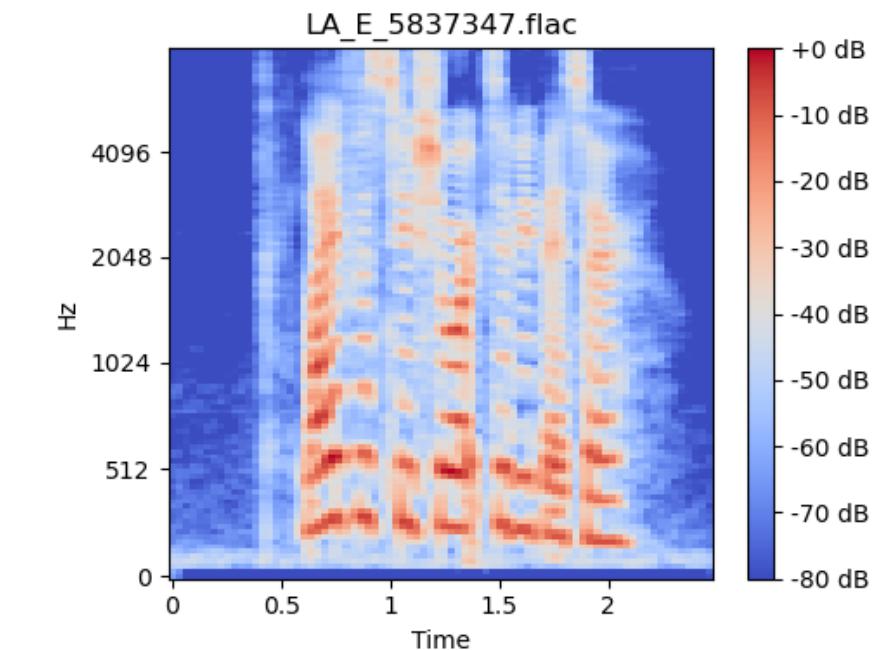
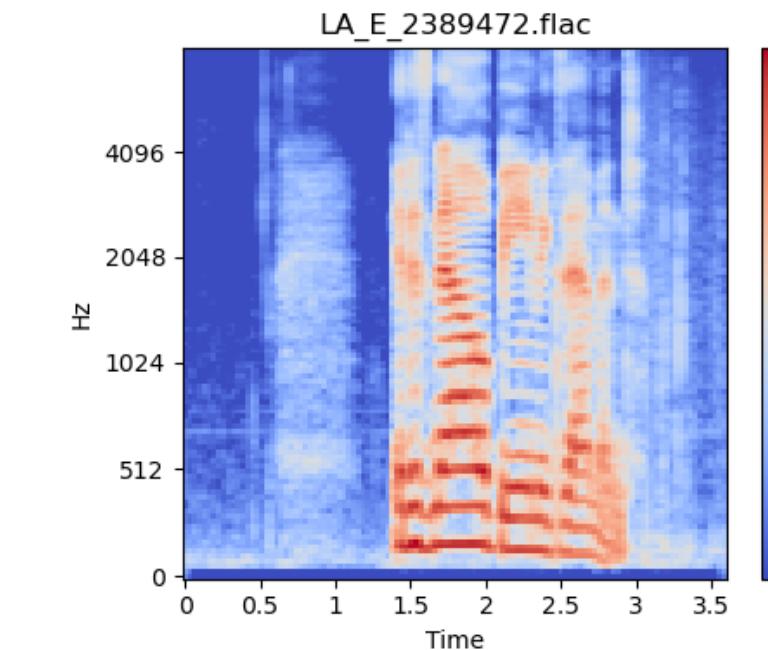
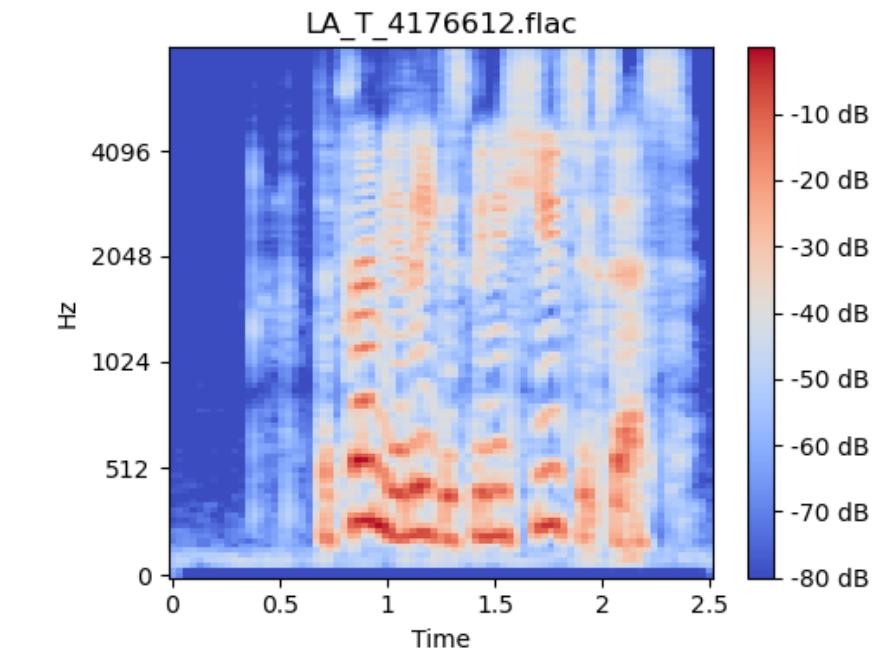
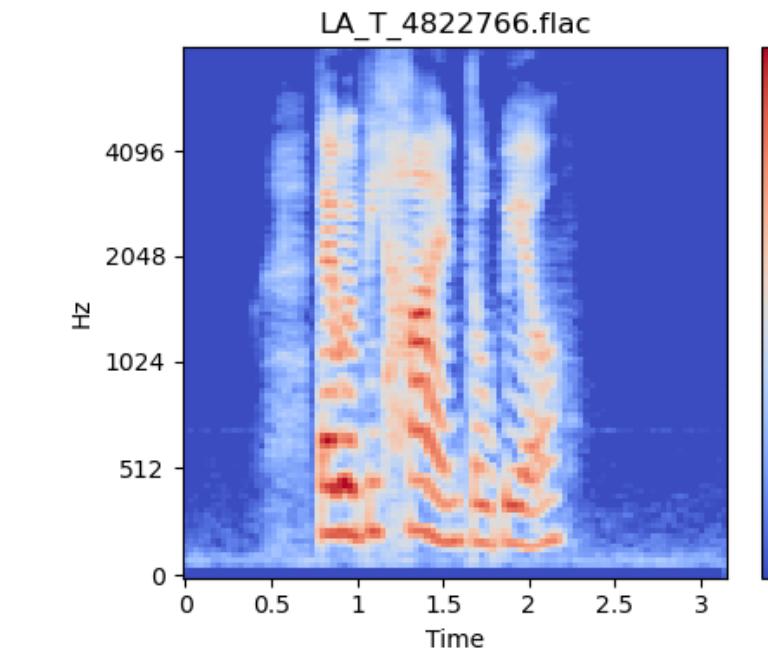


Ekstraksi Image

Spoofed

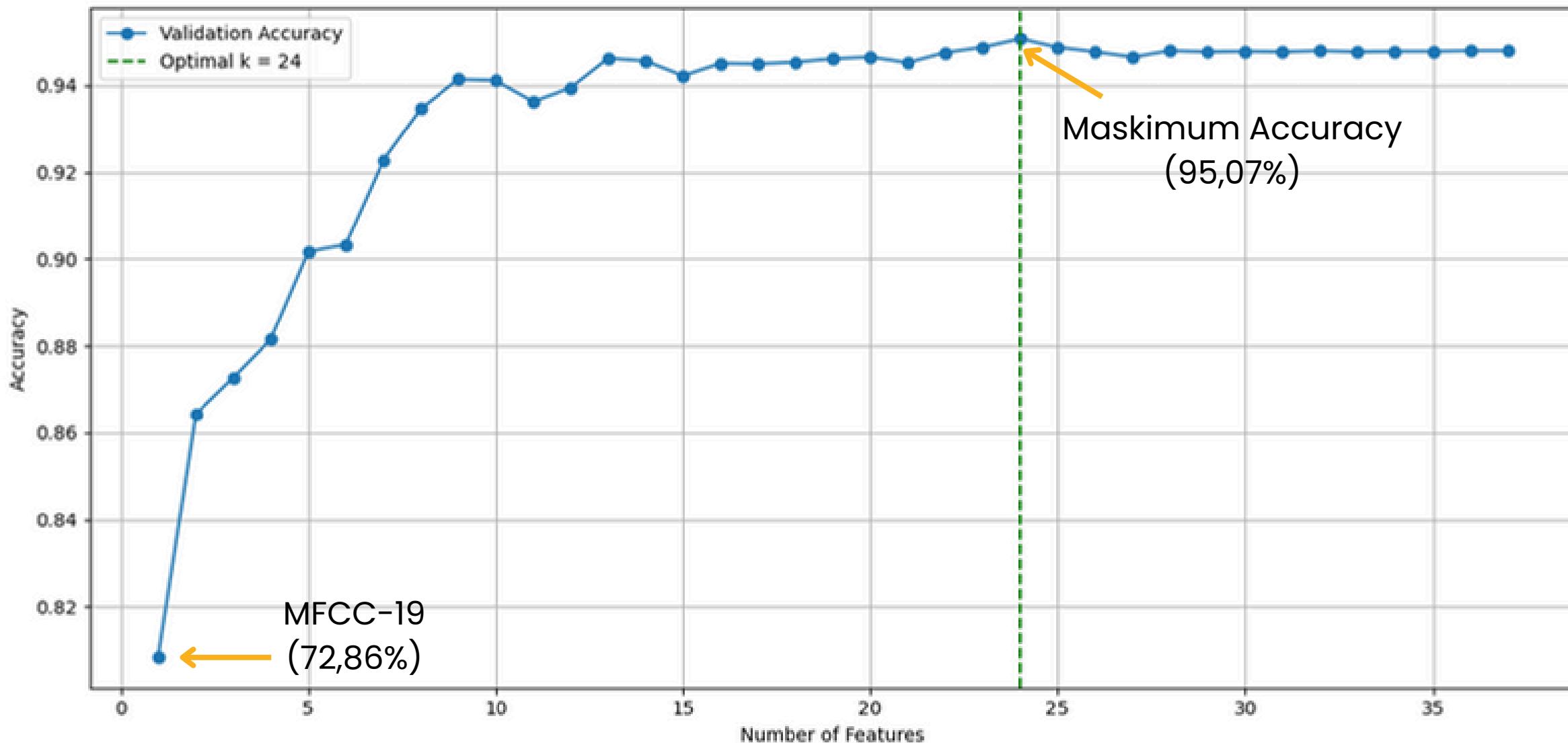


Bonafide



Pemodelan SVM

Feature Selection – Metode RFE



Fitur Terpilih

- Chroma-1
- Chroma-4
- Chroma-7
- Chroma-8
- Chroma-11
- RMS
- SC
- SS
- SR
- ZCR
- MFCC-1
- MFCC-2
- MFCC-3
- MFCC-4
- MFCC-5
- MFCC-6
- MFCC-7
- MFCC-9
- MFCC-10
- MFCC-14
- MFCC-17
- MFCC-18
- MFCC-19
- MFCC-20

Pemodelan SVM

Hyperparameter Tuning

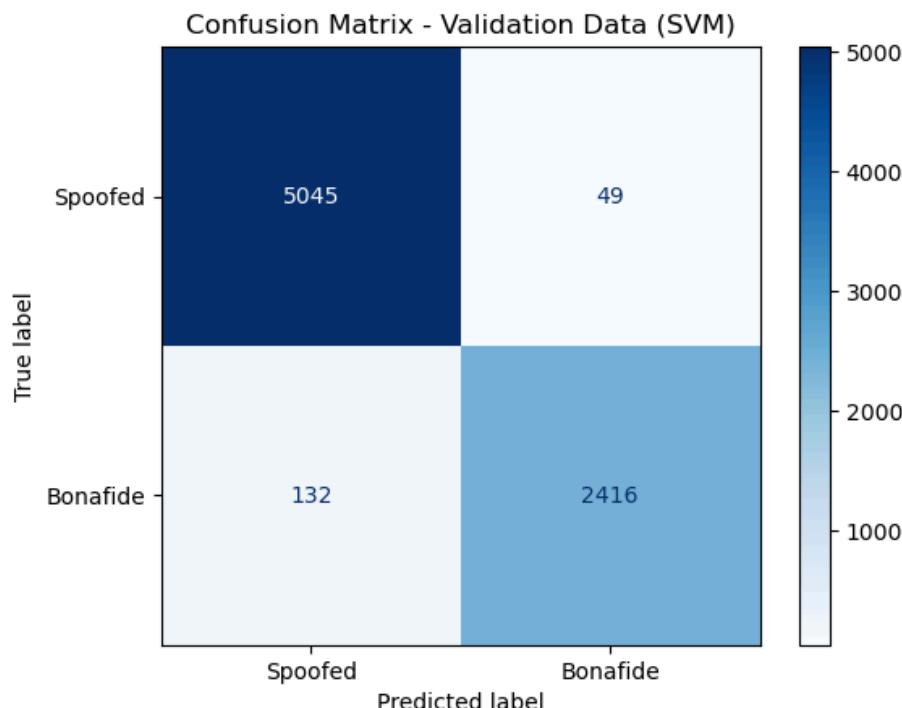
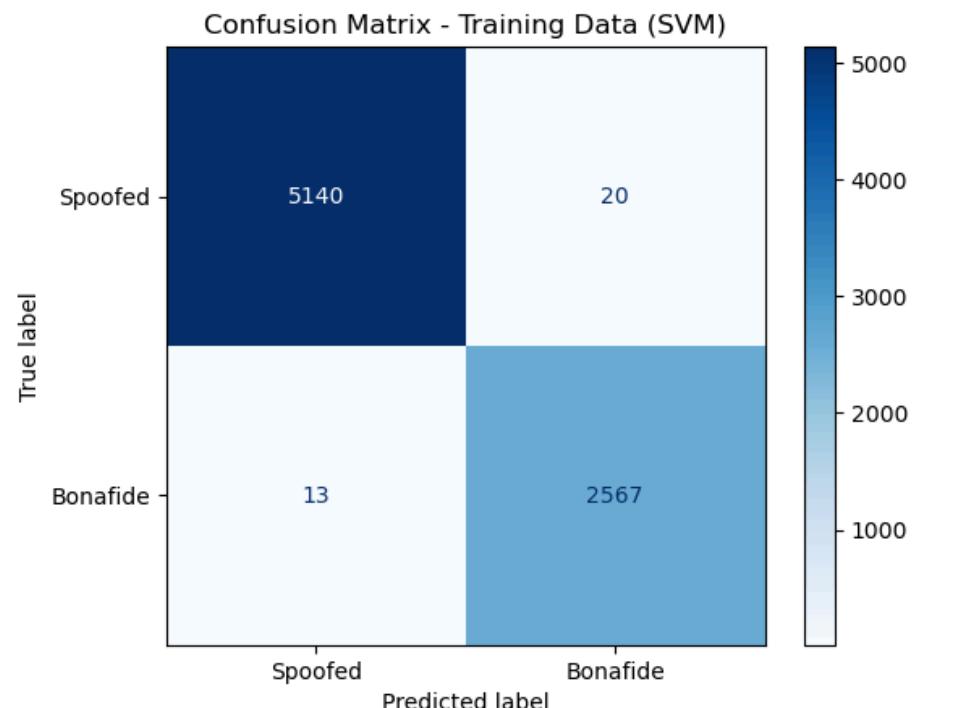
Konfigurasi Hyperparameter

Hyperparameter	Deskripsi	Konfigurasi
Kernel	Fungsi yang digunakan untuk memetakan data ke ruang fitur yang lebih tinggi	rbf
C	Hyperparameter regularisasi yang mengatur keseimbangan antara kesalahan pelatihan dan kompleksitas model	0,01 – 100
gamma	Hyperparameter yang menentukan jangkauan pengaruh data training	0,0001 – 1
class_weight	Menangani ketidakseimbangan kelas dengan memberikan bobot	[None, 'balanced']

Hasil Optuna:

- Percobaan ke-26
- C = 19,520
- gamma = 0,479

Performa Model



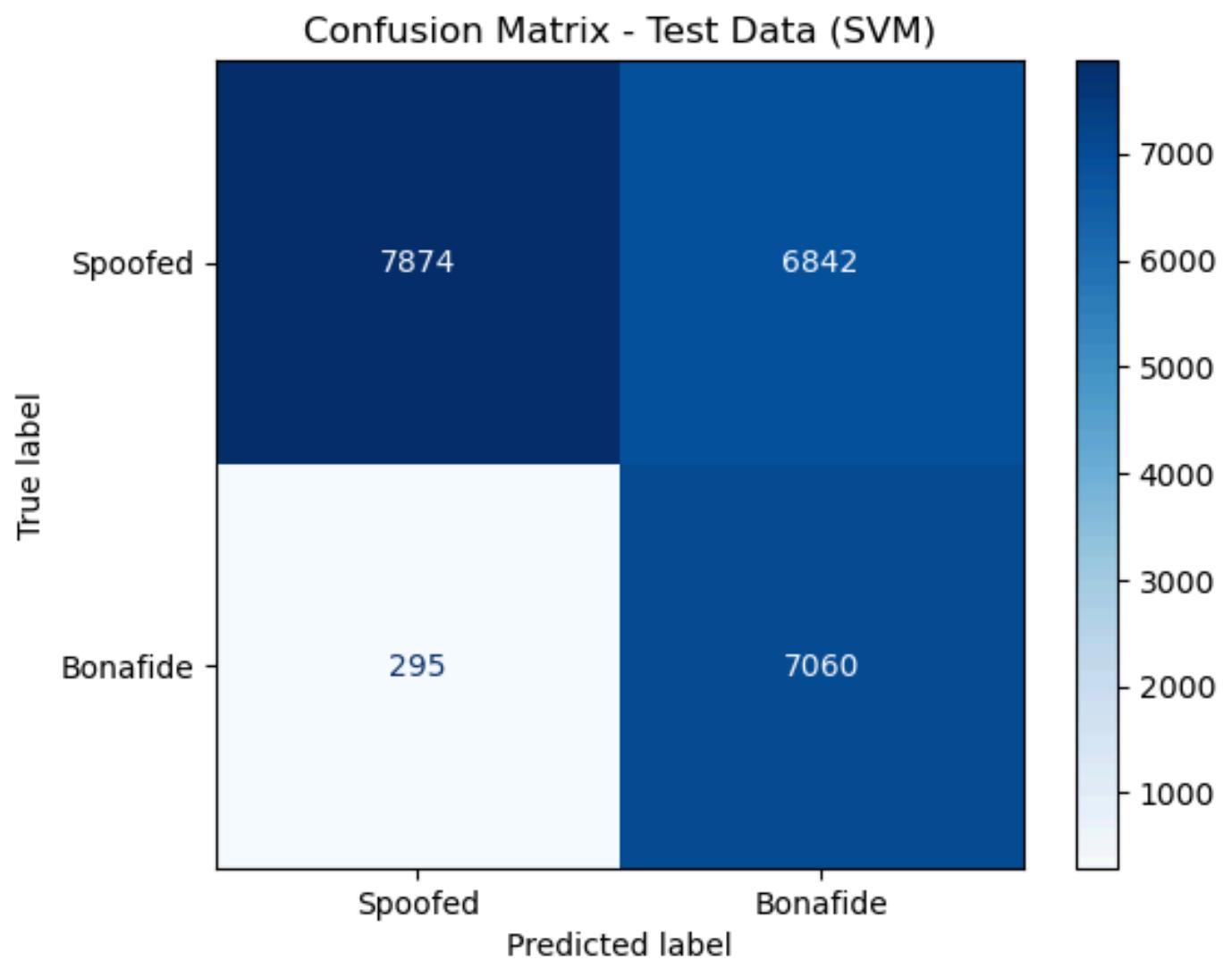
Akurasi Training
99,57%

Akurasi Validation
97,63%

Performa Sangat Baik
Tidak Menunjukkan Overfitting



Evaluasi Model



Efektivitas Ketepatan Klasifikasi

Kelas	Presisi	Recall	F1 Score	Akurasi
<i>Spoofed</i>	96,39%	53,51%	68,81%	
<i>Bonafide</i>	50,78%	95,99%	66,43%	
Rata-Rata	73,59%	74,75%	67,62%	67,66%

Efisiensi Waktu Prediksi

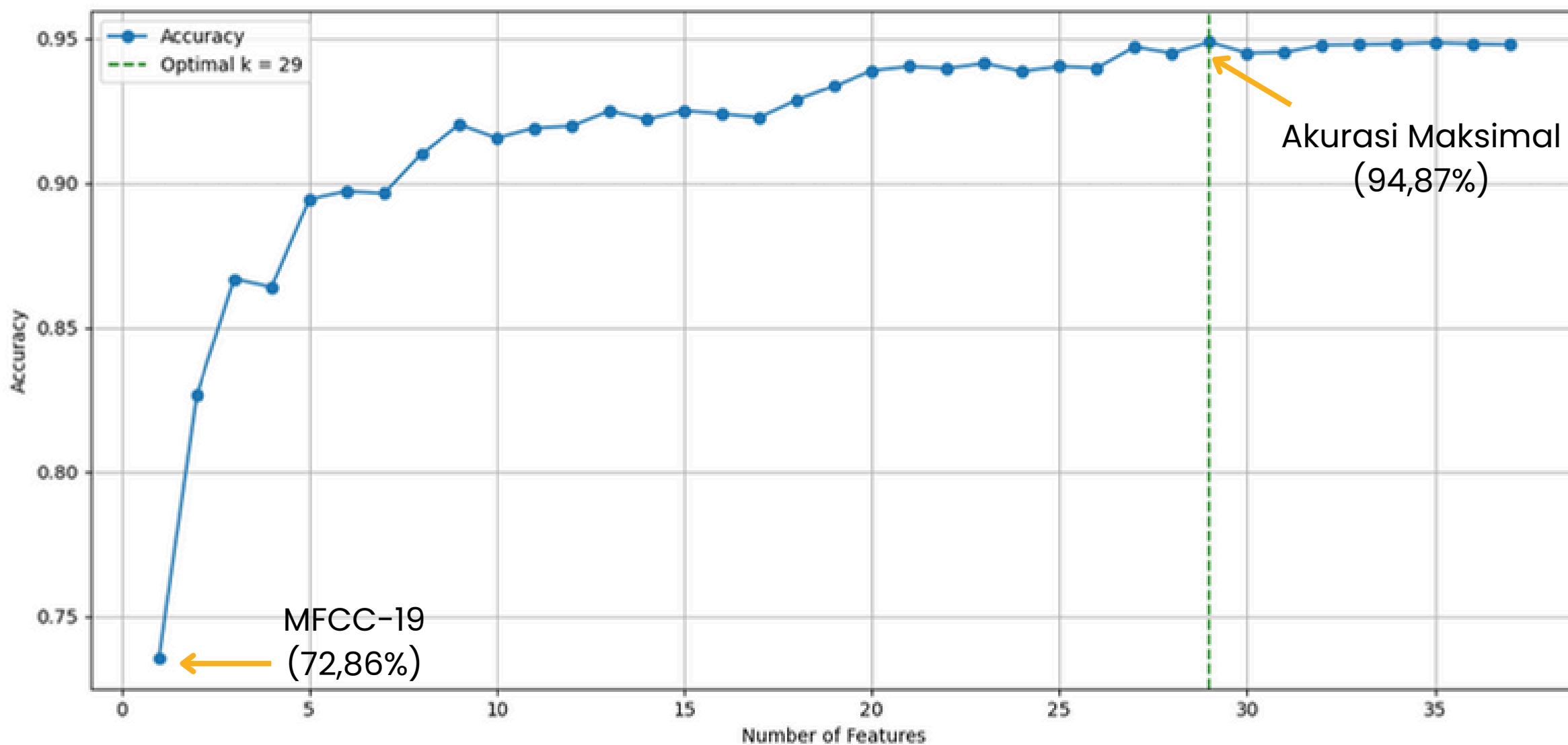
Waktu Prediksi

0,0010 detik/instance

Std = 0,0007 detik



Feature Selection – Metode RFE



Fitur Terpilih

- Chroma-1
- Chroma-2
- Chroma-3
- Chroma-4
- Chroma-8
- Chroma-9
- Chroma-11
- Chroma-12
- SS
- SR
- MFCC-1
- MFCC-3
- MFCC-4
- MFCC-5
- MFCC-6
- MFCC-7
- MFCC-8
- MFCC-9
- MFCC-10
- MFCC-11
- MFCC-12
- MFCC-13
- MFCC-14
- MFCC-15
- MFCC-16
- MFCC-17
- MFCC-18
- MFCC-19
- MFCC-20

Hyperparameter Tuning

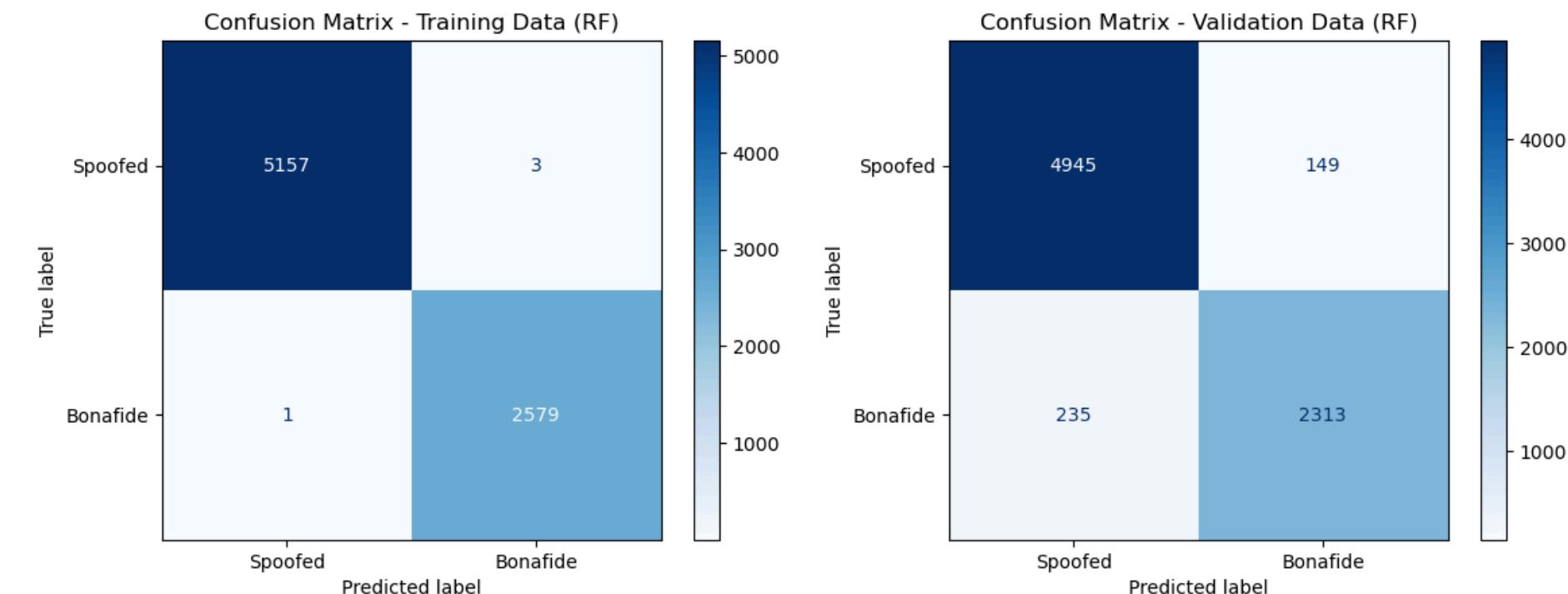
Konfigurasi Hyperparameter

Hyperparameter	Deskripsi	Konfigurasi
<i>n_estimators</i>	Menetukan jumlah <i>hyperparameter</i> yang dibuat dalam <i>ensemble</i>	100 – 300
<i>max_depth</i>	Mengatur kedalaman maksimum pohon keputusan	5 – 20
<i>min_samples_split</i>	Menentukan jumlah minimum sampel yang diperlukan untuk membagi sebuah <i>node</i> menjadi cabang baru	2 – 10
<i>min_samples_leaf</i>	Mengatur jumlah minimum sampel yang harus ada pada sebuah daun pohon	1 – 5
<i>max_features</i>	Menentukan jumlah fitur yang dipertimbangkan saat membagi <i>node</i>	['sqrt', 'log2', 0,5]
<i>class_weight</i>	Menangani ketidakseimbangan kelas dengan memberikan bobot	[None, 'balanced']

Hasil Optuna:

- Percobaan ke-41
- *n_estimators* = 111
- *max_depth* = 15
- *min_samples_split* = 4
- *min_samples_leaf* = 2
- *max_features* = log2
- *class_weight* = None

Performa Model

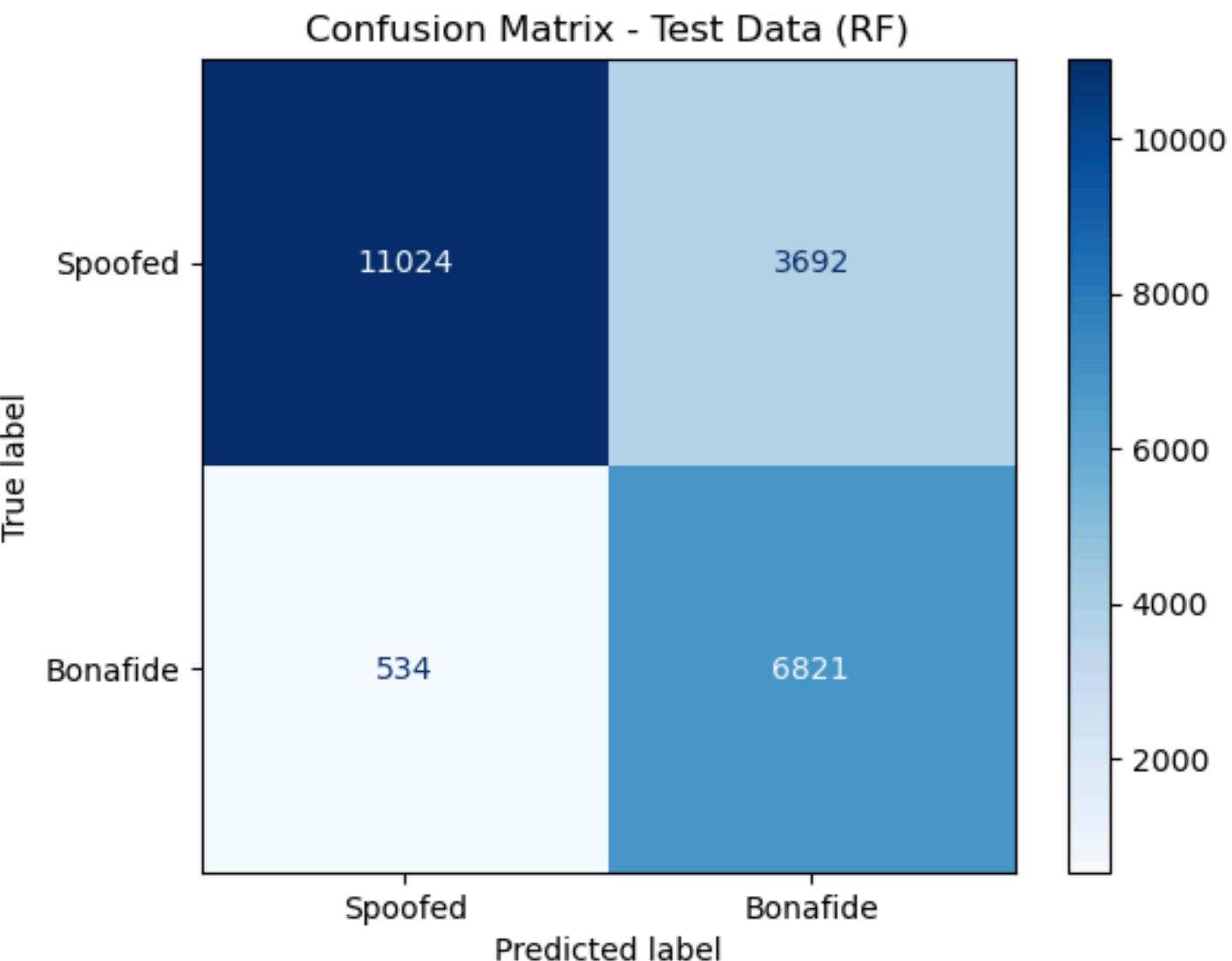


Akurasi Training
99,95%

Akurasi Validation
94,98%

Performa Sangat Baik
Tidak Menunjukkan Overfitting ✓

Evaluasi Model



Efektivitas Ketepatan Klasifikasi

Kelas	Presisi	Recall	F1_Score	Akurasi
<i>Spoofed</i>	95,38%	74,91%	83,92%	
<i>Bonafide</i>	64,88%	92,74%	76,35%	
Rata-Rata	80,13%	83,83%	80,13%	80,85%

Efisiensi Waktu Prediksi

Waktu Prediksi

0,0046 detik/instance

Std = 0,0015 detik

Arsitektur CNN

<i>Layer Type</i>	<i>Output Shape</i>	<i>Parameters</i>
<i>Conv2D</i>	(None, 126, 126, 8)	80
<i>MaxPool2D</i>	(None, 63, 63, 8)	0
<i>Conv2D</i>	(None, 61, 61, 16)	1.168
<i>MaxPool2D</i>	(None, 30, 30, 16)	0
<i>Flatten</i>	(None, 14400)	0
<i>Dense</i>	(None, 32)	460.832
<i>Dropout</i>	(None, 32)	0
<i>Dense (sigmoid)</i>	(None, 1)	33
<i>Total Params</i>		462.115 (1,76 MB)
<i>Trainable Params</i>		462.113 (1,76 MB)
<i>Non-Trainable Params</i>		0

Hyperparameter Tuning

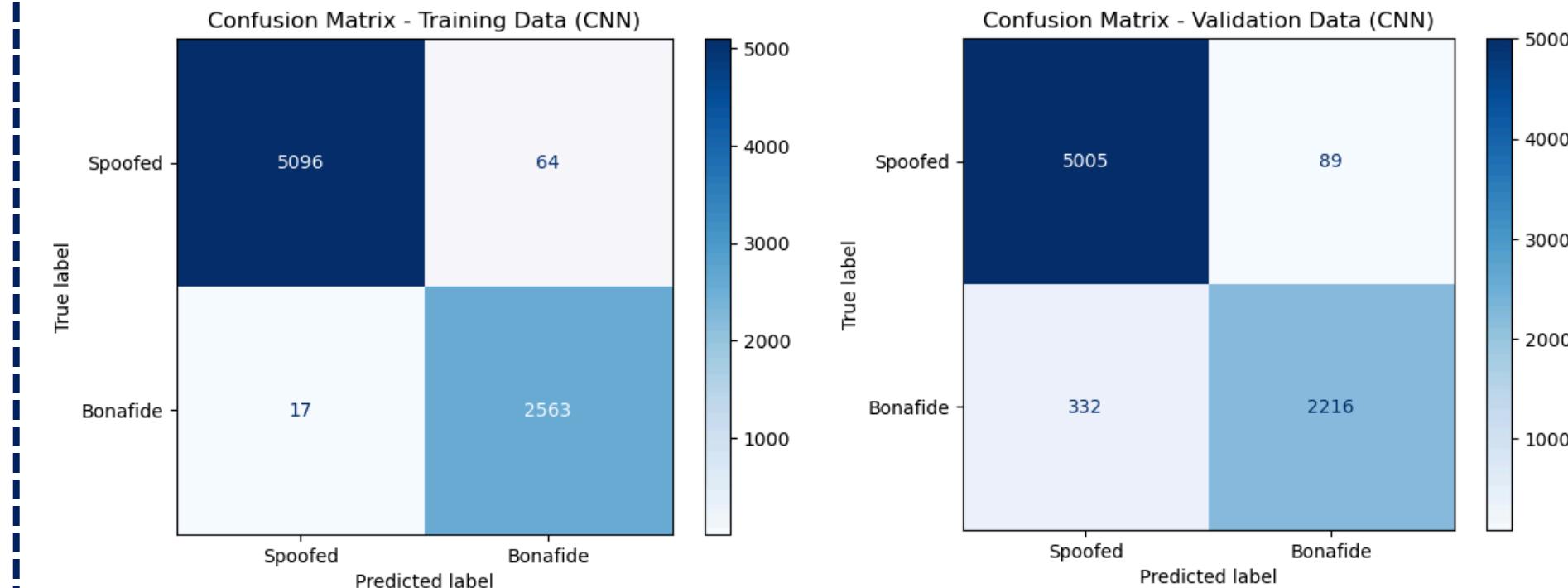
Konfigurasi Hyperparameter

Hyperparameter	Deskripsi	Konfigurasi
Optimizer	Algoritma memperbarui bobot saat pelatihan	Adam
Learning Rate	Kecepatan pembaruan bobot oleh optimizer	[1e-5, 1e-4, 1e-3, 1e-2, 1e-1]
Drop out	Persentase <i>neuron</i> yang dinonaktifkan	[0.1, 0.2, 0.3, 0.4, 0.5]
Batch size	Ukuran sampel yang diproses dalam satu iterasi	[32, 64, 128]
Epoch	Jumlah siklus penuh penggunaan seluruh data <i>training</i>	20

Hasil Optuna:

- Percobaan ke-15
- Learning Rate = 0,001
- Dropout Rate = 0,5
- Batch Size = 32

Performa Model



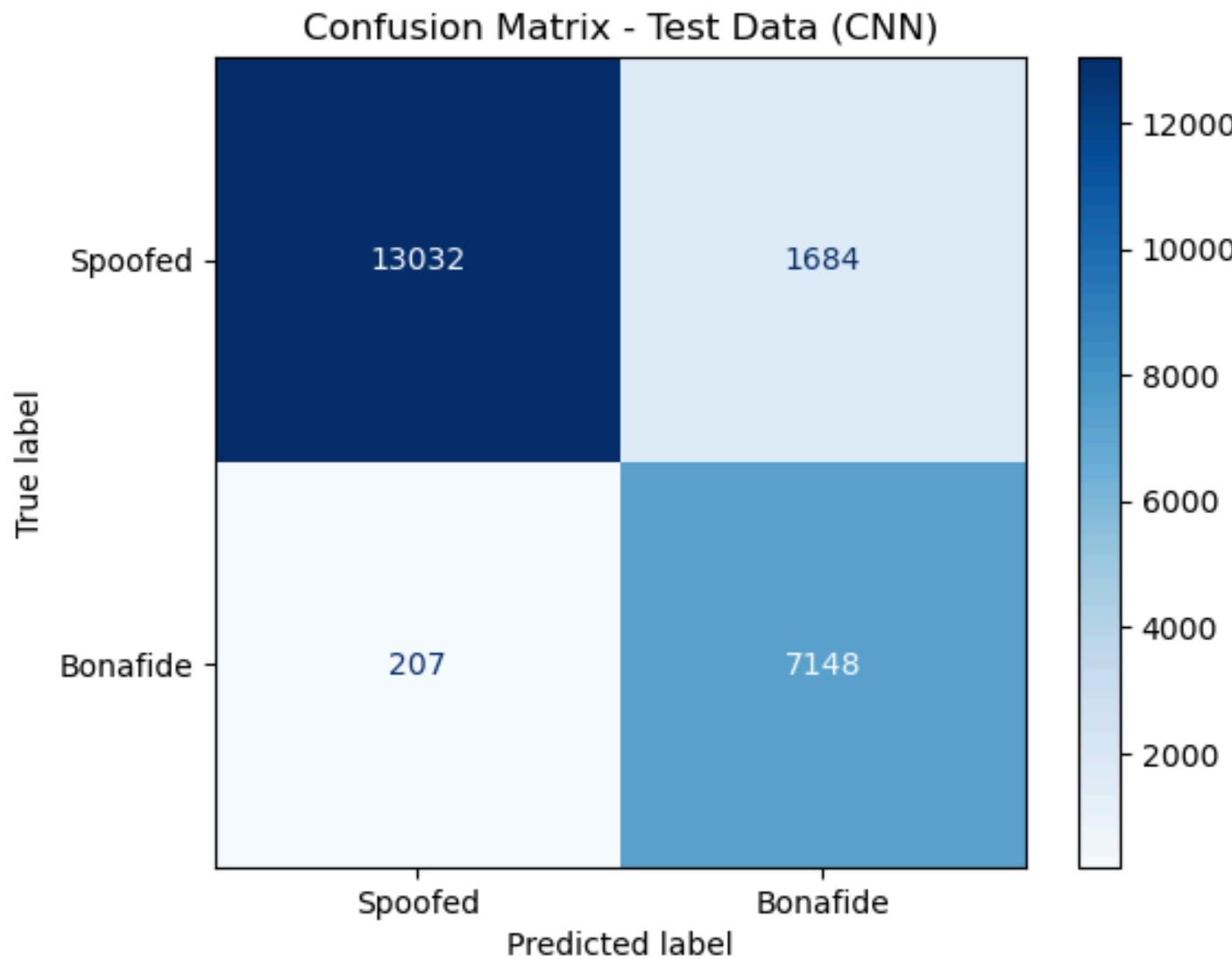
Akurasi Training
98,95%

Akurasi Validation
94,49%

Performa Sangat Baik
Tidak Menunjukkan Overfitting ✓

SLIDE 59

Evaluasi Model



Efektivitas Ketepatan Klasifikasi

Kelas	Presisi	Recall	F1_Score	Akurasi
<i>Spoofed</i>	98,44%	88,56%	93,24%	
<i>Bonafide</i>	80,93%	97,19%	88,32%	
Rata-Rata	89,68%	92,87%	90,78%	91,43%

Efisiensi Waktu Prediksi

Waktu Prediksi

0,1148 detik/instance

Std = 0,0429 detik

Pemodelan CNN-LSTM

Arsitektur CNN-LSTM

<i>Layer Type</i>	<i>Output Shape</i>	<i>Parameters</i>
<i>TimeDistributed(Conv2D)</i>	(None, 32, 128, 4, 8)	80
<i>TimeDistributed(BatchNorm)</i>	(None, 32, 128, 4, 8)	32
<i>TimeDistributed(MaxPool2D)</i>	(None, 32, 64, 2, 8)	0
<i>TimeDistributed(Conv2D)</i>	(None, 32, 64, 2, 16)	1.168
<i>TimeDistributed(MaxPool2D)</i>	(None, 32, 32, 1, 16)	0
<i>TimeDistributed(Flatten)</i>	(None, 32, 512)	0
<i>LSTM (return_sequences=True)</i>	(None, 32, 64)	147.712
<i>LSTM (return_sequences=False)</i>	(None, 32)	12.416
<i>Dense</i>	(None, 32)	1.056
<i>Dropout</i>	(None, 32)	0
<i>Dense (sigmoid)</i>	(None, 1)	33
Total Params		162.499 (634,77 KB)
Trainable Params		162.481 (634,39 KB)
Non-Trainable Params		16 (64,00 B)

Pemodelan CNN-LSTM

Hyperparameter Tuning

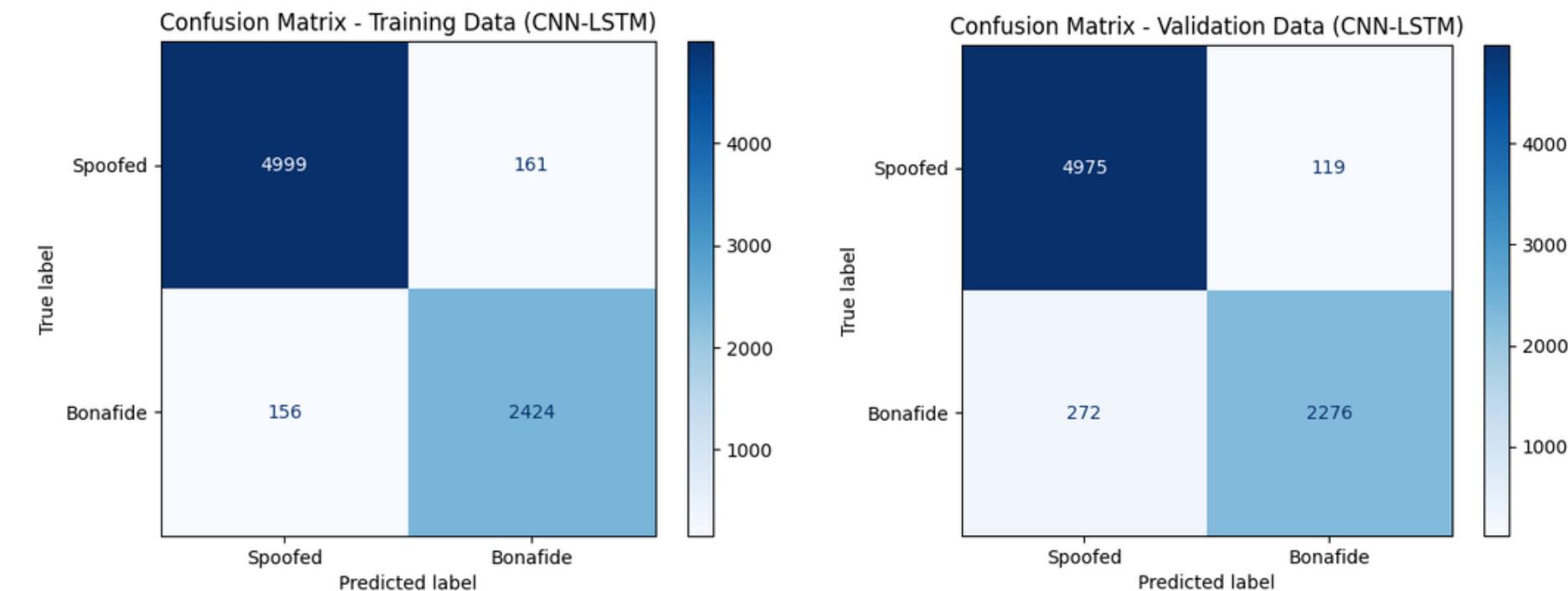
Konfigurasi Hyperparameter

Hyperparameter	Deskripsi	Konfigurasi
Optimizer	Algoritma memperbarui bobot saat pelatihan	Adam
Learning Rate	Kecepatan pembaruan bobot oleh optimizer	[1e-5, 1e-4, 1e-3, 1e-2, 1e-1]
Drop out	Persentase <i>neuron</i> yang dinonaktifkan	[0.1, 0.2, 0.3, 0.4, 0.5]
Batch size	Ukuran sampel yang diproses dalam satu iterasi	[32, 64, 128]
Epoch	Jumlah siklus penuh penggunaan seluruh data <i>training</i>	20

Hasil Optuna:

- Percobaan ke-6
- Learning Rate = 0,001
- Dropout Rate = 0,4
- Batch Size = 32

Performa Model

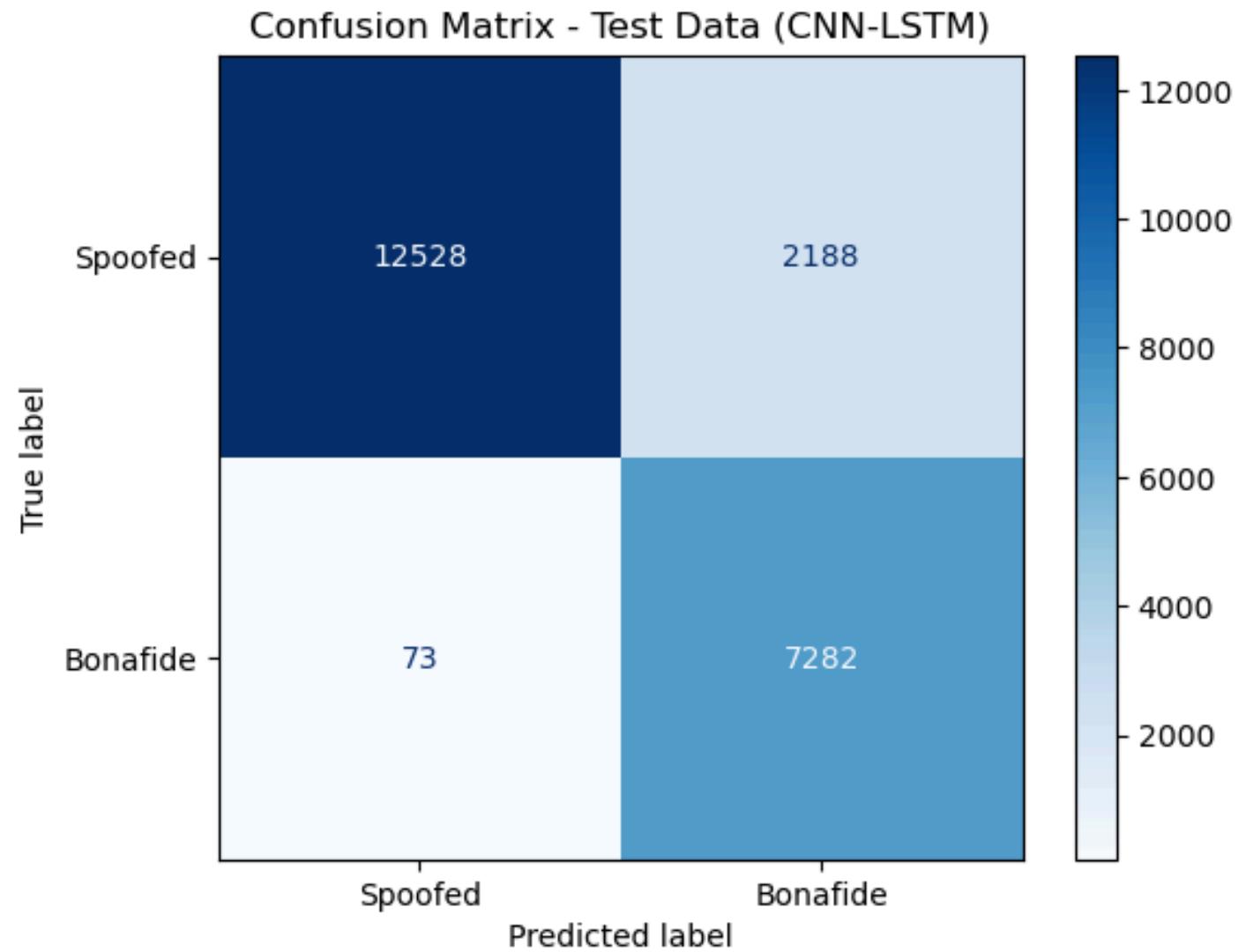


Akurasi Training
95,90%

Akurasi Validation
94,88%

Performa Sangat Baik
Tidak Menunjukkan Overfitting ✓

Evaluasi Model



Efektivitas Ketepatan Klasifikasi

Kelas	Presisi	Recall	F1 Score	Akurasi
<i>Spoofed</i>	99,42%	85,13%	91,72%	
<i>Bonafide</i>	76,90%	99,01%	86,56%	
Rata-Rata	88,16%	92,07%	89,14%	89,76%

Efisiensi Waktu Prediksi

Waktu Prediksi

0,1184 detik/instance

Std = 0,0385 detik

Perbandingan Hasil

Perbandingan Efektivitas Ketepatan Klasifikasi

Urutan	Model	<i>Training</i> Akurasi	<i>Validation</i> Akurasi	<i>Testing</i> Akurasi	<i>Testing Recall</i> <i>Spoofed</i>
1	CNN	98,95%	94,49%	91,43%	88,56%
2	CNN-LSTM	95,90%	94,88%	89,76%	85,13%
3	RF	99,95%	94,98%	80,85%	74,91%
4	SVM	99,55%	97,63%	67,66%	53,51%

Perbandingan Hasil

Perbandingan Efektivitas Ketepatan Klasifikasi

Jenis	False Pred Rate (%)			
Model	SVM	RF	CNN	CNN-LSTM
Bonafide	4,01	7,26	2,81	0,99
A07	0,00	0,00	1,59	0,00
A08	0,35	0,00	9,19	2,47
A09	0,27	1,68	0,00	0,00
A10	73,23	18,73	1,59	2,39
A11	53,00	13,78	1,15	0,44
A12	80,65	50,27	0,00	9,36
A13	76,06	17,93	0,53	6,71
A14	57,07	17,05	0,00	0,62
A15	78,80	27,39	0,18	4,06
A16	0,35	0,00	0,71	0,35
A17	88,52	83,39	56,89	97,79
A18	96,11	95,94	48,85	66,96
A19	0,00	0,00	28,09	2,12

A17

Link Suara
VC (Waveform Filtering)

A18

Link Suara
VC (Vocoder)

A07

Link Suara
TTS (Vocoder + GAN)

Perbandingan Hasil

Perbandingan Efisiensi Waktu Prediksi

Urutan	Model	Rata-Rata Waktu Prediksi (detik)	Standar Deviasi (detik)
1	SVM	0,0010	0,0007
2	RF	0,0046	0,0015
3	CNN	0,1148	0,0429
4	CNN-LSTM	0,1184	0,0385

Perbandingan Hasil

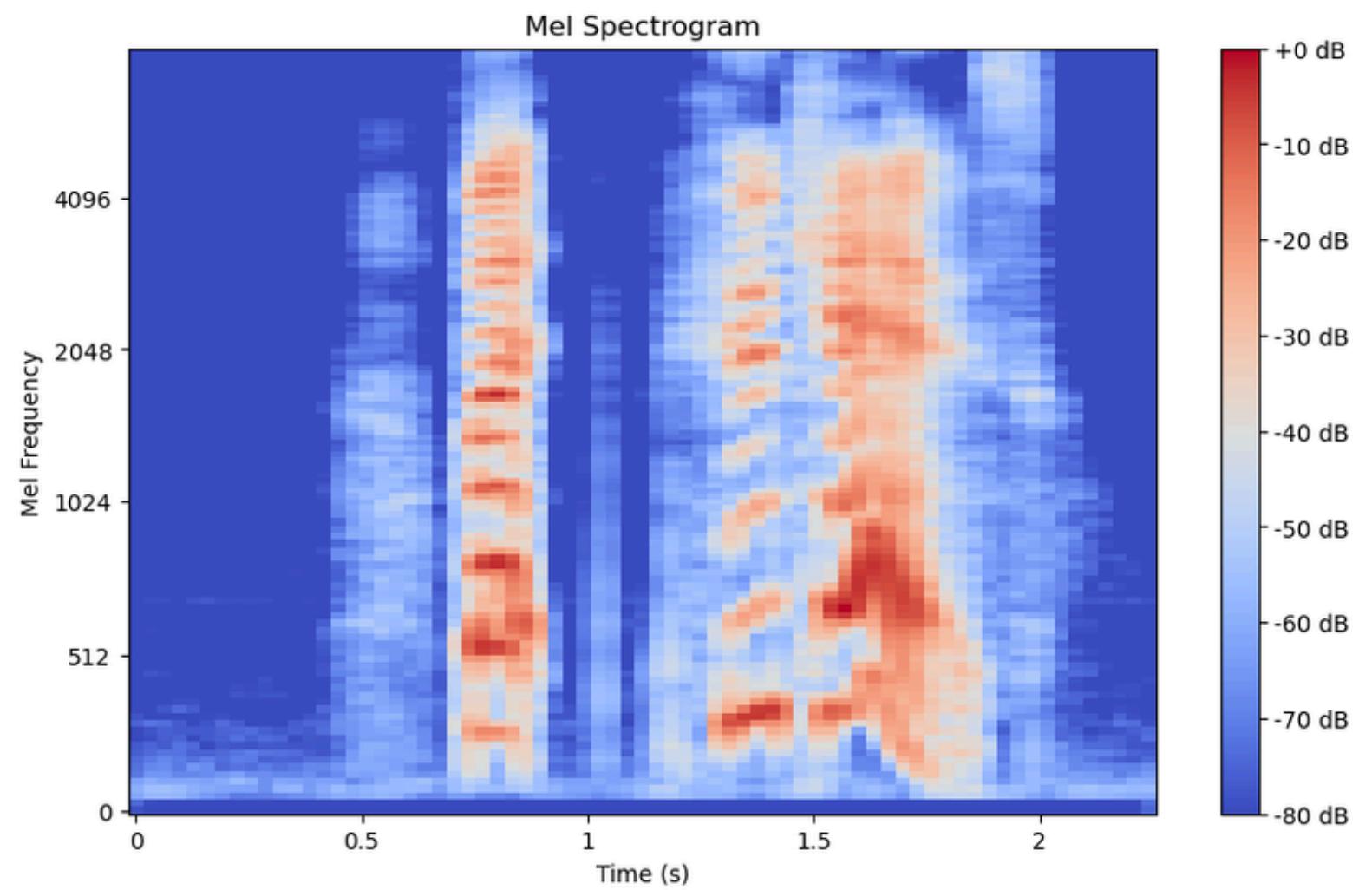
Feature-based

No	Fitur	Nilai
1	<i>Chroma-1</i>	0,3888
2	<i>Chroma-2</i>	0,3565
3	<i>Chroma-3</i>	0,3207
4	<i>Chroma-4</i>	0,4281
5	<i>Chroma-5</i>	0,5101
6	<i>Chroma-6</i>	0,5535
7	<i>Chroma-7</i>	0,4568
8	<i>Chroma-8</i>	0,4238
9	<i>Chroma-9</i>	0,4871
10	<i>Chroma-10</i>	0,4092
11	<i>Chroma-11</i>	0,4166
12	<i>Chroma-12</i>	0,3728
13	MFCC-1	-367,5911
14	MFCC-2	40,7555
15	MFCC-3	-19,8804
16	MFCC-4	11,5345
17	MFCC-5	-21,6711
18	MFCC-6	-4,9921
19	MFCC-7	-14,2397

No	Fitur	Nilai
20	MFCC-8	0,4120
21	MFCC-9	-4,8298
22	MFCC-10	-10,8160
23	MFCC-11	-2,6851
24	MFCC-12	-4,8315
25	MFCC-13	-3,7721
26	MFCC-14	-9,9427
27	MFCC-15	-7,4301
28	MFCC-16	-5,8291
29	MFCC-17	-6,4241
30	MFCC-18	0,5640
31	MFCC-19	-0,4950
32	MFCC-20	-4,7686
33	SC	2096,6777
34	SS	1851,4492
35	SR	4315,9111
36	ZCR	0,1442
37	RMS	0,0595

Fitur numerik hasil agregasi untuk keseluruhan frame

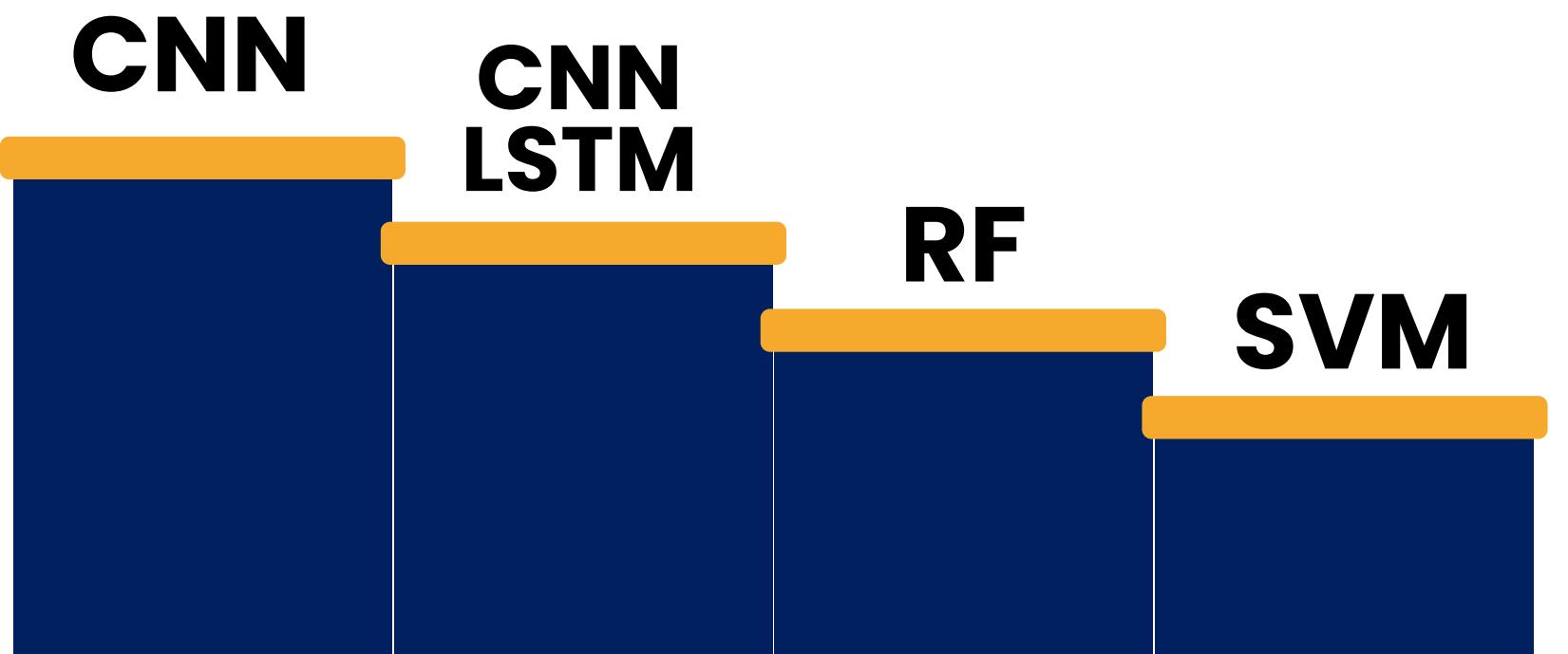
Image-based



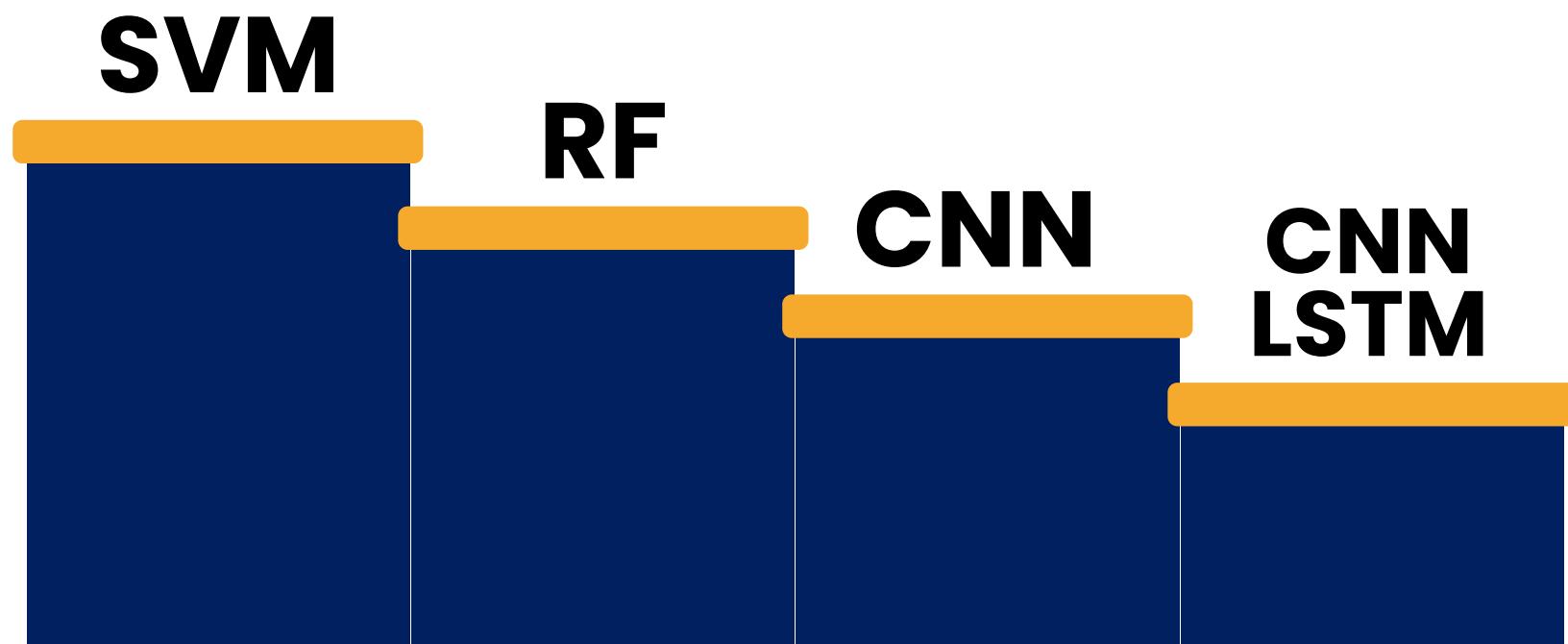
Mel-spectrogram yang mempertahankan pola temporal dan spasial

Perbandingan Hasil

Efektivitas Ketepatan Klasifikasi



Efisiensi Waktu Prediksi





KESIMPULAN DAN SARAN

SLIDE 69

Kesimpulan dan Saran

KESIMPULAN

Karakteristik Fitur

- Audio bonafide cenderung lebih stabil dengan standar deviasi rendah dan distribusi energi tonal yang merata, sementara audio spoofed menunjukkan variabilitas tinggi akibat distorsi dari proses sintesis atau konversi. Namun, secara visual (mel-spectrogram) sulit membedakan antar kedua kelas tersebut

Performa Model Feature-based

- Model feature-based (SVM dan Random Forest) menunjukkan kemampuan generalisasi yang kurang optimal dengan akurasi di bawah 81%. Namun, model ini unggul dalam kecepatan prediksi, dengan SVM menjadi yang tercepat.

Performa Model Image-based

- Model image-based (CNN dan CNN-LSTM) memiliki akurasi dan kemampuan generalisasi yang jauh lebih baik, dengan CNN menjadi yang paling akurat (91,43%). Namun, kedua model ini memerlukan waktu prediksi yang lebih lama.

Perbandingan Performa

- Hasil penelitian menunjukkan bahwa model image-based classifier, memiliki efektivitas klasifikasi dan kemampuan generalisasi yang baik, dengan akurasi dan Recall spoofed memiliki nilai yang tinggi. Meskipun demikian, model ini membutuhkan waktu prediksi yang lebih lama. Sebaliknya, model feature-based classifier menunjukkan performa klasifikasi yang lebih rendah, tetapi unggul dalam efisiensi waktu prediksi. Dengan demikian, pemilihan model yang digunakan sebaiknya disesuaikan dengan prioritas kebutuhan, apakah mengutamakan akurasi klasifikasi atau efisiensi waktu komputasi.

SARAN

- Penelitian selanjutnya disarankan menggunakan dataset yang lebih baru dari ASVspoof2019 untuk mengimbangi perkembangan teknologi AI yang sangat cepat.
- Untuk aplikasi di dunia nyata, evaluasi model di masa depan sebaiknya tidak hanya fokus pada akurasi dan waktu prediksi, tetapi juga mempertimbangkan efisiensi sumber daya, waktu ekstraksi fitur, dan kompleksitas pemrosesan.
- Disarankan untuk mengeksplorasi model ensemble hibrida yang menggabungkan pendekatan feature-based dan image-based untuk meningkatkan performa deteksi dengan menutupi kelemahan masing-masing model.
- Model dengan performa terbaik dari penelitian ini dapat dimanfaatkan secara praktis untuk meningkatkan keamanan pada sistem biometrik suara dan mengantisipasi penipuan berbasis audio palsu.



VI DAFTAR PUSTAKA

Daftar Pustaka

- Akosa, J. (2017). Predictive Accuracy: A Misleading Performance Measure for Highly Imbalance Data. *Proceedings of the SAS Global Forum*. 12, hal. 1-4. Cary, NC, USA: SAS Institute Inc.
- Alibrahim, H., & Ludwig, S. A. (2021). Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization. *2021 IEEE Congress on Evolutionary Computation (CEC)* (hal. 1551-1559). Kraków: IEEE. doi:10.1109/CEC45853.2021.9504761
- Almutairi, Z., & Elgibreen, H. (2022). A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions. *Algorithms*, 15(5), 155. Diambil kembali dari <https://doi.org/10.3390/a15050155>
- Aloysius, N. M., & Geetham. (2017). A review on deep convolutional neural networks. *2017 International Conference on Communication and Signal Processing (ICCP)* (hal. 588-592). IEEE. doi:10.1109/ICCP.2017.8286426
- Altalahin, I., AlZubri, S., Assal, A., & Mughaid, A. (2023). Unmasking the Truth: A Deep Learning Approach to Detecting Deepfake Audio through MFCC Features. *International Conference on Information Technology (ICIT)* (hal. 511-518). IEEE. doi:10.1109/ICIT58056.2023.10226172
- Bartusiak, E. R., & Delp, E. J. (2022). Frequency Domain-Based Detection of Generated Audio. *ArXiv*.
- Barua, K., Rahim, A., Parizat, P. S., Noor, M. U., & Jannah, M. (2021). *Voice Impersonation Detection using LSTM based RNN and Explainable AI*. Bengali: Brac University.
- Bird, J. J., & Lotfi, A. (2023). Real-Time Detection of AI-Generated Speech for Deepfake Voice Conversion. *ArXiv*.
- Borrelli, C., Bestagini, P., Antonacci, F., Sarti, A., & Tubaro, S. (2021). Synthetic Speech Detection Through Short-Term and Long-Term Prediction Traces. *EURASIP Journal on Information Security*, 2021. doi:10.1186/s13635-021-00116-3
- Box, M. (2024, September 30). Diambil kembali dari The Dangers of Deepfakes and Four Ways to Identify Them: <https://wellsaidlabs.com/blog/dangers-of-deep-fakes/>
- Boyko, N., Omeliukh, R., & Duliaba, N. (2022). The Random Forest Algorithm as an Element of Statistical Learning for Disease Prediction. *3rd International Workshop on Computational & Information Technologies for Risk-Informed System*. Neubiberg, Germany: CEUR Workshop Proceedings.
- Chen, T., Kumar, A., Nagarsheth, P., Sivaraman, G., & Khouri, E. (2020). Generalization of Audio Deepfake Detection. *The Speaker and Language Recognition Workshop (Odyssey 2020)*, (hal. 132-137). Tokyo. doi:10.21437/Odyssey.2020-19
- Dack, S. (2019). *Washington Edu*. Diambil kembali dari Deep Fakes, Fake News, and What Comes Next: <https://jsis.washington.edu/news/deep-fakes-fake-news-and-what-comes-next/>
- Ding, B., Qian, H., & Zhou, J. (2018). Activation Function and Their Characteristics in Deep Neural Network. *2018 Chinese Control and Decision Conference (CCDC)* (hal. 1837-1838). IEEE. doi:10.1109/CCDC.2018.8407425

Daftar Pustaka

- Giri, P. E. (2021). *Convolutional Neural Network: Konsep, Penerapan, dan Implementasi Dengan Contoh Eksperimen*. Bogor: IPB Repository. Diambil kembali dari <http://repository.ipb.ac.id/handle/123456789/111112>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press. Diambil kembali dari <https://www.deeplearningbook.org/>
- Halgamuge, M. N., Daminda, E., & Nirmalathas, A. (2020). Best optimizer selection for predicting bushfire occurrences using deep learning. *Natural Hazard*, 845-860. doi: 10.1007/s11069-020-04015-7
- Hamza, A., Javed, A. R., Iqbal, F., Kryvinska, N., Almadhor, A. S., Jalil, Z., & Borghol, R. (2022). Deepfake audio detection via MFCC features using machine learning. *IEEE Access*, 10, 134018-134028. doi:10.1109/ACCESS.2022.3231480
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques (Third Edition)*. Waltham: Elsevier.
- Hardle, W. K., Prasetyo, D. D., & Hafner, C. (2012). Support Vector Machines with Evolutionary Feature Selection for Default Prediction. doi:10.2139/ssm.2894201
- Ho, Y., & Wookey, S. (2020). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. *IEEE Access*, 4806-4813. doi:10.1109/ACCESS.2019.2962617
- Jin, Z., Zheng, H., Wu, H., & Zhang, Y. (2022). Deepfake Voice: Synthetic Speech Detection and Generation Techniques. *IEEE Access*, 10, 1-20.
- Kaplan, A., & Haenlein, M. (2018). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62, 15-25. doi:10.1016/j.bushor.2018.08.004
- Khochare, J., Joshi, C., Yenarkar, B., Suratkar, S., & Kazi, F. (2021). A Deep Learning Framework for Audio Deepfake Detection. *Arabian Journal for Science and Engineering*, 47, 3447-3458. doi:10.1007/s13369-021-06297-w
- Korshunov, P., & Marcel, S. (2018). *DeepFakes: A New Threat to Face Recognition? Assessment and Detection*. arXiv preprint. Diambil kembali dari <https://arxiv.org/abs/1812.08685>
- Kusuma, D. (2021). Teknologi Deepfake dan Implikasinya di Industri Media Sosial. *Jurnal Teknologi dan Inovasi*, 10(2), 45-52.
- Landini, E. (2021). *Synthetic Speech Detection through Convolutional Neural Networks in Noisy Environments*. Milan: Politecnico Milano.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444. doi:10.1038/nature14539
- Lee, C.-Y., Gallagher, P., & Tu, Z. (2015). Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree. *International Conference on Artificial Intelligence and Statistics*. San Diego.
- Lerch, A. (2012). *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Hoboken: John Wiley & Sons, Inc.
- Lim, S. Y., Chae, D. K., & Lee, S. C. (2022). Detecting Deepfake Voice Using Explainable Deep Learning Techniques. *Applied Sciences*, 12(8), 3926. doi:10.3390/app12083926

Daftar Pustaka

- Liu, T., Yan, D., Wang, R., Yan, N., & Chen, G. (2021). Identification of Fake Stereo Audio Using SVM and CNN. *Information (Switzerland)*, 12(7). doi:10.3390/info12070263
- Mcubaa, M., Singha, A., Ikuesanb, R. A., & Hein, V. (2023). The Effect of Deep Learning Methods on Deepfake Audio. *Procedia Computer Science*, 219, 211-219. doi: 10.1016/j.procs.2023.01.283
- Mondal, D. (2024, November 20). *Medium*. Diambil kembali dari Forward and Backpropagation in Neural Networks: A Mathematical Journey: <https://medium.com/@debspeaks/forward-and-backpropagation-in-neural-networks-a%20mathematical-journey-d8b503b0dd56>
- Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for Interpreting and Understanding Deep Neural Networks. *Digital Signal Processing*, 73, 1-15. doi:10.1016/j.dsp.2017.10.011
- Ning, Y., He, S., Wu, Z., Xing, C., & Zhang, L.-J. (2019). A Review of Deep Learning Based Speech Synthesis. *Applied Sciences*, 9(19), 4050. doi:doi.org/10.3390/app9194050
- Oppenheim, A. V., & Schafer, R. W. (2010). *Discrete-Time Signal Processing (3rd ed.)*. Prentice Hall.
- O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. arXiv.
- Proakis, J. G., & Manolakis, D. (2006). *Digital Signal Processing: Principles, Algorithms, and Applications (4th ed.)*. Pearson.
- Roberts, L. (2020, March 6). *Medium*. Dipetik October 16, 2024, dari Understanding the Mel Spectrogram: <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53>
- Sena, S. (2017, November 13). *Medium*. Dipetik October 16, 2024, dari Pengenalan Deep Learning Part 7: Convolutional Neural Network (CNN): <https://medium.com/@samuel.sena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>
- Singh, A. K., Singh, P., & Nathwani, K. (2021). Using Deep Learning Techniques and Inferential Speech Statistics for AI Synthesised Speech Recognition. *ArXiv*. doi: 10.48550/arXiv.2107.11412
- Soyander, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13). doi: 10.1142/S0218001420520138
- Stevens, S. S., & Volkmann, J. (1940). The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, 53, 329-353. doi:10.2307/1417526
- Suwajankorn, S., Seitz, S., & Kemelmacher-Shilzerman, I. (2017). Synthesizing Obama: Learning Lip Sync from Audio. *ACM Transactions on Graphics (ToG)*, 36(4), 1-13. doi:10.1145/3072959.3073640
- Tan, C. B., Hijazi, M. H., & Nohuddin, P. N. (2023). A Comparison of Different Support Vector Machine Kernels for Artificial Speech Detection. *Telecommunication Computing Electronics and Control*, 21(1), 97-103. doi:10.12928/TELKOMNIKA.v21i1.24259
- Widiputra, H., Mailangkay, A., & Gautama, E. (2021). Multivariate CNN-LSTM model for multiple parallel financial time-series prediction. *Complexity*, 2021, 1-14. doi:10.1155/2021/9903518



TERIMA KASIH

SLIDE 75



www.its.ac.id/statistika



@its_statistics

INSTITUT TEKNOLOGI SEPULUH NOPEMBER, Surabaya - Indonesia